Dynamic Information Management Methodology with Situation Awareness Capability

by

Vikram Sorathia

(200221005)

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

Doctor of Philosophy

in

Information and Communication Technology

to

Dhirubhai Ambani Institute of Information and Communication Technology

2008

Declaration

This is to certify that

1. the thesis comprises my original work towards the degree of Doctor of Philosophy in Information and Communication Technology at DA-IICT and has not been submitted elsewhere for a degree,

2. due acknowledgment has been made in the text to all other material used.

Signature of Student

Certificate

This is to certify that the thesis work entitled "Dynamic Information Management Methodology with Situation Awareness Capability" has been carried out by Vikram Samantbhai Sorathia (200221005) for the degree of Doctor of Philosophy in Information and Communication Technology at this Institute under our supervision.

Thesis Supervisors   Prof. Anutosh Maitra   Prof. Sanjay Chaudhary

Acknowledgments

I am indebted to my supervisor Prof. Sanjay Chaudhary for accepting me as a doctoral student and allowing me to join his research group. By means of courses and research projects, he introduced me to advance topics like SOA, Grid Computing and Semantic Web that laid a strong technical foundation of my research work. His unique motivating capability to be on the front line of the research kept me aware of the advancements. He has always provided me with the required resources and guidance, without which this research could not have been possible. He has given me opportunity to work as research assistant in a sponsored research project, and provided financial support. His leadership skills and result-oriented nature have created deep impression in my mind, and will continue to inspire me for the rest of the life.

I thank my co-supervisor Prof. Anutosh Maitra for providing excellent guidance throughout the tenure. Taking my ambitious research interest *"reducing chaos during disasters"*, he meticulously carved out research plan and lead me toward formulation of precise and manageable research questions. He allowed me to take up this research area and gently navigated through various phases of research that enabled me to complete research in due time. I have been provided with funding and other support required in academic and research activities. He has always supported me in various academic and administrative complexities and other difficult times that are part of student life. I really appreciate him, for showing immense patience in listening to ideas and making complex things simpler. He has always amazed me by grasping my initial thoughts, predicting and showing me all the possible roadblocks or prospects that may show up. I really thank him for all the guidance, support and help that enabled me to complete the work.

I am grateful to Prof. V.P Sinha for his continuous encouragement and mentoring through out the years. In earlier years of the program, his insightful directions lead me to explore the work of Barwise and others that laid theoretical foundation for the thesis. Prof. Samaresh Chatterji with his capacity as the Dean of academic matters, provided me numerous training, collaboration and research opportunities to enhance my knowledge and skills. I am grateful to him for all the generous help and support.

I thank Prof. Asim Banerjee for the ideas, insights and thought provoking questions regarding my research topic. Several teaching assistant opportunities with him enabled me to learn software engineering and project management principles that are incorporated in this research work. I am thankful to all the DA-IICT faculty members who have guided, mentored or supported me by acting on various academic committees.

My special thanks to Dr. Amit Anil Nanavati (IBM India Research Lab) for providing valuable feedback, suggestions and appreciation during his visits to the campus.

I sincerely thank the external reviewers of my thesis. In spite of busy professional schedule, they completed the review work in time and provided comprehensive recommendations improving the readability of the thesis.

I acknowledge the DA-IICT Resource Centre for ensuring best library, information and reference services that have played important role in my work.

I owe a special thanks to all the PhD students at DA-IICT. Especially, Zakir Laliwala has provided great companionship during the ups and downs of the student life. I thank him for introducing me excellent tools like Mind Maps and Year Planners that have helped me organize my work. Sunil Jardosh, Ratnik Gandhi, Pratik Shah, Guneshwar Anand, Divyang Rawal, Nilesh Bankar and many other current and past PhD students of DA-IICT have made the stay a memorable experience.

I sincerely acknowledge the help from Mr Arif Vohra (Senior Executive, GSDMA) for his guidance by means of resources and discussions on various disaster management related aspects. His kind and continuous help have revealed many important aspects of the application domain.

During my tenure, I have been fortunate to have interactions with many batches of M. Tech, B. Tech and M. Sc. courses, especially with the fellow students in DS Lab. I enjoyed communicating with them all and value their unique contributions.

The staff members at DA-IICT have provided lot of help and support. Lab assistants, faculty assistants, Help Desk and administration staff have always helped me with various issues. I acknowledge their kind cooperation.

At last, I thank my family that has been my driving force. With unconditional love, I am always ensured emotional, financial and any other kind support that I needed in my pursuits. I understand and acknowledge every one for all the compromises and sacrifices done to ease my efforts. I cannot thank my family enough for providing me a strong base on which I could stand still and was able to complete this work.

Abstract

In present world, information generation, handling and utilization is deeply integrated with various activities. From entertainment to serious professional activities, information content is managed in some ways. Appropriate information content is beneficial and sometimes becomes critical to life, business or the environment that supports them. Handling of information is therefore increasingly perceived as integral part of the activities. Systematic approach for handling information by individuals and organization is known as "information management".

A common approach towards information management is limited to recording and retrieval of data pertinent to various human activities or environmental processes. The resulting system acts as a computerized log of human activity transactions or phenomena recorded in natural environment. Systems supporting information management are defined based on project specific requirements with limited scope. The project-oriented or mission-specific engineering approaches employed in designing such systems also restrict the scope of supported information management activities.

It is important to realize that projects or missions are identified only as short-term objective that contribute to the long-term goals and vision of entities. A system defined to serve short-term goals may soon become incapable of meeting the emergent needs. Even with accurate estimation of potential change, external intervention cannot be avoided due to dependence of knowledge, authority and resource capability required to realize the goals. The dynamism exhibited by relevant entities, results in continuous change of needs resulting in constantly changing requirements.

With increasing amount of complex interdependence and dynamic behavior of relevant entities, the task of information management has become difficult. From the scale of effort required to address the complex interdependence, it is realized that information management cannot be carried out with systems created and managed by individuals and organizations in isolation. Also, from the dynamic behavior and evolution exhibited by relevant entities, it is realized that underlying systems must also undergo change at similar rates. In summary, a paradigm shift is required in information management strategy and information should be made available as critical infrastructure service.

This thesis argues that information management should be based on the goals of the involved individuals instead of the conventional activity-oriented approaches. The required approach not only should support the goal-specific information, but also allow identification of newer goals with emerging trends in the system. Information need of individuals and organizations is

uniquely identified in the form of situations. The state of having access to relevant information is defined as situation awareness capability. The situation awareness approach proposed for information management strategy identifies role of individuals in producing and consuming information. It is based on the realization that, no one can have the global picture of the situation, but can play a role in building the rich picture of situation by contributing the part of the situation known. The resulting coordinated effort allows realization of situation awareness capability to the contributors.

In order to support the argument, the situation theory and semantics is accepted as base. It is stated that small coordinated assertions regarding situations can be integrated to prepare a rich representation of the world. Qualitative and quantitative estimate of information needs are identified based on the commitment towards goals. Three problems are identified in identification of information needs of individuals and organizations. The implied goal-matching problem relates to the challenge of identification and handling of goals that are not explicitly expressed. The transient system element problem indicates the challenges of numerous short-lived entities that are relevant to the information needs. The third problem is regarding identification of the event space, a set of all possible events that are possible in given scenario.

As a solution to this problem, a conceptual modeling strategy followed by information processing strategy is proposed. The proposal utilizes the rich representation created with conceptual modeling process in meeting the information needs. It is established that scale and scope of work involved in conceptual modeling and information processing requires large-scale collaboration from various stakeholders. For consistency of collaborative effort, appropriate method content is provided. Reuse and traceability of work products are encouraged with unique situation awareness artifacts furnishing information about task and availability of reusable outcomes and other related information. The proposed information management facility is prescribed as a critical infrastructure service required in achieving the large-scale collaboration. Appropriate system architecture is introduced to facilitate realization of required domain specific middleware services.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Information management is a process employed to provide right information at right time to the right person in a right format [1]. Individuals and organizations can be seen as goal seeking entities that continuously evaluate their states. Based on the availability of information about their current status and possible future status, they determine next desired state. Making the desirable status their goal, they involve in activities that can support reaching the goal-state. Access to right information governs not only decisions about required actions in achieving the goals, but also affects the identification of goal state itself. Hence, it can be said that information management strategy is successful when it provides necessary information for continuous identification of goals, and its attainment.

With civilization, scientific development and other advancements, goals and activities of individuals are increasingly becoming complex. Skill-based activities of individuals have resulted in division of labor at scales that have resulted in specific disciplines of contribution. For example, individual actions recorded in primitive ages like hunting, gathering are gradually replaced with specific professional activities like: medical practice, marketing, teaching etc. In this scenario, an individual requires services of many others in various stages of life. This type of *division of labor* has increased interdependence of individuals. In meeting individual needs, various modes of collaborations, interactions and exchange of goods and services are employed by individuals in social environment.

In such scenario, information management to facilitate an individual in determining goals and required information for its attainment is becoming increasingly difficult task. Individuals continuously determine next goal-states from many known possibilities. For the selected goal at any given point, contribution from many other individuals can be possible. Information about all such contributors should be made available to the goal-seeking individual. In other words, the source and sink of information is continuously changing with the actions and intentions of individuals. Keeping track of this dynamism is a difficult challenge for information management strategy.

With civilization, access to certain goods and services are ensured to each individual. These are known as critical infrastructure services. By providing ensured access, a mechanism is made responsible for continuously monitoring availability of these basic services. Access to these services is affected by many natural and man-made interventions. Hence, information about everything related to uninterrupted access to services is identified as basic requirement. More and more products and services are increasingly added in the list of basic services indicating the development in civilized world. This implies increased amount of interdependence on various factors, resulting in increased load on information management strategy [2].

The logical collection of interdependent individuals, resources, environmental conditions and related aspects can be identified to form a system. A given universe of discourse (UoD) can be identified to have multiple such systems. Individuals or collections of individuals embodied as organizations play specific roles in the UoD. Based on the assumed roles they have to subscribe to appropriate information management strategy that allows identification, processing and delivery of information within the system. In order to support this information flow [3], information management strategy must be equipped with appropriate information system.

An information management strategy is required to meet their continuously changing needs. This includes a comprehensive methodology along with the system that can support required information flow. The methodology has to first establish the complex interdependence and dynamism prevailing in the system.

1.1 Complex Dynamical System Characteristics

Characterization of dynamism and complexity is the prerequisite for efficient system management. The first step towards characterization is to model the system as a collection of functional units and resources employed to attain a specified goal. In emerging systems, the collection is not limited to a small and obvious set of entities but have complex relationships and behaviors that are difficult to establish. Hence, modeling must characterize the system by identification of relevant entities, behaviors and relationships among them in the given environment.

Some of the known system modeling approaches [4] introduced basic system elements like system-environment boundary, input, output, process, state, hierarchy, goal directedness and information. It is observed that a system is not restricted to itself, but it may interact with other systems and with the environment and also may undergo evolution. The *Open System* character is studied in many natural and man-made systems like ant colony, vehicular traffic and financial markets. Some of these systems consist of multiple interacting agents exhibiting emergent behavior without the presence of a central control. They are classified as *Complex Systems* [5]. In some cases, it is possible to establish a law that allows determination of system status at a given time based on the previous status and characteristics of system entities. Such systems are known as *Dynamical Systems*. During evolution, systems may sometime experience sudden unpredictable changes or *punctuations* as opposed to long period of marginal changes identified as *Stasis*. In response to punctuation, the system undergoes major changes to adopt a new situation by getting help from within and outside. A system with this feature is known to exhibit *Punctuated Equilibrium* [6]. The dynamism results in continuous changes in the system boundary.

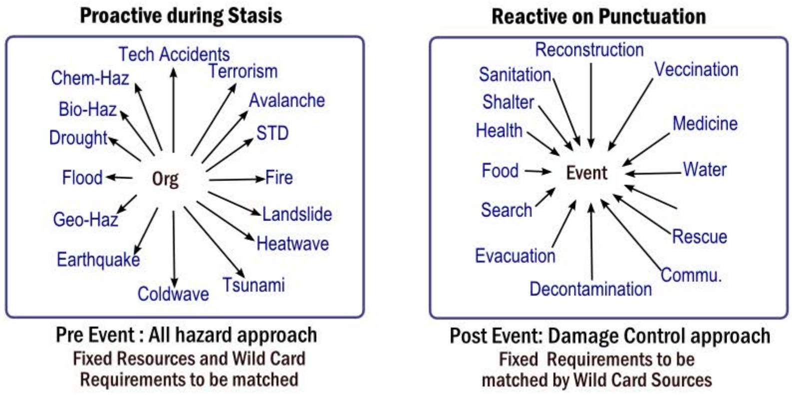

Figure 1.1: Pre and Post Event Focus

Figure 1.1 explains the stasis and punctuation handled by the disaster management system. In normal conditions, the disaster management authority is required to handle all potential hazards with available resources. In stasis the focus of management strategy is on proactive identification and updates of plans in addressing known hazards by gradually making required provisions. In case of disastrous event, the focus changes to reactive strategy, focusing on the damage and making provisions to restore various essential services. In this case requirements are known and required resources may be acquired with external help. Such major event may result in considerable changes in plans and provisions made during stasis.

The membership of a system can be defined based on its role. A system targeted at rendering some service requires specific natural and man-made resources. In absence of any of them the system is not able to render the service. This heterogeneous mix of resources is known as *Actor Network* [7]. This knowledge allows the determination of boundary as well as the status for the given system. It is also likely that members of one actor network can be members of other networks too. It results in overlapping system boundaries. In systems with overlapping boundaries, a status change of a member of an actor network may affect the status of many other systems. This complex dependence can be established by comprehensive representation of system entities, their interactions and relation within and outside the system boundary.

1.2 Situation Awareness

Access to right information, at right time, to right person, in right format provides awareness of the situation in the surrounding of an individual. In this process, apart from the individual, the system goal is equally considered critical in determination of what is right at specific time. The overall aim is to control the system state by furnishing appropriate information to the individuals, who are not the passive observers of the situation, but they are active agents that can have effect on the overall state of the system by means of actions. Their actions affect the system state and that is reflected back in their Situation Awareness. Hence, Situation Awareness acts as a dynamically changing representation of the system that is influenced by the nature, behavior and dynamics of itself and its surroundings.

1.2.1 System Point of View

The individual and goal centric view of achieving situation awareness guides the overall information management process. First step towards it is the identification the system that is prevailing in given UoD. Identification of the system with its overall boundaries is a difficult task, as there can be many systems identified in the same UoD. The goal provides a criterion to define a system by identification of relevant entities, environment and a boundary. All management actions are further carried out to this derived system, identified as a subset of the UoD. The objective is to interpret the current situation, decide future course of action according to the goals and furnish the information about required action to individuals. The modeled system, encoded goals and identified actions based on and continuously communicating to entities in the real environment in the given UoD.

Information management process creates a logical equivalent of the derived system to identify, process, and furnish required information among various elements. In summary, a system is identified to exist in given UoD, and corresponding to that, a logical equivalent is maintained to support processing and information flow with a facility of feedback control.

This strategy is successfully employed in plant automation environment in industrial systems that are close system as they are controlled with known finite set of parameters. Apart from few numbers of critical parameters, the number of individuals that are required to keep such systems in desired state is also very limited. However with changing nature and type of the system, the number of parameters to monitor, the required knowledge to control the system, number of individuals and nature of response may be larger.

1.2.2 Information Management Point of View

As the systems targeted for situation awareness continue to become complex with increasing number of critical parameters, the task of sustaining information flow becomes more challenging. The task of modeling, handling, processing the information; and decision-making based on that require consistent efforts from multiple individuals. A coordinated response is required to achieve control in the system environment. Hence, information management process itself, must support handling collaboration at larger scale. Coverage of individual effort introduces integration related problems that are to be addressed.

1.2.3 Information Management: the State of the Art

Critical factors in information management that affect the success of the systems have been studied. In response to failures and based on the findings, the discipline is continuously developing with contributions at various levels.

The theory pertaining to control and management of information have been developed to influence the information management practices. A broad classification of this theory is provided in [8]. A standardization process has taken place to address interoperability issues at various levels [9]. Standards and specifications are established by consortia and organizations using consensus-based process. Application level standards include business processes, transactions and related services. Communication level standards include standards of various wired and wireless communication techniques. Data Interchange standards includes graphics, video, maps, text, audio and other forms of digital content. Data management standards incorporate data handling, query, metadata, storage and related issues. For processing of information, standards are defined to enable distributed computing. For the visualization and user experience graphics and virtual modeling standards have been introduced. File level tasks for handling, storage, sharing and utilization are addressed by standards in operating system. Security and related issues like authentication, authorization, encryption etc. are also comprehensively covered in standardization process. For the development of software in collaborative environment, standards pertaining to software development related to software architecture, programming language, modeling and related aspects are covered in standards.

Collaboration tools are provided for group-decision and work functions in collaborative environment. As tracking, sharing, reusing of work products is important requirement identified from past experiences. Participating users may have dissimilar skills, operating procedures, knowledge and perception of tasks; hence method engineering discipline has emerged as solution for this problem. Method engineering provides rich guidance captured from experts and provided to carry out specific tasks [10].

Multiple cases of information failure are traced to the issues at information technology infrastructure level. Many solutions have emerged addressing these issues. Comprehensive architectural frameworks are suggested [11] to guide the process of software system development in distributed and complex environments. Certain functionalities that are commonly required for processing and handling of information are created and shared as libraries. These reusable components address such issues in distributed environment are available as middleware. They address specific issues of communication [12], data integration and access and other related aspects in software environment. As present systems are increasingly depending on and participating with other systems, integration of multiple such systems is an important issue. Enterprise integration strategies [13] address such issues, apart from strategies discussed at architectural modeling and deployment levels, there can be domain specific issues that must be dealt with at programming level. Patterns are provided as design solutions to given domain problems. Rich resources of patterns are published to help solving specific problems[14]. Patterns also address higher-level issues in enterprise integration like communication [13] and data access [15] in distributed environments.

While each critical success factors are dealt with by organizations having expertise in a specific area, a clear separation of concern is required. Sources of information can potentially be many, each has its own issues like coverage, specificity, quality and availability, they are also increasingly available as service. The advancement in instrumentation has improved collection, handling, communication, processing and storage efficiencies of information sources. As they are exposed with standard interfaces, the reliable access to information sources have became possible.

Figure 1.2 indicates the resulting interplay among various actors involved in various roles. Each plays specific roles individually or as a group, and carryout activities. For carrying out assumed role, one actor requires input from many stakeholders. As these collaborations are goal directed, the policies and standards play a central role in evolution of individual work product and therefore are continuously observed.

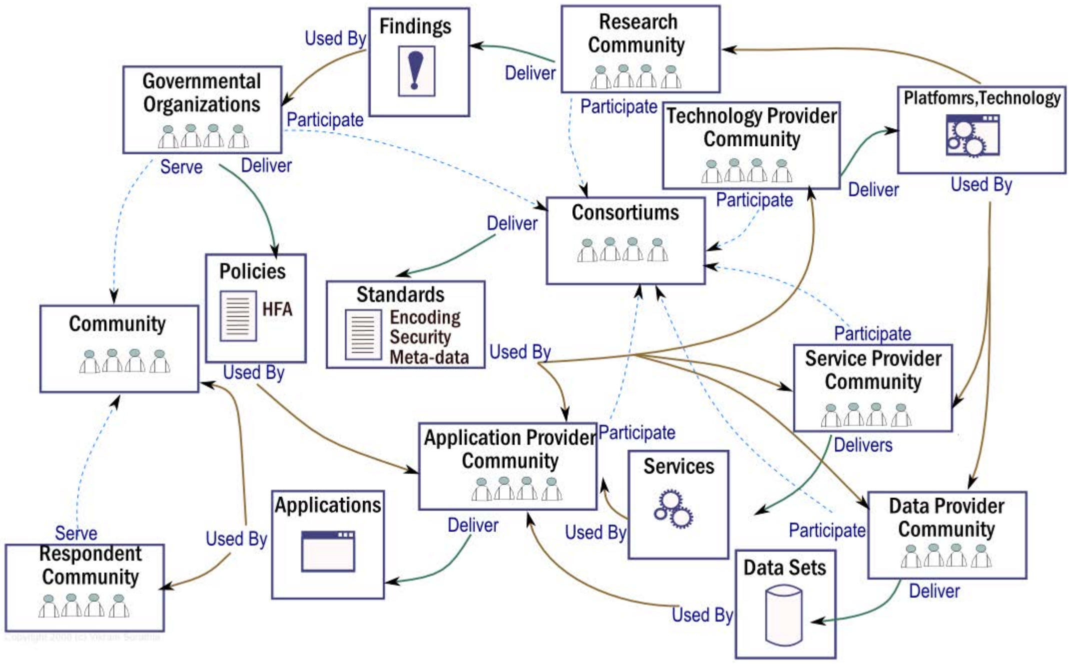

Figure 1.2: Interactions among Stakeholders

1.2.4 Application Domains

This research is targeted at application domains like environmental management, disaster management, critical infrastructure protection, security and other similar field. In these domains, there can be many active independent agents can be found with conflicting interests. As a result of their activity, many agents are either healing or harming the system.

Disaster Management Disaster management is a domain of activity that is employed in planning for and managing response to disasters. A disaster is identified as an emergency that exceeds the capability to manage and results in loss or destruction. The kind of emergency and the type of loss can depend upon the goal and associated resources for the achievement of the same. Considering the environmental management domain, an event of oil-spill in an ocean is a disaster, as it severely affects the elements of environment by loss of livelihood in area affected by oil-spill. In critical infrastructure protection, a collapse of communication network can be considered as disaster, as it impairs all the dependent activities in affected area. Hence, disaster can be of many types, many domain depending upon the cause and impact thereof. In management of disaster, the availability of latent dangers is identified and proper steps are taken to prevent possible disasters. When disaster happens, the focus of management is on controlling the damage caused by it and restoring the situation to normal conditions.

Following *actors*, *system*, *services* and *information* can be identified that relates to the information management task in the domain.

Actors: Positive actions are the natural processes in normal conditions. From human point of view, the positive agents are entities involving in planning, mitigating, preparing to prevent possible disasters and responding and recovering from disasters that might have happened. One important element in the domain is the administrative unit that is legally responsible for disaster response, known as Emergency Operations Center (EOC). Negative agents and processes are those that increase the risk of disastrous event.

Systems: Relevant systems support tracking of positive and negative agents and conditions that may cause or prevent disasters. The tracking involves system capable in monitoring various parameters of environments, resource, actions and processes that may be at risk. Availability of such systems is increasing with improved knowledge on interrelation among identification, monitoring and responding to all possible risks.

Services: Relevant services are the systems deployed with dedicated resources and actors that monitor and respond to the situations. With more number of risks identified in given UoD, higher numbers of service provisions are required.

Information Management: Information provides status of various elements in UoD, the plans and domain knowledge that guides the interpretation of the status. Information management for disaster management is critical task as continuous monitoring, interpretation, and prediction is required for all identified risks. As knowledge about hazards and impacts are identified, new systems are developed, and new risks are identified in the UoD, the responsibility of information management increases. Unlike other domains, uncertainty in disaster management increases manifold, as the involved entities, resources, impacts and other aspects are difficult to predict a priori. Disasters may cross many administrative boundaries, and involve response from volunteer with no uniform skill sets. Response phase provides very small window of opportunity in which, provision of appropriate information becomes challenging task [16].

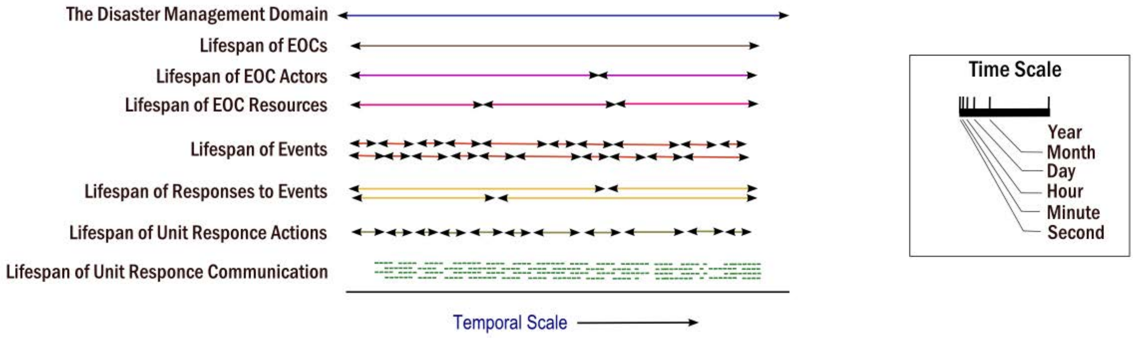

Figure 1.3: Comparative Life Span of Entities

Figure 1.3 shows comparative lifespan of entities in the disaster management domain. The upper most timeline indicates the domain. The lifespan of EOC is indicated in terms of few years. Their availability and structure is evolved over a period of time, based on development of the domain. Within lifespan of EOC, the EOC actors in form of emergency team members, resources like emergency vehicles, instruments and apparatus are acquired by the EOC. With this setup, EOC have to respond to event. Lifespan of event can be from momentary up to few months depending upon nature of hazard. Multiple arrows indicate that different types of events may take place simultaneously, that may even overlap.

In response to these events, the EOC has to engage the response work. The lifespan of response work is generally longer than the event itself. With multiple events, the lifespan also gets overlapped. During the lifespan of response work, multiple unit actions can be identified. This can be domain specific response action like medical examination, reconstruction, rescue, search, fire fighting etc. carried out by a respondent. In carrying out these actions, the respondents get engaged in communication with various other actors. They communicate to EOC to get information about the work, they talk to victims, and they talk to other respondents. The lifespan of these communications are very short in scales of few seconds or minutes.

In this illustration, it is argued that information taking place in respondent and victim communication are the sources as well as sink of information for information management strategy. They are very short lived, unplanned and cannot be modeled a priori. Second important issue is the load on the respondents. The same limited amount of respondents has to respond to multiple overlapping events. Management of them becomes very complex problem for information management point of view.

Critical Infrastructure Protection Critical infrastructure protection is a domain of activities that are focused toward protecting the infrastructure that are critical for the given society. An infrastructure is a collective mix of entities resources and action provided in the form of a service. Transportation network, telecommunication, electricity, drinking water, drainage etc. are example of common utilities that are found in typical urban environment. Figure 1.4 indicates a typical urban scenario in which various infrastructural services are visible. Individuals and organizations carryout various activities using systems built over these infrastructures. For example a medical service, business, education and financial institutions employ specific systems that required consistent connection to power, communication, transportation, drainage and other infrastructures. All recurrent infrastructures are therefore identified as critical infrastructure. Non availability of them directly has impact on all the systems and subsequently affects part of community that depends upon such system for their activity. Critical infrastructures are therefore considered as weak link [17] or soft target.

Similar to disaster management domain, many possible positive and negative agents, conditions, processes, and activities can be identified for critical infrastructure protection domain. Their dynamism and complex interdependence poses unique challenges for information management task.

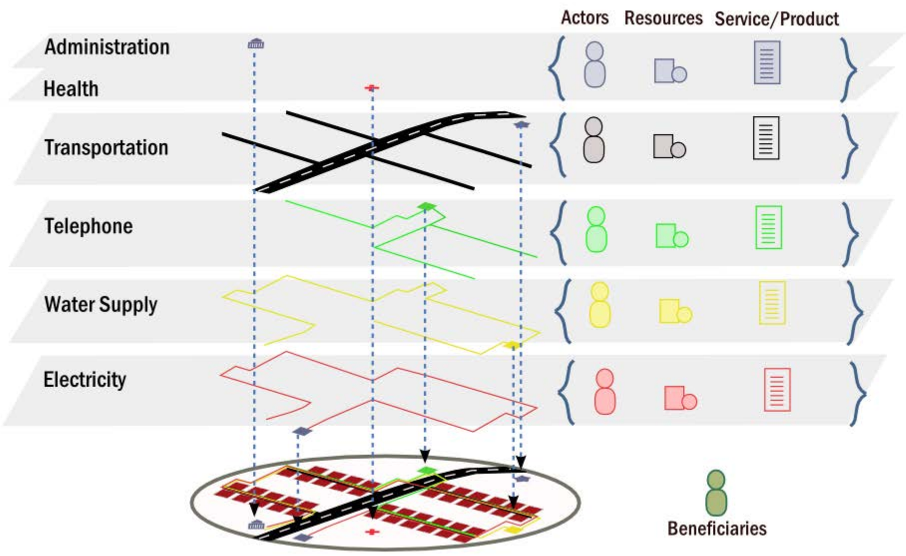

Figure 1.4: Common Infrastructure elements in UoD

1.2.5 Management and Information Needs

The task of managing a system often requires constant monitoring of the system dynamics. This includes monitoring of entities and processes that are harmful or beneficial for the desired system status. There can be many ways in which a system can be harmed and healed and there is no single specific solution. This management problem is known as a *Soft Problem* and can be solved with *Soft System Methodology* [18]. This is an iterative method and it requires a definition of the problem situation, creation of vivid representation of the system, enabling system evaluation from different point of views, identifying desirable change, and determination of recommendations from a single system management point of view. It is observed that after punctuations, a system receives internal and external help from its environment. From the management perspective, it is possible to predict the potential problems and build alternative measures, thereby improving the capability of the system to deal with punctuations within system. This is termed as *Dynamic Capability* [19].

There can be many interdependent systems visible to potential external contributors. Some of them are perceived to be in the need of external help. This help is provided to get them back to stasis period. The external entities who are able to identify such problems and get involved in the restoration process are known as *Active Publics* [20]. The involvement of the active publics is a result of a behavior that includes *Information Processing* and *Information Seeking*. This also allows identification of a desired information flow within and outside the system boundary. Information about the system status triggers some activity from the active publics that in turn change the status of the system that again is detected and reported back. *Situation Awareness* [21] is one class of application that deals with monitoring, comprehending and predicting system states to determine the decision of actions. This results in a feedback control loop; a common control system employed in closed industrial systems. The situation awareness must also be extended to deal with open system environment so that it can be made applicable to complex dynamical systems [22].

1.2.6 Critical Review of Existing Information Management Strategies

A wild card is a symbol, a word or an entity that can be substituted with any other one in given context. The context can be in the form of policy, capability or commitments. In organizational scenario, the provisions are made for issues and entities that are explicitly defined. The presence of wild cards make the management task difficult as they can be abstract and actual instances can be identified only on the runtime.

Reason 1 (Wild Cards in Policies)**.** *Policies of the stakeholder organizations contains wild cards that define the scope at very abstract level. Wild cards like "all risks", "every level" and "any relevant" may introduce unplanned "risk", "level" or "entity" that can be critical to the success of the system. At the time of designing the system based on such policy statements, the omission of any important part may result in failure.*

Reason 2 (Wild Cards in Capabilities)**.** *Database servers, equipments, computing resources, communication resources are argued to possess wild card capabilities. Mere availability of such resources does not serve the purpose unless specific provisions are made for their integration with the existing resource framework. As they can be employed in many different kinds of tasks during the punctuation phase, it is not possible for coordinating agency to individually attend each resource and make provision for their immediate and optimal utilization.*

Reason 3 (Wild Cards in Commitments)**.** *Actors also display commitments by using wild card in their capabilities. Asserting their willingness in the form like "provide anything required" or "do the necessary", the appropriate task assignment becomes difficult. While allocating suitable* *roles to available pool of such actors, it becomes difficult for coordinating agency to match suitability of the actor for specific roles.*

Reason 4 (Scope of Activity)**.** *Scope of any domain-activity is limited. For example, the software development activity, or system development activity, tries to cover limited scope of identified requirements. When overall coordination is required, mere integration of these building blocks with limited scope do not provide full coverage of required activity. This results in reveling gaps and inadequacies in the available provisions.*

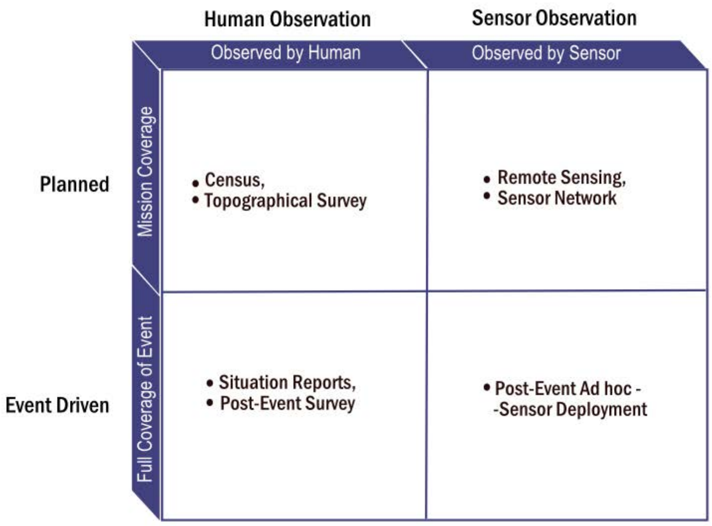

Figure 1.5: Quadraple of Information Collection

Reason 5 (One Size Fits All - Type of Solution)**.** *Contingency plans created by addressing specific risk, are assumed to be applicable for other risks in other situations.*

Reason 6 (Information Collection Planning)**.** *Information collection plan is limited to information sources identified at design time. The dynamism in the system may require incorporation of new information of sources.*

For example, In Figure 1.5, it is clearly indicated that present scenario, where source of information is commissioned scope only. When event takes place that requires information collection beyond the scope of the commissioned system, additional help is required. The picture also represents the human observation as a source of information for specific type of information that cannot be captured with sensor technology. Large-scale requirement of human observed information introduces challenges in information collection.

Reason 7 (Gap Analysis)**.** *Gap analysis is done only at system level during the designing of the system. Collaboration-wide gap analysis on the run-time is generally lacking.*

Reason 8 (Task Allocation)**.** *In collaborative environment, the delegation of task is generally voluntary, and hence it is delivered at different self imposed quality and quantity criteria. It is also delivered in isolation, such that information system or other volunteers have no record of such isolated activity, resulting in duplication of effort, wastage of resources and efforts and resulting in overall management failure.*

Reason 9 (Silos and Stove pipes)**.** *Coordinating agencies must rely on existing systems of collaborative actors. These individual systems are highly customized to suit the business of the specific collaborators, resulting in integration challenges of such silos and stove-piped systems.*

Reason 10 (Granularity)**.** *Granularity of basic artifacts restricts its utility at specific levels only. For example, a published report containing national level aggregate figures and statistical analysis has very coarse granularity and hence has very limited utility for a role working at city, town or village level.*

Reason 11 (Separation of Concern)**.** *It is being argued that task of all the stakeholders can be traced to and from less abstract to more abstract artifacts and work products. In spite of this dependence, the methods, skills and systems used by actors in carrying out activities are different, hence, proper Separation of Concern must be ensured by the system design.*

Reason 12 (Absorbing Innovations)**.** *Research efforts involved in issues related to the domain may continuously generate important findings that can be immediately applicable to the system. No strategy for adoption of newly generated knowledge indicates limitation of the system.*

Reason 13 (Standards Compliance)**.** *Standardization efforts involved in issues related to the domain may result in standards and specifications. No strategy for incorporating newly set standards in collaborating systems indicates the limitation of the system.*

Reason 14 (Dashboard View)**.** *No Strategy that allows visualization of the status of all the relevant actors, work product and services in dashboard-like environment indicates the limitation of the system.*

Reason 15 (Data Access Pattern)**.** *Data access strategy that may require two-side communication of information along with access to computing and storage resources for handling and management of the data. Lack of appropriate data access pattern may result in erratic information flow indicating the limitation of the system.*

Reason 16 (Problem Recognition)**.** *Lack of proper problem recognition mechanism that prevents correct identification of urgency and importance of the problem. This may result in assignment of task with lesser priority as defined in Figure 1.6.*

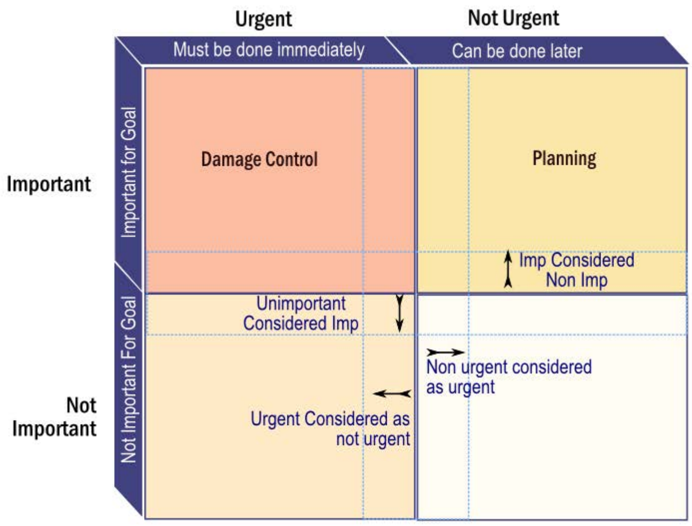

Figure 1.6: Quardaple of Task Recognition

1.2.7 A Plausible Approach

Identified complexity can be addressed with a suitable collaboration strategy. The collaboration from multiple individuals and organizations must be supported with clear delegation of task, responsibility with tracking mechanism to ensure monitoring and resource access. Identified dynamism can be addressed with both - proactive and reactive approach. For reactivity, event-driven response to the unfolding situation must be supported by the system. The proactive planning includes identification of potential risks and realization of system that support monitoring and continual adjustments in plans. The changing information needs can be continuously identified for each stakeholders of the collaborative system. Access to information meeting the identified need can be provided in the form of Situation Awareness. Continuous provision of actionable situation awareness must allow realization of feedback control loop in complex dynamical environment. The role specific actions may involve procedures and work flows that can be complicated and may require special training, skills and reference like standard operating procedures. Method engineering [10] is one approach that provides collection of guidance to aid individual tasks. Such guidance should be provided along with the situation awareness. In order to realize these functionalities, information system development process must meet additional considerations like interoperability so that the planned system can work coherently with other existing and future systems. A proper separation of concern is necessary amongst the interoperable environments to maintain the efficient interdependent system management.

Traceability: It is argued that it is possible to find traceability of work product outcome of every stakeholder identified in the given UoD. This traceability not only provides the link for establishing the identification of the requirement, but also provides basis for gap analysis, monitoring, evaluation and reuse in given configuration environment.

Granularity: Granularity of information must be appropriate to the specific role. Granularities must be calculated at required levels, in semantic, spatial and temporal dimensions.

Observability: Not all the information need can be matched by sensor technology or monitoring instruments. Human observation and intervention is required otherwise. The observation model and its realization in system environment is therefore necessary for the required observability of the system.

Adaptability: Unforeseen scenario includes involvement of actors at different geographical scales for undetermined time frame. This results in requirement of sustaining a Virtual Organization that may contain conflicting interests, procedures, and systems. The configuration must therefore be able to adapt to such changes on the run time.

1.2.8 Examples

Previous section identified some features of the potential solution towards information management in dynamical systems. The utility of these approaches is established by some example applications. They provide motivation to explore it further for their suitability for bridging the identified gaps in current practice. Following discussion introduces some examples of useful information management practices.

Information Generation: Information generation in collaborative environment resulting in continual development of content is demonstrated by Wikipedia[1] and related projects. The system provides basic feature like on-line text editing to the registered users. Yet, the strategy employed to handle the contributed content has made it possible to attract many users in creating, editing, accessing and handling the content as a reliable source of reference. The content is created and managed by multiple users on voluntary basis. When a user searches for a term, and if the desired term is not found, the strategy allows creation of a new content page for the searched term. The monitoring mechanism appropriately attributes the newly created topic for required content and corrections. Later, other users and volunteers, accessing the same topic, may have capability to extend the topic and hence they find the blank page or incomplete article as invitation to contribute. These mechanism serves as continuous monitoring and ***Gap Analysis*** that facilitate content generation in collaborative environment.

Content Management: The other Web 2.0 category portals provide content management capability by furnishing a meta data management service. The user contributed content is

[1] Wikipedia-a web based free encyclopedia. http://www.wikipedia.org

assigned tags, that later provides enhanced search and retrieval capabilities. The voluntary collaborators provide tags to describe the shared contents like photographs, videos, bookmarks etc. These tags are then used for searching, retrieval and visualization of the shared content.

Information flow The mash-up[23] applications like yahoo pipes[2] and similar services by other free and commercial service providers allows composition, aggregation and dissemination of information content. These services enable realization of information flow to suit the need of the individuals.

Events, Alerts and Notifications: Another important mechanism is content syndication, that allows flow of information according to subscriptions made to desired topics. Event-driven retrieval of information restricts the control and flow of information from multiple sources. The content retrieved in such manner is appropriately rendered to facilitate the visualization.

This feature allows realization of situation awareness feature by providing awareness of important aspect in the area of interest. GDACS[3] and RSOE [4] services are dedicated to the dissemination of disaster related event at global scales. A network of disaster monitoring stations situated across the globe has made it possible to retrieve occurrence of disastrous events in dashboard type environment.

Information Processing: With the technological advancement and availability of required infrastructure, the computational power is made available as utility that can be tapped by the user analogous to the utilization of electricity. The grid computing technology allows exposing heterogeneous compute resources that can be consumed using standard open protocols.

Global Policies: Policies that guides the information management at global levels are coming in to existence. Multiple collaborating organizations pledge to ensure required information flow. Hyogo Framework for Action[24] is one such policy framework that identifies action priorities required in the field of disaster management. Many of the identified priorities are directly related to information management strategy. Global scale coordination of humanitarian response is managed by Relief Web[5] international- a comprehensive information management system supporting the global response. The agency publishes information regarding situation awareness. It provides information in form of situation reports, *Who is doing what and where* (WWW), fund requirements, pledges and other response related information.

---

[2] Yahoo Pipes: http://pipes.yahoo.com/pipes/
[3] GDACS - Global Disaster Alert and Coordination System: http://www.gdacs.org
[4] RSOE EDIS - Emergency and Disaster Information Service: http://hisz.rsoe.hu/alertmap
[5] ReliefWeb International Portal: http://www.reliefweb.int

1.3 Problem Definition

Dynamism and complex interdependence amongst the entities and processes are established as primary challenges for information management strategy. Strategic provision for dynamism determines a dynamic set of entities to be considered for information needs at a given instance. A strategic provision for interdependence determines the resources, activities and outcomes related to the identified dynamic set of the entities.

In order to support these basic needs, the information system must realize the tasks of appropriately encoding, communicating, processing and interpreting information shared among the stakeholders. It is evident from the nature and scope of the problem that multiple instances of the same information system are required to meet the information need of all stakeholders at any time. Hence a meta-level system engineering strategy is required. It is expected that dynamism may introduce processes and outcomes that are not well understood with available knowledge or may not be supported by available technologies. Hence the strategy should support uncertainties and provide a scope of continuous evolution of systems. Stakeholders involved in specific domain activities contribute to the overall system evolution. The strategy must incorporate an event-driven mechanism that establishes traceability of work-products across stakeholder environments. By this means, continuous gap analysis and task involvement must also be supported. The targeted system must be enabled to observe, process and handle necessary information sources to support role specific information granularity.

Based on such requirements, the following issues are identified as research challenges:

- A situation awareness concept extended to complex dynamical systems
- Information management strategy to support modeling, instantiating, monitoring, evaluating, communicating and managing the system entities for required situation awareness.

The scope of the present research is to identify and deploy a strategy to enable the stakeholders in building, managing or utilizing a particular information system configuration that meets their instantaneous information needs.

1.3.1 Unsolved Issues

Based on the identified problem statement, the following research questions have been identified.

How to characterize system and its boundary? The given UoD consists of multiple entities. Depending upon the goal, it is possible to visualize a system. For example, for environmental management domain, actors, activities and resources that alter the state of environment are considered to form a system. It is difficult to draw exact system boundary due to various reasons. One of the reasons is the ramification of activities and resources. The state of affair can be altering due to many indirect or passive reasons. It is difficult to enumerate all of them.

How to identify the management strategy? Classical management strategies may not be sufficient to handle the level of complexities evident in the identified domains. The availability of resources and presence of agents that utilize the resources are dynamic in nature. It is difficult to predict the exact outcome of the agent activities as they are affected by numerous factors. Hence, there may not be simple applicable solutions to the identified problems. Many steps and activities can be achieved that may potentially lead towards the goal state. In other words, a feedback control loop should be realized in complex dynamical systems. Hence, the information management approach must follow a suitable management methodology.

How situations can be defined and possible worlds can be determined? Situations are part of the world evident in the given UoD. Depending upon the goal, situations can be identified as representation of relevant entities and their states. Since, any representation consisting of the instantaneous information of relevant entity is considered as a situation, there can be multiple situations identified with the same set of entities. Similarly, depending upon all the possible states assumed by these entities, number of possible world can be identified. Appropriate selection criteria for situation and possible world need to be identified.

How situation awareness can be defined for complex dynamical systems? Awareness of situation is important requirement for management. Situation awareness is classically defined and used for simpler domains. For complex dynamical systems, these definitions are not applicable. As per the goal, and emergent behavior of involved entities, situation awareness requirements may change. Hence, situation awareness needs to be defined with appropriate provisions for complex dynamical systems.

What modeling strategy will be useful to capture required roles, properties and rules? Determination of prevailing situation is possible by identification of important concepts, their properties. Each property can be measured on specific scales to determine the current status. This determination requires rules specific to the domain. Identification of all the relevant properties, specification of methods for consistent measurement and explication of rules that allows determination of the instantaneous state requires careful modeling of the system. The modeling task may span various domain of expertise and should be carried out for required geographical regions to cover all the required aspects. An appropriate modeling strategy needs to be defined that allows consistent conceptual representation followed by appropriate information processing tasks to meet the information needs.

How complex event processing can be achieved in complex dynamical environment? The complexity and dynamism are characteristic features of complex dynamical systems. According to the information-processing requirements identified for identification of the status of various entities, involves explication of rules. Some of the rules allow identification states based on the measured values. The values measured in a critical range may signify occurrence of an event. With entities having complex interdependence, the identification of event may require multiple measured values defined in the form of event profiles. The information management strategy must identify all the required event profiles that enable capturing of relevant events.

How coordinating agency can identify the setup and resource requirements ? For an agency it is difficult to determine the amount of work, resources and actions required to meet the goals. The number of risks, domains, stakeholders required to be handled for given application domain and kind of system infrastructure and resources required to handle them may continuously vary. The information management task required to support these activities must identify the setup of required infrastructure and resources required. With identification and provision for all possible reuse and collaboration, considerable amount of requirements can be satisfied. Hence, situation awareness must include such possibilities.

How stakeholders can identify the work? Identification of task in collaborative environment may depend of the emergent needs, capability requirements, knowledge, resources and other specific requirements. Information management strategy must furnish this information to facilitate the stakeholders in identification of required tasks.

How stakeholders can share the work products? Different stakeholders are continuously engaged in creating artifacts, work products and services. They can be reused in other situation later. But, their work products can be contributed in specific environments. Information management strategy must support mechanism to support cross environment sharing of work product outcomes.

How multiple systems and sources are integrated? It is possible that reusable components are shared and monitored by different systems in different environment. Depending upon the type of resource, the requirement of tracking may also be unique. Their availability and usage scenario may also be specific. Hence, it is difficult to integrate every types of work products and artifacts created by various systems and make them available for reuse. Appropriate integration strategy must be defined to solve this problem.

How stakeholders can track the work and get feedback and still get separation of concern? There can be many types of roles played by stockholders. Outcome of their work products can be taken as input to many others as indicated in Figure 1.2. From user point-of-view, the information about the availability of usable work product is therefore important aspect of situation awareness. From contributor point-of-view, the information about reuse of the shared work product is also useful in tracking the performance. This becomes difficult when stakeholders are active in separate work environments. Information management strategy therefore must recognize and establish cross environment traceability of work products. As all users should not be alerted for all types of work products; realizing separation of concern is important requirement for this feature.

How information requirements of the stakeholders can be identified? Stakeholders should be provided information not only suitable for their assumed roles, but also be provided information about required assistance and other information identified by the coordinating agency. Determination of information for specific individuals may be affected by many parameters. Information management strategy should be able to handle all the relevant parameters necessary to determine information need of the stakeholders.

Which new interaction patterns are required? Information needs are satisfied by interactions among the users and systems. There can be many types of interaction patters possible among them. Information management strategy should identify the interaction pattern suitable for entities in complex dynamical systems.

How dynamic set of stakeholders and their work products can be handled? The stakeholders and their outcomes can be seen as a dynamic set. Their presence, activity and absence and every state change may have implication on others and on overall system state. The information management strategy should be handled to minimize the impact of their absence, and maximize the utility of their presence.

How dynamic set of resources can be employed on the runtime? Based on the effectiveness of the employed information management strategy, the stakeholders are expected to deal with constantly changing set of roles, resources and activities. The information management strategy should be able to handle sudden availability or absence of the resources appropriately.

How information specificity can determined for different stakeholder hierarchies? Stakeholders assume specific responsibilities in system. The responsibility makes them accountable for specific tasks in specific area. For proper management of the activities, appropriate hierarchies are defined. Stakeholder working at various level of management hierarchy may require information at appropriate levels or granularity. The information may originally collected at a specific detail may later be transformed into other levels to suit the needs. Determination of all the required granularities therefore forms an important aspect of information management strategy.

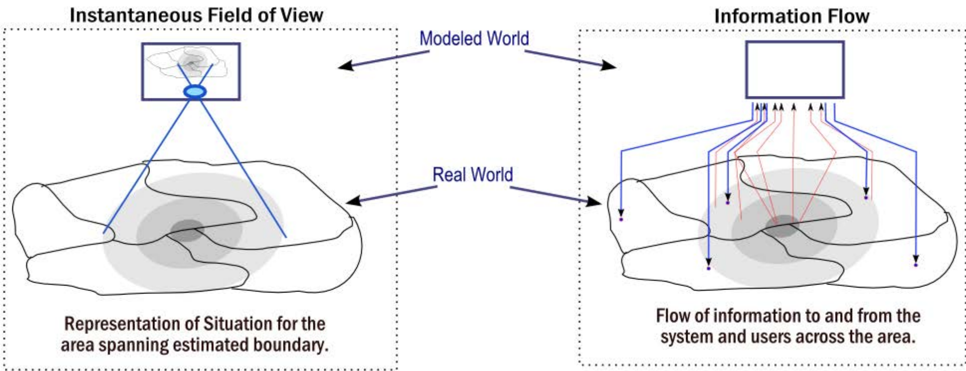

Figure 1.7: Instantaneous Field of View and information flow

1.3.2 Target User Group

In scenarios like a military conflict, disaster, international cooperation in peacekeeping, fighting terrorism, or protecting critical infrastructure, a unique nature of goal directedness is evident. Generally, the role of managing the individuals, groups and organizations independently contributing towards goal attainment is assumed by some coordinating agency. This is attained by providing a set of tools, resources, information, operating conditions and other specific requirements that may exist for carrying out the specific tasks to the involved stakeholders. The responsibility of information management is therefore to support the flow of information among coordinating agency and stakeholders as indicated in Figure 1.7.

The outcome of the research work is useful in building situation awareness for management of complex dynamical systems. Coordinating agencies involved in managing state of affair in domain like critical infrastructure protection and disaster management are the targeted users of this research work. As depicted in Figure 1.8, the information system of such agency acts as a *System of Systems*. The agency has to identify all the sources and sink of information and ensure flow of information among them. Depending upon the complexity and available infrastructure, the agency has to configure many information systems. They all may be created for specific purposes with limited scope. The challenge is to achieve required coverage by using existing systems, and making provisions of the missing ones.

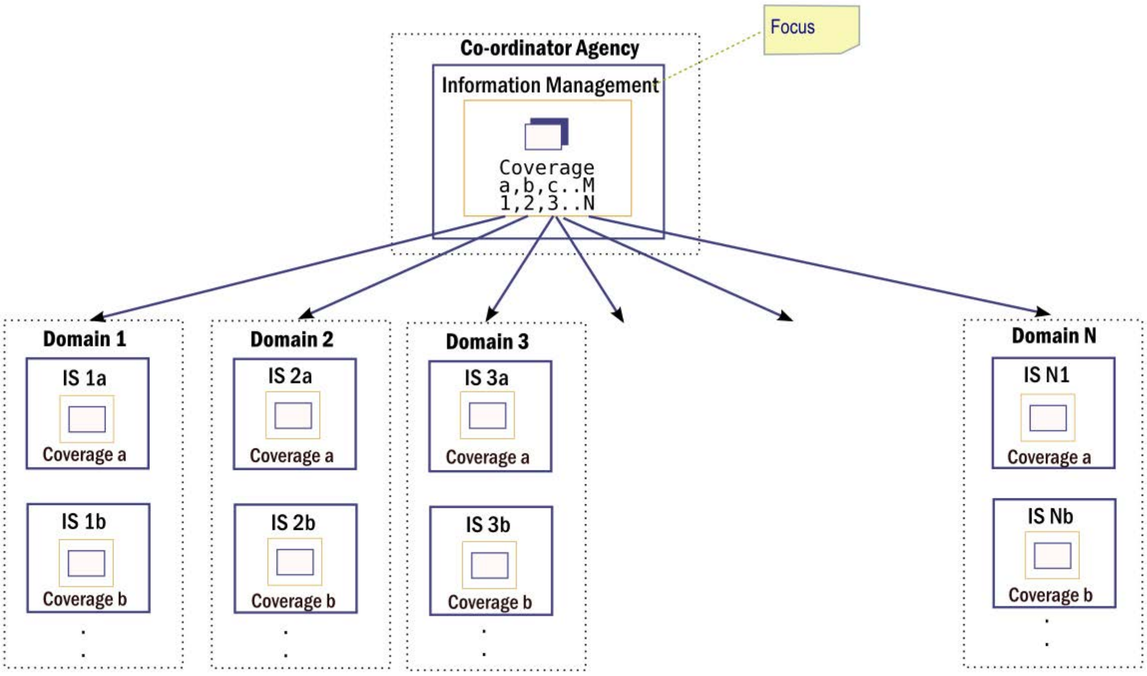

Figure 1.8: Research Focus

Coordinating agency can face multiple questions that designer of the soft system needs to answer while developing the solution.

- How can it connect to the stakeholder systems?
- How can it monitor all stakeholders and their services and work products?
- How can it detect event at all level and take reactive steps?
- How can it establish the interdependence among actor network?
- How can it connect to multiple distributed systems for information needs?
- How can it access repository of all possible kinds?
- How can it maintain, access and utilize various resources in resource pools?

1.3.3 Assumptions and Scope

It is a known fact that the collaboration of dynamic entities at such large scale can be addressed by many issues. The present research does not provide all-inclusive solution for potential problems. For manageability the scope is restricted to the design aspects considering some common issues discussed herewith.

Assumptions

This research is based on the assumption that coordinating agencies are obliged or motivated to adapt to novel systems that can help in large scale collaboration. Generally the target systems are expected to handle data that is classified as confidential and agency may or may not completely share the information to direct users. Except the plans made publicly available, governmental strategies may not be in favor of sharing the rules, operating procedures, priorities etc. While doing this research it is assumed that intergovernmental commissions are working towards the efforts of the governments opening up and sharing data, resources and processes. The research assumes that all parts of information are open for specific type of users. It is also assumed that they will be willing to base their all decision-making facilities on the proposed systems. The proposal is targeted at first identification of current practice in dealing situation awareness. It tries to understand the problem and proposes suggestive strategy to fix these problems that are in the domain of information system. This assumes that possible out come will be positive for the system.

- Coordinating agency is willing to adopt *Soft System Methodology*
- All stakeholders are source and sink of information
- All stakeholders are willing to participate in communication
- All stakeholders are willing to decide, execute and report actions accordingly

Issues Addressed

The objective is to sustain information flow among collaborators. For this, implicit and explicit status of the known entities can be found. Hence, a system that can support monitoring of all concrete and abstract entities and processes is required. The engineering of the domain middleware system, and creating a method to enable various users to customize, configure and use such system is therefore in the scope of the work.

- Conceptual modeling strategy to identify system, stakeholders, situations and needs

- Information processing strategy for role specific content
- Communication handling strategy to ensure information flow
- Method content to guide collaborative effort
- Domain middleware architecture to realize the system

The Scope of Work

This work relies on the representation of knowledge. But knowledge representation provided here is just an example hence the content completeness, quality and the accuracy is not covered in the scope. The proposed middleware is targeted to handle the real world observed data but any type of data collection, processing and analysis is not part of this work. Even though application is targeted to help organizations in monitor and executing their tasks, the work do not propose any management policy, standard operating procedures or similar artifact. Other issues that are out of the scope of this doctoral work are:

- Implementation for single issue for particular geographical area.
- Knowledge representation of all involved domains.
- Data sharing policy issues of organizations
- Misuse of information and other security issues
- Quality of service and performance issues of proposed middleware services

1.3.4 Contribution of this Thesis

Situation Awareness Domain Situation Awareness domain is extended for complex dynamical systems. Novel definition of situation awareness is established with emphasis on theories regarding individuals and organizations. It is argued that the conventional utterance of situation that is based on Boolean type - confirming whether the situation holds or not - may not be sufficient or possible for complex dynamical system. Hence, utterances of situation are governed by the system using types that accurately represent the entities and UoD. It is evaluated by the system to determine simple or complex situation.

Information Management Domain Availability of right information is identified with situation awareness. In order to support situation awareness, the required information management practice involves strategies for modeling, instantiation, communicating and processing relevant data among involved stakeholders and systems. The modeling strategy is proposed in the form of six conceptual and thirteen information-processing models that guide the realization process. With this information management approach, situation awareness is achieved with novel data access patterns realized for the users.

Method Engineering Domain For a complex dynamical system, UoD is classified into five stakeholder environments and traceability of activities among stakeholder environment is established. It is argued that every stakeholder activities must be appropriately represented with proper work breakdown structure. For each unit activity, guidance and provisions for allocation, monitoring, delivery and sharing should be coordinated to enhance consistency and reuse that are essential in collaborative environment. To support this feature, Situation Awareness Unified Process is proposed in the form of a process covering the content like roles, tasks, work products, disciplines, domains, delivery process and other conventional elements of method engineering domain.

Architecture Framework Domain Architecture frameworks are identified to bring important situation awareness capability in enterprise system environment. It is claimed that existing approaches can be extended with necessary design alternations to capture dynamism among the involved systems. Apart from enhancing the capability of known architectural products, the proposal includes novel products that establish traceability and gap analysis for the outcomes of various stakeholders involved in system.

Middleware Domain The realization of situation awareness capability is provided in the form of domain specific middleware services. This service allows identification and handling of transient resources that participate in information processing task to suit the complex interdependence and dynamism that may exist in given UoD. With the help of the automated generation, scheduling and handling of these tasks and appropriate configuration resolves the problem of wild card capabilities enabling middleware.

1.3.5 Organization of the Thesis

The thesis is organized in eight chapters.

The introductory **Chapter 1** discusses the complexities in real systems. Dynamism and complex interdependence are identified as characteristic feature of these systems. Conventional information management strategies for meeting information needs are reviewed for their applicability to dynamism and interdependence of systems.

Situation Awareness capability is recognized as an underlying requirement of information management and the theory of situation awareness has been studied in detail. **Chapter 2** characterizes situation awareness from stakeholder and organization point-of-views.

This characterization results in a unique approach for the modeling of elements and features of the given universe of discourse. **Chapter 3** introduces seven types of models for representing the information system capabilities.

Chapter 4 introduces Situation Awareness Unified Process (SAUP) providing guidance for the consistent modeling and realization in a collaborative environment. The approach provides interoperability and reusability of the models discussed in the preceding chapter.

Chapter 5 introduces Situation Awareness Architectural Framework (SAAF) - a set of architectural products that are targeted to aid software component development. The unified process and architectural framework together allow establishment of traceability and sharing of work-products among collaborating contributors.

Chapter 6 discusses a proposed system architecture that can support collaboration.

For demonstrating the applicability, a proof of concept scenario is discussed in **Chapter 7**.

Chapter 8 concludes the thesis with a description of the major research findings and directions for further research in the area of situation awareness targeted at complex dynamical systems.

1.4 Research Context and Motivation

Even in presence of syntactic and semantic interoperability and standards based systems; the individuals and organizations are still facing difficulties in meeting information needs. The enabling technologies that provided promising proof-of-concepts could not bring desired benefits when employed in large-scale collaborative efforts. For instance, shared Ontologies are reaching enormous amounts; web services and data services are exposed for open access; software libraries, process contents are freely available; yet all anticipated benefits are still not achieved. Many of these reusable outcomes are directly usable in newer projects, yet they are not utilized in absence of an appropriate mechanism. Hence, solution is not possibly in the isolated disciplines solving the problem individually, but it is in the strategy adopted by them in collaborative environment. With this realization, the present thesis introduces information management strategy that enables consistent coordination of interrelated stakeholder activities in collaborative environment.

Chapter 2

The Concept of Situation Awareness

The current research focuses on information management in complex dynamical system. A *Situation Awareness* capability is essential to provide effective information management for such system. In literature, the term *situation* is used in different contexts with varying objectives. The awareness of the situation therefore may also differ with its context. To address the issue of situation awareness, it is necessary to first define the theory and conceptual framework of situation, its awareness and related terms. As the purpose of studying situation awareness is to establish its role in solving the information management problems, it is necessary to investigate the relevant theoretical frameworks to characterize information, its management and related concepts. With explication of characteristic features of both disciplines, it will be possible to draw meaningful interchanges of theoretical concepts. These findings will be identified in the form of design principles, which will subsequently be adopted in the proposed information management strategy.

2.1 Background

The information management problem of selected domains can be addressed by first identifying the scope. The application domain is expected to manage some ***part of the world*** for the desired goal like disaster management, environmental management or critical infrastructure protection. Various entities can be identified in this world. The collection of various entities can be identified to form a ***system***. Many such systems can be identified, exhibiting various roles in given UoD. The proposed information management strategy is to be defined to solve the problem, by providing required information flow. The kind of information needs to be identified along with the sources, content and users. These users are ***individuals*** who are going to use the information in carrying out activities. Apart from that, there can be ***organizations*** that are involved in providing various products, services and activities. These organizations are unique entities in UoD, exhibiting specific behaviors, characterization of which is to be carried out separately. In given complex dynamical system, the status of the system can be determined by evaluating status of system elements and the responsive action from individuals and organizations. Hence, at any given time, status of these elements can be found to build a ***situation***. Study of situations is therefore also becomes imperative. The present discussion is therefore dedicated to identifying and characterizing systems, information, individuals, organizations and situations in given Universe of Discourse (UoD) to aid the information management strategy.

2.1.1 Systems

UoD can be visualized as a *system of systems*. Each system can be identified to possess specific characteristics. The components that are part of one system may also be found to play role in another system. It is possible to estimate the boundary of the impact of a system, and can be taken as single entity interacting with rest of the systems and its environment. The status of system can also be determined, along with state of their constituent elements. Based on the interactions, system can be found to have impact of internal and external elements in the environment. As identified earlier, the systems also demonstrate evolution and respond continuously to changes. Hence, it becomes necessary to study the systems to reveal possible types of systems, their characteristics and behavior to support the desired management control.

The systems can be seen from various domain viewpoints and can be classified appropriately. A UoD may contain systems that can be roughly classified to natural and man-made systems. Natural systems involve natural entities and processes and governed by the laws of nature- many of which are now established by the scientific community. Man-made system on the other hand, involves material resource, human processes and artificial conditions like business environment. They are based on the laws of the applied sciences and engineering disciplines. The business, industrial systems, machines etc. are man made system. With interactions of the man-made systems with natural systems for their dependence for inputs and environmental conditions, have deep impact on natural systems. Similarly, the products, services and processes involve many human beings, introducing social aspects in to the system. Hence, systems can be seen from all possible point-of-views so that their behavior, characteristics, and interactions can comprehensively be modeled.

System Behavior

System modeling is a widely addressed topic in scientific and research community. Different types of systems are identified and characterized to serve unique objectives. Theories are established to model their behavior. The entities in the system and interaction among them affect the overall system states. It is therefore necessary to capture all relevant entities, properties and behaviors in the given system.

In identified systems, individual are found to be active as agents. These agents are continuously interacting with each other. The systems represent emergent behavior. With interventions of agents and emergent behavior, system is continuously experiencing state transitions. The single control is not visible in the system, and involved stakeholders are found to determine course-of-action as spontaneous response. In contrast to the industrial and man-made systems, where actions, processes, and control results in definite outcome, in given scenario do not necessarily demonstrate the response in desired manner as a result of numerous variables.

Boccara [5] identified three properties that qualify the given system as a complex system. As indicated in Table 2.1, the presence of multiple interacting agents, emergent behavior and absence of central control is discussed to be present in the problem domains defined in Chapter 1.

	Property
1	Involve large number of interacting agents
2	Exhibit emergence
3	Emergent behavior in absence of a central controller

Table 2.1: Properties of Complex Systems [5]

The system can be considered to be in various states depending up on the status of its individual elements. At any given time it is possible to consider the status of the system. Systems with these features are identified as dynamical system[5]. Systems prevailing in the domain application selected for this study therefore also qualify as dynamical system.

	Ingredient
1	Phase space $\mathcal{S}$ (Set of all possible states)
2	time t (Discrete or continuous)
3	Evolution law (A rule for determining state at t from knowledge of previous states)

Table 2.2: Ingredients of Dynamical Systems [5]

Features of complex systems and dynamic systems are identified in given problem domain. Yet these systems are studied with limited environment with precise mathematical model. The system present in UoD can not sufficiently handled with mathematical models defined in [5] for larger system. Schellnhuber introduced [22] problem of complexity and dynamism; and termed the system as *Complex Dynamical System*. The mathematical modeling framework is also proposed that address the modeling of such system.

Design Principle 1 (Nature of the System)**.** *System is considered as complex dynamical system as defined by Schellnhuber [22].*

So far the basic system features have been identified. The ext task is to determine behavior and response of the system. The systems in UoD can be found to react to various changes exhibited by involved entities and the environment. The systems also exhibit gradual evolution. It is necessary to determine the factors affecting the evolution and reaction to various internal and external factors. The punctuated equilibrium theory[6] established that system tend to maintain an equilibrium state in existence of various forces. In general course, the system components slowly and steadily incorporate changes in structure to demonstrate evolution. When sudden massive event takes place, the system undergoes major changes to meet the emergent needs and soon come back to equilibrium state of steady growth. Thus, this theory establish that system respond to changes in specific manner, and always tries to achieve the equilibrium state. The long period of slow and gradual evolution is known as "stasis" and sudden massive change is identified as "punctuation". This is a useful result in designing strategy for the reactive systems. This helps information management strategy to determine course of action in normal (stasis) and special (punctuation) conditions as introduced earlier and depicted in Figure 1.1.

Design Principle 2 (Nature of Reaction)**.** *Punctuation is identified as source of event requiring considerable interventions to bring system in balance state again.*

System Elements

The overall system behavior is identified with relevant theories that also indicated existence of various internal and external elements of the system. These elements should be identified and characterized appropriately to maintain the desired states.

Systems have been identified as complex dynamical system. Yet, system elements need to be studied. There can be many components visible in the system. The next task is to specify the elements of the system that can be used for management purpose. Problem is in identification of all the relevant components that may affect the system states. Some of these elements are concrete entities, and others may be abstract or conceptual entities. The general systems theory[25] identified following elements of a system.

	Element
1	system-environment boundary
2	input
3	output
4	process
5	state
6	hierarchy
7	goal-directedness
8	information

Table 2.3: Elements of System [25]

Design Principle 3 (Elements of System). *Each types of system elements identified in general systems theory should be treated as potential source and sink of information.*

General system theory allowed identification of basic elements related to the system. For given application domain, the system includes complex man-made systems that contain elements and behaviors that are unique to the systems. The next task is to identify the elements of various man-made systems introduced earlier. From simple machines, plants and instruments to complex critical services, the man-made systems provide challenging mix of natural and man-made elements that makes the modeling a difficult task. These systems can be seen as a collection of heterogeneous entities and resources are involved. The problem is in taking decision regarding the membership of such entities and resources in to the collection. According to the actor network theory [7], all living and non-living things create a heterogeneous mix to render a specific service are considered members to form a system in a manner that any missing component will affect the status of overall service. These elements are collectively identified as actor network. While modeling the entities, and behavior, their actor network provides identification of a system by recognizing elements considered in modeling. Secondly, it also allows ramification by establishing other system elements that will affect if given entity will cease to exist.

Design Principle 4 (Grouping of Related Entities). *Actor network theory should be applied to determine boundary of a complex service as a single unit. This also guides the determination of ramification if any component undergoes state change.*

General system theory and actor network theory provided the identification of elements in natural and man-made systems. The application domain considered for this study includes complex dynamical system that requires identification of further system capabilities. In case of critical situation, issues like complex interdependence, dynamism and response become priority for the management of the system. Comfort [2] recommended seven measures for capturing this type of dynamism. Inspired by Kauffman's N-K system theory applied in biological system, Comfort extended this model for utilization in information management domain. Table 2.4 enumerates the identified features.

	Symbol	Measure
1	N	Number of participating actors in the system
2	K	Estimated number of interactions among actors
3	P	Shared goal of actors
4	B	Boundaries of the system
5	R	Response time of units
6	D	Duration of interactions
7	T	Types of transactions performed

Table 2.4: Elements of Complex Adaptive System [2]

Design Principle 5 (Elements of Complex Dynamical System)**.** *Complex interdependence among system elements and dynamism is identified with seven measures recommended by Comfort [2].*

System Evolution and Control

Systems and the elements with their unique features are identified with relevant theories. The proposed information management strategy is planned to support system evolution and control. Hence next task is to determine the appropriate theories concerning the evolution and management of the system. The theories should be identified that address the management issue in natural, individual and organizational setting. Achieving the control of the system to maintain the desired system status is an important feature for system management. Various issues related to control in the complex dynamical system needs to be identified.

If the system is seen from environmental management point of view, the management and control is achieved with the term *geocybernatics* that is a discipline devoted to appropriately controlling the complex dynamical earth system under all kinds of uncertainty. According to the geocybernatics concept, the control problem is defined in the form of three questions as indicated in Table 2.5.

Level	Question
1	*What kind of world do we have?*
2	*What kind of world do we want?*
3	*What we must do to get there?*

Table 2.5: Control Problem in Complex Dynamical System [22]

Simple questions represented in 2.5 provide requirements that can be difficult to achieve in complex dynamical environment. For example, the question: *What kind of world do we have?* requires comprehensive information about elements, their interrelations and characteristics that they exhibit over period of time. Similarly, the second statement identifies the requirement of goal state that implies the knowledge of desirable properties of elements in the system. The third statement requires information about plan to achieve that state. In summary, this allows identification of goal statement of information management strategy in complex dynamical system.

Design Principle 6 (Control Strategy in Complex Dynamical System)**.** *Schellnhuber's Control problem [22] is to be considered as goal of information management strategy.*

Schellnhuber's control problem [22] introduced the requirement of a plan to reach the desired status in the system. The next task is to address the selection of problem solving strategy suitable for complex dynamical system. Due to the nature of complex dynamical system, any management practice carried out may not result in desired outcome. The outcome can be affected by many parameters. Hence problem-solving strategy appropriate for complex dynamical system needs to be identified. Soft systems methodology[18] identifies such problem as soft problems and suggests a seven-step process for solving soft problem. Table 2.6 enumerates the sequence of steps to be followed.

Step	**Details**
1	Enter Unstructured Problem Situation
2	Express Problem Situation
3	Formulate Root definitions of relevant human activity systems
4	Build Conceptual models from root definitions
5	Compare models with real world
6	Define Desirable and feasible change
7	Take action in problem situation

Table 2.6: Steps in Soft Systems Methodology[18]

Design Principle 7 (Management Strategy)**.** *Management strategy guiding overall information management task is based on Checkland's soft system methodology [18].*

2.1.2 Information in Systems

Prime focus of information management is the concept of *information*. The management strategy therefore should appropriately recognize the term - information; along with its attributes, role and related features that affects the information management task.

Information

The dictionary definition of ***information*** identify the term as *"facts or details that tell you something about a situation, person, event etc"*[26]. In literature, information in defined vividly according to how it is perceived, handled and used. It is considered as outcome of data, signal, message or sign available in meaningful form. Braman suggested a hierarchy of information definition based on the role information plays[3]. When organizations utilize, process and handle information for various stakeholders as disconnected entities, the information is considered as *a resource*. When access to such information to various users provide them competitive advantage or power, the information is considered as *a commodity*. When information is enriched with context, found to have temporal dimensions and is affected by motive and changing environment, it is considered as *a perception pattern*. Going to a one level up, when information itself is used for creating context, and behaving as an actor responsible for change in environment, it can be considered as *a constitutive force in society*. While value of information is determined based on how it is employed, the form in which it is handled also helps determination of processing.

Apart from definitions, the ***attributes*** of information is claimed[27] to be more important for the understanding. When information is available as commodity, the cost of information is an important factor that is identified as commodification attribute of information. Information has a value attribute that can be variable for different users. Similarly, other attributes of information are: reliability of content, reliability of source, time, generality, novelty, subject domain, specificity, clarity, amount and instructional value.

Based on the attributes, the ***nature of information*** can be identified. Descriptive nature of information define the world the rules that govern various processes in it. The probabilistic nature information is model from limited observations to determine possibility of event. Explanatory nature of information can be in the form of unstructured text that describes the phenomenon. Unexpected nature of information can be the form that is undefined to the organizational users. Organizations may use Propaganda information to build attitude belief and behavior or stakeholders [28].

As information with specific attributes, forms and nature are identified essential for proper decision making, the absence thereof is considered impairing [29]. The kind of information required is known as ***information need***. It is determined by two factors: Uncertainty and Equvivocality. The uncertainty is lack of information required to execute the task. The equivocality is disambiguation about the content of the information. Addressing these two problems results in determination of information need. Apart from these two factors, critical success factors[30] are also considered as source of information need.

Information may be generated from many sources. Out of all the handled information only some part is relevant to decision maker for a specific decision. First, for information to reach to the user, the sources and destinations must be connected. Secondly, chunking and processing of information to identify only the required set should be done intermediately. For this, the information passes through a sequence of steps before it finally arrives to the decision maker. This transfer is known as ***information flow*** [31].

Information Management Purpose of information management is to help organization attain its goal [3]. It is realized in context of the organization. Information as an asset and goal of management is to maximize it use People who want to use, have to manage. Professionals and common people come and share. Organizations really benefit from information management.

Information management is define as a process of applying management principles in acquisition, organization, control, dissemination and utilization of relevant information organizational activity [32]. This general definition is further extended with reference to strategy of information use. For a given organization, information can be identified as commodity, resource, perception pattern or constitutive force based on its use. The management of information therefore affect significantly based on which role of information is adopted. Kirk [3] recommended four level of information management strategy based on each role of information. Table 2.7 indicates the information management approach for each level.

Information	**Information Management**
Resource	IT system
Commodity	Information resource management
Perception Pattern	Aligning Information Strategy and Business Strategy
Constitutive Force	Integrating Strategy formation and information formation and information

Table 2.7: Information Management Definitions [3]

Another related term Environmental Scanning - an activity of capturing and using information about events, trends, relations in external environment carried out in order to assist management in planning the future course of action [33]. Information gathered by environmental scanning is used to conduct communication, shared vision, strategic planning and management, future orientation with an objective to cope with the external change.

Information Management Practice There are numerous activities that lead towards management of information. They are carried out at different level of information abstraction by actors with unique objectives and methodologies. Classification of these activities and their interrelations are suggested to improve comprehension of the discipline. Wollnik's three level model of information management [1] is one such classification that reveals levels interrelated activity hierarchy. Top level is level of information use. Middle level involves ***Information Systems*** and lower level involves infrastructure that support them. Beyond this simplistic model of information management, a detail classification of activities is also available in literature. Schlogl [1] suggested one such detail classification by identifying management activities as Technology oriented, content oriented and knowledge aspects of information management.

Technology oriented information management highlights technology aspect of information. It is realized with computer based Information Systems by carrying out data management, Information Technology (IT) management and strategic use of IT. As data is utilized and generated in most of the stakeholder activities, the management of Data amounts to be a major concern in information management. The management of data involves data planning, accountability, policy, standard support and related activities known as data administration. The database administration activities like design, monitoring optimizing, tuning and securing are employed to manage data at physical level. The technological aspects like hardware, software and people related involved in handling data are identified as IT Management. The information system level and information infrastructure level in Wollnik's three level models are identified as IT management. Strategic use of IT is identified with activities that align the business activities in information management.

Another important approach is ***content oriented*** information management that have roots in library and information science. In this approach, the content is the prime focus of the management. The activities include records management, provision of external information and information resource management. As the information content is expected to influence behavior of users, the human behavioral aspect is also incorporated in this strategy. As the use of information content and technology increased among various actors, both were gradually found to involve special features. Learning from the behavior of actors, their use of information for decision making, and other interactions with the systems, the rules are identified to facilitate generation and provision of new information. This automated approaches extended capability of technology and content oriented management of information and are identified as ***knowledge management***.

Summary The term *Information* is defined along with its various characteristics and related concepts. In this section, the concept is revisited to reveal its nature and role in context of Complex dynamical systems. In theories that address controlling of such system, information is identified as common requirement. While general system theory just started information as an element of the system, the Soft System Methodology entrusted central role to the system in controlling the system state. The creation and use of rich picture identifies considerable requirement of information, its processing and handling. Three questions defining the control problem in Table 2.5 also indicated central role of information for answering the questions. The objective of this section is therefore to identify how information, information need, its communication and processing needs to change to meet the requirements in complex dynamical systems.

Information Need

Information need is identified as amount of information required by a specific actor to complete the assigned task [34]. The information need is identified with two important concept in the domain namely Uncertainty and Equivocality [35]. *Uncertainty* is absence of information required to accomplish the task. *Equivocality* is the ambiguity generated by the information. A proper solution should address both these problems to make sure the information needs of the user are successfully met.

Information Communication

Information theory is one of the most seminal work published[36] that have influenced the developments in information and communication disciplines. Figure 2.1 represent the basic model of the theory, depicting basic element of a communication system.

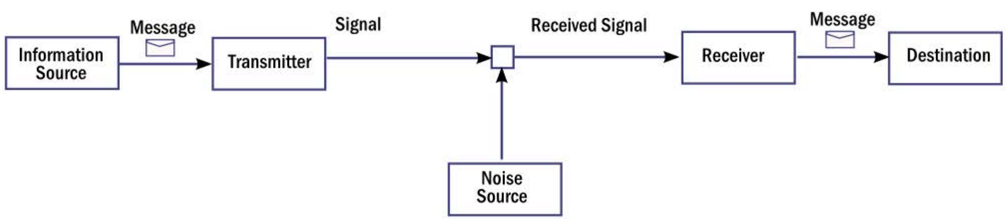

Figure 2.1: Information Theory Model

Apart from the basic model, the theory[36] also characterize problems in communication of information. Table 2.8 indicated three levels of communication problem along with the problem statement. Issues in communication are identified. Purpose of communication should now be established. The purpose of communication is to furnish information as and when required.

Information and communication is discussed in basic models. The next task is to identify how information is transferred among entities and systems. Such exchange of information is

Level	Problem Type	Question
Level 1	Technical	How effectively symbols can be transmitted?
Level 2	Semantic	How precisely the transmitted system convey the intended meaning
Level 3	Effectiveness	How effectively received meaning affect conduct in desired way

Table 2.8: Three level of Communication Problem [36]

known as Information flow. Depending upon the role of transfer, various types of information flow are identified [37]. From information management point of view, the information interaction patterns should be able to support the required information flow among the collaborating users.

So far, basic concepts of information and its flow are introduced. The next task is to determine the appropriate information dissemination strategy to reach the intended mass. An effective medium of communication should be identified for this purpose. Media richness theory introduced factors for determining effective of media selected for communication [38]. These factors are to be considered while taking decisions regarding medium of communication.

Design Principle 8 (Choice of Medium)**.** *Selection of available medium of communication based on factors established by media richness theory.*

Information Handling

When additional information is required, the new sources of information need to be identified. In a large-scale automated system, identification of all the required information sources can be a difficult challenge. An information source may provide only a part of required information yet, it can be critical for the execution of specific tasks. Identification of such small contributors from available stakeholders is difficult. Transitive memory theory suggests that individuals store memory in particular way. They have short term working memory and long-term storage memory. Apart from this, they also have "knowledge about knowledge" in the form of who knows what. Hence, members of organizations work as external memory instruments of others.

Design Principle 9 (Knowledge Reference Strategy)**.** *In selecting potential sources of information, transitive memory of members should be utilized.*

So far, sources of information, utilization of information, information flow and related aspects have been identified. The next task is to manage involvement of individuals as potential source and sink of the information. The strategy should identify the capability of individuals handling information. According to Miller[39], the individuals have limited information handling capacity. Individuals can process seven plus or minus two small chunks of information at a time. In case of additional information, either it is transferred to long-term storage or it is lost.

Design Principle 10 (Information Chunking)**.** *Cognitive load of individuals should be reduced according to information processing capability established by Miller[39].*

While individual's information processing capability is identified, the capability of organizations can be identified as well. Next task is to determine how organization information processing capability can be established and appropriately integrated in decision-making process. According to organizational information processing[40] theory, organization's capacity is determined by, information processing needs, information processing capability and the fit among them.

Design Principle 11 (Provisioning Compute Facility)**.** *Information processing capability of the organization should be able to meet the identified information processing needs of the organization.*

Information processing capability has been identified for individuals and organizations. The next task is to identify and support information exchange patterns. The strategy must be able to identify and support various information exchange patterns possible among stakeholders and environment. Among the known patterns, the ***Information Seeking*** behavior is observed when individuals identify what information is required to meet information needs, and actively ask the information from the sources. ***Information processing*** behavior, is observed where individuals are passively processing information available in the environment. These common behaviors should be appropriately employed to meet the needs of masses and active publics.

Design Principle 12 (Required Pattern identification)**.** *Various interaction patterns like information seeking and information processing are appropriately supported.*

So far, information is discussed with general features, without mention of the quality. In order to be useful, information content should provide appropriate details. The difficulty is in determining the level of details to be handled. Information specificity[41] indicates the level of detail at which information is handled. It is also known as granularity.

Based on the organizational capability, information management strategy should allocate task of environmental scanning for obtaining all the required granularities.

Design Principle 13 (Granularity)**.** *Multiple granularities of information is to be determined and appropriately handled by the system.*

2.1.3 Individuals in Systems

Individuals are critical elements in the system. They need to be characterized and managed appropriately by the strategy. Types of individuals, their characteristics, behavior, needs, role, impact on the system and factors affecting their behavior should appropriately identified.

Behavior

Individuals are expected to exhibit planned behavior in given UoD. The behavior is important to achieve for the control of the system. The behavior can be influenced to achieve desired control in the system. The task for information management strategy is therefore to provide appropriate information to trigger planned behavior.

Design Principle 14 (Influencing Individual behavior)**.** *Strategy should communicate to individuals to derive planned behavior.*

User may require information to conduct the assigned task in given system. The next challenge for the strategy is to provide appropriate information content to the user. The purpose of information provision is to improve performance of the user. According to cognitive fit theory [42] the presentation format of task and information has impact on performance. Various forms used in such presentations allow creation of rich internal representation of the problem. This along with the task specification improves the problem solving capability of individuals. Information management strategy for individuals can benefit in determining the content of information required by individuals in complex dynamical systems.

Design Principle 15 (Problem Identification)**.** *Provide information about the problem to achieve a clear cognitive fit to the potential actor.*

Role of appropriate information is identified for individual performance. The next task is to ensure that proposed information technology based solution is successful in achieving individual performance. Hence, effectiveness should be established for the proposed strategy. Task technology fit [43] theory addressed task technology and individual characteristics to improve the performance. Theory identified the quality, locatability, authorization, compatibility, ease of use, production timeliness, systems reliability, and relationship with users as critical factor for the success of the proposed technology framework.

Design Principle 16 (Technology uptake)**.** *Provide the technology and related guidance suggestion to the individual to improve productivity for allocated task.*

Systems include individuals and agents that exhibit some behavior. Next task is to control their behavior, in order to keep them consistent with goal of the system by providing proper information flow. The challenge is in determining the information flow to ensure required behavior from agents. According to the self efficacy theory [44] individuals choose the environment of the area where they belong to. Their behavior in given situation is governed by situational characteristics that affect the environment. It is also govern by their judgment of their capability to organize and perform the required course of action. This finding allows the coordinating agency to identify individuals as potential agents that can be utilized for keeping the goal statement. The individual's capability can be improved by providing proper training in using tools like computers in solving their problems. Also, the access to required course of action also improves coordinated effort from multiple individuals.

Design Principle 17 (Improve Technology Usage)**.** *Allotment of task and required information should be based on the self efficacy of the individuals.*

It is now identified that, individuals with recognition of problem and self-efficacy engage in activities. The next requirement is to track the individual activity. The challenge is in determining appropriate task monitoring strategy. Task closure theory [45] suggest that task completion is reported with end of communication. Actor may be select most effective medium to report the task closure. It also discussed utility of asynchronous communication to reach intended audience. This is a useful result for information management strategy as it first establishes the fact that individuals tend to finish the task with reporting it. Second, important observation is that they use most effective methods and medium of communication to reach intended audience. This ensures that in complex dynamical systems, collaborating individuals will use the communication strategy if they are assured about outcome of their report. This will provide them recognition about their contribution. From coordinating agency point-of-view, it will enable to solve many task allocation related problem like duplication of effort.

Design Principle 18 (Task Closure)**.** *Task closure feature should be incorporated by the strategy with provision of effective communication.*

Evolution

Evolution of contribution for required tasks is observed with appropriate information content provided to individuals. Several theories have established the factors governing the evolution among the action of the contributors.

Design Principle 19 (Information Provision for Action)**.** *Individuals should be provided situational information to improve their contribution.*

It is identified that individual behavior is influenced for triggering required action. In every punctuation scenario, the organization must make provision for information to influence the behavior of potential actors. But the potential users are not known. Hence, they must be identified from the masses every time the punctuation takes place. According to the situated theory publics [20], different types of actors in masses exhibit unique behavior. The theory suggested six variables that determine the membership of each person from the masses to a user category. Table 2.9 enumerates two communication behavior and four situational perception variable to characterize masses. The most favorable person exhibits high problem recognition, low constraint recognition, high level of involvement in the problem and have some referent criterion. They are designated as "Active Publics". The challenge for the organization is in imparting the capability to the potential users as part of planning process. As all six criteria

Type	**Type of Variable**	**Question**
Communication Behavior	Dependent	Information Seeking
	Dependent	Information Processing
Situational Perception Variable	Independent	Problem Recognition
	Independent	Constraint Recognition
	Independent	Referent Criterion
	Independent	Level of Involvement

Table 2.9: Situational Theory of Publics [20]

involve communication and situational perception, it comes in to the purview of the information management strategy.

Design Principle 20 (Information Provision Strategy for Respondents)**.** *Information seeking behavior of the respondent should be handled by the information management strategy.*

Design Principle 21 (Information Provision Strategy for Masses)**.** *Information management strategy should make provision for appropriate information processing behavior for masses.*

Design Principle 22 (Strategy to Derive Reasoned Action)**.** *Information management strategy should be able to establish reasoning for desired action.*

2.1.4 Organizations in Systems

The systems consist of another type of entity that is collection of individuals collaborating to provide specific product or service. Individuals, along with their products can be visualized to create a system within systems, with interactions with other individuals and the environment. They exhibit specific behavior and respond to changes. The membership to organization can be implicit or explicit. Same individuals and resources may participate in one or more organizations. The organizations themselves are governed by mission statements and goals statements. They evolve and respond to changes presented by other organizations and environment.

Since, they play valuable role in systems, it is necessary to characterize organizational features. To identify and establish features that can be managed to keep them in desired status.

This section discusses the factors that govern the behavior of organization. It identifies the factors that can be managed to evolve and protect the organizations from punctuations. It also addresses the tracking and controlling feature for the actors.

Elements

First it is necessary to determine elements of the organizations. Types of elements can be identified and characterized to facilitate the monitoring and management task.

According to adaptive structuration theory[46], organizations dynamically create perception about the role and utility of the technology, also indicating how that can be applied to their activity.

Design Principle 23 (Building Technical Processes)**.** *Technical structures should be built to fit the tasks relevant to the organization.*

Design Principle 24 (Identification of Sub Systems)**.** *The strategy should have plans for social, technical and environmental subsystems for given problem as source and sink of information.*

Stakeholders should be identified for given organization. Identification of all the stakeholders is critical task in information management. The theories regarding identification of stakeholder and their features are therefore important for information management strategy. Stakeholder theory [47] proposes criteria for determining the membership to organization. According to this theory, potential stakeholders have power and urgency of claim, legitimate according to moral, legal, property-based relation.

Design Principle 25 (Determining Who are Relevant)**.** *Information management strategy should plan to identify and cover all the stakeholders for their information needs.*

Behavior

Behavior of elements has important implications on information management task. Also, the information being provided to the individuals also contribute in changing behavior of the systems. Relevant theories therefore should be identified and incorporated in information management strategy.

Resource based view of firm[48] establishes that organizations create resources to gain competitive advantage. They can handle to do so until their resources are not imitated

Design Principle 26 (Work Product Tracking)**.** *Information management strategy should track work product types, related processes, actors and conditions.*

Resource dependence theory[49] stated that organizations collaborate with others for their resource needs. In such relationship more dependence equals less power.

Design Principle 27 (Resource Dependence Tracking)**.** *Strategy should be able to draw resource dependence clearly to facilitate ramification.*

Individuals and organization have been studied having commitment to specific goals. The next task is to determine factors that affect what one organization allocates task to other individual or organizations. According to agency theory, such task allocation may have difficulty when common understanding of task entrusted is not maintained. Also, such conflict may result from difference in perceiving risks involved in the task. In information management approach, to prevent this situation, it is required to use consistent method and contents to determine or establish same risk perception.

Design Principle 28 (Goal Specification Matching)**.** *Strategy should be able to provide explication of goal specification to facilitate precise goal matching.*

So far activities and resource creation is identified for organizational environment. The next task is to determine the movement of generated resources. Problem is in determination of factors that govern the sharing and reuse of the resources created by actors. The act of sharing will reduce competitive advantage and reuse of imported artifact will show as dependence.

Social exchange theory [50] establishes that they give to incur cost and take as reward in response that can be in the form of goods, prestige or approval. This give and take reaches equilibrium and determines movement of goods. Information management strategy should be able to establish individuals playing their role by sharing of work products and in need of resource thereby taking and giving both at the same time.

Design Principle 29 (Tracking Resource Movement)**.** *Strategy should be able to track movement of resources among actors.*

Information about business resources etc. are captured and treated appropriately. Similar to resource, the knowledge should also be captured, used and shared. Multiple entities including organizational culture and identity, policies, routines, documents, systems, and employees carries knowledge in different ways.

Knowledge based theory of firm [51] suggested that knowledge is one of the resource created by the organization, should be treated separately as there are unique issues attached to it. Knowledge management is important part of information management, and should be addressed by the strategy.

Design Principle 30 (Encoding Knowledge)**.** *Knowledge created in organization in forms that should be encoded properly to enable sharing and reuse.*

Knowledge is identified as important resource created that gives important competitive advantage. The next task is to identify source and sink of knowledge. Problem is to determine source of knowledge creation and its use in given organization. Organizational knowledge creation theory [52] establishes that organization create knowledge as a continuous ongoing dialog among actors. It captures tacit knowledge from experts in organization. This tacit knowledge is encoded and known as externalization. This generated knowledge is used by others in organization to carryout the task, hence expressed explicit knowledge is converted back to tacit knowledge. Information management practice targeted for knowledge management should take knowledge creation is continuous dialog between tacit and explicit knowledge.

Design Principle 31 (Externalization Strategy)**.** *Facilitate Externalization of knowledge in actors by providing method to encode and share content.*

Design Principle 32 (Internalization Strategy)**.** *Facilitate internalization of shared knowledge in actors by providing encoded knowledge to support actions.*

Evolution

So far, the creation and utilization of knowledge is introduced for organization environments. The next task is to identify the level up to which the organization can create and handle knowledge, and when it will resort on sources outside the organization. To draw the exact boundary, what organization should handle, and what should be dependent, as dependence means less power. This problem is supported by the absorptive capability theory [53]. According to this theory, organizations have limited capability to incorporate innovations and new knowledge. This is affected by prior knowledge, transfer and this capability is improved with individuals getting training. Information management has knowledge management aspect and organizations are found to handle knowledge. Challenge is in determining the scope of knowledge creation with internal Research and Development (R&D). Also, external sources may provide the knowledge along with data.

Design Principle 33 (Establish Boundary)**.** *Absorptive capacity for given organization is to be establish to draw boundary for knowledge creation.*

Organizations create and handle knowledge, exchange information for superior performance. The next issue is to prepare the organization for dynamic changes that happens inside and outside the organization. Dynamic capability [19] theory suggests that dealing with punctuations requires the organization to have prior preparation. This includes creating alterative back-up plans, product design etc. Information management should establish the provisions made for possible dynamic scenarios.

Design Principle 34 (Process Alternatives)**.** *Make provisions for handling pool of process alternatives to establish dynamic capability of organization.*

So far it is observed that created knowledge is internalized by others and resulting in benefit for the tacit knowledge within organizations. There are organizations continuously involved in research and development and produce innovations. The challenge is in reducing the time delay for incorporation of innovations relevant to the organization in procedures. Diffusion of innovation [54] identifies ease of use, relative advantage, compatibility operability and complexity as factors for diffusion of innovation. Information management strategy should utilize these results in improving the uptake of new innovations.

Design Principle 35 (Uptake of New Knowledge)**.** *Strategy should track and identify the innovations that can be immediately incorporated in organizational processes or decision making to reduce the potential delay of uptake.*

2.1.5 Situations in Systems

So far systems individuals, organizations and information flowing among them is discussed. Depending up on the nature of the surrounding, each of these elements is found to react and exhibit actions and state changes. In order to get instantaneous status of some interrelated entities, a comprehensive collective view is required. This view is identified as Situation. It is useful for comprehension of on going state of affair. There can be many situations identified in given UoD based on the selection criteria used that provide the context of the situation and restricts the entities and their states to manageable few. It can be said that situations are the view points or reports that describes part of the world that is very useful for a decision maker in tracking and managing the ongoing state of affair.

Dictionary definition of situation defines the term as *"'..a combination of all the things that are happening and all the conditions that exists at a particular time in a particular place..."'*. This is indeed a useful explanation as it defines the scope of information that is sufficient to comprehend the on going state of affair. For the information management strategy, identification, characterization, handling and provision of situation is therefore an important feature as it provides the qualitative and quantitative estimate of information that is useful for specific user in meeting the information needs.

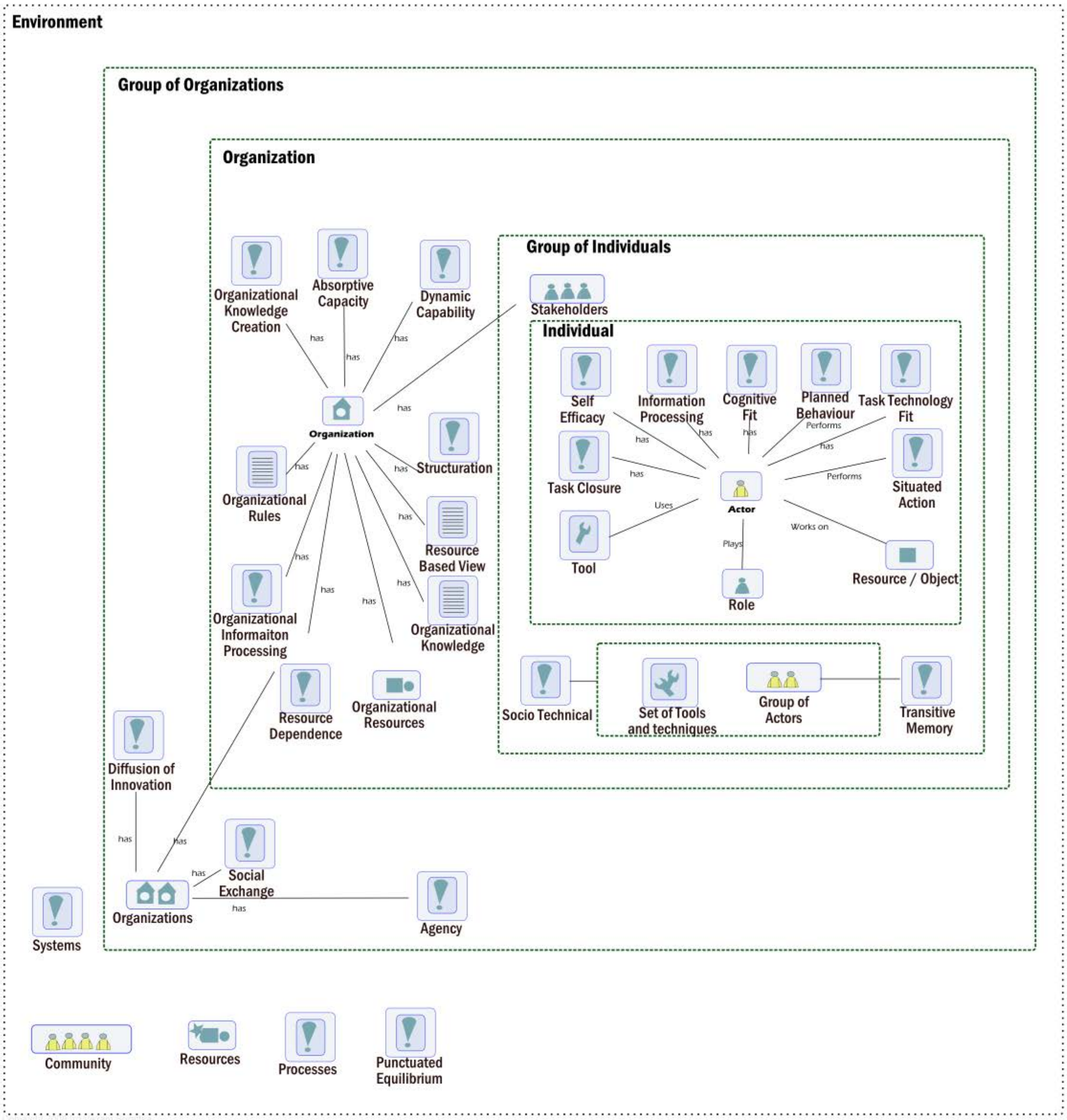

Figure 2.2: Information System Theory Summary

Design Principle 36 (Situation as a Scope)**.** *Use situation as a reference to restrict quality and quantity of information required by a specific user.*

As situation is expected to play a vital role in information management strategy, it is important to identify various features of situation. For example, different types of situations can be identified in given UoD. It is perceived by stakeholders in different ways. Characteristics feature of situation of each type, and how these features are affected by complexity and dynamism in the system. From information management point of view, how situation can be created, tracked, managed and provided to appropriate users.

The situation is discussed in early works of Barwise and others [55] and [56] in context of formal logic, language and semantics. Later the work is extended by Devlin [57]. The referred world is considered as set of objects, properties and relations. Objects not only possess properties but they stand in relation with other objects. Situation described as a limited part of the world that is used by various actors using form of a language. The natural language is used to provide information about the world for utterance or declarative sentences holding facts about the world. The basic unit holding the fact about the world is known as ***infon***. Barwise proposed situation semantics to address this practice and addressed situation theory as mathematical theory of meaning. Individuals, properties, relations, spatial and temporal locations, and situations form the basic constructs of the proposed theory.

Situation is perceived as an object representing part of the world. The theories defined the semantics and mathematical model about handing the situation. Next task is to use situation in information management strategy by considering it as basis of information exchange among various users. In given complex dynamical system, the scale and the nature of task may increase manifold. Situations change dynamically and may not be handled merely by language of natural communication. For using situation for feedback control loop type of control problem, the situation awareness should be appropriately defined.

Endsley defined[21] situation awareness as perception of elements in environment, comprehension of their meaning and prediction of future outcome. This definition identifies capability in three levels. Perceptions of elements provide facts and status about various entities in the situation. Comprehension of these inputs indicates the requirement of cognitive effort in order to determine the collected facts.

Design Principle 37 (Situation Awareness Capability)**.** *Strategy must provide information in form the of situation awareness to individuals and organization. Chunking of information should be guided by a unit situation determined suitable.*

2.1.6 Summary

The section provided introduction to relevant theories. The UoD is identified as a mix of heterogeneous objects. First system characteristics and elements are defied in light of known system theories. This includes the control of the systems. The information is identified as important entity in the system. Its origin, handling and utilization are established with help of theories. Individuals working as agents are second important ingredients. Their behavior, characteristics, and influence factors have been identified. Organization was the next complex member of the system. Complex behavior, characteristics and laws that govern evolution in organization is studied. As management and control is prime concern of this work, the situations are identified as one of the important aspect to study the system. The nature, scope and use of situation is established in relevance to the present discussion. To draw the benefits of discussed theories, each theoretical result discussed in reference to its implication in information management task and the important learning is established as design decision. The Design principles 1 to 37 are identified as important learning from the established theory that can be supported by the information management practice.

2.2 Nature of Situation

Situations are defined as set of information units that convey facts about part of the world. The set of information holding units should be studied in light of the theoretical framework as, the content of such set vary considerably with the sender (speaker) and the receiver (listener) and the world that is being described. In complex dynamical system, the situations should be identified and handled by information management strategy. This section provides introduction about various aspects of situation like representation, scope, their relation with the world and other related aspects.

2.2.1 Basic Representation

Based on the work of Barwise [55], [56], Zalta [58], Devlin [57] and others carried out at Center for the Study of Language and Information (CSLI) Stanford University, the concept of situation is discussed in this research. As indicated earlier, the term situation is utilized in various contexts. When situation is to be considered as basic unit holding the required representation of a part of the UoD, formal definition of the situation needs to be established. This section introduces characteristics, features, definition and basic notations utilized in representation of situation. Various types of situation and manners in which they are employed to represent part of the world is also included.

First, task is the characterization of a situation. The situation theory and semantics introduces the situation as a logical assertion containing properties and related objects.

$$\langle property, object(s), \pm \rangle \tag{2.1}$$

Equation 2.1 represent the basic elements of situation. It consists of a property and related objects.

$$\langle\langle R, a_1, \ldots, a_n, \pm \rangle\rangle \tag{2.2}$$

Equation 2.2 represents the situation in more formal way. Here, situation is given in the form of an infon σ, that consists of an n-place relation R, and objects $a_1, \ldots, a_n$. For given a situation s and an infon σ, it can be said that:

$$s \models \sigma \tag{2.3}$$

Equation 2.3 shows the situation s in the form of an infon.

$$s \models \langle R, a_1, \ldots, a_n, \pm \rangle \tag{2.4}$$

Equation 2.4 provides the template of an infon. It contains individuals $a, b, c..$ having identified to form relations $P, Q, R..$ to form situations $s_0, s_1, s_2...$

When assertion is done about the fact of the world, the speaker connect the individuals $(a, b, c..)$ to actual instances in the UoD. In situation theory, for tracking the important link between the abstract individuals and links seen among their instances in real world, the parameters are defined. Parameters acts as placeholders for specific entities and are defined as $\dot{a}, \dot{s} \ldots$

$$s \models \langle R, \dot{a}, 1 \rangle, s_1 \models \langle S, \dot{a}, 1 \rangle \tag{2.5}$$

Equation 2.5, demonstrates use of parameters, where a parameter $\dot{a}$ is held as a reference to a particular instance in UoD. Devlin [57] provided an example that further clarifies the use of parameters where a speaker is aware of link between Smoke and Fire and knows that the smoke is indication of Fire. Hence for the perceived smoke, it must be possible to connect the smoke to specific instance of fire that is directly linked with it. In this manner, parameters are useful in capturing such linkages.

$$s \models \left\langle rescue, inj\dot{u}red, 1 \right\rangle \tag{2.6}$$

Equation 2.6, demonstrates what is the content of a domain specific activity. It demonstrates a disaster management situation in which an injured individual is being rescued.

$$s \models \left\langle rescue, inj\dot{u}red, 1 \right\rangle \wedge \left\langle treat, inj\dot{u}red, 1 \right\rangle \tag{2.7}$$

Equation 2.7, demonstrates a compound infon. This is a very useful representation of situation, stating that multiple situations can be represented as single compound situation to provide rich representation of the situation that may exist in the UoD.

Design Principle 38 (Basic Unit of Communication)**.** *Basic unit of information handled by the information management strategy should be in the form of situation.*

There can be many such situations.

$$s_{dis} = \{s_1, s_2, s_3, s_4, s_5, \ldots, s_n | n \in \mathbb{N}\} \tag{2.8}$$

Equation 2.8 provides information about situations that may be defined in given UoD.

2.2.2 Scopes in Situation

In discussion so far, the simple situation is introduced with an example representation. The notation of situation is indicated holding some information about UoD. This information is also demonstrating the scope in specific context. The contexts are not universal, but are limited to some area implied between sender and receiver.

$$R \in D \tag{2.9}$$

Equation 2.9, demonstrates what is the content of a domain specific activity.

$$Domain = \{education, finance, health, security, science\} \tag{2.10}$$

Equation 2.10, demonstrates what is the content of a domain specific activity.

$$\mathcal{R} = \{R_{edu}, R_{fin}, R_{health}, R_{security}, ...\} \tag{2.11}$$

Equation 2.11, enumerates the possible relations that can be defined in domains.

$$R_{health} = \{diagnosing, treating, operating, counseling, ...\} \tag{2.12}$$

Equation 2.12, demonstrates what is the content of a domain specific activity.

$$treating^1 \Rightarrow \langle treating, \dot{a}, \pm \rangle \tag{2.13}$$

Equation 2.13, demonstrates single place relation in medical domain. This example suggest if an entity $\dot{a}$ is treating or not.

$$treating^2 \Rightarrow \left\langle treating, \dot{a}, \dot{b}, \pm \right\rangle \tag{2.14}$$

Equation 2.14, demonstrates the example of R^2 type of relation in medical domain. It suggest if $\dot{a}$ and $\dot{b}$ with relation treating holds.

Conceptual Dimension in Situation

Conceptual dimension identifies the conceptual hierarchies that may exist in Relations and Entities that are used in the system. in order to interpret the situation, sometimes it is necessary to consider not only the given concept but more detail or less detailed concepts in the dimensions. It may happen that one general term, should be used to cover all the specific terms and utilize it.

$$a \sqsubseteq b \sqsubseteq c \tag{2.15}$$

As indicated in Equation 2.15, it is possible to have hierarchies of the terms, and given term may have specific place in the given hierarchy. In other words it may have specific and general concepts in both the directions.

$$p \leftarrow (q \wedge r) \tag{2.16}$$

The use may be done as to avail knowledge encoded in the form as indicated in Equation 2.16. The rules are defined at specific concept level. Given concept may be part of this hierarchy, and hence the given rule are also applicable to it. Such scenario, requires consideration not only static matching, but also requires identification of rules in semantic dimensions.

Design Principle 39 (Strategy for Conceptual Granularity)**.** *Every concept referred in situation should be part of conceptual hierarchy defined in organizational knowledge. This should support various functions and rules that govern the membership to various abstract on concrete class.*

Temporal Dimension in Situation

Similar to concepts, the situation may include reference of time. A special type is used to denote time TIM for temporal reference. Situation may include temporal locations in form of t_0, t_1, t_2... They can be used as one of the members of n-place relation.

$$treating^3 \Rightarrow \left\langle treating, \dot{a}, \dot{b}, \dot{t}_{now}, \pm \right\rangle \tag{2.17}$$

Equation 2.17, demonstrates what is the content of a domain specific activity taking place at specific time instance. The situation observed and communication at different time may have unique interpretation. Multiple such related situations should be studied to reveal the change-taking place over period of time. The rules may be defined that indicates relation among time stamps.

$$t_a \prec t_b \prec t_c \tag{2.18}$$

As indicated in Equation 2.18, the timestamps may have the precedence relations. This also can be utilized to identify the duration by taking difference of reported time stamps.

Design Principle 40 (Strategy for Temporal Granularity)**.** *Every concept property referred in situation should be identified as known temporal coordinate in the organization knowledge. This should support various temporal functions and rules that govern temporal relationship of concept.*

Spatial Dimension in Situation

LOC: Location Reference is another important place holder for spatial dimension. The Spatial Locations are denoted using l_0, l_1, l_2.. in infons.

$$treating^4 \Rightarrow \left\langle treating, \dot{a}, \dot{b}, \dot{l}_w, \dot{t}_{now}, \pm \right\rangle \tag{2.19}$$

Equation 2.19 is an R^4 relationship with time and space reference. The spatial reference can be further beneficial by utilizing the spatial analysis.

$$l_a \circ l_b \circ l_c \tag{2.20}$$

Equation 2.20 indicates relation among spatial coordinates. This can as well be the spatial intervals or area covered. The area may have various spatial relations[59]. This includes topological relations like indoor, outdoor, near, far etc. This may also include cardinal relation like North, south, east, and west. The other type of spatial reference can be identified in form of orientations like behind, before, to the left etc. and distance in the form of within, near, far etc. Utilization of such spatial reference enhances the capability and accuracy of the asserted situations.

Design Principle 41 (Strategy for Spatial Granularity)**.** *Every concept property referred in situation should be identified as known spatial coordinate in the organization knowledge. This should support various spatial functions and rules that govern spatial relationship of concept.*

$$[SIT_1|SIT_1 \models \langle helping, \dot{p}, LOC_1, TIM_1, \pm\rangle] \tag{2.21}$$

$$[IND_1|w \models \left\langle person, IND_1, \dot{l}_w, \dot{t}_{now}, \pm\right\rangle] \tag{2.22}$$

Devlin [57] further introduced other types that can be utilized in specifying the situations. Equation 2.21 indicates the definition of situation type. Similarly 2.22 provide how individual type can be defined with same formalism.

2.2.3 Types of Situation

Situations hold the facts of the world as seen and uttered by the speaker. The exchange of information is done with reference to some mutually agreed context between speaker and the listener. Depending upon the nature, the utterance can be of three types.

When a speaker makes the utterance while having positioned in specific place and gazing at particular part of the reality for the observation, it is known as ***Focal Situation*** or described situation.

$$s \models \langle present, V, l, t, 1\rangle \tag{2.23}$$

Equation 2.23, demonstrate a focal situation where an injured individual is seen by the speaker at specific instance of time and space.

Design Principle 42 (Strategy for Handling New Reports)**.** *New reports from users can be identified as Focal Situation.*

When a common context is used in utterance, it is known as utterance situation. If a speaker reports to a listener that *a victim is at medical camp*. It is the example of a ***Utterance Situation*** u, where both - speaker and listener refers to the context. In any such utterance, the situation should be sufficiently rich so that the listener can identify all the salient features of the utterance as used by the speaker. It is possible that medical camp that is referred by the speaker, in a larger discourse can be any other medical camp present in the UoD.

$$\begin{aligned} u \models \langle utters, speaker, \Phi, l, t, 1\rangle \wedge \\ \langle refers - to, speaker, camp, C, l, t, 1\rangle \end{aligned} \tag{2.24}$$

Equation 2.24 demonstrates the speakers connection where Φ is the sentence *a victim is at medical camp*, and C is the camp that is fixed by u. Here, speakers utterance draws linage between the camp and the object C.

In a particular scenario, if speaker utters that *a victim I saw injured yesterday is at medical camp* the use of a situation that is uttered earlier is made to identify the victim at the medical camp.

$$\begin{aligned} u \models \langle utters, speaker, \Phi, l, t, 1\rangle \wedge \\ \langle refers - to, speaker, victim, V, l, t, 1\rangle \wedge \\ \langle refers - to, speaker, camp, C, l, t, 1\rangle \end{aligned} \tag{2.25}$$

Equation 2.25, Φ referees to sentence *a victim i saw injured yesterday is at medical camp* and where speaker is making use of r and the fact that V is the unique victim that is being referred here. In this case $r \models \langle injured, V, l', t', 1\rangle$. This is known as ***Resource Situation***. This is a useful from information management point of view, as all utterances can be consistently made by building upon the previous discourse. As situation perceived by the speaker is based upon the common knowledge about the world, it allows building of larger compound situation with help of multiple speakers.

Design Principle 43 (Strategy for Monitoring)**.** *Tracking and Monitoring should be done in the form of Resource Situation.*

2.2.4 Situations and World

By making appropriate utilization of situation and world theoretic views, the purpose of information management strategy is to develop a global picture. A common working picture that is consistently built and used can provide consistent reference. Now, the picture or representation of state of affair should be created such that it should depict the reality. All building blocks and their behavior should be constantly defined. It is known that situation provides basic mechanism for communicating unit information. This is just a building block for the whole picture. The theoretical framework should be known, how these building blacks should be integrated.

$$\forall x[x \trianglelefteq s \rightarrow Situation(x)] \tag{2.26}$$

According to the Theorem 3 in Situation theory[58], every part of situation is a situation. The part-of $\trianglelefteq$ relationship 2.26 is useful in understanding that situation can be broken down and integrated to build larger more complex situation using smaller constituent situations.

$$\begin{aligned} Maximal_1(s) &=_{df} \forall p(s \vDash p \vee s \vDash \neg p) \\ Partial_1(s) &=_{df} \exists p(s \nvDash p \& s \nVdash \neg p) \\ Maximal_2(s) &=_{df} \forall p(s \nvDash p) \\ Partial_2(s) &=_{df} \forall p(s \nvDash p) \end{aligned} \tag{2.27}$$

Situation have part of relationship 2.27. This is useful in understanding that situation can be integrated to build larger more complex situation. If and every SOA or it's negation is factual in s, the situation s can be considered $Maximal_1$. If some SOA and its negation are note factual in s, it is identified as $Partial_1$. If every SOA is factual in s, situation is considered as $Maximal_2$ and if some SOA are not factual in s it is identified as $Partial_2$.

Information management strategy must handle all the possible situation in given UoD. When a user is asserting facts about the world, the strategy must identify all the possible situations that can be triggered by the given assertion. In case of event detection, for given input, the system should identify all the possible events that can take place. Based on this, strategy can employ the utterance about the possible situations. It is therefore necessary for the strategy to determine all the possible situation in given UoD. From situation theoretic point of view, it must be determined when a particular situation is possible.

$$Possible(s) =_{df} \diamond Actual(s) \tag{2.28}$$

Situation s is possible if and only if it is possible that s is actual [58].

Information management strategy must attempt to model and maintain a logical equivalent of actual world in UoD. Individual situations must be used as building blocks to create a complete global picture of actual world. A comprehensive situation built using the individual situations asserted by the users. From situation and world theoretic point of view, it is necessary to identify when given comprehensive situation amounts to be equivalent to the actual world.

$$World(s) =_{df} \diamond \forall p(s \models p \leftrightarrow p) \tag{2.29}$$

It is established that a given situation s is a world if and only if it is possible that all and only factual state of affairs are factual in s [58]. This is known as the world theorem and w_α is considered as unique actual world.

Design Principle 44 (Building Modeled World)**.** *Information management strategy must build and continually update the model of unique actual world as defined in the world theorem.*

When constructing rich representation of world using uttered situations, information management strategy must ensure consistency of the facts. Any contradictory statements can be not included in building the situation. In situation theoretic point of view, the strategy must ensure that no inconsistent situations are handled by the system.

$$Consistent(s) =_{df} \neg \exists p \exists q[\neg \diamond (p \& q) \& s \models p \& s \models q] \tag{2.30}$$

Equation 2.30 is Theorem 17 in situation world theory[58]. It establishes that Situation s is consistent only when S does not make incompatible states of affair factual. If and only if no contradictory SOA are factual in S.

Now, the objective of information management strategy is to build the most comprehensive representation of the situation that is equivalent to the actual world in UoD. For this many possible states of affairs should be considered. The strategy must be able to derive actual world. From situation and world theoretic point of view, it must be defined that how many actual worlds that should be handled by information management strategy.

$$\exists ! w Actual(w) \tag{2.31}$$

Equation 2.31 is Theorem 18 in situation world theory[58]. It establishes that there is a unique actual world.

Information management strategy must consist of handling actual situations. From situation and world theoretic point of view the relationship from actual situation and actual world must be established.

$$\forall s(Actual(s) \leftrightarrow s \trianglelefteq w_\alpha) \tag{2.32}$$

Equation 2.32 is Theorem 19 in situation world theory[58]. It establishes that all and only actual situations are part of the actual world.

Information management strategies have to handle the situation asserted by the users. The statement being asserted may not be factual and therefore, considering it in building the comprehensive situation may lead to inconsistency. Hence, strategy must ensure if the statement being asserted is factual. From situation and world theoretic point of view, it must be determined if the uttered state of affair is factual.

$$p \leftrightarrow w_\alpha \models p \tag{2.33}$$

Equation 2.33 is Theorem 20 in situation world theory[58]. It establishes that state of affair (SOA) is factual if and only if it is factual in w_α.

In considering the situation being asserted, it must be determined by the information management strategy if the situation being uttered is necessary. From situation and world theoretic point of view, it must be established if the given situation is necessary.

$$\Box p \leftrightarrow \forall w(w \models p) \tag{2.34}$$

Equation 2.34 is Theorem 24 in situation world theory[58]. It establishes that state of affair is necessary if and only if it is factual in all worlds.

For information management strategy, it is important to verify that situation being stated is possible in the given world. From situation and world theoretic point of view, it must be established if the given situation is possible.

$$\diamond p \leftrightarrow \exists w(w \models p) \tag{2.35}$$

Equation 2.35 is Theorem 25 in situation world theory[58] pertaining to the possibility of situation. It establishes that given state of affair is possible if and only if it is factual in some world.

Zalta [58] provided detail overview covering various aspects including proof of theorems in Situation and World theory. Equation 2.26 to 2.35 are adopted from the comprehensive paper [58], providing twenty five theorems in situation and world theory.

2.2.5 Situations in Complex Dynamical Systems

Complex dynamical systems are mix of entities and their interactions in the environments. These entities are of many kinds and exhibit various behavioral patterns. Impact of their action may have visible effect in the environment and on other entities. The situation in the complex dynamical system therefore, has to cover facts about these details. The situation in such scenario is not merely exchanged for building state-of-affair. But the goal is to determine the status; its implication and the future course of action. This is also the overall goal of information management.

Hence for this reason, assertion of situation, not only contains information about objects and behaviors, but their meaning and action guidance as well. All these are in reference to one agreed upon reference goal statement. Hence situations are asserted in reference to these goals, and follow-up actions are also identified from the same. This also indicates that with change of goal statement also changes the situation.

For example, Situation about law and order is a common reference. When situation is asserted, it is within the frame of local security laws, local security personnel application at the time of utterance. Hence, when situation is asserted, it is common among the entities that exchange and interpret information. Hence, situation can be redefined as follows:

Definition 1 (Situation)**.** *Situation is a partial representation of the world restricted in space, time and semantic reference, communicated by a referrer with implied goal.*

A goal in some domain requires interpretation of many properties. Referrer may not have possibility to assess the designated situation individually. Truth-values can only be collected by utterance with many observers. The observers must be told about the time, space and semantic reference. There can be multiple such utterances are required.

Subjects of Situation

Situation is in the form of a representation defined using an n-place relation of some entity in universe of discourse. The subject of these relationships can be entities, processes or environmental conditions.

The entities can be of many types with their unique characteristics. Each type can be defined with unique n-place relation. There can be material or physical entities in form resources. These are static entities; remain unaffected until subjected to process. They do not exhibit any changes on their own. Conversely, there can be entities that are alive, exhibiting various behaviors. They have specific life span. Some living entities may move and exhibit various behaviors. They may affect status of others entities, processes and resources. Apart from the living and non-living entities, there are class of entity that are conceptual or abstract. They do not really exist on their own, but they are identified as special cases. or their definition belong to specific domain. The living or non-living entity may play roles of abstract entities.

Each type of entities identified is characterized with help of static properties. These properties indicate their status or relation that they stand for. The properties may be observable and can be assigned a value, with each value having specific interpretation. The observations may be referenced with time and space coordinate along with property values. Apart from these three kinds, the property may include standing relation with other subjects as well. Hence, property is directly used in the form of n-place relation that holds or do not hold in given UoD. This is what the information an infon holds.

Apart from static observation, the subjects may continuously exhibit behavior. These dynamic properties can also be modeled to predict the future outcome.

Contexts in Situation

In removing uncertainty with consistent reference to domain knowledge, resolving equivocality by referring consistent reference, various levels of details, and various properties are required to be handled. Rules regarding properties and their applicability are identified. This is the encoded in the form of knowledge. This also contain the rules that allows determination of states by evaluating assertions made about the subjects.

Complex Situation

A single infon with R^n relation determines the situation. In complex dynamical system, this may not be sufficient. The actual situation may require multiple such infons to be considered simultaneously. Also, they can be matched over period of time. For example, to establish if temperature raising or dipping, the situation asserted at two different time period is required. Sometimes, a single parameter is not sufficient, and complex infons can be required to be matched over a period of time.

Collaboration

According to information processing theory, single person cannot be sufficient to handle more information at a time. In given UoD, the person may observe a property for limited scope that may not be matching the whole. This happens when area to be observed is larger than the area covered by single observation. Hence, in order to build the complete picture, observation form multiple users are required. This is also true in the temporal dimension. Hence, various instances that can provide such required observation to cover the information need is required to collaborate in given UoD. These instances are to be handled appropriately to complete the picture of the situation.

Task Coordination

The required situation may span larger area for longer time. Breaking down of the task needs to be done that can be entrusted to individuals. This can be based on the available information about location, time and capability of individual. This helps in determining if the individual will be able to complete the task. Also, the allocation of the identified task is required. The task determination is done in a centralized place. The actors are placed remotely and should be informed about the assignment of the task. Once they perform the observation, assertion in form of utterance in the form of *Resource Situation*, can be processed to build the complete picture of complex situation.

The required situation may span domain concepts. The concepts in n-place relation may not be observed by all the instances; hence, the task distribution may also be done according to their capability to provide the domain specific observation as well. The logic of determining the ability of the instances to provide domain specific information is employed to breakdown the task. Once it is identified, the allocation is done. As instances provide information, the individual representations must be integrated back and processed to determine if the complex situation holds.

Complex Situation Determination

It has been discussed so far that, simple assertions, may have different effects when considered together. These are domain specific concepts and are defined in appropriate domains. Their implications are also defined in domain. They are also encoded as interpretation rules. They are stored in context. They can be evaluated once the complex situation is built as a result of collaborative effort of utterance.

2.3 Situation Awareness

Nature of situations possible in complex dynamical system is suggested. The next task is to use situation to guide actions. The situation needs to be asserted, collected and utilized. When actors are having access to situational information they are found to be having situation awareness. Missing component is the specification of required mediation. A mechanism is required to manage the process of the building and handling situation. Mediation required for building situation awareness among stakeholders and organizations.

2.3.1 Requirement of Mediation

The information is being exchange in the form of situation. To meet the instantaneous needs, what part of reality will be reported by whom and when is to be determined. To achieve this mediation is required mainly in the form of system. It is now established that in complex dynamical systems, the complex situations can be observed with the help of collaborative efforts of multiple instances. This evaluation is not continuously carried out. But this is triggered in response to some observed event. It is also not possible for a coordinator role, to identify all the instances, and breakup and rebuild the infons for each individual instance present in the UoD. Hence, an automated mediation is required to realize this task in rule driven form.

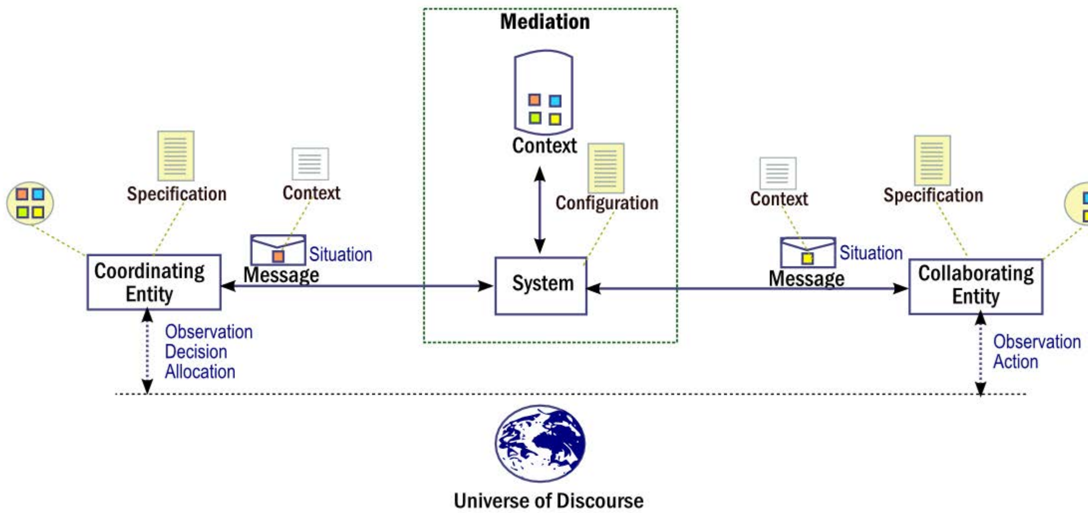

Figure 2.3: Mediation Required

Figure 2.3 indicates the role the mediator plays in order to meet the identified requirements. On the right side, a single instance is depicted. This individual comes with a specification of spatial, temporal and conceptual granularity that can be handled by them. Also they have their goal or intension to perform specific task. The left side, the coordinator instance is depicted with similar goal along with the specification. In the middle, the role of mediator is depicted. Firstly, it holds a repository of contextual information. They include the common reference to be used throughout for consistency of context. They also include rules for interpretation as well as rules for handling the task allocation. The system that accesses the rules is depicted to communicate using messages to other entities. The content of the message is the infons determined suitable for communicated. For example, an infon being send as message content to instance holds the information request, asking the instance to tell if that situation holds. This solves the problem of common consistent resource context across the instances.

Once instances reply, the message with observations, the integrated view is built by the mediator, and it is subjected to evaluation for complex situation. The resulting situation is communicated to coordinating agency that can take further decision regarding determination of future course of action. This is communicated back to the mediator, and provided to the instances that can perform accordingly. Hence, the mediator is made responsible for providing consistent contextual information and processing capability for overall situation awareness.

2.3.2 Nature of Mediation Required

Figure 2.3 provided information about role required to be played by the mediator. The main requirement is to furnish consistent references and to act based upon the rules defined to break-up, rebuild and process the task of situation creation in complex dynamical systems.

Quantitative Estimate of Situation

Quantitative estimates provide scale of complexity the system has to handle. The quantities can be affected by some parameters. Identification of these parameters are important for efficient design and handling for the system that support such functionality. Quantitative estimate of the situation is necessary to determine the load on mediation. The quantitative estimate is determined by answering the following questions.

Situation can be characterized with details about the State-of-Affair (SOA) using commonly understood terminology used for representation. Situation is therefore part of facts collected in reference to common domain. For example, situation of law and order in location x at time t. Situation may have multiple dimension but there are common conceptual dimension is the level at which it is defined. Spatial dimension provides idea about spatial coverage Temporal dimension provide estimate of how much period which time scale. In order to determine, if situation is completely covered, the associated domain knowledge must be referred. The representation may involve multiple roles with multiple levels of responsibility. Each may require

	Factors
1	Number of situations
2	Frequency of change
3	Possible break-ups in smaller situations
4	Complexity of situation
5	Number of required utterances
6	Time required for the evaluation of utterances
7	Spatial coverage of situation
8	Temporal coverage of situation
9	Number of stakeholders
10	Dynamism of the activities

Table 2.10: Factors Related to Quantity of Situations

unique level of facts. Hence different levels of granularities are required. In complex dynamical systems actors, agents and environments are continuously changing. The agent behavior is also changing the overall state. Here, situations are also found to be continuously changing. It is therefore necessary to determine the number of utterances required from observers to build the possible situations. The factors affecting the number of possible situations should also be identified. Some of factors include domain of knowledge, observable properties, complex interdependence among entity and dynamism. Hence, proper quantitative analysis is required in this regard.

Qualitative Estimate of Situation

Qualitative estimate provide the richness of content that must be handled by the systems. As quantitative estimate provides the estimate of load on the system, the qualitative estimate allows identification of cognitive load on the system. The richness of representation may require encoding of appropriate knowledge. Similarly, this increases the demand of cognitive load from individuals. The quality of information is established also by determining its ability to

	Factors
1	Domain involved
2	Goal of individuals and organizations
3	Complex interdependence among entities

Table 2.11: Factors Related to Quality of Situations

communicate and convey intended meaning without causing any ambiguity. This issue is known as equivocality of the system and it should be appropriately addressed. Situation must address the problem of uncertainty. Uncertainty is the non-availability of information required to take specific action. Representations the scope of action required and related information should be included in situation. Goal Matching is provided by the system. System may have multiple goals; similarly users may have and may assume various roles. The goals and intentions should be encoded in the situation. The newly identified goals should be made available as situations for matching.

Handling individuals in Situation

It is the fact, that uncertainty is only defined when task to be done is known. Task is known when situation suggest the system in specific state and there is a defined required course of action that needs to be taken. Hence an individual can first be identified that can take up the goal. Mediation is required to provide prospective entities and present them the state of affair, assuming that they will identify the goal. This is how building of active publics can be achieved. Apart from meeting information seeking behavior the organization can help the information processing behavior by providing information about ongoing situation and requirement of action. Based on provided information and available skills of individual, goal matching can be carried out. Individuals can be identified with new goals having matched the specification of action required. As individual have identified new goal, the course of required action is clearly known. Hence uncertainty about execution of the task can be addressed for the actor by providing related information necessary to execute the selected task. Also, the system ensures provision of information with quality that minimizes the equivocality.

2.3.3 Redefining Situation Awareness

Situation is defined earlier[21] in context of controlled setting, and hence, the focus is on quick decision making based on monitored parameters. In such scenario, perception of limited set of parameters are consistently carried out using sensors and instrumentation techniques, and hence, do not require attention from information management point of views.

In complex dynamical environment, the system boundary is not static. Within such boundary, perception of various elements, actions and processes are not easy as consistent monitoring instrumentation can be employed to track all of them. This is to be carried out with help of various stakeholders present in UoD. The resource situation is identified as mechanism for unit communication from stakeholders. In such scenario, focus of Situation Awareness is considerably divided on continuously determining the scope of situation, the stakeholders that will provide status and require information to determine future course of action and the information processing strategy that can support them.

Definition 2 (Situation Awareness)**.** *Situation awareness is a state achieved when information that is qualitatively and quantitatively determined by given configuration as suitable for assumed* *role is made available to stakeholder by engaging them in to appropriate information exchange patterns.*

2.3.4 Information Needs

So far in the discussion, situation is treated with system point of view. Entities are identified as situation aware when appropriate quality and quantity of information is provided to them. Hence, the next task is the determination of what information is relevant from user point of view.

The basic requirement of information can be identified in terms of information required by the individual to carry out specific tasks as per the assumed role. Non availability of such information is identified as ***uncertainty***. Hence, addressing uncertainty will lead towards meeting the information needs of individuals.

In order for the information to be useful to the user, it must be available at appropriate granularity or specificity. The granularity is discussed earlier in this section as hierarchical levels of details in spatial, temporal and conceptual dimensions. The system may handle information at multiple levels of details, yet user can handle and require information only at specific level depending upon the requirement and capability.

Upon availability of the information can determine future course of action, and corresponding to newly identified actions, the system have to address the uncertainty problem for the user. Hence, provision of appropriate information not only allows execution of task, but also helps selection of the required tasks. This is done on the basis of the goal identified by the individuals. Hence, goals can be treated as starting point of determining information needs.

Users may have goals pertaining to various aspects of life, and system may or may not cover all the aspects that the individual may perceive. Hence, the provision of information is done in reference to the aspects covered by the system configuration. Information management strategy therefore attempts to provide configuration to meet the needs of involved stakeholder and nature of the system. The configuration itself is affected by the goal of the organization. Hence, goal of the organization and the scope of configuration adopted by the information management strategy determine the scope of the system to meet individual's information needs.

The definitions offered by Bruce [34] and Maceviciute [35] provided in organizational context must be extended for complex dynamical system considering the dynamic set of involved entities, changing goals, and specific configurations.

Definition 3 (Information Need)**.** *Information need is an identified information requirement at a granularity, necessary to perform assumed role(s) by a stakeholder in a given configuration.*

This definition supports dynamism aspect as assumption of specific role by a stakeholder is a responsive behavior exhibited by stakeholders with changing dynamics of the situation. Similarly, identification of role and information granularity provides emphasis on the configuration that is continuously updated to suit the complexity and dynamism evident in the system. Hence it can be said that information need is continually identified and appropriately addressed as per the recommendation in the proposed definition. In order to establish utility of the proposed definition of information need an example scenario is offered.

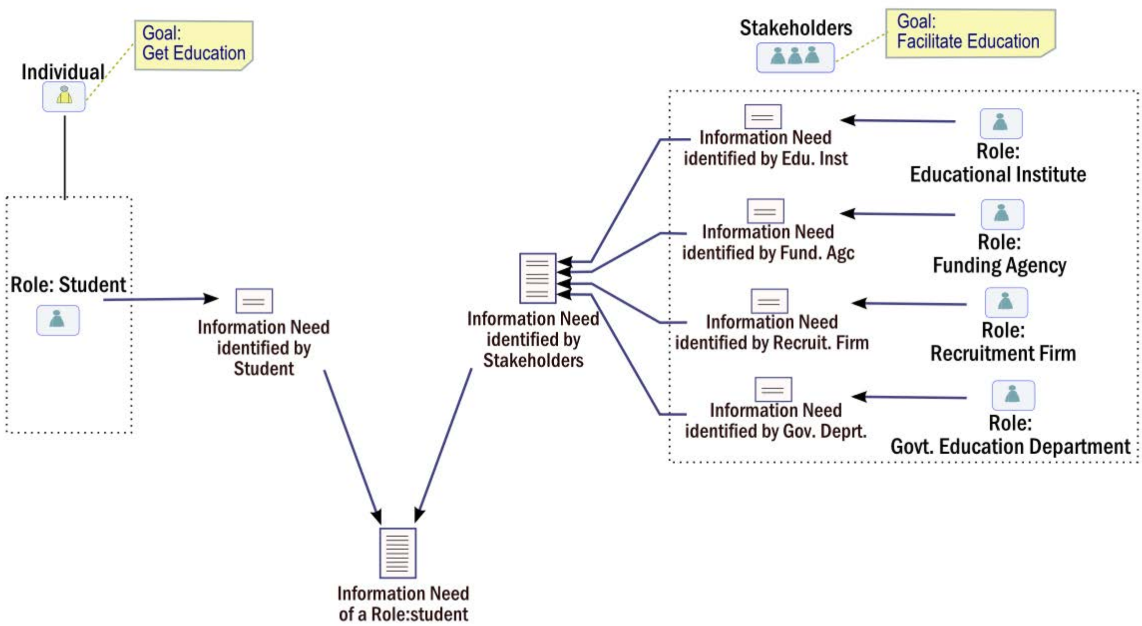

Figure 2.4: Information Needs an Example

In Figure 2.4, the information need of an individual is depicted. An individual decides a goal of getting educated and assumes a role of a Student. Based on this role, it is possible to identify information need. But at the same time, student must also be aware of certain things that are beneficial in achieving the goal. This information is determined by the stakeholders. Now, in any given area, there can be multiple stakeholders that are identified active in providing educational services. The examples of such stakeholders are government education department, funding agency, private coaching class, tutors etc. These are organizations provide services based on the available domain knowledge in education domain. It may include knowledge about use of various education and teaching methods, educational content, etc. Some research to improve specific aspects of education results in creation of promotional and support schemes. Hence, it can be said that the more knowledge available in the form, more organizations are likely to adopt and render services. And more such organizations are present; it increases the information need of an individual.

Quantitative Estimate of Information Need

The example provided the hint about the factors that may affect amount of information that may be provided to meet the individual's information need.

$$\underbrace{(I_1, I_2, I_3, \ldots I_m)}_{Goal_1} + \Big(\underbrace{\underbrace{(I_1 + I_2 + I_3)}_{SH_1} + \underbrace{(I_1 + I_2 + I_3)}_{SH_2}}_{Goal_1} \Big) \tag{2.36}$$

$$(Goal + m.I_{goal}) + \sum_{i=0}^{r} \sum_{j=0}^{s} SH_i I_j \tag{2.37}$$

Equation 2.37 indicates quantitative estimate of information need for a single goal statement. In real world, an individual entity may have more than one identified goals. Therefore similar estimates can also be calculated for all the identified goals.

Active Agents Point of View: The value of r, the information need is the number of stakeholders is directly proportional to number of active agents. These are the active agents in UoD either helping or harming the system state. They do so by affecting entities, processes, resources and environments in the given system. They may include Stakeholders SH. More the number of stakeholders present in the system, higher the information need to be handled by the system. This is because, individuals should be made aware of them, as well as they needs to be made aware of individuals.

$$r \propto n|\text{n= Number of active entities} \tag{2.38}$$

Available technology providers and organizations: Number of active agents is directly proportional to number of organizations committed to provide coverage, and have engaged people in specific domain activities. Availability of number of stakeholders in given UoD is based on how technology providers and organizations are involved. Access to various enabling technology and its benefits known to the stakeholders will improve the count of SH providing services in given UoD. Many organizations involved in providing services and products. Thus, the available technology, know-how and recognition, improves the number of stakeholders.

$$r \propto o|\text{o= Number of organizations} \tag{2.39}$$

Domain Knowledge Point of View: This is directly proportional to number of domain concepts available in knowledge. Product and service provider organizations are providing services based on the available knowledge in specific domain. The domain knowledge establishes the benefits of specific product or service. With more amount of domain knowledge available in the form of content, its application etc, higher the possibility that this growing knowledge is taken up by product and technology organizations to create solutions. This further facilitates the usage, promotes the service and build network of various stakeholders for their solution.

This is a common trend observed in past few decades. The developments in domain knowledge of mobile and wireless communications and its availability as published work have attracted many technological organizations to develop products. With availability of products, many service provider organizations have come up to take the communication service to the potential users. With this development, the urban areas are now having multiple stakeholders in form of service providers, sellers, repairers, advertising and related professionals. This subsequently leads to improved information need of individuals as potential customers.

$$r \propto k|\text{k= available domain knowledge} \tag{2.40}$$

As a consequence of this relation, it can be said that amount of information need is affected by number of active agents, that is the stakeholders, product and technology organizations, and domain knowledge. All of them are part of same UoD yet distinct in their process of work products.

$$IN_{qty} \propto r \tag{2.41}$$

Given UoD can span a large geographical area. The spatial coverage of organizations and stakeholders may be contained in this larger area. Also, it is likely that individual and stakeholder may be active at global level, and given UoD can be a smaller part. The spatial footprint if individuals, stakeholders, organizations and system can be identified from the specification extracted from goal. This spatial footprint must be considered as one of the decision parameter for matching goals.

The information about the level of detail is required. Goal matching process is done by a mediator based on matching of various parameters expressed in specification. In case of organizations and organizational individuals may have clearly expressed the service specification. But individuals may not have detail about the specification of services or activity they can offer. Hence, the semantic scope in context must be defined and appropriately used in goal matching process.

Individuals and organizations may assume role as response to some factors in ongoing situation. For example, emergency medical service is needed by an individual in case of injury obtained in disastrous situation. In response to same event, medical professional may offer emergency medical services to meet the emergent needs. The span of medical attention required might be very small as compared to span for which medical service offered. Hence temporal footprint is also a critical factor in determination of match. Both co-occurring in temporal dimension should be taken for matching decision.

In summary, the discussion just stated that information needs are affected by goal and it's explication in the form of specification. Various parameters used in specification pertaining to spatial, temporal and semantic dimensions provide support for accurate matching for activities needed and offered.

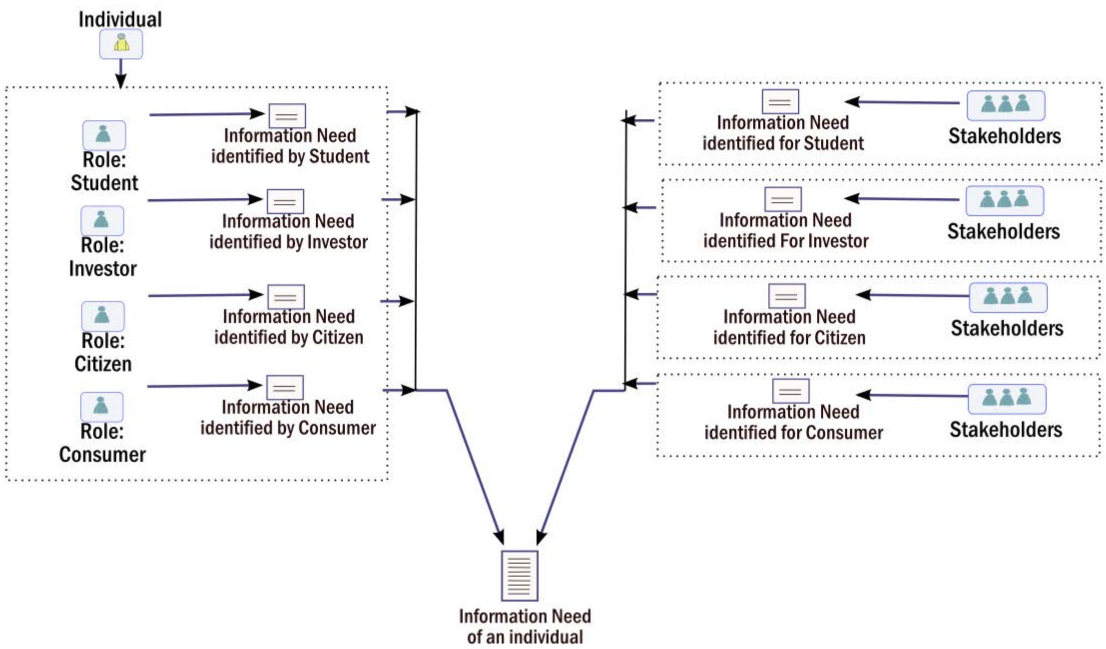

Figure 2.5: Information Needs of an Individual

Qualitative Estimate of Information Need

Quality and content of information is also important. This section describes, what should be the content and how it should be provided to the user, so that it improves the chances to meet the goal.

Representation of Situation: This is the primary requirement, as it should give representation of all relevant entities and their status in given UoD. Sometimes, this can be inhibitively large, and the user can have cognitive load of processing information. Yet, this is the representation of part of world, as it is-required considering the fact that, use may extract information out of the representation that is not captured by the system. Information in the form of alerts, event notifications and open requests are also required. This is regarding some important events that systems have identified. Alert regarding an action requirement is also issued to the potential contributors. For the required quality or work product, appropriate guidance is required by potential contributors. An actor may have limited information, or have different perception of quality and time frame desired in performing the required action. Hence, this information needs to be provided by the coordinating agency. The organization might have updated, or have some specific criteria. For this, user must have access to information about execution of the required tasks.

Information is required to facilitate monitoring and tracking of relevant elements in the UoD. It requires monitoring and tracking related information that enables the actor to keep updated regarding the outcome of collaborative activity. It also allows registering task closures, allocations and other important aspects.

$$IN_{qual} = \langle rep, guidance, trackingevents, alerts \rangle \tag{2.42}$$

Information Need and Situation Awareness

It is critical to determine whether the individuals are situation aware. By virtue of goal or mere presence may affect the information need. Information need can be determined based on goal, intentions etc. Determination of information need is possible with identification of tasks, goals etc. for overall situation awareness.

Information Need and Events

Dynamical systems, events, triggers, event, punctuations, response, actions, and tracking events many others elements contribute to the information need. With any state change recorded, identification of events is possible. Constant monitoring is required for occurrence of various types of events. Hence events are one of the basic elements that trigger information needs. Events mark the starting point of information flow, as well as the ending point. Responsive course of action identified for detected event provides basis for creation of new collaboration. Events may start the punctuation phase in the system. They allows the systems to start a group of masses becoming active publics, and cease to be so, based on the events detected. Hence, events are primary determinant of actions.

2.3.5 Actor Situation Awareness

Individuals in complex dynamical systems require basic awareness of ongoing situation and therefore exhibit information processing behavior. With information processing and commitment to goal, they may assume new goals. With their self-efficacy, they indicate their ability to use technology to accomplish the task. Any additional information required relevant to the assumed role is met with information seeking behavior. By providing information about the problem system ensures the mental cognition of the problem. The additional information about the use of technology demonstrates task technology fit. With availability of such information, the individual is converted in to active public and engage in problem solving behavior. Individuals demonstrate ability to perform the task. While solving problem, individual provide status of local environment. Upon task completion, the individual requires to send task closure information. In summary, an individuals' goal for situation awareness is to identify required task, indicate the intension to offer the task, get the information required to perform the task and once task is completed, must get due recognition.

Definition 4 (Actor Situation Awareness)**.** *Actor Situation Awareness is a state achieved when actor is provided with representation of the universe of discourse at specific space, time and conceptual granularity along with the specification of the required action, events, alerts and guidance determined suitable for carrying out the assumed role(s) in given configuration.*

Figure 2.6 describes the situation awareness requirement for an actor. The actor is engaged in various activities, and is part of the universe of discourse as one of the stakeholder. Actor has a personal goal and commitment towards achieving it. The goal provides the specification about the capability and intensions of actor. In order to meet the information need actor is identified

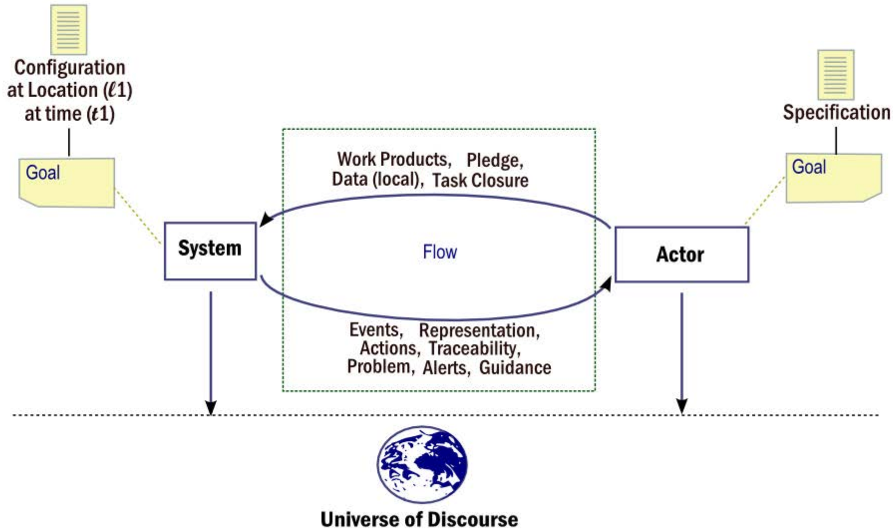

Figure 2.6: Characterization of Information Need for SA

with the system. The system is hosted by an organization to support the stakeholder needs. The system has goal in meeting information need based aligned with organizational goals that has hosted the system. Based on organization's jurisdiction, the system can be identified to have a specification. This covers the kind of information the system is configured to handle, the area and temporal granularity covered by the system. For monitoring and resource allocation, system may also be seen as continuously connected as entity in UoD. Based on the monitoring, and matching it with organizational goals and rules, the system is able to identify the events; required actions are other relevant information. The system requires the help of the actor in creating actions in UoD. In order to support the action, the system provides the user with information identified matching with actor specification. The action request, the representation of the world, the guidance to take action, alerts about task and required resources etc. In response to this, User is also required to report back to the system, regarding its ability to respond to the call for action. The action may also require reporting of commitment towards task, completion status of the task, local conditions, etc. This information is provided by the user to the system.

2.3.6 Situational View of the Organization

For organizational situation awareness, it is important to establish information required by the organization itself, and the information the organization want to share with the stakeholders. In Barwise's point-of-view, the organization stands in specific relations with others. Others can be individuals, organizations, and special kind of elements. Idea is not to display the world as it is, but to enable organization to achieve the goal, by creating and organization information content that will lead it towards this achievement. Quantitative estimate provide the information about the frequency at which the information will change. Spatial estimate is required to identify relevant changes taking place in other geographical area. The action requirement in creating plans to enhance dynamic capability, finding new actors by providing required information to build active publics, sharing and accessing resources exhibiting the resource dependence, utilizing the transitive memory of involved stakeholders. With longer lifespan and extended resource allocation, the issues like diffusion of innovation demonstrating penetration of technical advancements, methodological alternatives, and standards compliance is also supported. To enable the reuse of the work products, the task of publishing in public registry of resources is also required. Hence the situational view of organization covers all these identified aspects.

Definition 5 (Situational View of Organization)**.** *Situational View of Organization is a view specifying the prevailing view of organizational dependence, status of stakeholders of given organization and situation awareness thereof.*

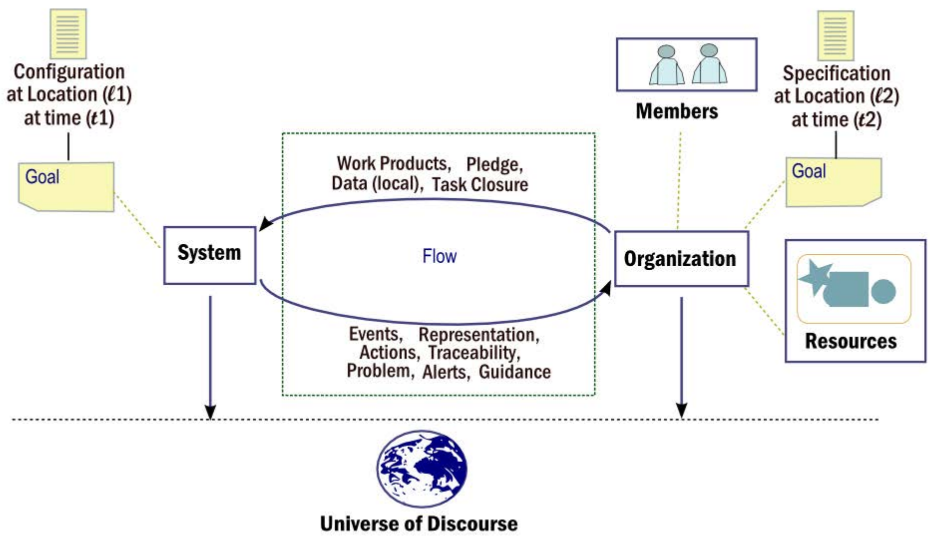

Figure 2.7: Characterization of Organizational Information Need for SA

Figure 2.7 describes situation awareness requirement with reference to an organization. The organization is active in given UoD as one of the stakeholder. The organization has specific goal and the commitment required towards achieving it. This goal statement provides specification about capability, coverage and intensions of the organization. To support this organization is found to acquire resources. Also the members of the organization those are individual supporting organizational activity. This also indicates that dependence of organization for resources are actor from outside organizational boundaries.

Yet organization situation awareness is recommended as a view or representation. The source is same as the individuals but as organizational information processing capability is higher, the information is kept in hierarchical form. All the work products and related status are in the form of hierarchy and not isolated information containing the status. Similarly, based on the role of organization, the representations, and actions alerts can be many. All of them are organized under proper hierarchy, so that it can be retrieved properly.

2.3.7 Gaps

In previous sections, actor situation awareness and organizational situation awareness is defined from individual point of views. It assumes that both are aware of specific goals and parameters. Yet, in complex dynamical systems, such specifications are not known, and even if known, they keep on changing frequently. Hence, there is an additional responsibility on the mediator system to also track the changing goals, along with the responsibilities identified earlier.

Implied Goal Matching Problem

Situation is always communicated with context. The goals, plans, policies provide this contextual reference. As they are driving the calculation of situation awareness, the important part of mediation is to carry out goal matching. This can be a difficult challenge in complex dynamical system. Figure 2.8 depicts the explicit goal scenario. This is a typical service consumption scenario. Here an individual is having a goal. That can be expressed clearly in the form of goal statement and the specification of service consumption. Another role is the service provider with

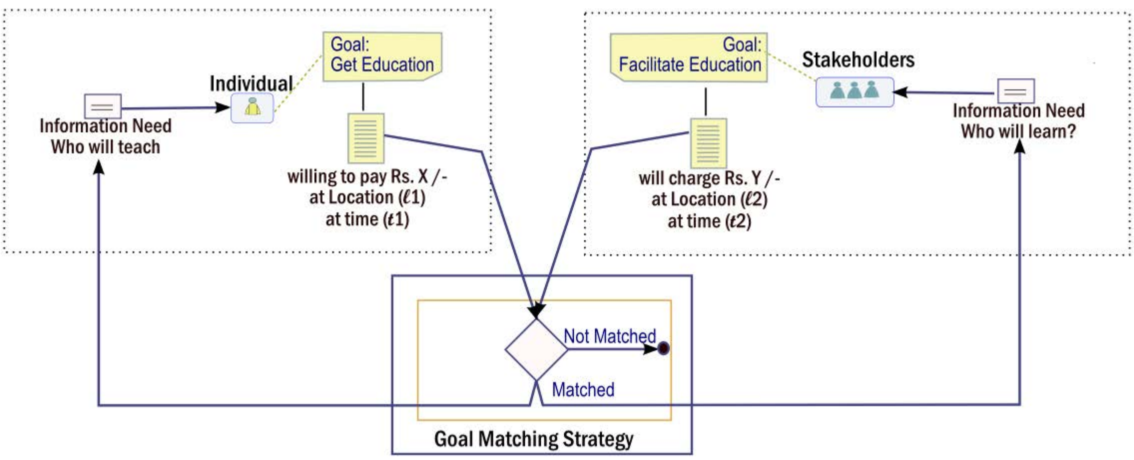

Figure 2.8: Explicit Goal Matching in Service Consumption Scenario

a goal and the specification of service. For an information management strategy it is easy to

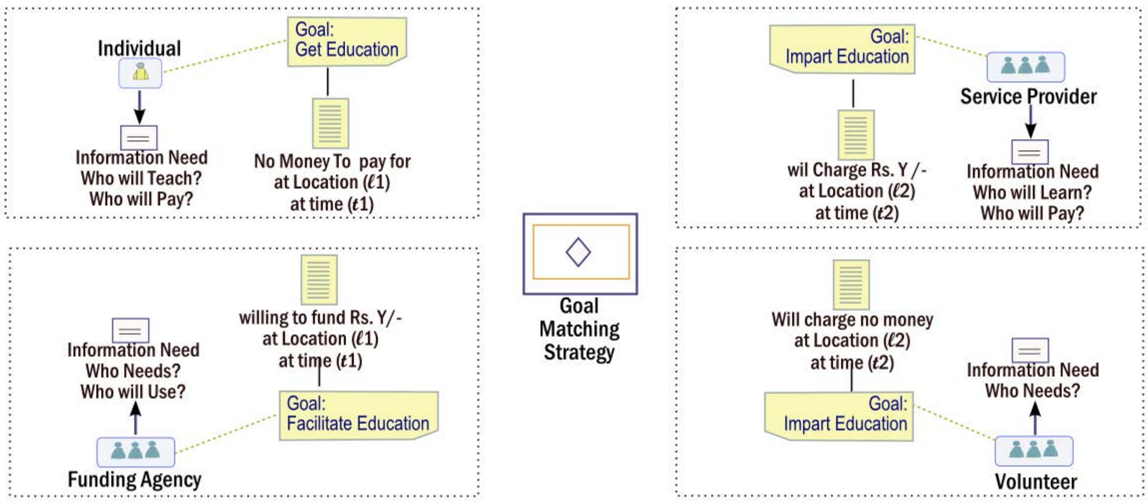

Figure 2.9: Explicit Goal Matching in Voluntary Scenario

match goal of both parties based on explicitly expressed parameters. The complementary goals and similar parameters allow the matching process with little effort. The decision box decides if goals are complementary and parameters in space time and other specified domain are also matching if found matched. The matching is considered to meet the information requirement of both roles. Hence, the matched entities can be informed for further action. This is possibly a general scenario in consumer of basic and common known services.

Figure 2.9 depicts the service consumption scenario with help of volunteer and welfare Agents. It is generally found that certain services are considered essential service for critical service in civilized society. Access to such basic services to all domiciles is ensured by society, government is made responsible for ensuring access to these services. They either provide service, or facilitate the services. Apart from the services provided by government or service providers, the service may be offered by volunteers. These organizations recognize the needs of the beneficiaries and may provide service their role is different from service provider as they do not charge. They are different from government as they are not legally responsible or liable for providing such services. For an individual, there are now three options for getting the same service. For the goal matching strategy there is more than one complementary goal that should be identified and matched. All three types of entities contribute to information need of an individual. The information about the individual should also be passed on to the related counterparts.

Figure 2.10 demonstrates implied goal matching in co-ordination scenario. The upper right part of picture represents the agent instead of a service provider. This agent may be an entity with positive intent or negative intent that can be identified with outcome of its behavior. This agent is an entity in UoD that may either harm or help the desired status of the system. This can be industry involved in production of goods they may help in from of creating products or services that can be useful to individuals. At the same time, they can harm the individuals by generating wastes and polluting the resources. Their goal is to maximize their economical gain or profit at least operational cost. There can be many such products or service providers. They may not know impacts.

Another entity in the Figure 2.10 is an individual. Individuals have general goals, and no specific parameter known to them. Example of this state is the goal to get benefit of many of the available services or to be protected from harmful impact of potentially hazardous activities. For this reason, exact parameters cannot be provided as well as goals. Both are essential requirement for matching logic.

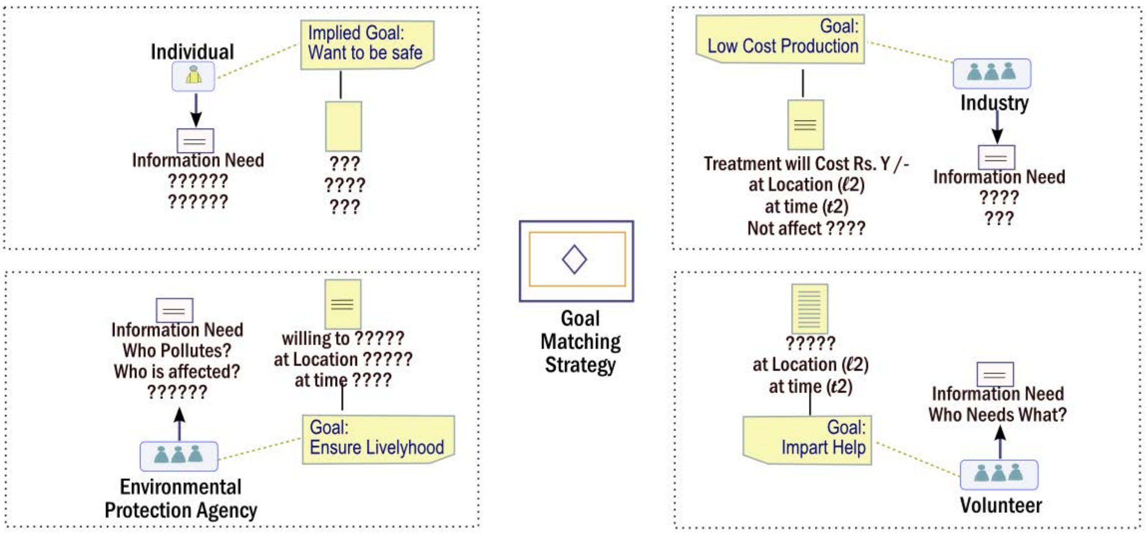

Figure 2.10: Implied Goal Matching

Third entity is governmental or coordinating entity that too has general goal statement like ensure safety, ensure livelihood. This may not know individual agents that may prove to be hazardous harming the individuals and environments in the surroundings. For such agents, exhibit behavior for smaller period of time. Volunteers also are capable to furnish help. But constantly ongoing events it is difficult for them to decide which service is required more attention. Hence they also cannot provide concrete goals and parameters that can be appropriately matched with complimentary goals. In all four cases, information needs are difficult to establish and goal matching strategy have to work with implied goal statements and parameters that are frequently changing with behavior of individual entities.

Transient System Element Problem

Over a period of time, many elements come to existence, play specific roles and cease to exist. Each kind of entity exhibits specific life span. The mediation should make provision for these issues. Entities are in the form of natural, artificial or organizational environment. They can be concrete or abstract based on how they are defined. The entities have specific life span. During this life span, there undergo many state changes that are important for overall system dynamics. In given information management practice, each entity identified in the UoD must have corresponding instance (copy) in the supporting information system. To make it appropriate, this logical equivalent in the system, should emulate the same behavior and life span as in the UoD. Thus, for all entities in UoD, corresponding transient instance should be managed and monitored by the system. This is a required feature, as these transient entities affect and gets affected by various activities and processing in the system. Hence, in order to track the system state, and define the required course of action, the status of these entities provides valuable input to decision-making process.

While management of transient properties are very important for control of the system, handling of them is a difficult problem. In real life situation, apart from the concrete entities physical entities that are easy to track, some entities play various roles that can amounts to be treated as a separate abstract entity. An instance of the human, during a long lifespan, may play a specific role of volunteer, respondent or victim. For the system, the exact starting point, end point and various states pertaining to such abstract role is considerably difficult to handle.

Event Space Problem

The transient nature of entity and dynamism of the system both indicate possible state transitions in the UoD. These state transitions are sources of Events. Events mark the start point, endpoint or intermediate state of a dynamic process, and hence, tracking of event is useful predecessor of decision-making process. Any state change can be an event, and there can be many of them, increasing the amount of information required to be processed. Hence, Event must be classified and handled based on its importance in determination of required action. Therefore event are sometimes referred to as *important state changes*, restricting the scope. In complex dynamical system, the dynamism drives the number of events while complex interdependence allows identification of importance of event in overall system state. It is necessary therefore, to determine ***Event Space*** - set of all possible events in given UoD.

In order to determine event space information management must address following issues. First, the all the sources of information must be identified that produce events. Second, is to establish the criteria that separate the insignificant transitions and important events that require attention. Third issue is in the establishment of a mechanism required to collect and process the information. The fourth part is to incorporate detected event with the information need of appropriate stakeholders.

In implicit goal matching scenario depicted in Figure 2.10, the event can take place in all four types of user environments. Actions or process by agent (industry) may have complex outcome that may trigger multiple events. One of the outcomes may be experienced by individuals in the surroundings. Hence, the events related to process triggers events related to individuals that are affected from the outcomes. With knowledge of affected individuals, the volunteers may start activity-indicating event in volunteer environment. The process by industry, impact of individuals and response from volunteers all three are relevant for the coordinating governmental agencies. This in collaborative organizational environment ensures tracking of activities, status of entities and resources are events in organizational environment. All these coordination is handled with information management process. As indicated in previous section, the corresponding transient resources are created in information system and tracked. Hence, creation and handling of instances in system environment are also sources of events. Thus Event Space in the example scenario involves the various environments, and hence information management practice should sufficiently make provision to identify events and address the four issues related to event space.

2.4 Dynamic Information Management

Information Management was introduced in Chapter 1 with brief introduction to general concepts. In Chapter 2, Information Management issues are discussed in parallel to setting up background in complex dynamical systems. While discussion of theoretical concepts pertaining to system, information, individuals, organizations and situations provided identification of Design principles, the detail characterization of information needs revealed some gaps. The problem of implied goal matching, transient entities and event space introduced additional challenges created by complex dynamical nature of the system. Next task therefore, is to identify strategies to fulfill the gaps and provide specific recommendations for information management strategy. The purpose of information management is identified to remove uncertainty of stakeholders of a coordinating agency to meet identified goals in complex dynamical systems. This is to be done by furnishing information that supports decision requirements. Also furnish information to provide sufficient guidance to successfully carry out the work.

Definition 6 (Information Management)**.** *Task of enabling users in creating, accessing, handling and provisioning information, information sources, targets and technology that is involved in them in order to meet information needs of all the stakeholders for given goal oriented system.*

2.4.1 Dynamic Strategy

Definition 7 (Dynamic Information Management)**.** *Dynamic information management is an information management strategy that is employed to meet the information needs of the stakeholders for their implicit and explicit goals in complex dynamical systems.*

Dynamism in Technology Oriented Management

Planning, storage, and provisioning of data for system, users, and stakeholders is identified as requirement. The purpose of technology oriented management approach is to ensure physical access to information by providing technological solutions for provision of information. Historically, information management was earlier known as data management as in many situation data is considered to be equivalent to information. While distinction is clear between utility of data and information, the data continue to play important role as much of the information is generated, handled and provided in form of data. Data source are originating source of data, they can be in the form of sensors, process or human. Data resources are the instruments or physical entities that physically hold the data. Data Management task include policy, planning, physical level design, operation level management issues. Based on the dynamism of the system, the availability of data sources and data resources is generally considered inconstant. Hence, the information management strategy should. It should support multiple dimensions of data that is required to appropriately describe situation.

Design Principle 45 (Data Management Strategy)**.** *Planning, analysis and operational level management of data should be made event driven. Data sources and data resources should be handled as transient resource, and managed by the system.*

Data management should be supported by the appropriate IT infrastructure. To integrate use of data with working of users, the information systems should be appropriately defined. The complex dynamical systems should provide sufficient IT infrastructure for required data sources and data resources, processing it and provision to the appropriate users.

Dynamism in Content Oriented Management

Records are created for holding information. As it is expected to provide situation awareness in addition to organization specific information, the situation related information should be included in the records. The dynamically changing situation must be captured and published in the form of situation reports. This dynamic creation requires other important aspects as well.

Records management practice is important part of information management strategy. This includes creation and handling of records that can be utilized by various stakeholders. They can also be used as references suitable for execution of the assumed roles. Situation has specific impact on content and context of record management activity.

Task of record generation is driven by dynamism in situations observed in given UoD. As dynamism triggers the generation of records, the complexity of entities involved in the situation provide reference for representations and other content captured in the records. Hence, situations allow identification of two design principles.

Design Principle 46 (Situation for Creation of Records)**.** *Generation of records should be driven by situation.*

Design Principle 47 (Situation as Content of Records)**.** *The content of various records should provide situation awareness appropriately addressing situational information.*

Provision of external information is a critical part of the information management strategy.

Design Principle 48 (Strategy for External Information)**.** *Identification of appropriate external sources of information should continuously be carried out.*

Dynamism in Knowledge Management

Knowledge creation is a continual process resulting from externalization of individuals' knowledge or skills that are useful in performing some action. As system is continuously evolving, the producers, response and behavior of the actors also change. This newly acquired knowledge must be externalized, so that as and when required, the other users can also apply. Similarly, various branches of pure and applied sciences, also is enriched with experimentations, research, discoveries and developments in the field. Newly created knowledge should be continually made available to the potential users. Schlögl [1] discussed set of features typically supported by a knowledge management system identified by Maurer.

When knowledge management is carried out in an organizational environment, the subject of knowledge is similar to the organization's interest. Hence, the individual's knowledge is

	Feature
1	Provide information of an individual potentially useful to others
2	Learns as users utilize
3	Triggers actions and furnish information not asked explicitly by users
4	Create new information from existing

Table 2.12: Features of Knowledge Management System [1]

within the boundary of organizations domain of interest. Also the potential users are in the same organization, and in the same work environment. In this scenario, externalization of the knowledge of experienced user and utilization of this knowledge can be managed with relative ease due to the homogeneous nature of the environment. In case of complex dynamical systems, there can be many users with varying expertise. Externalization of their expertise and management of such knowledge in appropriate specific area of domain becomes difficult. From users perspective, out of vast amount of encoded knowledge, recommendation of specific knowledge relevant to the individual need becomes difficult.

Design Principle 49 (Strategy for Coverage in Knowledge Management)**.** *Relevant branches of domain knowledge must be identified, and potential sources of knowledge containing externalized knowledge created as a result of encoding of expert's knowledge.*

As the users make utilization of the encoded knowledge, and the experts continue to gain experience in handling various actions the learning should take place. In complex dynamical system, this practice may not take place as regular interval, as most action may be triggered as response to some event. Hence, the learning from usage can take place in response to event that may exist for very short period of time. Those places where, occurrences of events are higher, and response includes the same activities, the learning can be observed to be more as compared to other places. Hence, Knowledge management practice should consider the effect of dynamism in capturing learning from various sources to obtain the coverage of required knowledge.

Design Principle 50 (Strategy for Selecting Knowledge Sources)**.** *Externalized knowledge can be selected from sources that frequently deal with similar problems and exhibit desired response.*

Design Principle 51 (Knowledge Generation Strategy)**.** *Information management should address creation of new information on the basis of ongoing events and response.*

Design Principle 52 (Knowledge Provision Strategy)**.** *Information management strategy should include recommendation of relevant information to the users based on their assumed role.*

2.4.2 Situation Awareness System

Requirements of dynamic information management are identified. Situation awareness is identified as an important feature for dynamic information management. For building system to support required information management capability, various design principles have been identified. These design principles should be realized in software environment. Figure 2.11 depicts the role of the proposed solution.

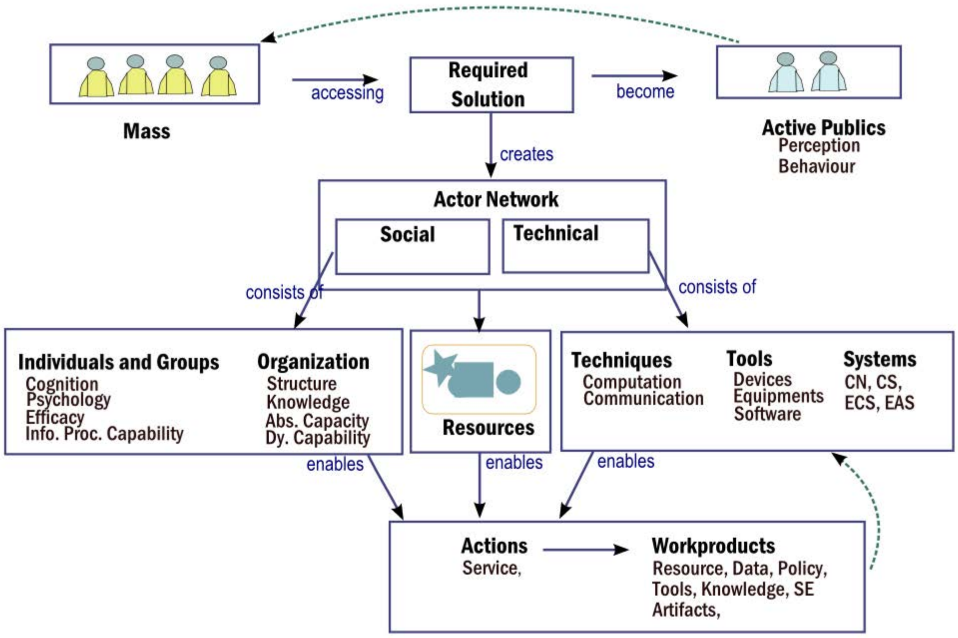

Figure 2.11: Organization Behavior Point of View for Situation

Definition 8 (Situational Awareness System)**.** *Situation awareness system is an information and communication technology based system that is necessary to impart situation awareness capability to individuals and organizations in complex dynamical systems. The system is based on design principles identified as information management strategy for situation awareness.*

Design Principle 53 (System Strategy for Information Management)**.** *Situation awareness system needs to be utilized for required processing and communication of information.*

2.4.3 Situation Awareness Configuration

Situation awareness system is projected as a system realizing dynamic information management methodology to support stakeholder situation awareness. The scope of the system is provided in the concept.

Definition 9 (Situation Awareness Configuration)**.** *Situation awareness configuration is an* *execution environment that provides tractability and holds enterprise continuum, communication environment for actors spanning multiple situation awareness systems.*

Design Principle 54 (Coverage Strategy for Situation Awareness)**.** *A situation awareness configuration is required to establish live traceability of work products to facilitate their management and reuse by appropriately connecting all the required situation awareness systems.*

2.4.4 Situation Awareness Process

The requirement of building situation awareness configuration leads to the identification of additional requirement of consistent collaborative efforts. Such requirements include explication of a process guiding the collaborators in carrying out the allocated task. The process content provides information about various roles, their responsibility and guidance to carryout the specific tasks. The guidance information is created and shared by the domain experts. The process also allows task management, monitoring, traceability and other features that facilitate collaborative effort.

Definition 10 (Situational Awareness Process)**.** *Situation awareness process is a method guidance recommended for stakeholders in meeting their information needs specific to role(s) assumed in given configuration.*

Design Principle 55 (Process Strategy for Information Management)**.** *Rich process guidance is to be defined and utilized in building situation awareness configuration to meet the information need of stakeholders in complex dynamical system.*

2.5 Summary

In this chapter, the salient characteristics of the complex dynamical systems are identified. With reference to relevant theoretical framework, various features and elements of the system are introduced. Handling of these elements with information management strategy is also discussed. Based on the theoretical framework, the identification of situation awareness for complex dynamical system is carried out. The situation awareness and information need are characterized for determining the new strategies for information management. Three problems are introduced that poses challenge in identification of information needs. Design principles are identified based on relevant theories and their applicability in achieving situation awareness. Situation awareness is defined for individual and organization point of view.

Chapter 3

Modeling for Situation Awareness

3.1 Modeling Strategy

In previous chapter, situation awareness capability is identified as a core requirement to maintain the desired status in complex dynamical systems. It was established that situation awareness requirements are unique for individuals and organizations. With multiple entities continuously exhibiting dynamic behavior, their impact on other entities, resources and overall state of affair therefore is also changing. It causes constant change in situation awareness requirement. To identify and characterize such requirement, the concept of information need is discussed. Impact of the relationships of various roles, events and other aspects on information needs has also been identified.

With this realization, the next task is to engineer a situation awareness system with the features identified in Section 2.4.2. From information management point of view, the task spans all three aspects of dynamic information management introduced in Section 2.4. This includes technology oriented, content oriented and knowledge oriented management of information for complex dynamical system. As a common reference point, knowledge should be captured and encoded from technological, organizational and domain specific aspects. Also, the utilization of this knowledge base in handling the actions and behavior of individuals is to be achieved.

A consistent representation of all the relevant elements in UoD is required. The resulting knowledge base should not only establish the complex interdependence and dynamic behavior of entities, but the rules towards controlling the behavior is also required. The requirement is therefore a comprehensive conceptual representation and an appropriate information processing strategy that can realize required information flow in given UoD. A conceptual representation should be created with a modeling strategy that allows capturing and encoding to characterize features of various elements. In order to make the resulting representation useful, an appropriate information processing strategy to be defined that allows instantiation, monitoring and appropriate handling of the elements in UoD. In other words, a strategy is required to answer the second topic in problem definition identified for this thesis in Section 1.3.

The task is challenging due to scope and nature of the system. There can be many elements in the systems that can be found relevant from different aspects. The same elements can be characterized from multiple domain point-of-views. This requires a conceptual modeling effort from multiple collaborators. Some representations may already be available that can be reused. Certain domain specific aspects and general characteristics are common in systems all over the world. Such common conceptual representation is reusable, whereas some features are local-specific and must be modeled individually. The conceptual representation derived from reuse and modeling of the required components should be consistent so that it can be appropriately employed for information management. The creation, processing and delivery of information chunks should also be handled consistent to the modeling strategy. This includes identification of instances, monitoring and detection of events, handling of information exchange and related information processing aspects in real time.

This Chapter introduces the conceptual modeling required for situation awareness. The discussion is not only restricted to conceptual modeling of all possible concepts in UoD, but is also introduces information processing strategy that allows instantaneous information processing required to sustain the desired information flow as defined in the rules. As an outcome of these strategies, the realization of desired feedback control loop in complex dynamical system is explained.

This chapter discusses the following aspects:

- A modeling strategy for conceptual representation of all possible entities in UoD along with complex interdependence and dynamism.

- An information processing strategy for the runtime for all available entities in UoD as they exhibit the complex and dynamic behavior.

3.1.1 Purpose of modeling

There can be many ways of modeling the entities and their behavior to meet the identified requirements. The requirement may include modeling of real-life or abstract entities with their possible states, state changes, events, and related aspects for identified objective. By following the modeling process, a system can be modeled from many point-of-views with specific accuracy and highest possible frequency. But if the employed modeling process is not based on the required specification, it may result in an underestimate or overestimate situation that subsequently affect the information management. It is a common practice that conceptual modeling and physical design is carried out in initial phase of information management practice. All the provisions for information processing and handling is based on this initial commitment to model. Any change discovered later may require considerable changes in the whole system that depends upon that. Hence, it can be said that conceptual modeling should be handed with due estimation of the requirements.

As present discussion is targeted towards complex dynamical system, the complex interdependence and dynamism are expected to continuously evolve over period of time. Hence, any modeling strategy employed must not only handle the changing requirements, but it must facilitate evolution so that emerging system behavior can be characterized. Apart from this evolution feature, few additional features are imperative for the purpose of situation awareness, and therefore they must be recognized as one of the core purposes of the modeling process.

Modeling for Complexity The model is targeted to represent the UoD by accurately capturing the state of affairs. This includes complex relations and interdependence that may exist among entities.

Modeling for Dynamism The model is targeted to capture the dynamism that is evident at various levels. Different types of entities have unique lifespan and may exhibit dynamic behavior. Such dynamism is core element of situation awareness and should be captured accurately in appropriate spatial, temporal and conceptual dimensions.

Modeling for Ramification Modeling should be able to reveal how one state change will cascade in system of complex interdependence. Ramification feature is also core to the situation awareness capability as it establishes the effect of changes being propagated among the interrelated entities. As discussed in Chapter 1, the coordinating agency has to monitor every possible dynamic entity that may harm or heal the over all system state. The ramification feature allows estimating the impact of observed individual behavior. This behavior may be very short lived, but its impact on the system and the response required may be at different scales.

Modeling to Specify Granularity As dynamism indicates specific granularity in temporal scale, all possible granularities and other relevant dimensions must also be considered. Spatial, temporal and conceptual are three basic dimensions required for situation awareness. Modeling process should capture various granularities and hierarchies that can be identified in each dimension.

Modeling to Specify Traceability The purpose of situation awareness also includes the actions required for desired outcome in the system. The UoD is therefore monitored for entities, processes, activities, resources and outcomes. Outcome of one entity may act as input to another and sometimes can be useful to many others for longer period of time. Information about reuse and re-employability of the available outcomes therefore contribute to the situation awareness of related stakeholders. By monitoring and registering the activity, its outcome and any reuse of the same, allows the establishment of traceability. This traceability feature enhances the situation awareness capability of stakeholders involved in allocation, creation, sharing, or reuse activity.

Observability Important aspect of modeling is the provision for empirical evidence. Specific elements of the UoD can be observed by sensor, but some of them cannot be monitored without human observation. Hence, observable properties of the system components must be identified along with appropriate specification of observation. This purpose will explicate not only the property, but also the specification of how property can be measured. And in given modeling scenario, how the modeler wants specific property to be observed, so that it can be consistently measured by the sources of information.

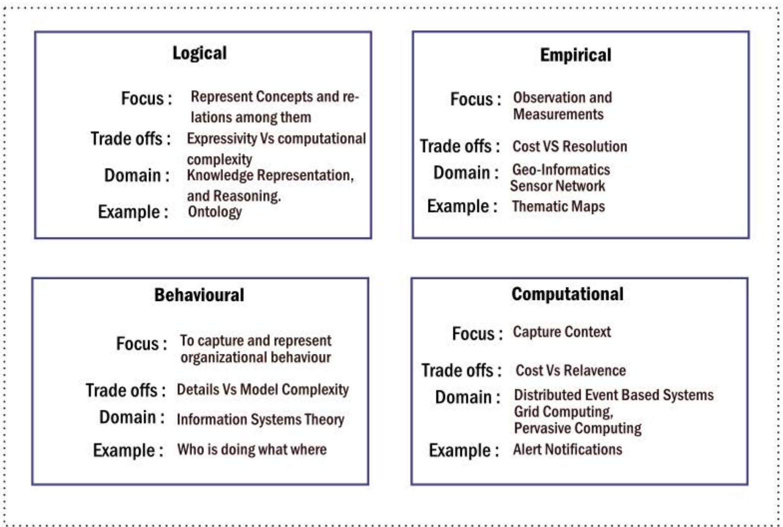

Figure 3.1: Domain View

3.1.2 Dimensions of Modeling

Dimensions are viewpoints of looking at the system. The same systems can be viewed and modeled from various domain view-points. These viewpoints may be diagonal to each other, having no common concern among them even if they allow modeling of the same entity. Yet, each viewpoint facilitates the purpose of modeling by characterizing specific aspects, behavior and properties of a given entity. Following domain specific viewpoints are found relevant for the modeling process:

Empirical Dimension The observability feature introduced earlier refers to the modeling of entity from empirical dimension. This is a discipline involved in setting methods and techniques for carrying out accurate observations and measurements for observable properties of entity. Accurate measurement of property is essential for identification and evaluation of instantaneous state of the observed entity. An observation and measurement made enhances situation awareness capability of related stakeholders. Such observations made in standard manner improve the reuse of information and facilitate consistent information processing at required granularities.

Logical Dimension Conceptual model includes capturing complex relations, interdependence and dynamism among entities. Such aspects must be consistently modeled and represented. This requires careful selection of formal representation that allows features like consistent notation, reasoning, verification and consistency checking.

Computational Dimension The task of information processing is complicated as stakeholders involved in monitoring, processing, and using the information may not be located at a single site using the same system. The generation of information, processing, handling, and communication is to be carried out on computing systems scattered across larger geographical area. Such tasks can be scheduled for execution at specific temporal frequency. The information generated as outcome of such tasks should be stored and made available to suit the need of the users. Hence, consistent effort is also required from computational dimension to support the situation awareness capability.

Organizational Dimension Evolution of organization, participation and behavior of individuals, stakeholders and publics are found to follow principles established in organizational behavior domain. Information management strategy must consider these aspects as they determine the scope of situation awareness of individuals.

Collaborative Dimension From the four domain viewpoints introduced so far, it is evident that the required information management practice involves considerable amount of collaboration of many individuals and organizations. For sharing of responsibility, in providing coverage and for other reasons, tasks needs to be allocated among collaborators. Hence, interaction, sharing and other aspects should also be monitored and handled to complete the information management from collaboration point of view. For example, modeling of entity property cannot be completely achieved by single user or organization. Organization can only provide local and domain specific aspect of property; rest should be accepted from the contribution of others. The collaborative aspect is required to integrate the effort in the different domain to support consistent modeling and information processing involved in situation awareness.

3.1.3 Modeling Prerequisites

Modeling procedures cannot start without first determining the context. As indicated, situation awareness is highly affected by the goal of the individual and organization. The prevailing entities, processes have roots in the identified organization specific terminology. Procedures and required actions are also therefore captured in such policies. Hence, to define the scope of the system, characteristics of the system, it is prerequisite to have access to existing frame of reference. Items that provide such frame of referees are discussed in what follows.

Goals There can be multiple frame of reference a universe of discourse may exhibit. Goal is something that provides specification of status that system should be in. Hence it determines the desired outcome. Hence, goal is the first requirement toward modeling process. It also allows tracking the activities growth and development congruence is necessary.

Plans Plans provides step-by-step methods for achieving the identified goals. It says how organization is willing to achieve the desired goal. Plans consists both qualitative and quantitative parameters for desired outcome. It is also necessary in determining the boundaries. They provide context and common frame of reference for activities. They indicate commitments and adherence to specific standards. They provide important information like local facts, standard operating procedures and organizational role hierarchies. They are based on legal obligations of the organizations and therefore must be followed.

Existing Resources Survey of available and employable models, systems, infrastructure that can be directly utilized in information management task. No system is built from scratch. Some already exist; some will be created and made available in future. Hence, it forms a dynamic set of a system. All the existing entities must be incorporated or at least studied.

3.2 Entity Model

The coordinating agency is one special type of organization that has responsibility of maintaining desired system state in the given Universe of Discourse (UoD). Information about elements in UoD that is relevant to organization is encoded in the policy. Similarly, possible states, behaviors and appropriate responses of the agency are also defined in the form of plans. It also includes the information about various roles and the involved resources. The allocation of responsibility regarding decisions, actions and resources are also explicated along with standard operating procedures.

As defined in possible world theory, there can be many possible ways the world can be in at given time. The policy, plans, and goals provide boundary for comprehension of the instantaneous state. They restrict the scope to identified domain knowledge and provision that are made. Hence, such information is very critical in covering the complete scope yet; restricting it to what is necessary as identified by the agency.

For example, the scope defines how organization has made plans to provide specific services or products in area under its jurisdiction. There can be many services imparted to an area as practiced and known in the other parts of the world. Yet, depending upon the local policy and resource availability, the organization may restrict the scope to manageable few. Organizations having similar scope may adopt different approaches to achieving the targets. They are defined in the form of organizational strategy. Policy is information about critical decisions or standpoints that the organization may take when conflicting interests arises.

Once the policy is derived, the next task for information management is to fulfill information needs of the stakeholders. This is done by first identifying and capturing important entities available in the given UoD. They can be real world or abstract entities; they can be identified on the basis of relevance to plan. For their lifetime in UoD they should be monitored, communicated and handled as per the plan.

The challenge is in identification and selection of all the relevant entities. The selection strategy should involve identification of various qualitative and quantitative features of the selected entities. Among these tasks, identification of organizational entities and stakeholders are missing. These are the individuals that are related with the goal of the organization. They may be abstract or concrete objects playing various roles. They may have certain attributes and behaviors. They can participate in or may affect organizational activities. The missing link here is the criteria for the selection and characterization of all the participating entities.

A modeling strategy is required to characterize the behavior of involved entities. This strategy is required to select the relevant entities from many available from UoD. For selected entities, criteria must be specified for identification of required properties for further characterization. There can be many types of entities in UoD. Those entities should be handled by information management strategy; hence the model should be able to characterize the nature, behavior and interactions of entities. It should be able to come up with all properties that can be monitored. It should also provide method to collect values and other parameters to enable situation awareness.

The required entity model is expected to provide basis for comprehensive representation of the UoD. This characterization and strategy will generate list of concepts and their properties relevant in given context. This selection strategy for entities bridges the missing gap of unidentified relevant elements in UoD and facilitates further tasks towards meeting the information needs. It allows identification of all long and short-lived concrete and abstract entities.

The modeling strategy has specific prerequisites that are mandatory for the identification process. As the criteria for selection of entity comes from the policy and related documents, the appropriately written policy documents are one of the important prerequisites. The domain knowledge covering important disciplines like science, technology, management and other aspects provide important framework for capturing domain specific entities. Given the policy, standard operating procedures, goal and other references, the entity model should be able to find out all the relevant entities in given UoD. Apart from the visible entities with physical features, the model should also be able to characterize and capture logical or abstract entities as defined in references discussed earlier.

In any information management practice, conceptual modeling task is basic starting point. From simple database management application to data warehouses supporting complex analytical capabilities, the effectiveness of the application is determined by the conceptual modeling carried out.

Basic elements of the entity modeling strategy are the entity, general properties, and role-specific properties. The role-specific properties include the utility footprint, related states, transitions and other features indicating static and dynamic behavior of the entity.

The challenge in entity modeling task is due to dynamic set of entities that may show up and disappear on the runtime. The complex relations are established by domain experts, are sometimes considerably complicated to handle when realized in system environment.

Entity model is proposed as a block diagram. This diagram provides a visual reference for identification of various aspects of entities in given UoD. This template also allows identification of related entities and processes. Figure 3.2 represents outcome of an entity process. The entity modelers may include or exclude the entity template parameters to suit the application specific needs. These are the typical parameters that will help fulfill the need of information. All entities

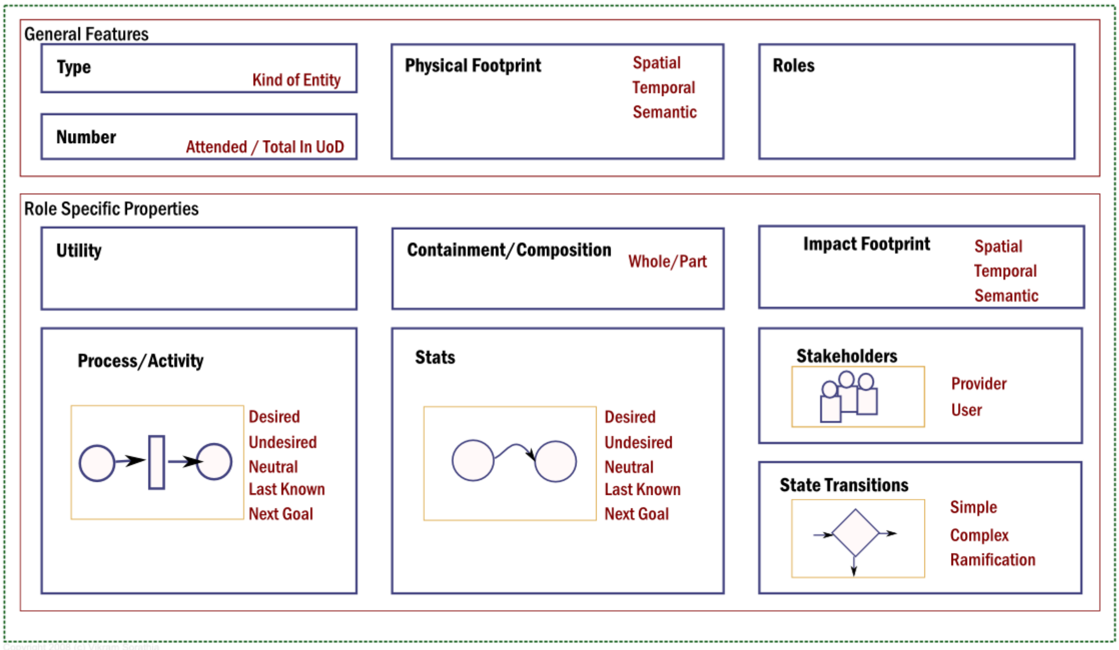

Figure 3.2: A Sample Template Selected for Modeling Entity

are denoted with e, are member of an entity class $\mathcal{E}$.

$$\mathcal{E} = \{e_1, e_2 \ldots e_n | n \in \mathbb{N}\} \tag{3.1}$$

Equation 3.1 includes all entities visible in UoD or are abstract as identified in roles, plans, references applicable to given UoD. For example, a person is an entity. The role assumed by some person is a logical or abstract entity.

3.2.1 General Features

As indicated in the Figure 3.2, entities visible in UoD or those mentioned in goal or policy statements have some general features. These general features include physical property, type etc. These terms provide identification of relevant domain and required knowledge. Especially the roles and impact, behavior and characteristics are specific to an entity.

$$e_{type} = \{abstract, concrete\} \tag{3.2}$$

Entity count is the property that provides the important information about the population density of the entity being modeled in the UoD. This also specifies how they can be handled for specific purposes. As the outcome of number counting is always a non negative integer, the e_{count} is always considered to be member of $\mathbb{N}$ as indicated in Equation 3.3. In case precise number is not relavent, then ordinal scale can also be useful. For example, {10s, 100s, 1000s, 10000s.. etc}. In case resouces are to be modeld as entities, they can also be assigned units of quantities.

$$e_{cont} \in \mathbb{N} \tag{3.3}$$

In certain cases, the utility of the static entity can be in the form of service. In such case a fixed area is covered under is provided specific utility for certain time period. This scope of utility is known as footprint.

$$\begin{aligned} e_{p-fp} &::== \langle Semantic|Spatial|Temporal \rangle \\ Semantic &::== \langle Concept \rangle \,|\text{where } Concept = \text{role assumed in UoD} \\ Spatial &::== \langle Area \rangle \,|\text{where } Area = \text{space occupied in UoD} \\ Temporal &::== \langle Duration \rangle \,|\text{where } Duration = \text{time visible in UoD} \end{aligned} \tag{3.4}$$

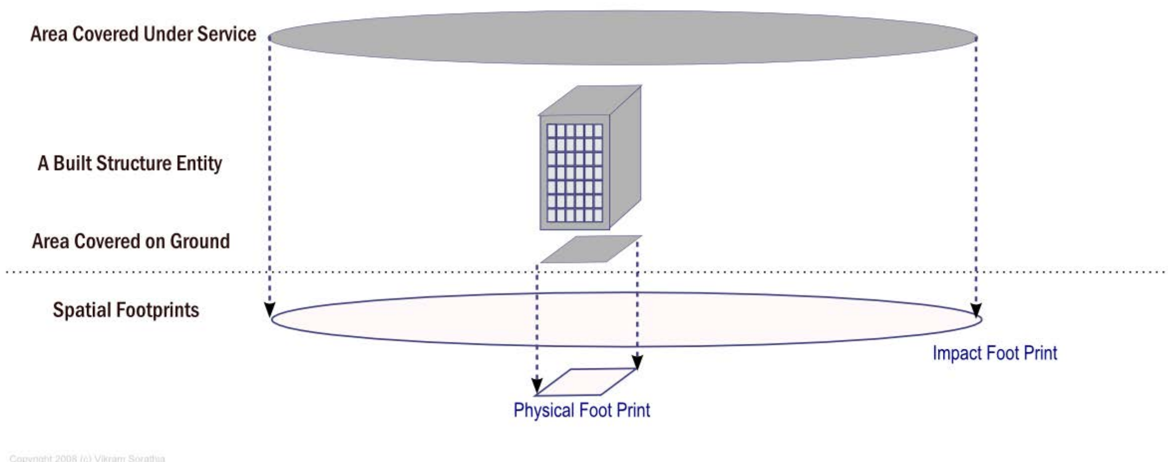

Figure 3.3: Example of Spatial Footprint of a modeled entity

Equation 3.4 provide role footprint or impact footprint. Entity is visible in UoD for some period of time. It may occupy some space and it can be classified or identified as object or instance of some class defined in organizational knowledge base. This is identified as footprint of role.

Spatial span in the physical property is the area covered. This can be a point, line or polygon that is covered. It may have spatial relation with other objects or instances as well.

Temporal span in physical footprint is how long the role will be visible with the same properties in the given universe of discourse. This information is required to determine the frequency at which particular entity should be monitored. This also provides overall life cycle of a single instance of the entity. The quantity therefore provide estimate about level of effort required to deal with it.

Concept footprint is the word, or context defined. For example concept part of physical footprint of a person seeking education is Student. The footprint can be more expressive, like a graduate student, a medicine student, a student of advanced degree with specialization branch. Concept physical footprint is the name the object is identified. As the similar concept or the closest match should be identified and recognized under some domain of reference. Entity can play different roles in given UoD. Roles playing can be true for some period of time. Roles are based on goals, and capabilities to carry out on given time. It may play more than one goals. For more than one role, based on the roles its impact on information flow can be seen. It may require information it may provide information and itself can be subject of information of other entities. For monitoring and control purpose as role has unique implications they are separately modeled as role specific features.

$$e_{role} \quad \mathbf{R} = \{r_1, r_2, r_3 \ldots r_m | m \in \mathbb{N}\} \tag{3.5}$$

3.2.2 Role Specific Features

As represented in Equation 3.6, the role can be any domain specific abstract entity that participate in domain process and provide some utility. Utility is a unit action or contribution imparted by entity in UoD. A utility can be in the form of a service nature for example, given in Equation 3.6, an instance of a vehicle is an entity. This entity can provide transportation utility when an entity is providing role specific utility; it may be for some time.

$$e - r_{util} = \{Transportation, Communication, ...\} \tag{3.6}$$

$$\begin{aligned} e_{i-fp} &::== \langle Semantic|Spatial|Temporal\rangle \\ Conceptual &::== \langle Concept\rangle \,|\text{where } Concept = \text{type of impact} \\ Spatial &::== \langle Area\rangle \,|\text{where } Area = \text{area of impact} \\ Temporal &::== \langle Duration\rangle \,|\text{where } Duration = \text{duration of impact} \end{aligned} \tag{3.7}$$

Impact footprint is defined in similar way to the physical foot print as indicated in Equation 3.7. Yet the values are difficult to determine, as these values are mostly the estimated values, and may differ in actual terms. One of the components of the footprint is in the form of spatial footprint. The utility rendered by the entity may cover an area. This area not necessary be the administrative unit, but can be a boundary. Sometimes the impact or utility cannot be identified with crisp actual boundary, and some estimated buffer is considered as area of impact. The distinction among fiat, bona fide and estimated boundaries are depicted in figure 3.4.

Spatial part can be an area denoted by any of these three boundaries. The temporal impact footprint is the time of duration between which entity provide utility. This is temporal footprint active as a role in given UoD. The conceptual impact footprint is the conceptual representation of utility provided by entity, and its impact on other entities and processes in given UoD.

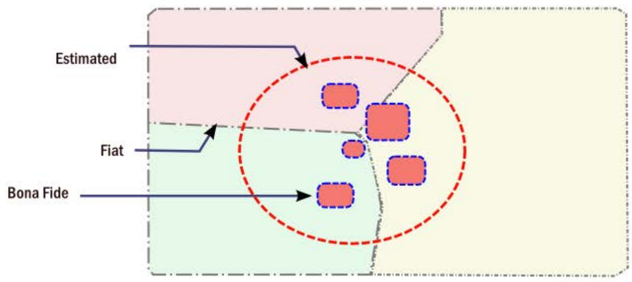

Figure 3.4: Types of Spatial Boundaries

$$e_{states} \quad \mathcal{S} = \{s_1, s_2....s_n | n \in \mathbb{N}\} \tag{3.8}$$

Various states are defined in Equation 3.8. These are set of identified distinct sates the entities can be in. Based on domain knowledge and complexity of entity, it can be found in unique state of existence. These states are result of some process or activity. The entity is subjected to the identification of state can be complex but important as it provides valuable information about role, especially for monitoring and tracking purpose. The state can be found either in desirable ($\mathcal{D}$), undesirable ($\mathcal{U}$) or neutral ($\mathcal{N}$) states. These disjoint sets of possible states are vital in identification of current status and next desired states.

$$\begin{aligned} \mathcal{D} \cup \mathcal{U} \cup \mathcal{N} &= \mathcal{S} \\ \mathcal{D} \cap \mathcal{U} \cap \mathcal{N} &= \phi \end{aligned} \tag{3.9}$$

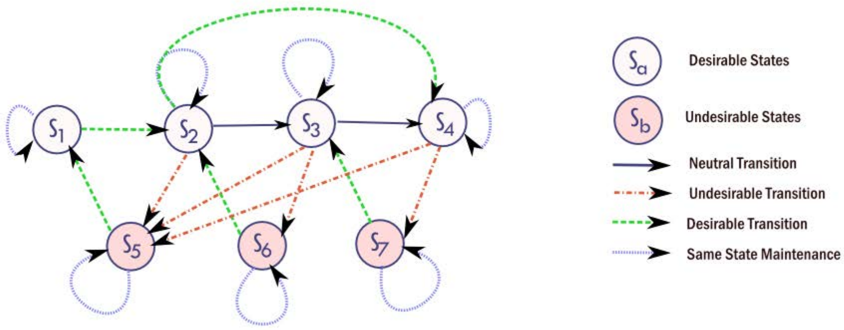

Figure 3.5: State Transition Experienced by Entity

Transactions are outcomes of activity that changes the state. Not all transactions are necessary to monitor. Yet some transactions that are from desirable state to undesirable state are not desirable, and hence they are sources of event. State transitions are therefore important and must be tracked.

$$e_{StTran} \quad \mathcal{ST} = \{st_1, st_2....st_n | n \in \mathbb{N}\} \tag{3.10}$$

Design Principle 56 (State Transition)**.** *A state transition signifies outcome of a process on a specific entity. It is necessary to establish how these transactions are reflected by change in property values.*

Processes that affect entity states the transitions cannot take place in absence of any external force. Or to keep the same state, much external help and resources are required. Both indicate existence of process or activity.

$$e_{Process} \quad \mathbf{P} = \{p_1, p_2....p_n | n \in \mathbb{N}\} \tag{3.11}$$

Processes that entity affects include those, which utilizes entity as input or as some utility attached to it. That is important for ramification. Identification of process also allows identification of other entities, processes and roles and possible states. As process requires other entities in form of input, output roles associated with it they provide identification of other entity and process. Hence, process property is important for modeling as well as tracking perspective of information management.

3.2.3 Example

Example modeling process may result in identification of multiple roles with associated entities and processes as indicated in 3.6. The picture indicates different roles played by instance of

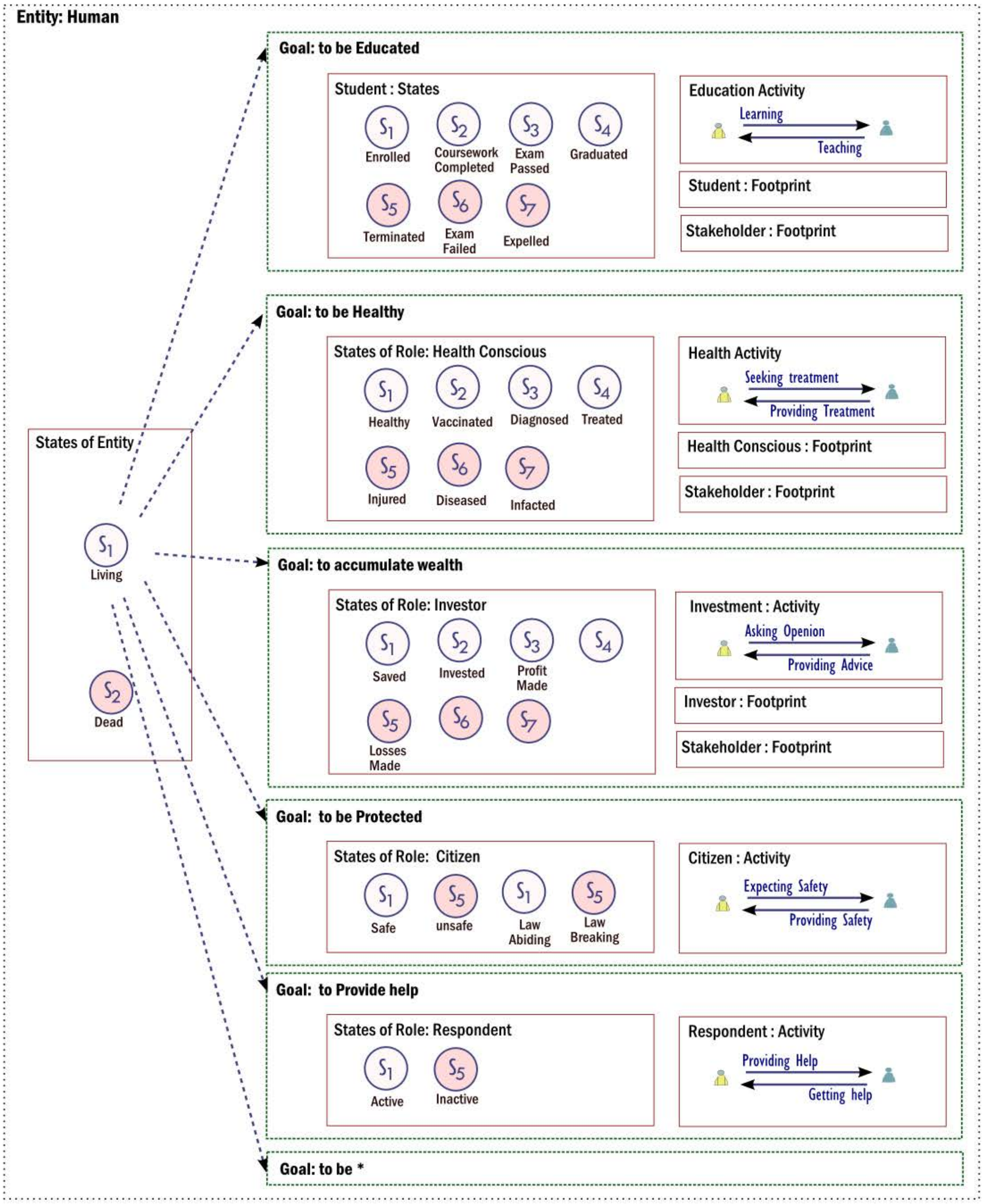

Figure 3.6: Entity Model Example

an individual human being. The desirable state is living state. But that does not stop there in living during the lifespan but it plays various roles. A goal to get educated, if selected then it comes to special type of relationship with stakeholders-the other human being with complementary goals. There are specific states, activities and transitions associated with the role similar to this. More the roles an entity assumes more it is likely to come in to relation with goal, role related concepts. And hence multiple concepts and properties can be identified for each one of them.

In given example, an instance is known to participate as role of student, an investor, a health service consumer, a citizen, and a respondent. In all these cases, desirable states and undesirable states can be identified. The assumption of role marks the event for which, further monitoring is required. Therefore, it has specific lifespan under overall lifespan of entity. This also includes event space. Information needs of individual and respondents and stakeholders.

Equation 3.12 represents an instance of a human entity playing various roles. Each role along with the current state is represented as e_{role}^{status}. Hence, an entity playing multiple roles at given time, is represented with a set of roles and corresponding status. Observation of role and status of a given entity over a period of time signifies the situation and events.

$$\begin{aligned}
\text{at} \quad t_1 &= e_{student}^{enrolled}, e_{citizen}^{safe}, e_{health-conscious}^{healthy} \\
\text{at} \quad t_2 &= e_{student}^{enrolled}, e_{citizen}^{unsafe}, e_{health-conscious}^{injured} \\
\text{at} \quad t_3 &= e_{citizen}^{safe}, e_{health-conscious}^{healthy}, e_{investor}^{invested} \\
\text{at} \quad t_4 &= e_{citizen}^{safe}, e_{health-conscious}^{healthy}, e_{respondent}^{active}
\end{aligned} \tag{3.12}$$

Summary of Entity Model Approach

The purpose of modeling strategy discussed here is mainly to identify all relevant entities in given UoD keeping the organizational documents as reference. The outcome is a list of entities with features. The modeling is not yet complete, as there are many other entities, which will be identified in following steps of the modeling process. The modeling is not to be compared with traditional Entity Relationship (ED) Diagrams, as there is no mention about the representation strategy. A suitable strategy for representation is identified in subsequent steps.

3.3 Process Model

Entity modeling process is carried out based on plans, goal and related reference. The model decided for specific purpose allows capturing specific features of the entities. This includes footprint of entities and state transitions. This allows the impact on entity from others and also impact of entities on others. Both ways if one entity is covered in modeling process, it allows identification of others.

Next task is to enable identification of various processes and their characterization. Entity model included state transitions. It is the result of the process or activity. Also entity plays role, have some utility. Hence it also triggers activity of some kind. Activities are also in form of standard operating procedures (SOP). All together provide rich source of content for the processes. All relevant processes and activities are to be identified and appropriate content should be collected.

Selection of all activities and processes are not limited to identification but also determine characterizing properties that will be required for identified goals or plans. Priorities and importance of specific concepts and properties are further identified. For identified concepts, the interrelations of entity and process should be defined. A proper strategy for identification of such aspects needs to be defined.

For meeting these requirements a model is required for comprehensive characterization of processes and activities. Model facilitates Identification strategy to find activities and important properties. The characterization of process along with dependencies needs to be established. The model included appropriate representations strategy establishing the resource dependence. Among the involved resources, some resource consumable and other reusable resources may be available for long running processes. Model should appropriately establish the constant inflow of consumables and scheduled availability of reusable resources for planned executions.

This model will provide guidance that will fulfill the selection problem and thereby allows comprehensive identification of process details that is the next goal. The model provides template with specific information blocks meeting information need of specific aspect including information on instantaneous status, guidance, schedule, resources etc. It also helps demonstrating dependence with other process. Here, the distinction in natural and human process is made clear based on the identified dependency.

As a prerequisite to this modeling process, complete understanding of various processes, activities and states is required. Involved entities, resources and any other requirement of the process should be known. A suitable entity model that allows identification of relevant entities and their property is also a prerequisite as, they allow identification of processes. Many of these processes being model can be domain, industry or organization specific. Single person cannot provide all the inputs at single instance of time. A mechanism of repository is needed that allows modeled subcomponents for activities and process. Resource allocations specific to the process are according to the priorities defined in plans or policy; hence all the applicable policy documents are prerequisites.

Model outcome is template that allows characterization of services, volunteers, and natural processes. The template includes characterization by means of parameters required to review features of process model reveals distinct features. Dependency and inter relation with other processes. It allows representation of guidance and other management related aspect of the system.

Listed services with their dependence are used by the knowledge engineer in creating various assertions, relations and rules regarding the service. Identified services and related knowledge can be consistently collected and updated by knowledge engineer. This fills the information gap (uncertainty) regarding the process or activity. This enables identification of process / activity and related guidance and info to user.

Various difficulties can be identified in this regard. The decision regarding process like how many or which process should be included. The decision regarding the characterizing features to use to define different types of process or events. Identification of levels of detail in the form of process steps is also a critical challenge. The decision regarding information requirement task that needs to be embedded with each process step is also critical. The decision regarding provisions for tracking the progress also introduces difficult engineering challenge.

In decision enumerated above, numerous options can be identified. Possible types of process are: industrial, natural, human activity, volunteer activity routine activity etc. Many relevant theories related to the involvement of actors can be identified. This allocation should include information state from entity model.

The proposed modeling strategy for modeling process in given UoD is recommended individually for natural processes and human activities.

3.3.1 Simple Activity Model Template

A simple process in nature is shown in Figure 3.7. This simple natural model involves natural entities and resources. They are natural as they are performed by force of nature, follow universal laws of nature and outcomes are certain. This is manifested in form of change in natural resources. They can be monitored managed and controlled with very little capability.

The natural processes are triggered yet they indirectly/directly affect entities identified by entity modeling exercise.

The activity includes input output and a staged process. Input is natural resources or power used. Working environment or climate is necessary for the process to take place. The workflow is the known or established scientific fact. As this follows laws of nature, it cannot be much tempered but it can always be monitored in different stages. At the end of process, outcome can be observed.

As indicated in figure the preconditions, trigger, resources required are vital input. The output provides information about outcome of process. The state change in system is also stated in model. This information is useful in estimate ramification of the event happens responding to ongoing process.

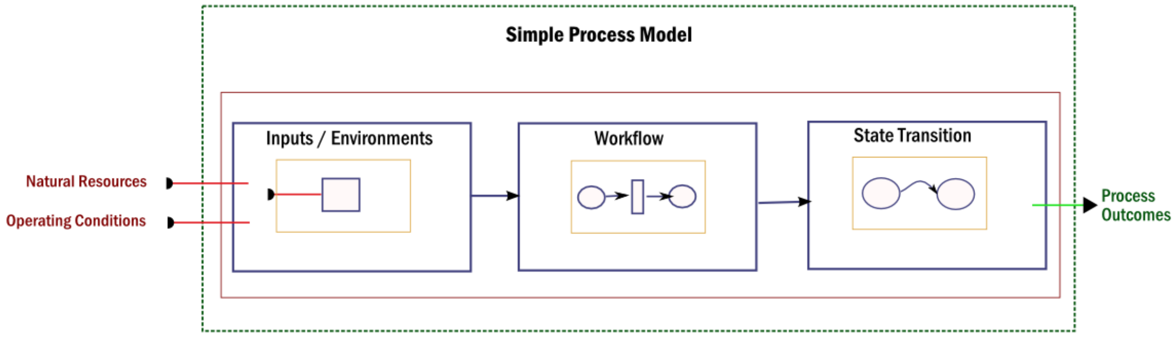

Figure 3.7: Process Model Simple

3.3.2 Human Activity Model Template

Human activities are more complex and do not follow laws of natural and not so predictable. The outcomes are also not certain. They can be affected in different ways hence modeling strategy should be able to capture this aspects. As they do not follow laws of nature, the theories about factors that govern human activities must be included in the model. Human activity can take place as isolated process or can take place on ongoing basis. As a result of organizational commitment the model therefore contains two parts of features.

Human aspect

Role is the participation of individual in executing or carrying out specific activity. Role is a logical or implied relation that specifies the kind and nature of participation. It may provide resources, make utilization, control, start, stop, or facilitate a process. A process may have single role multiple role or even hierarchy in the role. Specific task needs to be completed and if not the whole process may get affected, the role is defined in process document or can be inferred one person may play more than one role due to delegation. This can be also a self-imposed activity as well.

Information need is another logical concept that specifies the requirement of information a role player may require to complete the role. The interaction pattern is information seeking behavior. The term is also associated with uncertainty. The characterization of uncertainty for given process is encoded as information need block. The information need include information about entities associated with the process, the states of the entities, the state of inputs required to carry out the task also status of output expected out of the task. This also include information regarding role delegation or to attract information seeking to commit.

Based on requirement and assumed role, an individual may identify the problem and task that needs to be done. If it is acceptable he will commit to the task by another logical component like a decision box. This follows the concept of active publics. If a person commits to play a role is based on many known factors problem cognition can take place and can be encouraging to an individual to take up the role. If role is taken it is noticed in task management component.

The task management is required component as many roles may get information about job needs to be done but all many not be employed to complete the task. Thus who is doing what on what date and what is status is done in task management part. This block holds information about who is doing what and when and where, and the work schedule. If the task is a long running activity, the role like resource provider, delivering inputs may have created a schedule till the end of the task.

System Specific Component

The proposed model has certain system specific components. Input and output are critical, as they must meet the requirement specified as an input. It may require another service consumable resource. In simple requirement reusable resources should be present there for all subsequent use.

Workflow is the sequence of task to be followed. The typical workflow consists of many intermediate tasks, entry and exit conditions and other critical aspects governing the process. Long running tasks are to be intermittently reported for the monitoring of progress. The status is stored in task management. It is possible to derive workflow from plan. Information about workflow is the reference, in information need of user.

State transitions are the outcomes of the process. They signify changes in the property, thereby allows establishment of entity state changes. If a particular state change is reported in an entity, it is important to establish the workflow or activity responsible for the given state change.

In summary, the proposed model, components for modeling human activity or process. They not only provide typical components like input output model, but it also contain organization management aspects of the task. It identifies the information needs. To remove the uncertainty in carrying out task, this is carefully designed to meet the information exchange behavior of publics. It supports information seeking by providing information about process and its details etc. This information is utilized in scheduling to support those who already are part of active publics. By this means, it allows reduction of duplication of efforts.

It also support information processing behavior of masses by publishing status of task, process it allows and the recognition that something needs to be done. The information need also include workflow and guidance required to complete the task. This reference helps consistency in process. Rule drive access to information and role players are asked to follow steps. In case of change in work flow it is immediately reflected, thereby provides facility to upgrade to suit the dynamism.

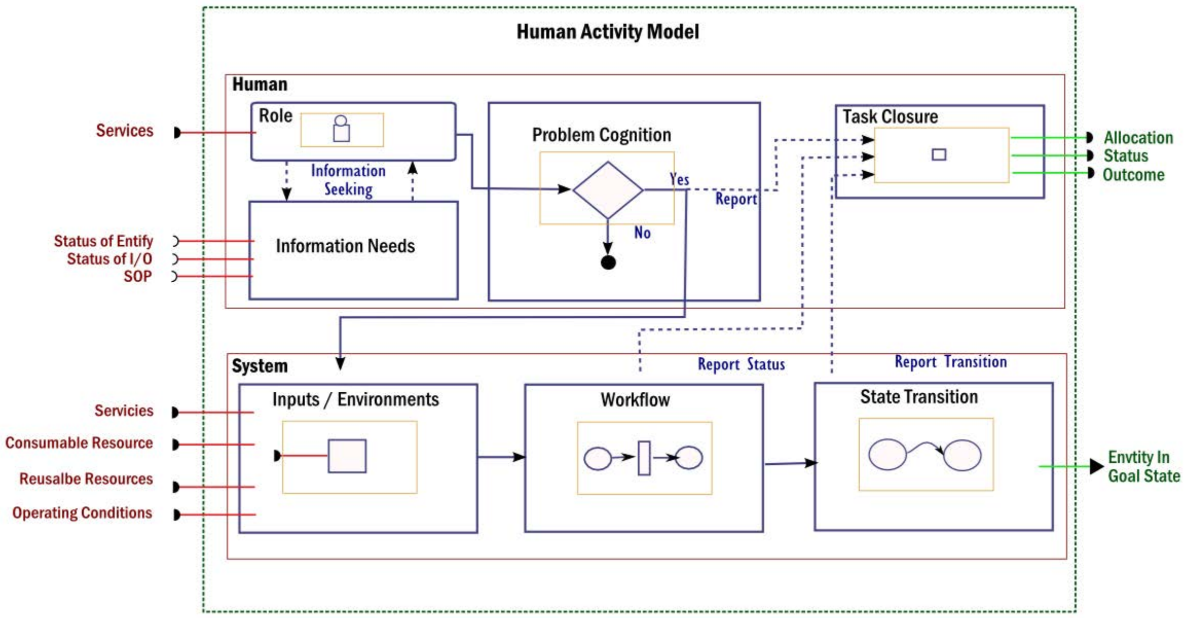

Figure 3.8: Process Model Human Activity

3.3.3 Process Representation

The goal is to reveal the inputs, as they are the enablers or preconditions for process. To initiate their status and availability is important part of reference. To reveal the output the nature and schedule of output with knowledge of current state is also another important aspect. To reveal the intermediate goals that governs the input, the output the behavior or role players. They are the decision-making blocks that affect and control the workflow. There are many possible ways to represent process.

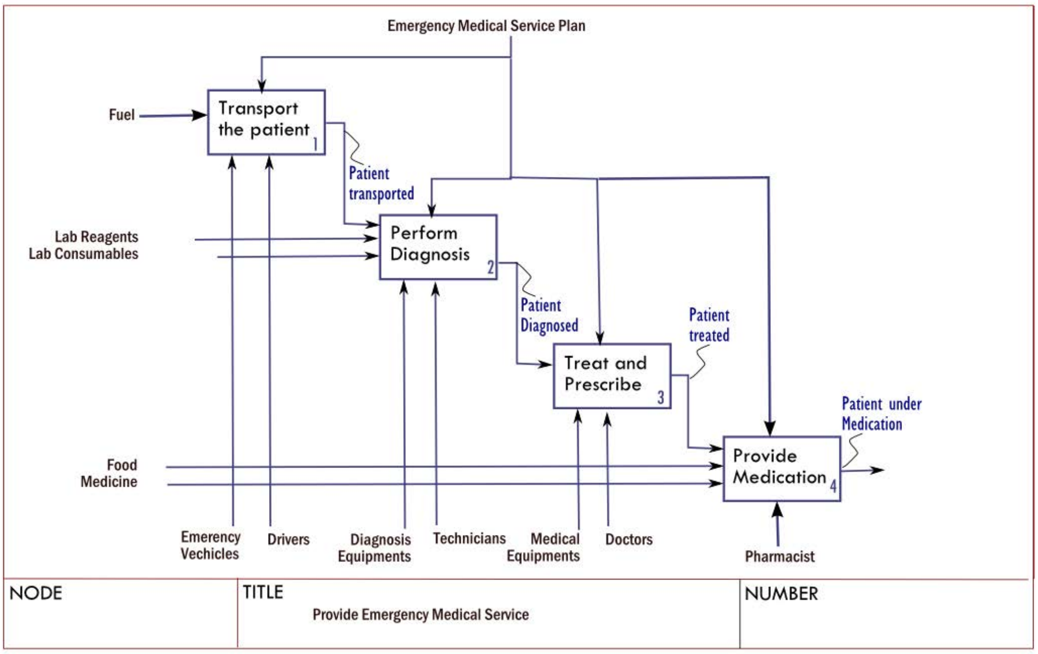

Figure 3.9: Process Modeling with IDEF0

IDEF standards are defined by IEEE to facilitate the functional modeling process. They are set of standards provided in the form of IDEF0, IDEF1, IDEF2 . . . addressing the modeling of process from different aspects. Functional block, the center shows state of the object, the left contains resources requirement required as input, the bottom displays roles associated with it and the upper input provides guidance in form of plans. Output is provided as arrow from right bottom.

A sample IDEF0 diagram in shown in Figure 3.9. It indicates IDEF0 diagram created for emergency medical service. Process guidance is available in the form of plan identified by the organization. The given example contains five stages of object: Patient. For patient states various users are active, requires various resources guided by standard processes, resulting in desired state change. For example, perform diagnosis state requires guidance for diagnosis and the resources like reagents, lab consumables etc. to complete the task. It requires diagnosis equipments and trained professionals. Such process representation allows handling of pluggable units to create complete actor network.

Design Principle 57 (Representing Actor Network)**.** *Actor network should be represented with*

Process Model with IDEF0 representation as building blocks of unit activity.

3.3.4 Modeling Infrastructure Services

Concepts related to entity and processes are introduced. Processes typically have various inputs, roles and environment. Some are recurrent and are found to be required for many other processes. These basic omnipresent processes, services or activities are useful in almost everywhere and hence, due to critical dependence is established. In case of undesired state change, they become critical point of failure and have multiple cascading effects on overall system. They are sensitive to disastrous events. Most of the time, irrespective of cause, the priority after such event is to keep them in desired states, and hence first entities to be restored. Examples of such critical services are transportation, communication, supply of potable water, electrical power, fuel and other such essential services.

Referring the definition of Actor Network, these services, can be visualized as mix of heterogeneous entities and can be impaired by non-availability of any of the component. An ingredient entity may exist in form of complex service. For example, a medical service requires transportation network as basic dependence for the transport of medical team to those who requires it. Hence, the transportation network is one of the critical infrastructures required for other critical services. Due to these reason, generally the boundary found to be blurred among entities and processes.

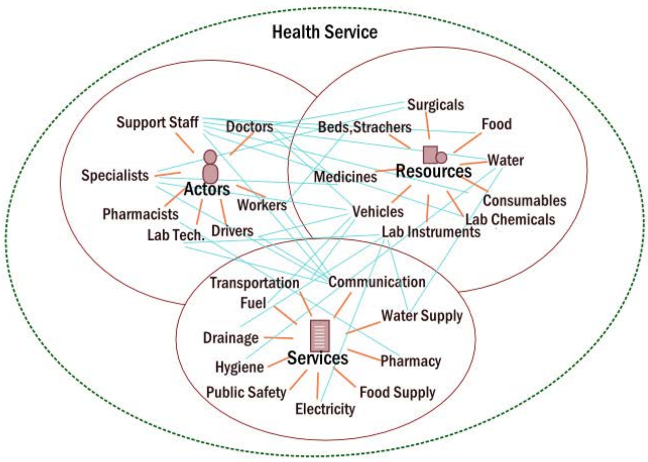

Figure 3.10: Health Actor Network

3.3.5 Example

Modeling of critical infrastructure service is common requirement. Conventionally [60] the interdependence is depicted as indicated in Figure 3.10. This representation does not reveal the ramification and chain of dependence. The proposed model recommends extensible parts of tasks or activities with associated resources, roles and any other requirements for successful completion of the overall task. Hence, a complex workflow can be considered as outcome of many small units. An Emergency Medical Service (EMS) involves basic components like medical treatment, transportation, consultation, diagnosis etc. The transportation activity is common service that might have captured earlier. Hence components of that service are directly utilized as one of the components. Similarly all the other required facilities are indicated as input to the EMS service. In case of unavailability of transportation service, the ramification will reveal the affect on Emergency Medical Service. Hence, proposed methodology is instrumental in identification of specific deliverables and interdependence of deliverables by various stakeholders in UoD.

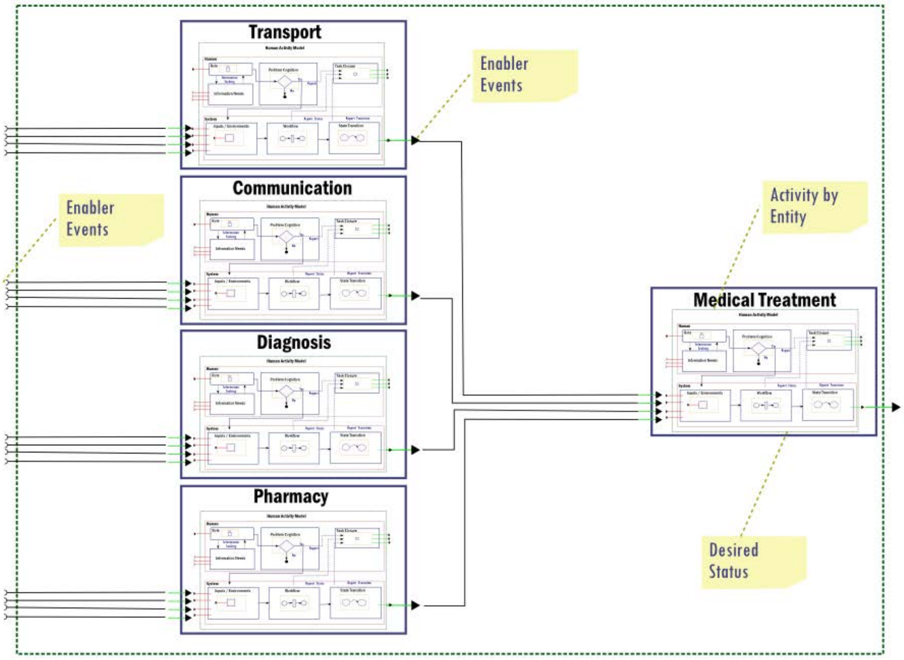

Figure 3.11: Process dependence in Emergency Medical Service

3.4 Representation Model

Policies, goals, standard operating procedures applicable to the planned situation awareness configuration are known. Based on policies, goals and other references, entities and processes are identified. Entity modeling process resulted in identification of set of entities, roles and associated features. Process models demonstrated workflow, process, dependence and procedures to achieve goal state for entities.

Next task is a proper representation of entities, processes and organization rules in machine processable format. The concepts to be represented may have complex interdependence and exhibit dynamic behavior. Therefore, appropriate representation is required to support consistent representation for further information.

Strategy to consistently represent identified entities and processes is missing that can appropriately capture their interdependence and complex behavior. There can be many possible ways this can be achieved. Identification of appropriate method is missing. It should be able to address questions like: what are the exact requirements from representation? How the resulting representations can be utilized in future? What is the best strategy to build, share, discover and utilize the reusable knowledge?

The model is required to characterize the logical representation of entity, with identification of types of relations. It is required to specify representation attributes of language and strategy for collaborative building of knowledge representation. It helps to formally represent captured entity and process along with the property and behavior. Model should be able to specify a methodology that can be followed to generate formal representation consistently. Model should also characterize the procedure to handle knowledge of operating procedure, goals, and rules etc.

Considering the model requirements, the proposed representation model is planned to provide reference for definition, incorporation of identified concepts, properties and rules in the knowledge base (KB). This will generate representation template that can be used in describing identified entity that will fulfill the next need to define KB, fulfilling the need of dynamic information management. The model strategy is very critical role in this regard. It is the starting point of making provision for reactive and proactive behavior required for the system.

Prerequisite for representation knowledge include knowledge about various methodology, their pro/cons in capability of building KB. Outcome of entity modeling and process modeling tasks as defined earlier are input to the representation model. Knowledge of various formalisms is required that can be employed in knowledge to suit application specific requirements. Knowledge about how the captured knowledge is going to be utilized by the information processing strategy during the runtime is also required.

Outcome of the representation model is a logical template that allows definition of newly found concept to be represented in formal manner. This identifies logic-based template that can be used to express nature and property of captured concepts. It represents standard operating procedures, conditions etc. It provides formal mechanism to capture the knowledge. This is done in form of rules. Based on this new instances can be created and handled.

Collaborating knowledge engineers (KE) will utilize these templates to capture and encode the required knowledge. In dynamic information management task, the capturing of business logic, domain knowledge and general concept is done as a part of knowledge management. This is responsibility of knowledge engineer who either collaborates with subject matter experts or domain experts. This is also used by application developer or system engineer, to facilitate usage and handing in system environment.

Formally represented concepts along with their properties, relations, hierarchies, rules etc are some of the elements commonly handled by the proposed representation model.

The requirement for represent strategy includes capability for generating rich representations of the situation that may be evident in given UoD. The required representation is not only limited to logical relationships, but also requires quantitative and qualitative expression of various properties. These values allow evaluation of rules to facilitate decision-making process.

It is known that rich representation generated using advance language features affects the reasoning capability of the approach. The trade off between expressiveness and tractability of reasoning makes the logical representation a challenging task.

The representation model proposes the representation of knowledge using description logic(DL) based approach. The benefit of using DL is selection of alternative DL flavors to suit the representation needs. From basic representation of Attributive Language AL to various flavors of DL by adding union of concepts U, full existential quantification E, Number Restrictions R, negation C etc. for improving the expressivity.

3.4.1 Conceptual Representation

Conceptual modeling approach is realized with the development of Ontologies, which are considered useful in modeling process as it provide the links among the state of affair in the world, metaphysics, and human cognition [61].

Use of Description Logic(DL) is widely employed in building knowledge representation in the form of Ontology. Brachman [62] provides overview of representation approaches. Multi dimensional approach [63] including temporal, spatial and conceptual dimensions are discussed, providing some information about scope of representation. Staab [64] discussed application of specific flavors of logic is useful in various application domain; it provides overview of how knowledge representation have evolved. Egenhofer [65] and Longley[66] provides information about spatial and temporal aspects that must be supported in analysis and handling of representation, Hadrich [67] provides information about how knowledge infrastructure can be set up to meet the knowledge requirement of an enterprise.

A comprehensive modeling approach required for knowledge representation is discussed by

Borgida and Brachman [68, page 359]. The collected information about entities and processes are to be represented using this approach. The recommended represent indicates following aspects.

The starting point of the representation is the explication of objects and relations that they stand in with other objects. These relations are binary in nature. The basic relation seen in the given application domain is between the stakeholder and the work product. And standards organization `OGC` sets `GML` standard to standardize the representation of geographical features. A binary relationship can be identified between these two concepts. `GML` specification is `standardizedBy` `OGC` or `OGC` is involved in `isStandardizing` relationship with a standard `GML`. The relationship represented in form "$a\ R\ b$" or $R(a,b)$ is read as object a exhibits relation R with object b. Hence, for given example, it can be said that `isStandardizing(OGC,GML)` or `standardizedBy(GML,OGC)`. The relationship can be partial or full depending upon the context. A property may or may not have a standard specified for it. But for an available standard, a specific standardization organization can always be identified.

In complex dynamical systems, handling of individual is quite important feature, as their existence and availability affects the overall state of affair. Individuals exhibit identity in given UoD and can possess features that distinguish themselves with others. They may exhibit a countable property. While representing such individuals, object and values like integers, strings, and lists should be treated separately. Another important feature is the possibility to identify the object from its relationship. This means that `OGC` should be identified by one of its relation stated in the form of the entity that standardize `GML`. From all such relations exhibited by the individual, it should be recognized.

In similar manner concepts should be identified with essential and incidental properties. The representation should support reified concepts, meta-roles, and relationship dependent concepts. The sub concepts should be handed with disjointness features. Primitive and defined sub concepts along with features dynamic membership and structural hierarchy should also be identified for given domain.

Similar to the concepts, and individuals, the relationships among them may exhibit complex features. For appropriate representation of complexity, the nature of complex relationship should be characterized appropriately. All possible hierarchies in relationship should be identified. For example, `isLegallyResponsibleFor` $\sqsubseteq$ `isResponsibleFor`. Other relationship related features like relationship roles, materialization, part-whole aggregation, constrains should be included.

Knowledge representation created with such formalisms allows creation of a knowledge base

(KB). In DL based system, terminology box known as TBox contains the intentional knowledge in the form of terminology. Terminology or taxonomical representation of concept is based on the principles of some domain. This representation is static, and once defined, it can be used and reused as reference. Assertion Box known as ABox contains extensional knowledge or assertion knowledge. It is representation specific to individuals in UoD. The individuals in UoD are subject to change and exhibit complex and dynamic behavior. Hence, a box is utilized to handle transient entities visible in the UoD. Another important aspect in knowledge base is the representation of rule. Rules can be in many forms. Production rules are famous in expert system domain. Event Condition Action (ECA) rules are used for defining event driven systems. These are the formalisms employed that provide capturing of domain knowledge, business process or are used to guide processing of newly added information. Knowledge captured in this manner can be accessed with appropriate query mechanism. It allows reference and retrieval capability similar to database domain and allows selection of instances based on properties values or relations defined earlier. The resulting knowledge base (KB) consists of TBox, ABox and Rules and hence, it is defined as $\mathcal{K} = \{\mathcal{T}, \mathcal{A}, \mathcal{R}\}$.

3.4.2 Spatial Representation

Apart from these general properties identified for entities in the UoD, the explication of attributes like footprints involves representation of spatial attributes. The concepts of spatial representation is and reasoning[65] can be adopted from geographical information system domain[66].

Spatial properties that are useful in inferring spatial relation like contain, near, overlap, similarly temporal modeling proceeds, follows, overlaps, starts, finishes, and other relative temporal domain [65]. Spatial hierarchy is also important feature. Malinowsk [69] recommends representations of spatial hierarchy levels from location, to city, state, country, continents etc.

3.4.3 Temporal Representation

Apart from the spatial dimension, the concept of footprint contains reference to the temporal dimension. Temporal representation can be of two types. One is exact temporal coordinate that provide time. The other is the duration between two marked temporal coordinates. When temporal aspects are incorporated in representation and there exists rules that utilize the temporal representations.

Temporal analysis can be carried out based on these representations. Relation in temporal domain is extensively studied and utilized [69]. The relation includes before, after, during, starting, ending. In order to utilize the formally represented representations, appropriate mechanisms are required temporal calculus, algebra-providing interpretation based on the axioms. These formalisms are useful during interrelations, decision making during runtime when instances are taken for inference or asserted.

From implementation or realization aspect, the temporal features can be access with appropriate query support. As a common requirement, filtering based on temporal relations that may exist among the available instances. Various temporal relationships and their handling is discussed in [69] and [59]. Providing a comprehensive strategy [59] for handling of these formal concepts, temporal data types including time, instant, interval, time span, interval bag, instance bag, simple time, complex time etc.; temporal predicates like meets, overlaps, contains, intersects, equals, covers, starts, finishes etc. are also useful in representation of spatial relationships.

3.4.4 Collaboration Aspects

To cover all the relevant aspects of the UoD, considerable amount of effort is required. Specific skills and domain knowledge is required to enable representation of characteristic behavior. Expertise to represent the captured knowledge using popular and effective formal representation approaches is also required. Expertise of the content of knowledge that is being encoded is required. Ability to constantly update the concepts to reflect new findings in research and development and practice is also needed. These tasks cannot be done individually at single place, by single person. Work distribution is required to achieve the goal.

Depending upon the content, the task of knowledge representation can be distributed. Any given concept can be identified as a member of domain independent, domain specific, local specific or application specific domain. Following discussion identifies the possibility of task allocation based on this categorization.

Domain Independent Representation

Use of some concepts do not belong to a specific domain, and generally used for representation of knowledge by providing general level terms. They should be defined carefully as domain specific terms are explained as extensions to these basic concepts. Consortium, research community, modeling experts can classify and represent such domain independent concepts. Such concepts include basic notion of time, space, and entities. Generally they are available in the form similar to classification systems of living and nonliving entities. They play an important role, as they provide basis for detail knowledge representation. In research domain this is realized with upper level Ontologies. Standard Upper Level Ontology, SWEET, OpenCyc are some example of the upper level Ontologies that provide such domain independent representation. Such representations can be directly reused in knowledge representation task.

Domain Specific Representation

Terminology, taxonomy, rules determination of membership to specific class etc. are example of domain specific representation of knowledge. Research group, consortium, associations etc., involved in specific domain can provide such knowledge representation. Domain specific representations are also reusable components that can be directly adopted by the representation strategy.

Terminology rules etc. that belong to a specific geopolitical area only are known as local specific concepts. Local authority organizations are responsible for identification of the same. This includes facts about local area, organizational hierarchy, and standard operating procedures for handling emergency management rules. The terminology, roles, name, workflow, alerts, communication systems also forms information content specific to an area. They can only be handled, and updated by the organization and may not be reusable in other parts of the world. With identification of entities and processes, with their characterization, the role of representation model is identified. The model established suitable templates to aid the capturing of formal knowledge representation. It was identified that formal representation is useful in

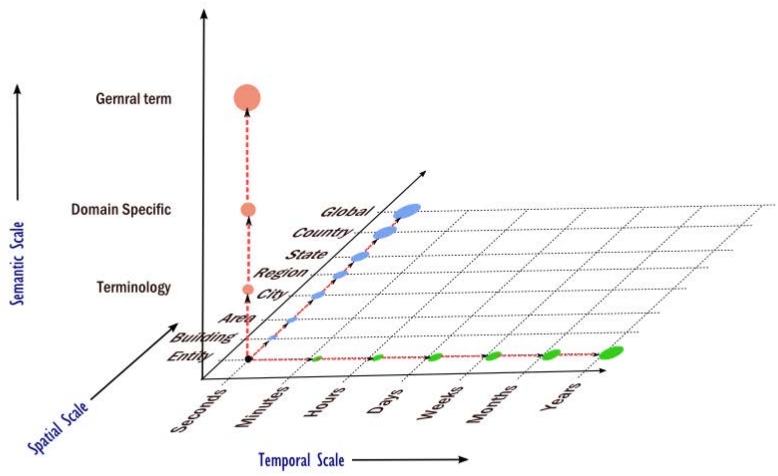

Figure 3.12: Space, Time and Conceptual Granularity Levels

order to make the collected list usable. Concepts, properties, relations should be established appropriately for concrete domain. Based on asserted values query filtering should be supported. The resulting granularity can be represented as Figure 3.12. This says the terms used in representation. Also include multiple possible levels of granularity that can be accessible to suit the information needs. This includes most general to most detailed level representation of captured knowledge.

In summary, the representation model provides estimate of rigorous effort required to accurately represent entities, domain knowledge existing and applicable in given UoD. It identifies how accurate representation strategy provides basis for design information processing to meet the information needs in complex dynamical system. Representation approach also highlights the distribution of work, responsibility in building and updating the domain specific concept in collaborative environment.

Also, spatial, temporal, and semantic granularity is identified for all types of knowledge representation. The selection of rules to capture business logic, policy or domain principles are also important decisions towards making automation possible for later reference. Concepts, rules, knowledge expressed in such manner provide basis for continuous KB management and upgradation in collaborative manner.

3.5 Measurement Model

So far, the representation model is defined that provide specification for modeling of entities, processes and properties from various domain perspectives. Assuming that many collaborated according to the model, and generated model template provide guidance about content specification for concepts and processes representation business logic or goals captured as role facilitate interpreting current status and determining the future course of action.

Next task is to assign value by carrying out measurements that is suitable for purpose. Generated representation should be used by diverse people for situation awareness needs by creating assertions and about concepts, properties and their values. These values are concrete and contain values according to users' perception. The user perceived values are to be appropriately assigned and reported. The same is to be used for processing and summarization. Such summarized data can directly be usable for decision makers in executing their roles.

Missing element in the modeling process is therefore a strategy that will be selected for proposed measurements. This also guides the transformation of measurements if done using uncommon techniques to enable the usage. The model establishes possible ways the properties can be measured, transformed and aggregated.

For various concepts identified in UoD many have unique properties that may require unique ways toward measurement. If the measurement is done with non-standard technique, its utilization and reuse becomes difficult. It may cause many problems causing inaccurate and erroneous usage of the same. To help this process, proper mapping rules must be defined that can guide the measurement process. Rules are also required to provide change in representation unit. These rules establish references that can be used by providers and users.

For known parameters, it allows capturing of dimensions scale, unit of measure etc. Representation strategy provide exhaustive list of domain properties. All that is known is captured and represented. The observers who assign values to these concrete property and users who interpret these data and further process, all use the same reference. The strategy is required to specify this task.

The proposed model will provide guidance to specify different ways in representing measurement. The observer can select from the available measurement approach to suit the requirements. Adoption to representation model, allows assignment values and units. In case of change, the model provide conversations, type change or mapping with other useful representations.

Prerequisite to measurement model includes entity and property whose values should be measured, decision-maker requirements and references to available measurement standards. Domain knowledge about values standard methods related to measurement, scale and unit of measure specific properties also provide critical information for measurement. Knowledge about *de facto* and *de jure* standards, knowledge about mapping and conversation, possible representations among units, practice of summarization, type change, aggregation possible in the range is required.

Outcome of the representation modeling process is in the form of template representing property from domain point-of-views. It also provides templates for various rules regarding conversion, mapping and type change possible for given representation. Outcome of measurement is in the form of additional representation rules. Dealing with domain properties and associated standard methods, it also provide application specific role generation that allows type change, type conversion.

This will help knowledge engineer in defining all possible observable and measurement property in standard format it also support capturing knowledge of conversion. In information management, the provision for capturing knowledge, handling and process is done for concrete domain. Representation rules for handling such properties are responsibility of knowledge engineer. For an analyst role, outcome is useful in handling new, conversions made available in the runtime.

Basic element of the proposal include property template, rule templates for property value measurement, conversion, and transformation. Challenges in identifying these elements are due to many possible ways to monitor, measure, and assign scale to same property. Continuous research outcomes establishing effective measurements should be bring in to practice. Availability of various standards must also be incorporated in the process. It is difficult to keep track of all relevant advancement and incorporate them on the runtime. Hence provision is required that will guide this process. It is difficult to provide possible representations, for given property decision regarding the strategy that standard organization will incorporate new conversions how analyst will provide new ways for aggregation strategies and visualization strategy to improve the decision making process.

Proposal is to incorporate measurement details according to *de facto* and *de jure* measurement methods. Apart from domain specific property measurements there are defined by domain experts, there are some general guidelines to be followed by knowledge engineer in representing concepts of concrete domains. Spatial and temporal properties are basic component in them.

Temporal measurements can be very important source of information criteria in rules that are based on temporal relations. Time captured as point or period, they can be utilized for query, filter and analysis. There can be many ways for handling time as described in literature [69, 65]. An isolated event can be handled with single experience time. Multi experience time is useful in tracking subsequent events. Some critical properties should be monitored continuously and hence continuous time is suitable for them. Special scenario may require multiple temporal perspectives. Some regularly recurring event can be modeled with cyclic time. When temporal reference bifurcates in multiple branches, branching time is useful. They have unique implications in analysis and aggregation.

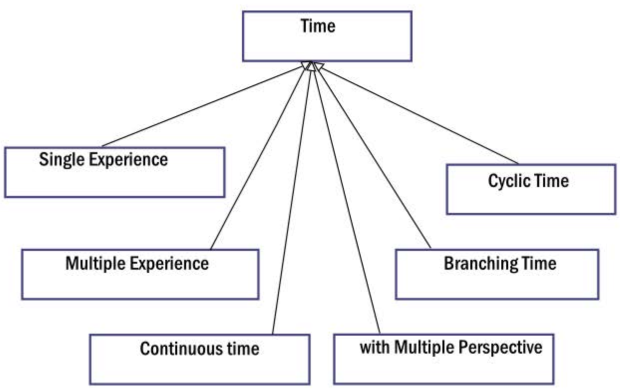

Figure 3.13: Catagory Time

Spatial measurements are also carried out with some referencing system. The spatial coordinates should be following standard projection and datum system so that spatial data from multiple sources can be seamlessly integrated.

Figure 3.14 indicates possible value a filed can assume. The field value once observed can be assigned nominal attributes. Example is the existing land use on land visible in UoD. This is called nominal, as there are no arithmetic operations supported by these attributes. In some special cases, the names assigned are arranged in particular order of sequence. Like showed in example poor, very poor, average, very good, etc. are ordinal attributes. Observables like temperature pressure and many such properties that are bound in some range. Ratio attribute provides support of arithmetic operations and have absolute zero. Weight is one example of ratio attribute. Some properties are measured by sensors as continuous attribute. Absolute attributes are also useful in certain domain specific application.

These field values come with specific unit of measure (UoM). Assertion of UoM is essential along with the measurement. Assertion as it provides information about interpretation and further conversions. These are being established by standards organization based on consensus building process.

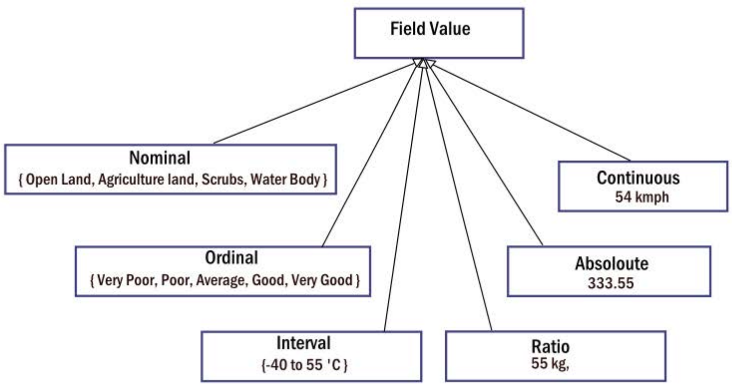

Figure 3.14: Catagory Attributes

Adoption of appropriate standards needs comprehensive research about operations, procedures and references. Adoption may result in selection of specific approach. For competing alternatives, adoption and facilitation is required by research, industry and consortia to make it popular at large scale. The factors affecting the penetration may include domain specific standard methods, availability of specific equipments and resources like reagents, availability of references, and conformance for accuracy, speed, and ease of use. The competing methods are selected and alternatives are also suggested in case of unavailability, accuracy, speed, and ease of use. Requirements are the deciding factors in use of standards methods. Mapping therefore is required for conversations.

3.5.1 Conversion Rules

Conversion rules or conversion tables is a popular handy tool in literate population. Conversion table is limited to in regular practice. While building knowledge base, all identified properties must be searched for set standards and common practice in field. Example for such competing representation practice is adoption of metric and ISO properties. When information sources are following different scenarios, like kilo-pound, kilometer-mile and Fahrenheit-Centigrade, the provision should be made for conversation of scaling systems.

Continuous values are providing precise reference with accuracy, but in decision making in larger scale, many observation reported in this scale proves less useful in analyzing data. In geographical information domain this is primarily used practice as indicated in Figure 3.15. A value with interval attributed is converted in nominal to enhance the extreme. Not all the attributes types can be converted to nominal or ordinal set. The set can be in specific number of class hence while building KB it is necessary to define such rules for all attributes types. This will facilitate automated type change conversion on the runtime.

3.5.2 Aggregation Rules

Summary reports are useful cognitive artifacts used by decision makers. The property values stored collected at record granularity may not be much helpful if used at larger scale. In such cases there are many records and analysis becomes difficult. Aggregation can be done on many ways. It can be done in temporal dimensions. Summarizing data collected at specific frequency aggregated at given time period. Spatial dimension is also useful in which a large area covering many small spatial blocks are taken as single representative. Aggregation function therefore must be defined based on the property type aggregation take place. Figure 3.15 indicates two example aggregation recorded temperature value. This can be aggregated only as average temperature. If same values are representing population, the aggregation function is selected as sum.

3.5.3 Type Change Rules

Some times summarization may not be useful as it looses the outliers. Record formats may be not useful to retrieve spatial relevance, a pattern or correlation of some kind. Changing the available data type can provide visual advantage for analysis. This is primarily used in geographical information systems (GIS) and Digital Image Processing (DIP) domain.

In GIS domain, when data covering multiple records, assigning unique color do not prove to

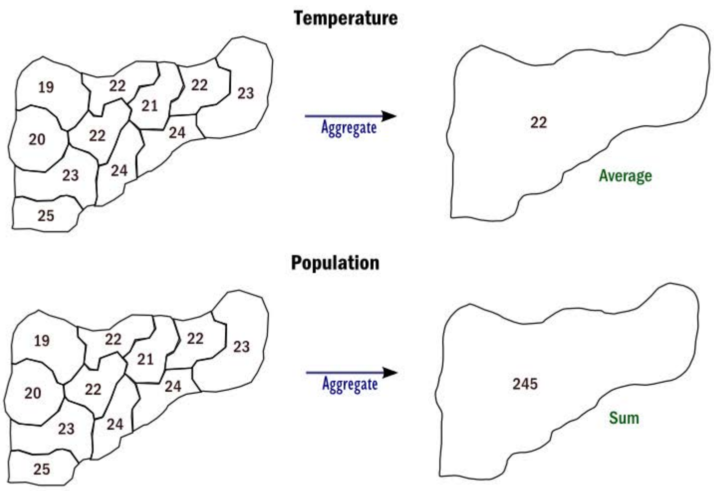

Figure 3.15: Aggregation Rules

be a useful visualization strategy. For better perception of outliers, special cases, a range can be defined. This requires specification of classes and range values that determines membership to each class. The color codes assigned to each class can be unique color or graduated color or it can be unique symbol or graduated symbol. If given property can accurately defined with unique color or symbol can be defined in form of rules. As generally, this task is done by GIS analyst role, if this is to be done automatically on the runtime, this knowledge should be encoded in the form of visualization rules. This also indicates guidance on creation of range and selection scheme for colors and symbols.

In image processing domain, this is achieved with supervised and unsupervised classification. Classification is used as type change operation as it converts the recorded values to identified classes. Each class is then defined color or gray scale value. In unsupervised classification, it is roughly classified in to number of classes. In supervised classification, a training set is provided, as a basis for determination of membership in specific class.

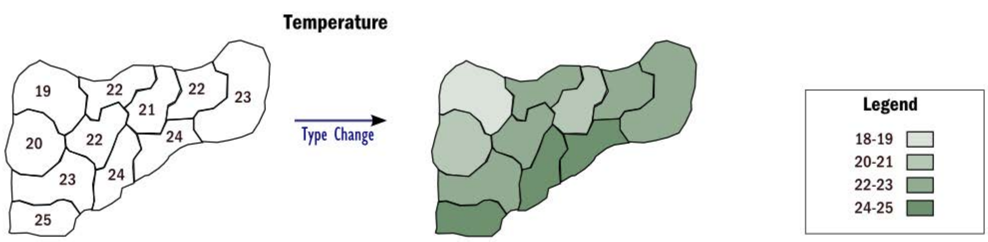

Figure 3.16: Type Change Rule

3.5.4 Summary

Measurement model is the suggested practice to be carried out for defining the measurement specific aspect. It introduces which aspects of measurements should be incorporated in representation KB. Rules are defined to improve visualization and to reduce cognitive load on users. They should be defined for each observable property, as they are prerequisites for automated processing and visualization of recorded data.

3.6 Observation Model

Concepts, properties and domain specific knowledge including measurement strategies are defined and covered in model phase. Various properties and suitable method is introduced so that they can be defined and evaluated. Spatial and temporal footprint is defined and a visualization strategy is also specified with rule generations. Basic framework necessary to generate snapshot of the situation is also introduced.

Based on the knowledge instances should be monitored for change in values of observable properties. This change measured is further used for processing and appropriate task should be assigned. Sensors should be configured or human observer should be notified to provide value ranges.

Identification of observation specification to suit the information needs of many potential users. Each property must be monitored as defined interval with defined method mapping value range. Each entity property is unique applicable for space, time and concept. Identification of them is missing in the modeling strategy.

The model is required to bridge the gap from models discussed earlier and the identified domain knowledge. This model specifies the temporal and spatial coverage of observations. It characterizes the rules specified for observable property. It identifies strategy that allows determination of scope of observations. Based on domain knowledge it allows parameters specification about observations.

This will provide identification of property specific observation templates/intervals in space-time and other characteristics that can be further utilized by communication, data collection, processing this will subsequently help detection of simple or complex situation. This is difficult task as each property has unique granularity. As defined in domain knowledge, the intervals in space-time and conceptual levels are useful, as it can ensure the coverage of information collection type. This is collected at record granularity.

Domain property with updated knowledge of standard methods scale unit of measures and interpretation, this also includes how fast property change and how much it can be generalized. For example some observed measurements are valid for how much area for how long period. This is essential in determination of actual intervals for observation, rules appropriately defined. It helps in establishing how required sampling rate and identification of that rules cost without harming loss of information detail.

Outcome of the model is a template for observation need for domain specific observation. The rule template will be filled by domain expert, for each domain properties apart from the observer details and measurement the important aspect is to define scope of observation in form of footprints defined earlier.

Observation model provide guidance to configure sensor if it is a sensor observable property or it provides reference to human observer or collecting in planning how often it will take to appropriately monitor the desired domain property. In information management domain, this configuration and allocation management of external information source is responsibility of the configurer and automated generation of source specific observation templates helps the configurer determine it for transient resources automatically.

Basic elements of the model are observer, observable properties, type of observation, scope of observation, frequency, range etc. Observer is the entity that observer the property. It can be sensor or human observer. Domains have many observable properties and some many not. The observation scope is the property that is similar to footprint discussed earlier. They provide spatial, temporal and semantic coverage of observation. This also provides type of measure for exchange. While type interpretation and observation of observable property is fixed, what is most difficult is to know and establish the granularity. This is the single most difficult part for which this model is defined. It allows even conversation and type change rules defined in measurement model the requirement may be different for different users hence specifying one will change other dramatically. In case the granularity for record is not identified properly this may result in loss of details. These details might be important in accurate and timely comprehension of situation unfolding in UoD. In case observable properties are not defined with alternative. This may lead to complete loss of information capture from ongoing event. With this loss, many vital signs are lost that may have lead to accurate interpretation of ongoing situation.

Proposal is to extend the conversion and aggregation rules defined for properties to further extend. Which following features human sensor based on measurement model the measurement scope. The scope of measurement will provide most vital information about the observation. This define the t_u, t_u^+, when occurrence will take place at what frequency measurement should

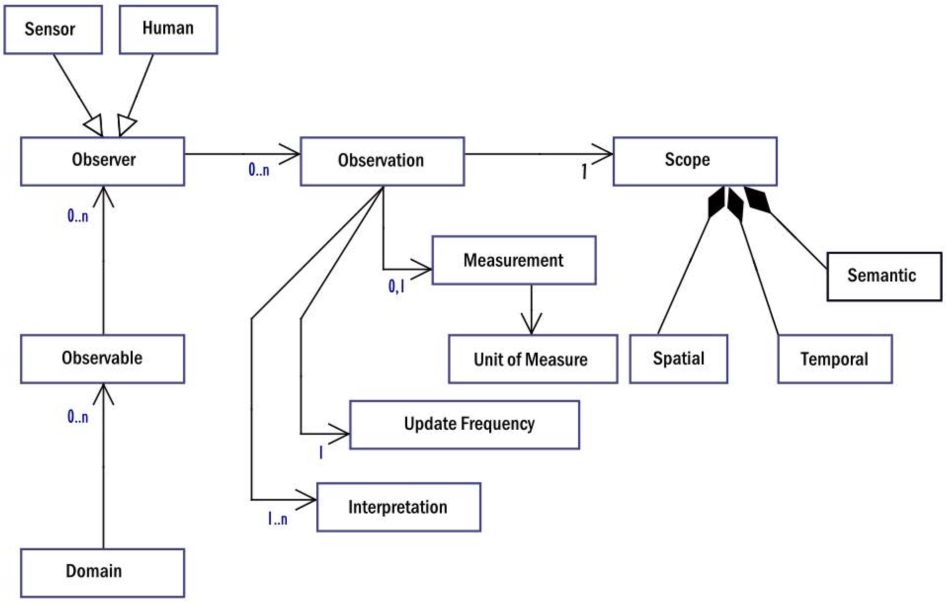

Figure 3.17: Observation Model

be taken in reference to identified situations at t. This bring about interaction pattern how human-human, human-system are different from each other. Because from sensor observable property it is possible to select constant polling strategy, but that will not be suitable for a human observer. Scope of the measurement in spatial domain determines this can be valid for how large area. Temperature, pressure wind speed etc. are valid for few kilometers of range whereas motion and smoke detection is required in each isolated shell in constructed building.

Figure 3.18 indicates the entity state change trace. How over a period of time it change if the change is observed as soon as it take place corrective action can be taken in reasonable time period.

Information need determination based on these intervals information need identification is possible for specific roles. Information need of an EOC is to constantly monitor the inflow of victims But as it is observed by respondent and (s)he is also having responsibility to carry out the response work. The rules created based on observation model must reduce to workable solution or interval. To improve situation EOC may incorporate automated entry for victim as they are admitted in to the camp. This way frequency can be improved. It can also be made event driven, thereby reducing the workload of the respondent. To provide comparative representation of system in a response scenario.

The observation model should be defined in form of rules. They capture items displayed in Figure 3.17. As indicated for given entity and selected domain, all the observable property should be included. For each observable different type of observers should be identified. As

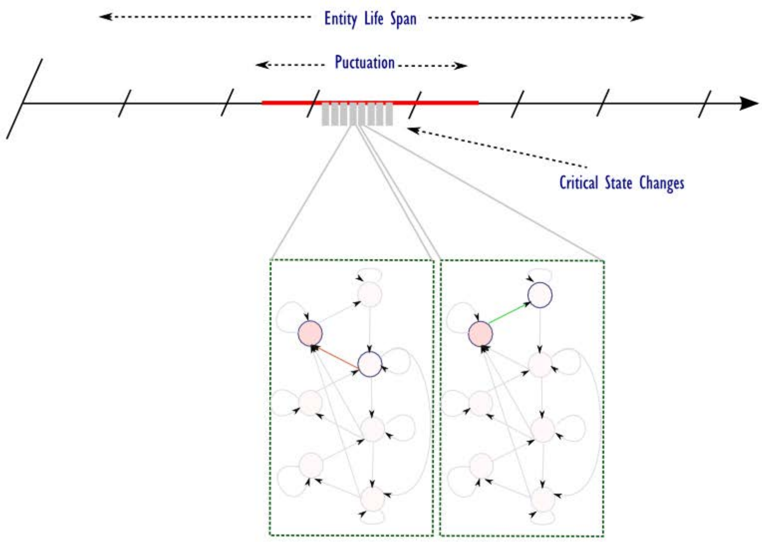

Figure 3.18: Entity State Change Trace

indicated they can be human or different types of sensors. For a unit observation expected from observers, unit of measure and interpretation are defined for possible observable ranges. The scope in spatial, temporal as well as semantic dimension is defined with footprint conversion as indicated in the entity diagram.

In summary, observation model work toward more structured representation of property value in concrete domain by identifying rules they govern assertions about domain specific entities.

3.7 Eventing Model

Dynamism can be characterized in form of events. The response required for the event is many times driven by the nature of event. Hence, situation awareness also includes not only information about various states of entities, but also requires handling of events. There are outcome of abstract or concrete property generally followed by cascading impact and many secondary changes. The handling of event should be done as explained in [70] and [71].

Information needs are unique for different actors based on their assumed roles, location, their legal status and implicit or explicit relation with others. Information need determination is based on and all about changing situation. But situation in general but certain situation are more meaningful they signify event. Barwise considers the situation in action as events. Events are basic elements in complex dynamical system. There can be many possible state changes and many possible events. It is therefore necessary to determine how individual events should be monitored, handled and conveyed to appropriate role player any pro-active or reactive steps can be taken. These are not known *a priori*, but must be decided on the runtime and a strategy is missing toward it.

The model characterizes basic event and related terms, necessary to utilize it. Event based system should be characterized. It allows the identification of how many events can take place in given configuration. An evening strategy is required to derive event space. A strategy suitable to capture dynamism is also required. It establishes the process indicating how eventing can be carried out with provision of logical models and identification of eventing related parameters.

It also allows identification and characterization of mechanism that is required to realize the event-based system in distributed nature. Event definition provided by this model will enable determination of system requirements for information processing. It will also lead to identification of system/application specific concerns that will facilitate building of event-based system.

Prerequisites for eventing model include representation, evaluation criteria, and rules including desired state. Explication of pro-active steps and reactive response is also required. Outcome of representation of domain knowledge provide information about concepts and relations. Outcome of measurement model helps to determine rules how property value measurement should be handled. Outcome of observation model is required to determine interval and build event trace. Outcome of entity and process model leading to identification of state, transitions and required actions.

Outcome of the eventing models are templates, strategies, rules definitions of event and its related concepts. The strategy is required to identify simple and complex events. Underlying data required how to handle that information flow. A method is also provided to establish the eventing model.

The eventing model is to be used by the roles responsible in developing situation awareness system event driven components. Resulting event profiles, entities and tasks should be part of application specific concept that is essential for the realization of the functionality in system environment. This is also useful for appropriate roles that realize the functionality based on rules created by domain knowledge engineer.

Basic elements are definitions that are required to be redefined for complex dynamical systems including concepts like event and related concepts. Rules should be defined for :

1. Checking if event has occurred
2. Managing related - required entities and monitoring task
3. Carrying out the identified tasks.

There can be many possible ways in which events can happen. Various levels of events, some of them are monitoring related, some of them are feedback related, and some of them are system or application specific. Each event is providing important change in status of concept.

Definition 11 (Event). *Event is a significant change recorded in observed situation for which specific responses are defined for associated entities in given configuration.*

Here, all and any state change is not considered as event. If it is considered as event, there is a follow up action identified and roles instances are alerted that can execute the identified response. In this case, a state-change of interest in also not considered unless it is implicit or explicitly expressed in knowledge base. Event is a state change recorded that amounts to provide feedback so that necessary action can take place. Hence, it is based on some rules in knowledge base. It has a subscriber that takes decision for identified event.

Definition 12 (Event Granularity). *Event granularity is the state change recorded in the situation observed at specific situation granularity.*

For example various levels of event monitored in programming domain include event detection at debugging, information, warning, or error. The actions and roles are defined at different levels of response from actors as defined in process model, granularity of event detection is therefore specified at that levels.

Definition 13 (Event hierarchy). *Event hierarchy represents the granularity levels at which events can be detected for given situation.*

As indicated in same process model, the process may have hierarchy and therefore event should be different for different levels. For each level event should also be defined at appropriate granularity from very specific to very general level. This results in identification of event hierarchy. Event hierarchy tree is important tool allows navigation to different granularity of event monitoring and notification. This hierarchy is important to define for filtering and thereby adjusting the load of system as well the subscribers of the event.

Definition 14 (Event Profile). *Is a criteria provided for a situation to be considered as an event.*

It is the rule; the event driven system uses to check if event has taken place. Method for checking if event has taken place can be known by tracking the changes in value of properties. These values change demonstrate the outcome of an action, event or process. Some process outcomes are simple and associated events can be monitored by simple rules. The simple rules can be in the form of range query. If particular property value is measured in specific value range event has occurred. This event can be considered as single parameter function. As indicated in Figure 3.19. As depicted event profile maps present reading of an entity to an event in set of all possible events. This reading at specific time can be identified as a simple event. Sometimes sudden state change is also a separate event. Current state and previous state both are required

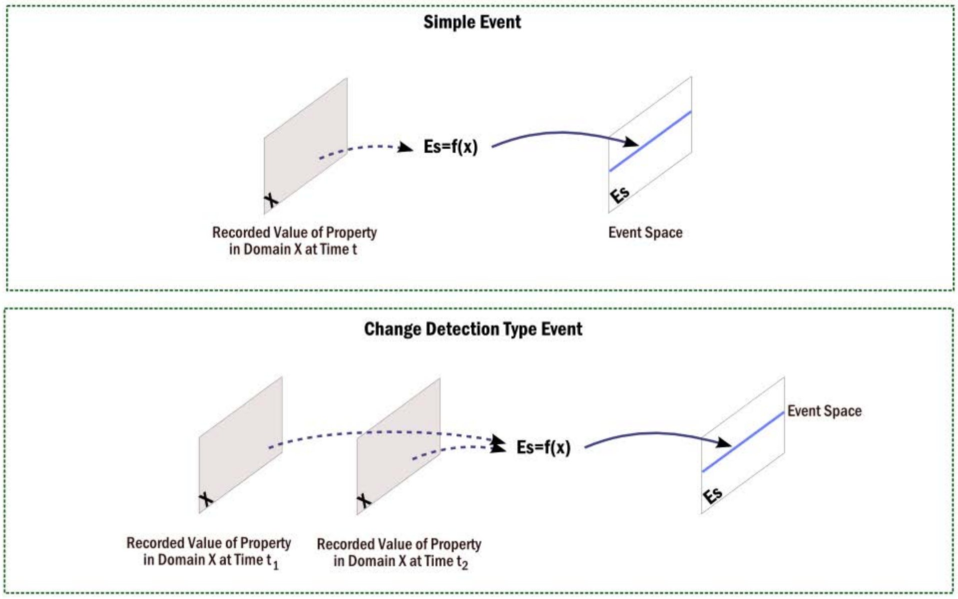

Figure 3.19: Single Property Event Profile

to determine if event has occurred. This is popularly known as a Change Detection event. If change is reported beyond some value range, event can be considered to have happen. Time precedence is shown as $t_1 \prec t_2$.

In many real world situations or events involve more than one property values. The event profiles therefore are functions of more than one parameters. Figure 3.20 includes pictorial representation of a multi-property event. In order to determine if event has taken place, multiple properties event profile must be provided observed value of required properties. Multi property events may have complex events if considered over period of time.

Mostly the profiles are unique for domain specific concepts and properties. Evaluation of profile is done to check if event has taken place based on implications defined in previous models. They are based on information needs. In information need it is determined that what is relevant for whom at what frequency granularity. The what part contains the present status. But it also contains much more things than just values. They provide if instantaneous readings considered together, indicates new information. Hence event profile provide utilization model for information collected.

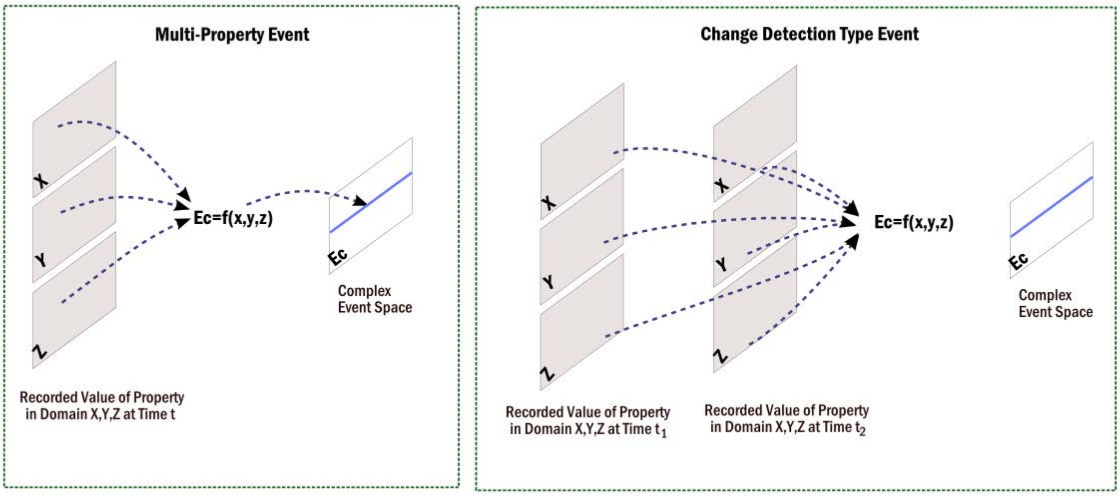

Figure 3.20: Event Profile Multiple Property

Definition 15 (Event Profile Matching Strategy)**.** *is a strategy that governs evaluation of observed property values for available event profiles.*

As explained earlier, event profile allows determination of event occurrence. They can continuously evaluate or at defined interval. The determination of interval is based on update frequency of involved property. Hence profile matching is considered as recurring task.

Definition 16 (Event Space)**.** *is a set of all possible events that the current configuration is capable to detect.*

Each profile defined provides mechanism for detection of event. Number of profiles defined is based on number of individual properties known plus the dependencies are known of given property with itself over a period of time as well as with other properties. Hence, based on the number of event profiles defined, number of all possible events can be known. This is known as event space. Event space provides information about systems capability in handling event. This is kind of resolution or system capability. If event are known but no corresponding profiles are found, than system engineer can identify the gap of profile availability and contribute appropriate profiles in the system.

The number of event profiles is defined, the time new task will be performed and the time new task will be created. Event space is a dynamic set, constantly changing along with the state of affair in the UoD. As new events are reported responsive actions creates new patterns and corresponding entities in system. Hence at any given time, event space provides set of all possible events that includes all transient entities in instantaneous UoD.

3.7.1 Sources of Event

Source of event can be direct checking of property on representation or it can be on recorded event trace in case of change detection event. Some times transient provisions and respondent in the UoD may act as source of event

3.7.2 Evaluation of Event Profiles, Task Rules

Primary: Primary event profile matching is done to track on-going event. Also as an outcome of the event how other entities of processes are demonstrating event. Apart from this, the purpose is also critical as stakeholders can take actions hence level 2 responses are those taken in response.

Secondary: Response events are these types of events. In maintaining all these systems come into play and system also hence events like when new instance are created etc. They are done at specific intervals determined by rules. They are done as scheduled task to be carried out.

3.7.3 Eventing Roles and Life Cycle Management

Apart from actual role played by entities, they assume eventing roles. When they produce, consume or handle events. When their behavior is source of event, they are called event sources. When event source is detected event, the rule is referred to provide information about occurrence of event. Hence they are the sink of event one can also map source to an event publisher, and sink with subscriber. To support the communication system specific objects are also required for example channel router, splitter etc.

3.7.4 Subscription and Notification of Event

Subscription to an event is necessary in completing the information flow, which started by monitoring and detection of event. The determination of appropriate target and delivery of information is realized in the form of event notification. The subscription mechanism is necessary to get notification about the event profile-matching outcome. Subscriptions can be role specific and rules must be defined for making role automatically subscribe to event. Yet system may match many profiles and may create many messages. How these are provided to user is a design decision. Based on severity of event, it should be filtered accordingly. If some individual is playing multiple roles, event notification at client side becomes challenging issue.

3.7.5 Eventing and Messaging Patterns

With identification of filtering mechanism based on rules, subscription and related roles content of the message type of the channel etc. should be appropriately defined. Various interaction patterns can be realized to suit different scenarios should also be specified in form of rules. For example publish subscribe model, request reply model consistent polling client, secure channel etc can appropriately utilized to suit the requirements.

3.8 Information Processing Models

Models are defined to capture the complexity of entities, interdependence and dynamism. The model allows explication of goals, procedures and strategies that can later be utilized. By establishing entity model, features of abstract and concrete properties are identified. With establishing process model, complex interdependence of various simple and complex services have been identified. With establishment of observation model, detail of collection of observation is recognized for domain specific observable properties. Representation model, enumerated standard scales, unit of measure and type of measurements that will enable accurate conversation and handling of observation. By establishing event model, means of situation evaluation is specified. Profiles Matching is introduced that can allow identification of simple and complex events. In summary, various aspects important for the representation of situation is identified and explicated in the design time. This will provide a reference for further information processing task.

Next step is to build a system that accesses the outcome of the modeling process. The system should handle the developments in UoD by allowing assertions to the system. System in response should be able to identify the newly added fact, and find out the ramification, by identifying how it is going to affect the goal of the system. In response, it may ask for more data to be collected about specific features of the UoD. Or it may allow identification of tasks or action that needs to be carried out in order to keep the system in the goal state.

- Knowledge base (KB) is vast, holding many concepts, properties. All cannot be handled at single place and single time by single user.
- Instances who will act as users, their assumed roles and defined on the runtime, only when they show-up in UoD.
- In situation awareness point-of-view, different types of relations are known, but instances are not defined. These instances have implication on information need of themselves and

others.

- Even if it is identified that all instances are available, how communication will take place is not defined. It is also not specified which information will be supplied by whom.

- Once the information is collected, different situation and varying granularities should be created and evaluated. This has implication in computation. Computations are carried out as individual jobs. Issuing of jobs, executing, monitoring and handling of outcome, are also individually identified for each case. The strategy to handle all these activities is not defined.

- Once situation or complex infons are created, if they holds or not must be evaluated in the form of event.

3.8.1 Assertion Management

As information-processing requirements are being considered from implementation perspective, the first task towards it is the creation of instances of appropriate concepts captured in KB. In this direction, so far concepts, properties and appropriate rules are identified and represented with specific observation and measurement specifications. The interpretation of their state is also available in from of event profiles.

The next task is to create instances of observations that are being reported by the user. The user may report events or status of entities, resources, processes, or activities. They may be as observed and perceived by the user. The user may not be aware of all the observable required to accurately characterize the element being reported. Therefore, the system has to interactively handle the reporting from the user, in order to create an appropriate assertion in KB. The user assertion may be in the form defined in Equation 3.13. In this assertion user in reporting specific State of Affair (SOA) where an individual element a in UoD is *burning*.

$$\langle burning, \dot{a} \rangle \tag{3.13}$$

$$Eventtimestamp = t_e \tag{3.14}$$

$$T_{obs} = t_{s-obs} - t_{e-obs} \tag{3.15}$$

$$t_{rpt-dc} \tag{3.16}$$

Instance creation is a difficult challenge in complex dynamical systems. It marks the beginning of a object life cycle, or it reports an event that will trigger many procedures and actions. A strategy must support handling of assertion initiated by the user as system have to collect the required properties to accurately classify the term being asserted as object of specific class. Once this is done, next task is to find implication of the user assertion. As this is to be one during the runtime, there is no expertise from domain expert or knowledge engineer is available. This task needs to be done automated. Hence, the strategy that determines the handling of assertion is a requirement that needs to be specified.

The model is required to characterize the activity of assertion from user. It allows specification of how user will be interactively asked. This requires selection of a strategy that allows determination of property relevant to the present context.

Model will generate criteria and a template to determine the instance for present context. This will fulfill the missing gap of an appropriate strategy to determine only relevant properties and concepts from potentially vast amount of related terms defined in the knowledge base. This will in turn allow realization of the next step of automated handling of instance creation process in runtime.

One important prerequisite for this model is inclusion of rules that allows identification of relevant properties. This is captured from domain knowledge, local policies and standard operating procedures.

The outcome of this modeling process is a procedure determining the interactive reporting from user. This allows consistent reporting of user experienced/observed events.

In information management domain, the task of managing data sources, and providing visualization with user friendly interlace is played by specific roles. This model is useful for designing user interface for assertion. This model is also used by Knowledge Engineer in making provisions for handling new assertions. Overall it will be useful in content oriented management aspect of information management.

Basic elements in the proposed model are strategy, decisions and input and output as depicted in Figure 3.21. Input to the process is a string submitted by the user `(I 7.1.1)`. The user term needs to be evaluated for corresponding concept captured in KB. This is achieved with a lookup task indicated as `(T 7.1.1)`. For corresponding concept in KB related concepts are identified with a semantic query represented as `(T 7.1.2)`. For each identified concepts, basic properties are identified with another semantic query indicated as `(T 7.1.3)`. For derived concept-property graph, needs to be recorded with the function `(T 7.1.4)`. Output is the `PrimaryInfoNeedList` indicated with `(L 7.1.2)`

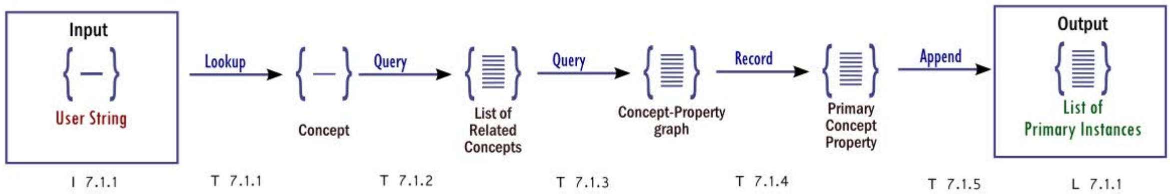

Figure 3.21: Abstract Strategy for Assertion Evaluation

Algorithm 3.1: ASSERTION EVALUATION

input : String from User
output: List of Required Concept with Properties

Accept the term from User UserString ;
Concept ← `Lookup(` *UserString*, $hasConcept$, $?x$`)`;
RelConcepts ← `Query(`*Concept*, $hasRelatedConcept$, $?y$`)`;
Append Concept to RelConcepts ;
foreach *Entry in the RelConcepts* **do**
    RelConceptPro ←`Query(`$Entry$, $hasRelaventProperty$, $?z$`)`;
    foreach *Entry in the RelConceptPro* **do**
        Append Entry to reqInstanceList ;
Load reqInstanceList ;
foreach *Entry in the reqInstanceList* **do**
    PrimaryConceptProp ←`Record(`$Entry$,$Context$`)`;
    Append PrimaryConceptProp to PrimaryInfoNeedList ;

Procedure to handle user assertion is provided as Algorithm 3.1. Input is a user string. output is the list of concepts and properties from KB.

First task is to scan the user input and look-up the input in available KB to find corresponding concept. This is represented by a query `(I 7.1.1)`. This is a semantic query and function `Lookup()` is employed to retrieve concept from KB. Once the concept is identified, the next task is to explore rules. These rules allow identification of relevant properties and other concepts. This is achieved by `Query()` function. The outcome of the semantic query is a `resultSet` that provides a List of relevant concepts `(T 7.1.1)`. The result set may contain more than one concept. For each concept, relevant properties for context are to be identified. This is achieved by another semantic query `(T 7.1.3)`. This results in concept property maps. Identified mappings are loaded in temporary list residing in the memory. Once the determination is complete, the identified list must be stored in the KB. This task is archived by creation of another loop as defined in Line 8-10. Each Entry is loaded, and with help of `Record()` function, it is written in KB.

To understand the working of the strategy, consider that a user enters the word "burning" to the system. Function `Lookup(burning)` results in identification of `FireEvent` Concept in KB. `Query(FireEvent)` allows identification of related concept. The result set contains, `PremiseOwner`, `Police`, `FireStation`, `Ambulance`, `Respondent`. For each entry another query `Query` results in identification of properties like `(Name, PhoneNumber, Address)`. For each concept, related properties are similarly identified. Concept property map is appended in temporary required instance list. As these instances reside in the memory, they must be asserted in KB along with the context. Hence each entry is read and recorded in to KB with reference.

The respondent has observed an event and willing to report that by means of asserting a fact. It is about a concept a. The concept have property p. It is related with other concepts $b, c..$ as indicated in Figure 3.22. The final outcome of the algorithm is the primary information need list `(L 7.1.1)` in Figure 3.21. The count of entries can be defined as follows.

$$\underbrace{(P_1, P_2, P_3, \ldots P_m)}_{C} + \Big(\underbrace{(P_1)}_{C_1} + \underbrace{(P_1, P_2, P_3, P_4)}_{C_2} + \underbrace{(P_1, P_2)}_{C_3} + \ldots \Big) \tag{3.17}$$

$$(Concept + m.P_{Concept}) + \sum_{i=0}^{r} \sum_{j=0}^{s} C_i P_j \tag{3.18}$$

3.8.2 Information Need Determination

Information management strategy so far addressed the issue of handling the reporting of events observed by the users. It includes primary information about type of event and elements involved.

The emerging situation in UoD must be appropriately characterized. For this purpose, the current status of relevant attributes of reported instances must be collected. The user reported instances are entities and processes that can be comprehensively defined by capturing all the attributes recommended in entity and process model discussed earlier. But, in appropriate characterization of present situation only selected attributes are required. Identification of

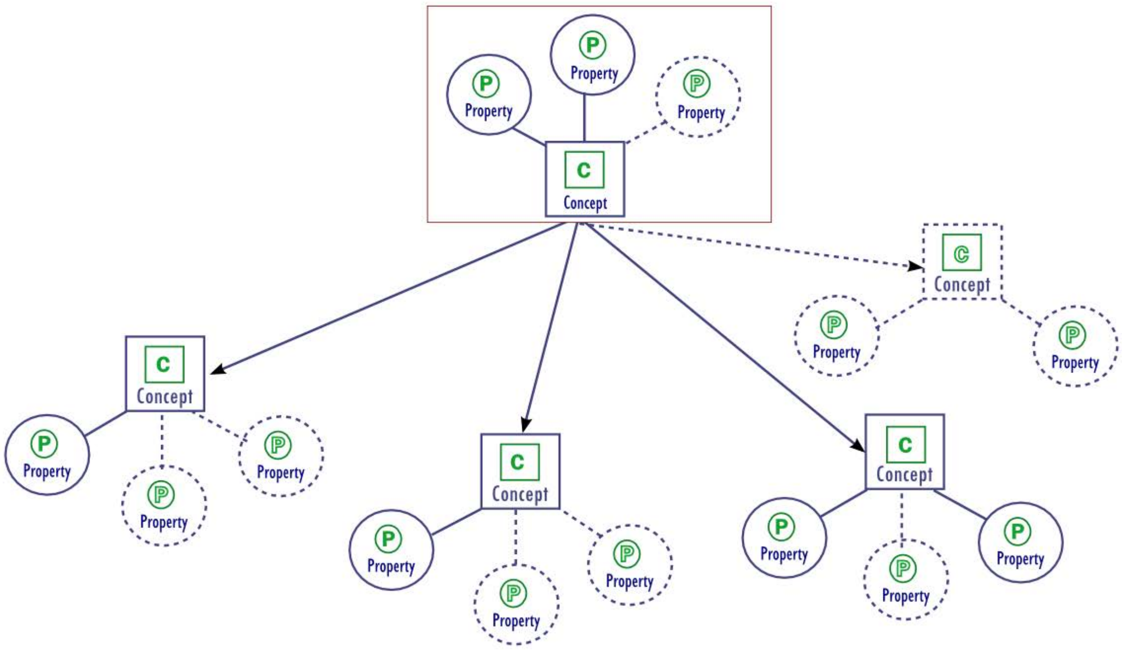

Figure 3.22: Model for Assertion Evaluation

these limited attributes is the next challenge that must be addressed by the strategy. In other word the uncertainty needs to be specified.

$$\begin{aligned} \langle Rescuing, a, \pm \rangle \\ \langle Treating, b, \pm \rangle \\ \langle FireFighting, c, \pm \rangle \\ \langle Reporting, d, \pm \rangle \\ \vdots \end{aligned} \tag{3.19}$$

A strategy is missing that allows identification of only required attributes for each instance reported by the user. This strategy is useful as it allows specification of all the required attributes that are necessary for proper situation awareness. At the same time it also prevents collection of those attributes that may not be relevant. Collection of irrelevant attributes not only increases the information-handling load, but also increases the effort of information sources in reporting them.

A model is required to characterize the information need for situation awareness in given UoD. The model allows specification of exact attributes and observation details. By doing so the proposed model provides a strategy to determine information need from vast amount of concepts and rules available in KB. This will provide a template that fills the gap of attributes and concepts required for characterization of situation. This identification will enable the strategy to further make provision for collection of information. Property and concept relations defined in such a way that can facilitate determination of interdependence. This is already achieved by defining all the activities and processes according to entity and process model followed by representation that can be referred.

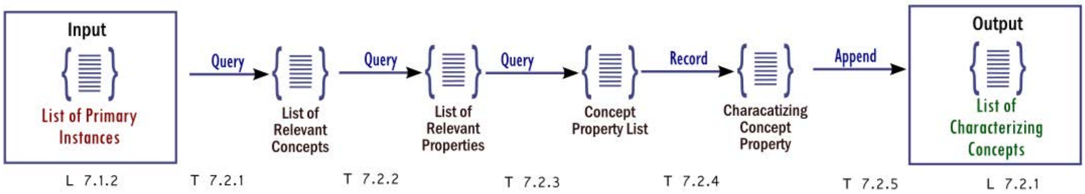

Figure 3.23: Information Need Specification

Desired outcome from this information-processing model is a list of characterizing concepts. This list includes the set of concepts and properties required for accurate characterization of instances for further processing.

This is used by knowledge engineer (KE) by an automated process to determine relations that allow creation of infons.

Basic elements in the proposed model are in the form of strategy, decisions and input and output. Input to the process is list of primary information need derived from previous strategy, indicated as `(L 7.1.1)`. A semantic query `(T 7.2.1)` is required to identify relevant concepts occurring to the rules. For each concept derived based on the rules, the relevant properties are identified with another semantic query `(T 7.2.2)`. The concept property graph is build with `(T 7.2.3)`. This temporary graph is then recorded with function `(T 7.2.4)`. Output of the process is the updated list `(L 7.2.1)` that include characterizing concepts for situation being reported.

For determination of information need, Algorithm 3.2 INFORMATION NEED DETERMINATION is proposed. It takes primary information need list `(L 7.1.2)` as the input. Output of this strategy is a list of characterizing concept and properties `CharConceptPropList` as indicated with `(L 7.2.1)` in Figure 3.23.

Line 1 of Algorithm 3.2 indicates the loading of the input to the memory. This list contains concept property map identified for further characterization. Each member of the list should be referred to identify relevant properties. Loop in line 3-6 is maintained for this purpose. For individual concept, related property is to be identified. This is achieved with a semantic query `(T 7.2.1)`. This retrieves all dependents for the given concept. Resulting concepts are stored in a `resultSet`. These are just concepts and may have many properties, out of which only few are important in present context. Loop is created 4-6 that will query `Query() (T`

Algorithm 3.2: Information Need Determination

```
   input  : PrimaryInfoNeedList
   output: List of Required Concept with Properties
1  Load PrimaryInfoNeedList ;
2  foreach Entry in the PrimaryInfoNeedList do
3      ResultSetConcept ← Query(Concept hasRelConcept ?x);
4      foreach Entry in the ResultSetConcept do
5          ResultSetConceptPropery ←Query(Concept hasRelProperty ?y);
6          foreach Entry in the ResultSetConceptPropery do
7              Append Entry to ConceptPropList ;
8  Load ConceptPropList ;
9  foreach Entry in the ConceptPropList do
10     CharConceptProp ←Record(Entry, Context);
11     Append CharConceptProp to CharConceptPropList ;
```

7.2.2). The resulting concept property map is appended in a temporary `InstanceList`. This instance list is in memory, and must be included in KB. Once concept property map of all identified concepts are completed, the resulting temporary list is loaded in memory. Line 8-10 describes the strategy to record them in KB so that it can later be utilized. Each entry containing the concept property map is recorded in KB with the context of the identified event. The individual mappings are recorded on a list `CharConceptPropList` in Line 10. This enables tracking of required individual mappings are created for characterization of state of affairs.

Rescuing, treating are the concepts available from primary information need list. These concepts are now further processed to identify characterizing information needs. Rescuing and treating both are important activity, yet their information needs are different. The query to identify *hasDependence* for rescuing activity results in identification of few more concepts like (rescuer, mask, etc.). The outcome of the same query for Treating will be more complicated for the "Treating" concept as it will result in list of Medical Staff, Medical Equipments and treatment workflow. Hence the number of concept as a outcome of query `(T 7.2.1)` will depend upon the number of activities the concept is participating in.

The Notation 3.20 and 3.21 provides quantitative overview of the process. For identified each concept in (C), corresponging concepts (D) are identified. Each concepts in (D) possess several properties from (P) that can be relevant for characterization of the situation.

$$[\underbrace{\underbrace{(P_1+P_2+P_3)}_{D_1}+\underbrace{(P_1+P_2+P_3)}_{D_2}}_{C_1}+\underbrace{\underbrace{(P_1)}_{D_1}+\underbrace{(P_1)}_{D_2}}_{C_2}\ldots] \tag{3.20}$$

$$\sum_{i=0}^{r}\sum_{j=0}^{s}\sum_{k=0}^{t} C_i D_j P_k \begin{cases} r = & \text{Number of Concepts } (C) \text{ in Rule} \\ s = & \text{Number of Concepts } (D)\text{in Actor Network} \\ t = & \text{Number of relevant properties } (P) \text{ of Concept} \end{cases} \quad (3.21)$$

Equation 3.22 represents that time taken for identification of relevant concept and properties can be defied as T_{ini} that is derived from taking difference of time instance t_{s-ini} and t_{e-ini} that denotes starting and ending point of the given process.

$$T_{ini} = t_{e-ini} - t_{s-ini} \quad (3.22)$$

3.8.3 Information Need Specification

In a typical runtime scenario, a user report has been handled to describe the observed event. The assertion management strategy specified interactive reporting from the observer. In the reported concept, the information need determination strategy allowed dependence to determine the ramification of the observed event; resulting in a list of concept and properties that might be affected and therefore be monitored. This is done with reference to the domain knowledge captured in the form of rules. Yet, the identified information need is not complete, as it only provides class and property without any reference to the observed event.

Next task is to convert abstract information need in to concrete information need specifications. Along with the name of concept and property, the specification, must also incorporate known instances collected earlier. Depending up on the event, the property may have large spatial and temporal footprint. The resulting specification must contain exact temporal and spatial reference as indicated in Equation 3.23.

$$\begin{aligned} \langle Rescuing, a, l_{w1}, t_{t+\triangle t}, \pm\rangle \\ \langle Rescuing, a, l_{w1}, t_{t+2.\triangle t}, \pm\rangle \\ \langle Treating, a, l_{w2}, t_{t+\triangle t}, \pm\rangle \\ \langle Treating, a, l_{w2}, t_{t+2.\triangle t}, \pm\rangle \\ \vdots \end{aligned} \quad (3.23)$$

It can be expected that for major events, the impact of event is visible for larger area for longer period of time. The reporting is an atomic infon holding information about a small part of the world. To cover the exceeding area and time, multiple such infons should be collected.

The strategy to determine the intervals in space and time domain is missing. While multiple information sources are involved with observable properties of various domains. This requires a strategy for consistent specification of information need.

This model is required to characterize time space and instance specific requirements of the information need. A strategy is required to identify various time intervals at which information needs to be updated. Similarly, the value properties are to be collected for spatial coverage, and hence spatial intervals are also to be identified. There can be many alternatives to manage each of them. The proposed model is required to realize a strategy for these requirements.

A solution to identified problem, a template can be generated that holds observation and measurement specifications for each property for identified time and space intervals. This will fulfill the gap of instance specific identification of information need. This in turn will facilitate the collection of information from available instances as identified in KB.

Observation and measurement specification are unique for specific properties from various domain point-of-views. Hence, the knowledge representation encoding the Observation Model is required to determine spatial temporal and semantic coverage of the observation. The assignment of specific observed value is the task that must be based on the knowledge encoded using Measurement model. This allows determination of the Unit of Measure (UoM), attribute type and other important feature of the measurement. Spatial intervals are to be determined based on the geographical representation of the UoD. Hence a spatial database covering the UoD and made accessible for query is important requirement for this model.

The outcome of the proposed Information Need Specification model is the algorithm proposing the strategy to generate specification according to the identified need.

This strategy is useful to the content related role of information management practice. The role is responsible for the identification of information sources in given configuration.

Basic elements of the model are depicted in Figure 3.24. Number of specifications required to be generated. Following questions are difficult to answer: How they are going to be managed? What are the content and the structure? How it should be generated? How it should be recorded and managed in permanent storage?

INFORMATION NEED SPECIFICATION Algorithm characterizing information need is proposed as Algorithm 3.3. It takes `CharConceptPropList` as input and provide `InfoNeedSpecification` as output. In Line 1 `CharConceptPropList` is loaded as input to the process. In Line 2, the estimated duration `EstiRespDuration` and estimated area `EstiRespArea` of response is read. Each concept in `CharConceptPropList` should be processed further to determine information need based on the estimated values of response. For this a loop is maintained. A semantic

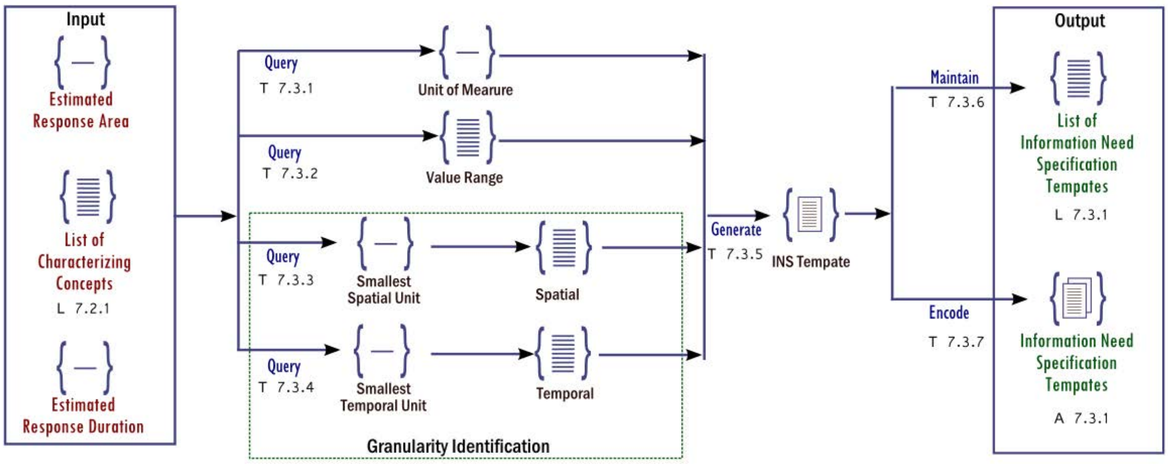

Figure 3.24: Information Need Specification Abstract

query is utilized for determination of Unit of Measure for given property. Similarly Value range is determined for the given property. For estimate of spatial span of information need, required spatial instances are calculated with the help of smallest spatial unit and `EstiRespArea`. Similarly required temporal instances are calculated.

For each identified spatial interval, information needs to be collected at specific temporal granularity. Hence, all the spatial instances are recalled in the memory. For each spatial instance, individual time-instances identified earlier are loaded. At this level, the instantaneous temporal and spatial interval is utilized along with previously identified property related parameters like unit of measure, possible value range to build the specification. The generate function utilized to create information need specification template using these identified values.

Each of these instances is available in memory, and they should be appropriately encoded to suit the application requirements. This is done by calling the encode function for all the INSTemplates in temporary list. To maintain a register of all the generated specifications, an entry of encoded specification is appended to information need specification list.

Notation 3.24 and 3.25 represent the scale of computation required in this effort. As depicted, total number of specification is affected by number of properties identified. For each property, required spatial and temporal intervals are identified based on scale and duration of estimated response. Hence, larger the scale, more number of information need specifications will be generated.

$$[\underbrace{\underbrace{(T_1 + T_2 + T_3)}_{S_1} + \underbrace{(T_1 + T_2 + T_3)}_{S_2}}_{P_1} + \underbrace{\underbrace{(T_1)}_{S_1} + \underbrace{(T_1)}_{S_2}}_{P_2} \ldots] \tag{3.24}$$

Algorithm 3.3: INFORMATION NEED SPECIFICATION

```
input  : CharConceptPropList, EstiRespDuration, EstiRespArea
output: InfoNeedSpecification and entry in InfoNeedSpecificationList
Load CharConceptPropList ;
Read EstiRespDuration, EstiRespArea ;
foreach CharConceptPropList do
    UoM ← Query(Property hasUoM ?x);
    ValueRange ← Query(Property hasValueRange ?y);
    SpatialUnit ← Query(Property hasSmallestSpatialFootprint ?z);
    RequiredSpatialInstances ← Calculate(SpatialUnit, EstiRespArea);
    TemporalUnit ← Query(Property hasSmallestTemporalFootprint ?a);
    RequiredTemporalInstances ← Calculate(SpatialUnit, EstiRespDuration);
    foreach RequiredSpatialInstances do
        foreach RequiredTemporalInstances do
            INSTemplate ←
            Generate(name,Property,UoM,ValueRange,TimeInst,SpaceInst);
            record INSTemplate to INSTemplateList ;
Load INSTemplateList ;
foreach Entry in the INSTemplateList do
    Encode InfoNeedSpecification ;
    Append to InfoNeedSpecificationList ;
```

$$\sum_{j=0}^{r}\sum_{k=0}^{s}\sum_{l=0}^{t}\Big(P_j S_k T_l\Big)\begin{cases} r = & \text{Number of Properties } (P) \\ s = & \text{Number of Spatial Intervals } (S) \\ t = & \text{Number of Temporal Intervals } (T) \end{cases} \tag{3.25}$$

Equation 3.26 represent the duration T_{insp} required in calculation of information need specifications as a difference of two time instances t_{s-insp} t_{e-insp}

$$T_{insp} = t_{e-insp} - t_{s-insp} \tag{3.26}$$

3.8.4 Messaging Pattern Generation

Information need specification strategy is defined in Algorithm 3.3 enables generation of templates that are used for information collection. With time and space coordinates defined as per the configuration, the collection of information is facilitated with the provisions of specifics. The strategy allows automated identification and specification of required templates in response to recorded triggering events.

Templates are generated and stored in the vast KB. The users who can assign observed values in these templates are scattered across the UoD. Form Situation awareness theory view-point, the users are expected to report information as indicated in Equation 3.27.

$$\begin{array}{c}\langle telling, a, \pm\rangle \\ \vdots\end{array} \tag{3.27}$$

Comprehensive information need is identified in the system, and users are not aware of all the information need they are expected to report back to the system. As a solution to this problem, systems have to inform user about the required information need at required granularity as specified in the templates. The information flow is therefore; initiated from the central system that holds the repository of templates, to the individual users reporting the observation back to the system for defined time period. This information flow subsequently supports the required interaction patterns among system and the instances. Realization of interaction patterns in system environment requires specification of a messaging framework[12]. Hence, the immediate next challenge turns out to be realization of the messaging framework that supports various interaction patterns.

Figure 3.25 provides basic interaction pattern among user and system. Basic component of messaging framework are the roles of Sender and receiver and the message. The same entities may play different roles during interactions. The content of the message also is specific to the information need specification. Apart from these basic roles that can be sufficient for simple

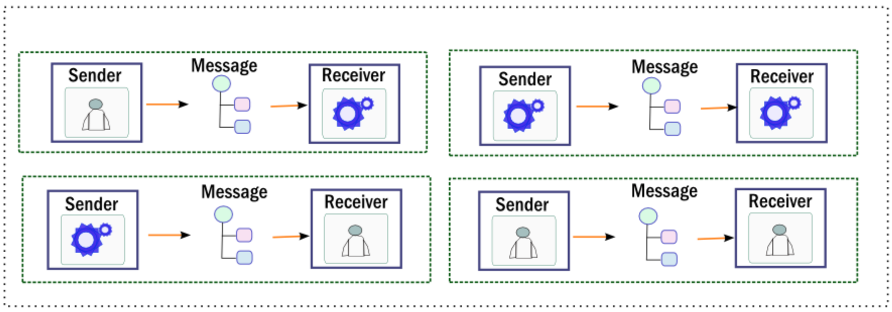

Figure 3.25: User System Communication

interaction, special provisions are sometimes necessary in meeting special information needs. Request-reply model, publish-subscribe model, constant polling client, type channel are some of the examples of the specialized role that are assigned to meet specific requirements. These are refereed to as messaging patterns [13], and must be employed to specific scenario.

Specific messaging patterns are required to meet the information need specified in the templates. Large amount of specifications are automatically generated. Identification of appropriate messaging patterns for the templates is a difficult challenge. Individually identified patterns do not serve the purpose, until each is linked with appropriate other pattern to support the interaction indicated in Figure 3.25. Therefore, finding out mapping and specification of the same is also equally challenging task.

This model is required to identify specific messaging patterns required to support interaction patterns. The characterization of patterns individually created from information need specifications. It provides strategy for instantiation of messaging patterns. Model is required to discover and establish mappings among generated pattern instances.

This will allow automated creation and mapping of messaging patters necessary to sustain desired information flow. Pattern maps fulfill the gap of required messaging framework that was identified as challenge. This will in turn allow realization of next task that is to support interaction patterns to fill the information template with observations from the user at identified granularity. The prerequisite for message pattern generation includes identification of users that may act as source of information. The outcome of observation model and measurement model are also prerequisites to this task. Message pattern generation is carried out with specific types. Measurement model provides specification on type. The time and space interval requirement must also be identified which is supplied by the information need specification templates. Apart from this information useful in creation of the pattern, the knowledge about messaging patterns is also required. The rules defined in the KB provide specification for creation of pattern for specific purpose. It also provides guidance on how multiple patterns are appropriately mapped and handled to meet the interactions.

Outcome of the modeling process is a procedure to generate patterns map for supplied information need specification.

Information Management involves handling of external information sources. Information is collected from these sources with help of messaging provisions. The messaging related aspects are considered by the communication engineer role. This strategy is useful for this role in creation, haling and monitoring of messaging patters for instantiations information needs.

Figure 3.26 depicts the elements of proposed messaging pattern detection strategy.

First decision is to determine the source of information that can be used as input to this strategy. If only information need is considered, it does not provide granularity. If information need specification is considered, they are many for each reading, and for all of them individual messages are not required to be created. The strategy to determine messaging pattern for given information need specification is therefore a difficult challenge as there can be many types of patterns suitable for given input. The decision about selection of pattern maps, as for given patterns how many other pattern must be linked. It is also difficult to determine the strategy

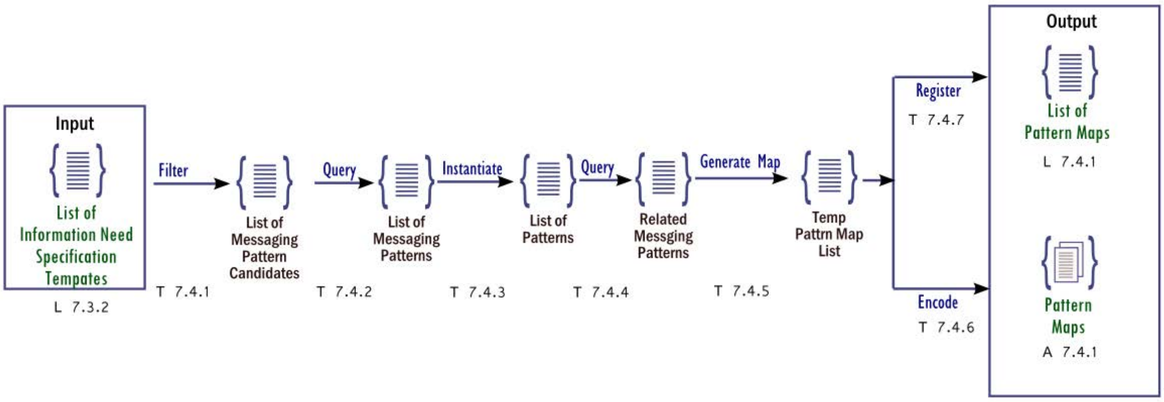

Figure 3.26: Messaging Pattern Detection

to specify the mapping. As the mapping is used by the system to automatically configure and handle the interaction during the runtime.

Life span handling is difficult as decision regarding instantiation up to destruction of a pattern have many implications. Created entities do not exist in isolation, but they participate in communication. When life cycle is completely managed by automated procedure, the participation needs to be held automatically as well. This allows identification of mappings among patters.

An Messaging Pattern Generation is proposed as 3.4. This algorithm guides creation of messaging patterns. It takes `InfoNeedSpecificationList` as input. With successful execution, the algorithm results in creation of messaging patterns. These patterns are mapped to support interaction pattern. The outcome of the algorithm is therefore *PatternMapList* that contains a record of generated pattern mappings.

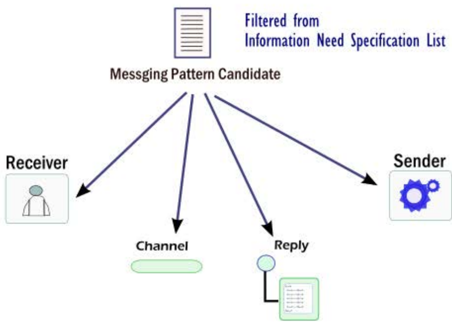

Figure 3.27: Messaging Pattern Generation

First task is to select the source of information to identify the message pattern requirements. Since specific information needs are already identified with space and time granularity, this information can be used for further reference. The `InfoNeedSpecificationList` is therefore selected and loaded to the memory as indicated in Line 1. Each member should be processed for pattern generation. Hence, the Line 2 indicates initiation of a loop that reads each entry one by one. For the loaded entity, the corresponding messaging pattern is identified by accessing the message creation rule for that property. This is realized with a function providing semantic query. The same property may require more than one message pattern; hence, another loop is required to individually treat each of the identified members. For each identified pattern type, corresponding pattern needs to be created. The patterns are created with `Instantiate` function and recorded in the pattern list `PatternList`. This list contains all isolated patters. To realize the interaction pattern, they must be mapped. For this reason, the members of the list are loaded in memory. Mappings may result in identification of many pattern maps. Each of them is again treated with a loop. There each component is utilized to prepare a `generateMap`. This is also recorded in the KB.

Algorithm 3.4: MESSAGING PATTERN GENERATION

```
    input  : InfoNeedSpecificationList
    output: PatternMapList
 1  Load InfoNeedSpecificationList ;
 2  foreach Entry in the InfoNeedSpecificationList do
 3      RSMPCandidate ← Filter (Entry);
 4      Append RSMPCandidate to MssPatCandidateList
 5  Load MssPatCandidateList ;
 6  foreach Entry in the MssPatCandidateList do
 7      resultSetMP ← Query(Entry MessagePattern ?x);
 8      foreach Entry in the resultSetMP do
 9          Instantiate(Entry);
10          Append to PatternList ;
11      Load PatternList ;
12      resultSetRelMP ← Query(Entry RelatedMessagePattern ?x);
13      foreach Entry in the resultSetRelMP do
14          GenerateMap(Entry);
15          Append to tempPatternMapList ;
16  Load tempPatternMapList; foreach Entry in the InstanceList do
17      Term ←Query(Entry);
18      Append to PatternMapList ;
```

Notations 3.28 and 3.29 represents total number of messaging patterns generated and corresponding mappings identified among them.

$$\Bigg(\underbrace{\underbrace{P_1 + P_2 + P_3}_{c_1}}_{(I_1)+(I_2)+(I_3)+(I_4)+(I_5)} + \underbrace{\underbrace{P_1 + P_2}_{c_2}}_{(I_1)+(I_2)+(I_3)} \Bigg) \tag{3.28}$$

$$\sum_{i=0}^{r}\Big(\sum_{j=0}^{s}C_iP_j\Big)+PM_i\begin{cases} r = & \text{Number of Candidates}(C)\text{ identified for set of templates}(I)\\ s = & \text{Number of Patterns}(P)\text{ Generated}\\ PM = & \text{Identified Mapping among patterns }(PM)\end{cases} \tag{3.29}$$

The time taken by the strategy is identified as T_{mpgn}, derived from difference between the time instance t_{s-mpgn} and t_{e-mpgn} each represents starting point and ending point of the process.

$$T_{mpgn} = t_{e-mpgn} - t_{s-mpgn} \tag{3.30}$$

3.8.5 Assignment of Communication Role

With the help of procedure identified as Algorithm 3.4, Messaging patterns are identified based on the information needs. The strategy not only provides identification of the messaging patterns only, but also links individual patterns with map. Hence, the patterns maps can be employed in hosting communication for a set of information need templates.

Next task is to enable usage of the messaging pattern. From information management and situation awareness point of view, individuals should be assigned specific roles in mapped pattern. As indicated in Equation 3.31, the infon should be holding an instance that will provide a utterance situation.

$$\begin{aligned} u \models &\Big\langle Utters, \dot{a}, \dot{b}, \dot{l}_w, t_{\dot{t+\triangle t}}, \pm\Big\rangle\\ &\Big\langle refereesto, \dot{a}, \dot{b}, \dot{l}_w, t_{\dot{t+2\triangle t}}, \pm\Big\rangle\\ &\Big\langle refereesto, \dot{a}, \dot{b}, \dot{l}_w, t_{\dot{t+3\triangle t}}, \pm\Big\rangle\\ &\vdots \end{aligned} \tag{3.31}$$

In realizing the requirement of the next goal, user reports must be collected at desired temporal intervals. The instances are known, the messaging patterns are known, but identification of instance that will carry out with messaging task using generated patterns is not known. Hence missing feature is specification of complex infons in the form of Equation 3.31, that contains instance, time and message information.

Model is required to characterize all the required communication roles for situation awareness. Model is required to specify the content of communication. For identified roles, the model generates appropriate task specification. The model is also needed to provide strategy to collect the parameters for the task. The model is required to identify mapping among actual instances and Communication Roles identified in pattern map. A strategy is required to link the roles with instances in KB. The lifespan of roles and possible instances may not be exactly same. Hence, a strategy is required to schedule the messaging task for planned communication to appropriate instance.

The prerequisites for this process are the concepts and rules in knowledge base, instances, and specific value properties that provide critical input to the decision required in the process.

Outcome is in the form specified communication tasks that includes involved roles, messages, and communication requirement to build the required situation awareness. Realization of various communication and information exchange patterns is the responsibility of Message oriented Middleware Engineer. The information management strategy requires input from MoM engineer in the form of messaging framework over which the information flow can be realized. The automatic generation of required messaging tasks facilitates the task of MoM engineer.

Figure 3.28 depicts the elements of proposed communication role assignment strategy. Decision

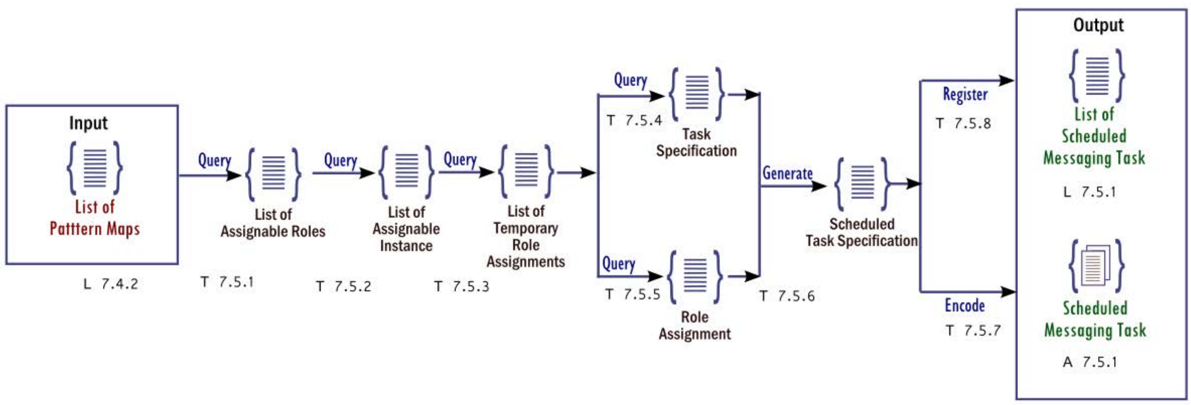

Figure 3.28: Communication Role Assignment

regarding selection of source of information, assignment of role, determination of required interaction patterns, and publication of role assignment and scheduling of communication tasks poses difficult challenge.

Communication Role Assignment model is proposed as Algorithm 3.5. This takes `PatternMapList` as input. With the processing, it assigns specific role(s) to individuals. The individual messaging task that needs to be carried out is specified as scheduled messaging task and record is maintained in `ScheduledMssgTaskList (L 7.5.1)`.

The starting point of the procedure is to select the pattern maps that are to be assigned an individual instance. This information is stored in `PatternMapList`. Line 1 indicates the loading of the `PatternMapList` to the memory. Not all patterns can be assigned roles, hence, for selected pattern map, to decide if the pattern needs to be assigned a role, a function is required. This is achieved by a Semantic Query indicated as `(T 7.5.1)`. For given pattern map, there can be one or more assignable patterns. They are stored in a list `(RSAssgnRole)`. Next step is to determine proper individual for identified role. Hence, each member of the list is to be taken and a role should be assigned to an individual instance. This is achieved by maintaining a loop for this task as indicated in Line 4-6 of the algorithm. The query is required to select assignable instance. This is achieved with a semantic query `(T 7.5.2)`. Identified individual instance is stored as `(AssgnInst)`. Identified Instance is recorded in an `InstanceList` along with the messaging pattern role. For each identified roles in the pattern map, other individuals are similarly identified. The resulting `InstanceList` is then appended to `tempRoleAssignmentList`. The list is in memory and available in execution environment. It must be properly encoded and stored in KB.

The assigned roles are made available for reference, yet when particular role is required to take action must also be identified. This is derived from identified information needs. Information need specifications include the temporal specification along with the other details of required observation and measurement. The temporal intervals are identified with the help of a semantic query. The result contains information required at all identified time intervals. For each interval task needs to be carried out, hence a Loop is maintained for this purpose. `GenerateTask` method is used to schedule the task at each interval. The scheduled task is in the memory, and should be placed for permanent reference. First requirement is to encode the property of the task specification. Encode function in line 15 is defined for this purpose. Another requirement is to record the identified task in to a list as indicated in line 16.

$$[\underbrace{\underbrace{(STsk_1 + STsk_2 + STsk_3)}_{Role_1} + \underbrace{(STsk_1 + STsk_2 + STsk_3)}_{Role_2}}_{PM_1} + \underbrace{\underbrace{(STsk_1)}_{Role_1} + \underbrace{(STsk_1)}_{Role_2}}_{PM_2} \ldots] \tag{3.32}$$

$$\sum_{j=0}^{r}\sum_{k=0}^{s}\sum_{l=0}^{t}\left(PM_j Role_k STsk_l\right) \begin{cases} r = & \text{Number of Pattern Maps } (PM) \\ s = & \text{Number of Identified Roles } (Role) \\ t = & \text{Number of Scheduled Tasks } (STsk) \end{cases} \tag{3.33}$$

Equation 3.34 provide time taken by the strategy in determining and scheduling the communication tasks. This is defined as interval T_{cra} derived by taking difference of time instance t_{s-cra} and t_{e-cra}.

Algorithm 3.5: Communication Role Assignment

```
input  : PatternMapList
output: AssignedPatternMapList and ScheduledMssgTaskList
Load(PatternMapList) ;
/* ********* Identification of Instances for Messaging Patterns *********  */
foreach Entry in the PatternMapList do
    RSAssgnRoleList ←Query(Entry isAssignableRole ?x);
    foreach RSAssgnRole in the RSAssgnRoleList do
        AssgnInst ←Query(Entry isAssignableInstance ?y);
        Append (RSAssgnRole, AssgnInst) to InstanceList ;
    Append InstanceList to tempRoleAssignmentList ;
/* ********* Specification of Role Assignments *********  */
Load tempRoleAssignmentList ;
foreach Entry in the tempRoleAssignmentList do
    RoleAssignment ←Query(Entry);
    Append RoleAssignment to RoleAssignmentList ;
    /* ********* Specification and Scheduling of Communication Tasks *********  */
    TaskRqList ←Query(Entry hasCommTasRequirement ?z);
    foreach  TaskRq in TaskRqList do
        SheduledTaskSp ←GenerateTask(RoleAssignment, TaskRq);
        SheduledTask ← Encode (SheduledTaskSp);
        Append SheduledTask to ScheduledMssgTaskList ;
```

$$T_{cra} = t_{e-cra} - t_{s-cra} \tag{3.34}$$

3.8.6 Representation Generation

According to the information processing strategy defined so far information is communicated as per the specification. The scheduled tasks created for each role provides guidance for carrying out communication. Various messaging patterns are used for this purpose. The message received contains the required information.

The message template was build with identified information need specification and the response is holding assigned values as observed by the users. These messages are in messaging environment and therefore individual content should be extracted from message patterns and incorporated in KB as assertions. These assertions can be handled to facilitate the next procedures.

The availability of required values can be estimated from the list of scheduled messaging tasks. Messaging task is scheduled in a future time instance, which is known in the system. The system has to wait until the scheduled time arrives. The time duration spend in waiting for the scheduled task is denoted as Equation 3.35.

$$T_{wait} = t_{sc-tsk} - t_{sc-chk} \tag{3.35}$$

Notation 3.36 indicated the time instance t_{c-rcv} at which, the communication is received by the observer. This is the time instance at which the observed is requested the observation.

$$t_{c-rcv} \tag{3.36}$$

Upon availability of observation request, the observer carries out measurement of specific property. The measurement activity may take some time denoted with T_{msur} as indicated in 3.37.

$$T_{msur} = t_{e-msur} - t_{s-msur} \tag{3.37}$$

Once the measurement is over, the information request is replied, that is denoted with time instance t_{c-rply}.

$$t_{c-rply} \tag{3.38}$$

In processing further, access to information about replied messages is necessary. This will enable processing of the message for extraction of required information. All individual messages are processed to create a complete representation. At any given instance, there can be numerous messages that can potentially be useful for this purpose. A strategy that allows automated handling of message extraction is missing.

The representation model is required to characterize the representation suitable for situation awareness. It is needed to specify the content details of the representation. It provides strategy to extract information from messages. It also specifies strategy to prepare representation parameters. Adoption of representation strategy will enable messages extraction to enable representations. Representation derived as result fills the information gap, therefore allows further processing the collected data.

Prerequisite for representation generation model include the outcome of information need specification model, assigned patterns etc. Information need specification suggests details of the information requirements. Outcome of algorithm allows identification of roles and patters responsible for reporting specific messages. The scheduled task list provides time reference when the records will be available. With this information extracted message can be utilized for generating representations.

Outcome is the procedure to prepare representation from communicated message. Procedure takes list of scheduled message tasks and extracts the information to create representations.

Information management task of handling external information sources is handled by communications engineer role. This is achieved by extracting communicated messages and preparing the representations. These representations are provided to the decision makers. The automated strategy for generation of representation helps the role in providing the same.

Figure 3.29 depicts the elements of the proposed representation generation strategy. Difficult

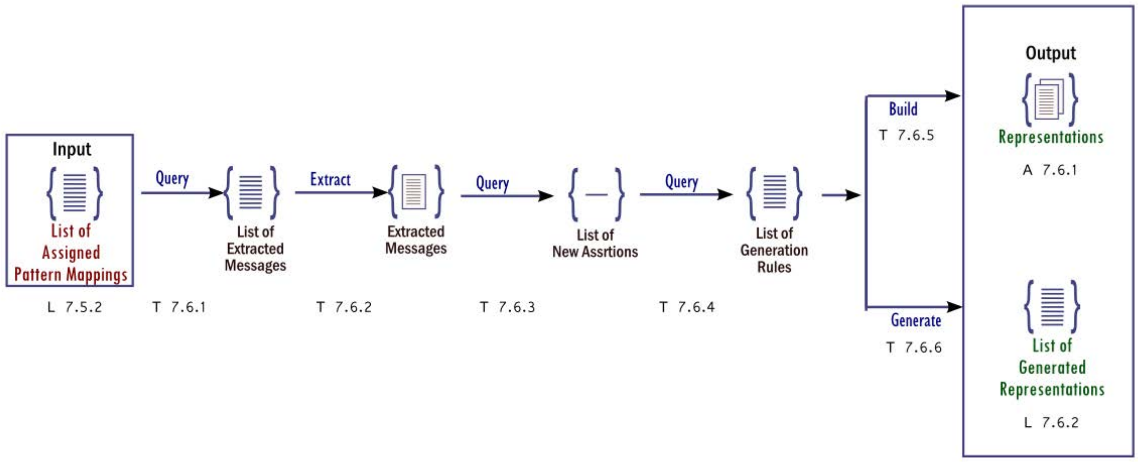

Figure 3.29: Representation Generation

part is to determine time of availability of the message. Decision is also difficult regarding extracting of the message and using it further. They can be created as new record that can further be integrated. This can increase space requirements. The templates are already generated and publication of filled template also requires the additional space requirement.

Representation Generation procedure is proposed in the form of Algorithm 3.6. Input is the `ScheduledMssgTaskList` as it contains the information about who will provide what information and when. The output of the procedure is Generated representation recorded in the list `GeneratedRepList`. `ScheduledMssgTaskList` is loaded in the memory. Loop created in line 2-6 is maintained to process each entry on the list. For given entry in scheduled message task, it is required to identify pattern instance that is holding the message. The outcome is the message pattern that is subjected to `Extract` procedure to enable extraction of the message. The extracted message is containing the property along with the observed value. This property value pair must be stored in a temporary instance list.

Once this task is completed, the list is loaded again in the memory. For each entry in the

Algorithm 3.6: REPRESENTATION GENERATION

```
   input  : ScheduledMssgTaskList
   output: GeneratedRepList
 1 Load ScheduledMssgTaskList ;
 2 foreach Entry in the ScheduledMssgTaskList do
 3     Message ← Lookup(PatternInstance);
 4     MessageContent ← Extract(Message);
 5     PVPair ← Build(MessageContent);
 6     Append PVPair to InstanceList ;
 7 Load InstanceList ;
 8 foreach Entry in the InstanceList do
 9     InfoNdSpInstance ←Query(InformationNeedSpecification);
10     Update(InfoNdSpInstance,PVPair);
11     Record InfoNdSpInstance to GeneratedRepList ;
```

instance list containing the property value pair, corresponding information need specification is looked up. The resulting instance is stored as `infoNdSpInstance`. This information need specification instance is already containing a blank template. `Update` procedure is called with extracted property value pair. This enables assignment of the observed value in the template. The template filled with observed value, may now act as a representation, and can be used in further analysis. The updated template is recorded so that other programs can notice. This task is done by function defined in Line 10. The information about the updated instance is recorded in the `GeneratedRepList`. Notation 3.39 and 3.40 represents number of messaging tasks scheduled at given time instance holds message. Each message holds observed values required to be extracted.

$$[\,\underbrace{\underbrace{\underbrace{(V_1)}_{M_1}}_{STsk_1} + \underbrace{\underbrace{(V_1)}_{M_1}}_{STsk_2}}_{T_1} \ldots] \tag{3.39}$$

$$\sum_{j=0}^{r}\sum_{k=0}^{s}\sum_{l=0}^{t}\Big(V_j M_k STsk_l\Big)\begin{cases} r = & \text{Number of Observed Values (V)} \\ s = & \text{Number of Messages (M)} \\ t = & \text{Number of Scheduled Tasks (STsk)} \end{cases} \tag{3.40}$$

Equation 3.41 represents duration of time T_{rpgn} taken by the process in generation of the representations.

$$T_{rpgn} = t_{e-rpgn} - t_{e-rpgn} \tag{3.41}$$

3.8.7 Event Detection

In information processing strategy so far, discussed generation of representations representation according to identified information needs. These representations are filled templates of information that are to be used for further evaluation.

The next task is to read and evaluate the representation for detection of possible events. This defines the factual reference to possible events. The representations individually or in combinations are to be evaluated for specific events. This is done with event profiles generated in Section 3.7.

Missing link is the strategy that governs the evaluation of generated representations. There are many representations being generated and automatically being updated. The strategy for profile matching needs task is missing.

The model is required to determine which events can be detected based on collected and generated representations. The strategy is required to determine the time for event profile matching. The model is required to specify the parameters that are required and strategy to collect and specify the parameters for the representation being generated, or scheduled to be generated.

Event Profiles and tasks generated with this model fill the gap of what to test, how to evaluate, and enable interpretation of current state of affair based on observed representations.

Primary prerequisite is the outcome of eventing model. The outcome include the concept of event, various event profiles, and concepts related to eventing [72] defined in terms of situation awareness system. The representations are created with a value assigned to a observable property. The Measurement model is also a prerequisite as it establishes the scale, unit of measure and other specification about the generated representations. Hence observation model is required that specifies when assignments are carried out[73].

Outcome of the proposed model is a procedure that allows determination of event detection tasks for generated representations. The tasks can be for detection of simple and complex events.

Information management domain have task in analysis of the data. The role responsible is analyst or decision maker. The role is responsible for evaluating collected data. For continuous stream of data it is difficult. Proposed strategy allows automated detection of event as and when the required representations are available.

Figure 3.30 depicts the elements of the proposed event detection strategy.

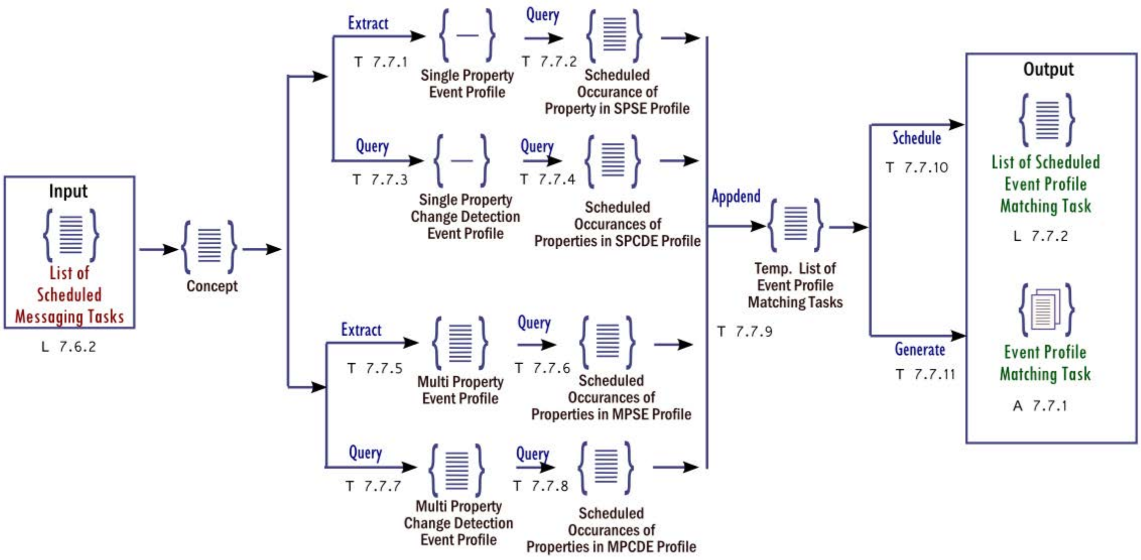

Figure 3.30: Abstract Information Processing Model: Event Detection

Event Detection strategy is proposed as Algorithm 3.7. The algorithm takes scheduled message task list `ScheduledMssgTaskList` as input and generates Event Profile Matching Task List `EventProfileMatchingTaskList`. In line 1 it reads the scheduled messaging task list and takes one entry for processing. It identifies the concept in the list to determine the possible event profiles to match. This algorithm is limited to four types of event profiles as discussed below.

Single property Simple Event: In line 4 the algorithm issues a query to retrieve single property event profile for given concept. All returned profiles are stored in one array. Each profile is then searched for schedules and resulting scheduled tasks are stored in an array. All such profile matching tasks are then strored in temporary profile matching task list.

Single property Change Detection Event : In the next concept new query is issued retrieving change detection type of profiles. This time the change is while scheduling the task; it is necessary to determine when second reading will be made available. Rest of the logic remains the same. The identified profile matching tasks are appended to task list.

Multiple property Simple Event: The same concept can also participate in profiles that requires multiple parameters. In this type of events it is determined which other concept is

Algorithm 3.7: Event Detection

```
input  : ScheduledMssgTaskList
output: Scheduled Event Profile matching task and entry in
        EventProfileMatchingTaskList

Load ScheduledMssgTaskList ;
foreach Entry in the ScheduledMssgTaskList do
    Concept ← Query(Entry);
    SnglPropSmplEvPrpf ←Query(Entry SnglPropSmplEvPrpf ?x);
    SchOccurSnglPropSE ← Query(ScheduledOccurences SnglPropSmplEvPrpf ?x);
    foreach Entry in the SchOccurSnglPropSE do
        Append SnglPropSmplEvPrpf to tmpEventProfileMatchingTaskList ;
    SnglPropCDEvPrpf ←Query(Entry SnglPropCDEvPrpf ?y);
    SchOccurSnglPropCDE ← Query(ScheduledOccurences SnglPropCDEvPrpf ?z);
    foreach Entry in the SchOccurSnglPropCDE do
        Append SnglPropCDEvPrpf to tmpEventProfileMatchingTaskList ;
    MultPropSmplEvPrpf ←Query(Entry MultPropSmplEvPrpf ?a);
    SchOccurMultPropSE ← Query(ScheduledOccurences MultPropSmplEvPrpf ?b);
    foreach Entry in the SchOccurMultPropSE do
        if notDuplicate then
            Append MultPropSmplEvPrpf to tmpEventProfileMatchingTaskList ;
    MultPropCDEvPrpf ←Query(Entry MultPropCDEvPrpf ?c);
    SchOccurMultPropCDE ← Query(ScheduledOccurences MultPropCDEvPrpf ?a);
    foreach Entry in the SchOccurMultPropCDE do
        if notDuplicate then
            Append MultPropCDEvPrpf to tmpEventProfileMatchingTaskList ;
Load tmpEventProfileMatchingTaskList ;
foreach Entry in the tmpEventProfileMatchingTaskList do
    Term ←Query(Entry);
    Append to EventProfileMatchingTaskList ;
```

also necessary in profile matching and when it will be available for matching. Based on this information, the multi-property task is scheduled in the same manner.

Multiple property Change Detection Event : Multi property change detection is similar to the single property change detection, with a difference in number of properties to consider. For all required properties, it is determined, when earliest second and beyond values will be available and hence, at that time the change detection type event can be called upon. The resulting scheduled tasks are appended in the master list.

At the end, the temporary list is added four types of event profile matching tasks. Yet these tasks are in the memory and hence they will be asserted in the main task list in the knowledge base, so that it can be referred by other procedures. Notations 3.42, 3.43 , 3.44 and 3.45 provide estimate of involved task in event detection using various types of profiles.

$$\sum_{i=0}^{r} P_i \begin{cases} r = & \text{Number of Single Property Simple Event Profile P at Time } (T) \\ P = & \text{Profile} \end{cases} \tag{3.42}$$

$$\sum_{j=0}^{s} P_j \begin{cases} s = & \text{Number of Single Property Change Detection Event Profile at Time } (T + Tcd) \\ P = & \text{Profile} \end{cases} \tag{3.43}$$

$$\sum_{k=0}^{t} P_k \begin{cases} t = & \text{Number of Multiple Property Simple Event Profile P at Time } (Tmax) \\ P = & \text{Profile} \end{cases} \tag{3.44}$$

$$\sum_{l=0}^{u} P_l \begin{cases} u = & \text{Number of Multiple Property Change Detection Event Profile P at } (Tmax + Tcd) \\ P = & \text{Profile} \end{cases} \tag{3.45}$$

Equation 3.46 denotes time taken by profile detection tasks T_{ed}. The scheduled detection of event with complex event profile results in detection of events at corresponding time intervals.

$$T_{ed} = t_{e-ed} - t_{s-ed} \tag{3.46}$$

3.8.8 Action Detection

Active instances of individuals are known in the KB. With help of strategy recommended strategy, representations are created, providing information about state of affairs. The representations are evaluated with event profiles according to the schedule determined with proposed algorithm that enables detection of simple and complex events. The detected events provide determination of state. Complex change detection provides the changing trend of the system status. This also captures the impact of processes or action emergent in the system.

The next task is utilization of rich representations and complex events for determination of next course of action. The purpose of representation generation is to support decisions about determination of action. Instances have assumed various roles that can take the identified actions. Based on the information collection at specified granularity, the next task is to determine the system status. With help of encoded organizational policy for the identified state, the future course of action needs to be determined. Identified action requirements should be conveyed to appropriate instances.

Identification of the prevailing system status, that allows mapping of the current system state with one of all the possible states identified in the entity modeling exercise. In case the system state is determined to be in unfavorable status, identification of the next desired status is missing. Many actions and processes may be required for the transition to next desired state. Identification of actions, the roles involved with the action are missing.

The model is required to characterize the action requirement. This is a complex problem due to cascading events that may have complex ramification. The model is also required to specify the content of an action requirement. It is also required to identify parameters for of the action requirements with reference to instances in KB.

Action requirement generated with this model will result in identification or role specific action requirement request that can be notified to appropriate instances. This will fill the gap in identification of prevailing system step and action requirement. This will provide who will do what information in system and to the individuals. With this, it is possible to efficiently use the generated representations in determining the future course of action in given system.

The evaluation of the state is possible with known entity states captured in entity modeling strategy. Along with this information, transition to the next desired state is also furnished. Identification of activities or processes required for desired transition is also encoded in the entity model. The activity and process details like required resources, roles, and other dependence necessary to carry out activity can be identified based on the outcome of process model.

For given representations, and encoded knowledge regarding the evaluation, the outcome of the action detection model is the procedure that provides identification of action requirements and role specific action requests.

Information management task involves decision-support. The Information deraived by this strategy is used by decision maker role by determining action recommendation to specific actor instances.

Figure 3.31 depicts the elements of the proposed action detection strategy.

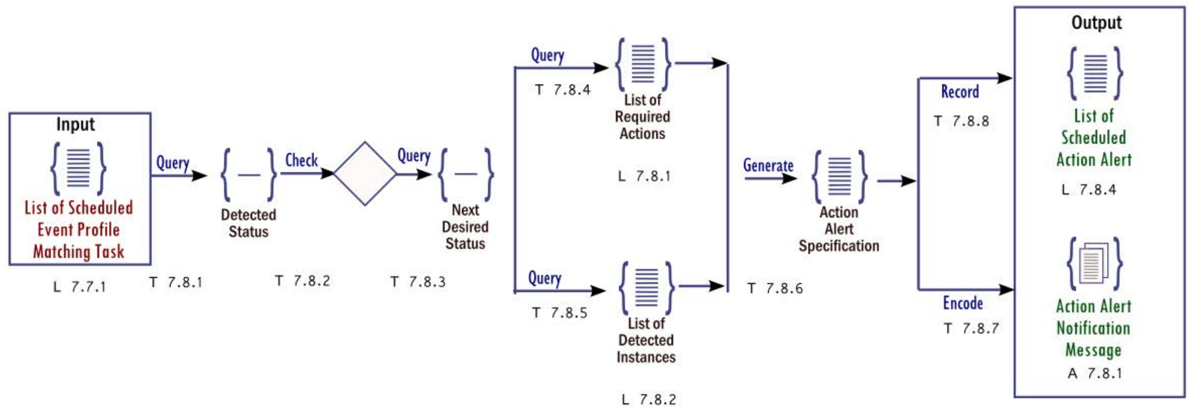

Figure 3.31: Abstract Action Detection

Decision regarding election of present and next state, decision regarding required action, decision regarding related role and corresponding instances poses difficult challenge.

The proposal of the Action Detection Model is in the form of Algorithm 3.8. It takes invent profile matching task list as input argument. It generates action alerts with a corresponding entry in action alert list.

The actions are required only when an undesired status change in detected. If the event has taken place, or next task is known only when profile are matched. Hence action detection strategy task can also be scheduled only after profile matching is completed, and resulting is following some negative pattern. In Line 1 of the algorithm, the Event Profile Matching Task List is loaded. For each individual member it is to be decided if next goal is a new state. This is achieved by a semantic query result is stored in status. The status has action requirements. This action may have dependency. Hence all action to be carried out is identified by another query. For all identified actions, identification or role instance that exist in KB. This is available based on actions identified in process/activity model. For each role instance action message must be identified with space-time and context. This alert message is generated by query the message specification. This is stored in temporary list and recorded in appropriate list. This message is in memory and must be asserted in the KB. All identified action alert notification message are loaded again that must be encoded with application specific schema. They are generated using the encode function and recorded with an action alert notification list.

Notation 3.47 and 3.48 indicates factors affecting identification of required action.

Algorithm 3.8: Action Detection

```
input  : EventProfileMatchingTaskList
output: ActionAlertList

Load EventProfileMatchingTaskList ;
foreach Entry in the EventProfileMatchingTaskList do
    Status ← Query(Entry hasEvaluatedStatus ?x);
    if Status is undesirable then
        ActionList ←Query(Status hasAction ?y);
        RoleInstanceList ←Query(Entry hasRelRoleInstance ?z);
        foreach Entry in the RoleInstanceList do
            ActionAlertSpPara ←Query(Entry ActionAlertMsgPara ?a);
            ActionAlertSp ← Generate(ActionAlertSpPara); ;
            Append ActionAlertSp to tmpActionAlertSpList ;
Load tmpActionAlertSpList ;
foreach Entry in the tmpActionAlertSpList do
    ActionAlertNotMsg ← Encode (ActionAlertSp);
    Record ActionAlertNotMsg to ActionAlertNotMsgList ;
```

$$[\underbrace{\underbrace{(Role_1 + Role_2 + Role_3)}_{A_1} + \underbrace{(Role_1 + Role_2 + Role_3)}_{A_2}}_{State_1} + \underbrace{\underbrace{(Role_1)}_{A_1} + \underbrace{(Role_1)}_{A_2}}_{State_2} \ldots] \tag{3.47}$$

$$\sum_{j=0}^{r}\sum_{k=0}^{s}\sum_{l=0}^{t}\left(State_j A_k Role_l\right) \begin{cases} r = & \text{Number of States } (S) \\ s = & \text{Number of Action } (A) \\ t = & \text{Number of Roles } (Role) \end{cases} \tag{3.48}$$

Equation 3.49 represents the duration of time T_{ad} that is required for the action determination.

$$T_{ad} = t_{e-ad} - t_{s-ad} \tag{3.49}$$

The time instance is identified as t_{noti} when the notification about the requirement of action is issued to the identified role.

$$t_{noti} \tag{3.50}$$

3.8.9 Granularity Calculation

In information processing strategy discussed so far results in generation of information of representations representing the UoD. The recorded values of the property of various concepts in UoD are identified based in information need specification depending upon the concept footprint.

The representations generated are to be utilized by various stakeholders in respective decision-making tasks.

The representations at specified level of detail may be missing. The decision makers have specific impact footprint. They take decisions based on the information at granularity matching with their footprint. It is possible that information collected at record granularity may not be directly useful to all the stakeholders. Hence, representations at appropriate levels may be missing.

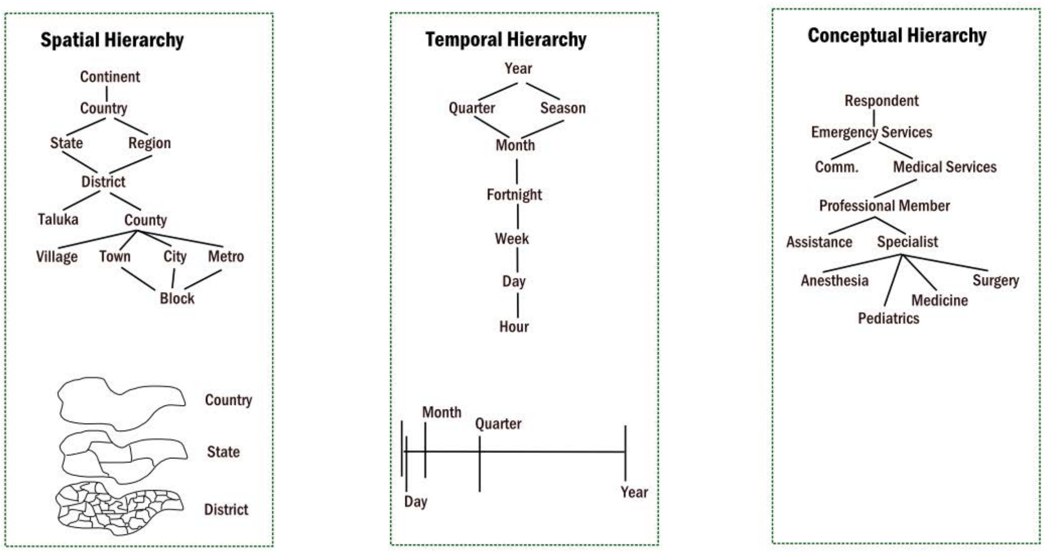

Figure 3.32: Granularity Generation

A model is required to identify the relationship among various granularities. The model is also required to establish methods to facilitate the calculation of various granularities from the available one. The calculations of such derived data should be done continuously to meet the information need of decision makers. This feature is commonly known as calculation of levels in data cube according to the dimensional hierarchies in On Line Analytical Processing domain. The explication of hierarchies guides the process of aggregates calculation at various levels identified in hierarchy[74]. Such hierarchies are commonly identified in spatial and temporal dimensions [59]. As the representations here are based on concepts in KB, and concepts can also have hierarchies, the calculation is also possible in the third dimensions of conceptual representation.

As depicted in Figure 3.32, the KB includes all the possible levels in hierarchies in spatial, temporal and conceptual dimensions. The information need is having specific footprint, may not require consideration of all the levels depicted in the figure. In knowledge base there can be many levels of hierarchies identified in all three dimensions. Calculations may not require at all known levels, but only required on levels at which the instances of decision-makers are identified. Hence, a part of hierarchies relevant for present configuration is required to be explicated. The identification or required hierarchy fills missing gap of required information granularity thereby enable Next task of providing user specific data is benefited with the proposed strategy.

The KB must be holding various hierarchal levels in all three dimensions. The conceptual hierarchy is captured in the form of a general concept to specific concepts that can be derived from it. The spatial hierarchies [69] can be based on the administrative boundaries, starting at the largest area like country or continent to smaller ones like state, region, city, village etc. Temporal hierarchy included identification of levels from years, to months, weeks, days, hours etc. Outcome of this modeling process is a procedure for generating granularities for all observed properties covering space time and conceptual dimensions. The hierarchies are depicted in Figure 3.33 are required hierarchies identified for specific footprint in given UoD. It depicts collection granularity and other required granularity levels to suit the stakeholder needs. This also includes aggregation required at each level. In information management strategy, the

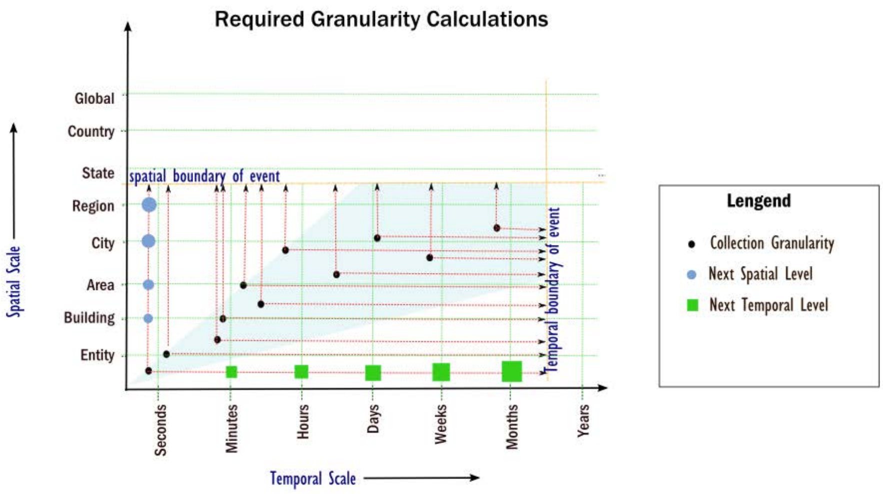

Figure 3.33: Required Granularity Calculation Levels

tasks of business intelligence and reporting, OLAP done by an analyst role. This model is useful for the analyst and multi-dimensional data warehouse engineers in determining levels for roll-up and drill down to facilitate the analysis.

Figure 3.34 represents the basic elements required for the strategy. Input is the information need specification list `(L 7.3.1)` generated using Algorithm 3.3. A query `(T 7.9.1)` is utilized to load the concepts from the list. For each concept the possible hierarchies needs to be determined. A query `(T 7.9.2)` is utilized to derive all the relevant spatial levels for the given concepts. From identified all levels, those following within the estimated spatial footprint are determined with `(T 7.9.3)`. Selected levels are utilized to build spatial hierarchy with required levels. In the same manner, conceptual and temporal levels are identified employing functions `(T 7.9.4)` up to function `(T 7.9.10)`. Function `(T 7.9.11)` is utilized to identify aggregation function appropriate for the given property. Outcome of these functions are utilized by a Generate function `(T 7.9.12)` to build required dimensional hierarchy for the given concept. Decision regarding selection of property, decision regarding building hierarchy, decision

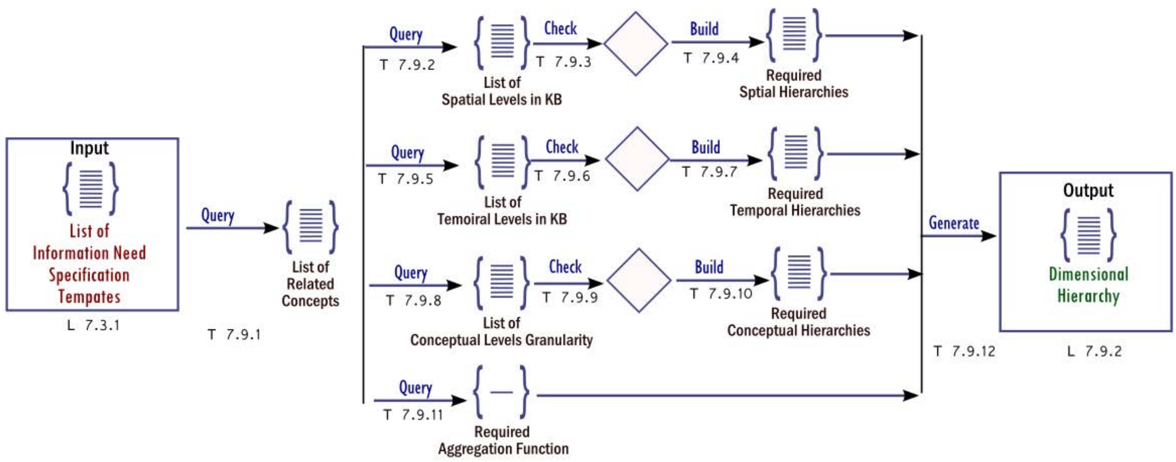

Figure 3.34: Granularity Calculation

regarding encoding and generation of hierarchy is challenging task as it needs to be defined automatically.

The proposal is Granularity Determination strategy given in the form of Algorithm 3.9. The algorithm takes the input from `InfoNeedSpecificationList` as it includes the information about the concept properties that are observed and recorded at the granularity determined suitable by the algorithm. This record level granular information needs to be aggregated at various other levels. Hence the algorithm starts with querying a concept from the list. With the help of a spatial query, a `SpatialLevelArray` is built that covers all the spatial levels identified in the KB. As recognized earlier, all these levels are not required in given situation. The estimated spatial boundary `EstSpaBoundary` is identified for given event, and only those spatial levels falling within this estimated boundary are selected and stored in `SpatialDim`. In this manner, temporal and conceptual levels are also identified.

As representation at various levels can be calculated with the help of aggregation function, the aggregation function appropriate for the selected property is identified as `AggregationMethod` . All the identified levels in three dimensions and the aggregation method stored in temporary list `tempHierarchy` are utilized to generate a `Hierarchy` with the help of `BuildHierarchy` function.

Algorithm 3.9: Granularity Determination

```
input  : InfoNeedSpecificationList
output: Hierarchy

Load InfoNeedSpecificationList ;
foreach Entry in the InfoNeedSpecificationList do
    SpatialLevelArray ← Query(Entry hasNextSpatialLevel ?x);
    EstSpaBoundary ←Query(Entry hasEstSpatialFootPrint ?a);
    foreach Entry in the SpatialLevelArray do
        SpatialDim ←Query(Entry hasInstance ?a);
        if withIn EstSpaBoundary then
            add SpatialDim to tempHierarchy ;
    TemporalLevelArray ← Query(Entry hasNextTemporalLevel ?x);
    EstTmpBoundary ←Query(Entry hasEstTemporalFootPrint ?y);
    foreach Entry in the TemporalLevelArray do
        TemporalDim ←Query(Entry hasInstance ?z);
        if withIn EstTmpBoundary then
            add TemporalDim to tempHierarchy ;
    ConceptLevelArray ← Query(Entry hasNextConceptualLevel ?a);
    EstConBoundary ←Query(Entry hasEstConceptFootPrint ?b);
    foreach Entry in the ConceptLevelArray do
        ConceptualDim ←Query(Entry hasInstance ?c);
        if withIn EstConBoundary then
            add ConceptualDim to tempHierarchy ;
    AggregationMethod ← Query(Entry hasAggregationMethod ?d);
    add AggregationMethod to tempHierarchy ;
Load tempHierarchy ;
foreach Entry in the tempHierarchy do
    Term ←BuildHierarchy(Entry);
    Append Term to Hierarchy ;
```

Notation 3.51 provides factors effecting the identification of various granularities for given concept property.

$$\sum_{i=0}^{r} P_i \Big(\sum_{j=0}^{s} C_j \sum_{k=0}^{t} S_k \sum_{l=0}^{u} T_l + A_i \Big) \begin{cases} r = & \text{Number of Concept Properties (P)} \\ s = & \text{Number of Levels in Conceptual Hierarchy(C)} \\ t = & \text{Number of Levels in Spatial Hierarchy(S)} \\ u = & \text{Number of Levels in Temporal Hierarchy(T)} \\ A = & \text{Identified Aggregation Function (A)} \end{cases} \quad (3.51)$$

Equation 3.52 depict the time required in calculating the granularity hierarchy as $T_{Assertion}$.

$$T_{Assertion} = t_{rep\_fin} - t_{rep\_st} \quad (3.52)$$

3.8.10 Extract Transform and Load Specification Generation

Within information processing model so far, datasets are collected at record granularity are available for further processing. This is at the bottom of the identified granularity hierarchy. The upper levels are defined in three dimensions as discussed in granularity calculation strategy in Section 3.8.9.

The next step is to generate derived data for recorded one at identified hierarchy. To specify the aggregation function and create dataset to furnish the created data to the users where they want.

In achieving the next step, critical information about the processing is missing. The identification of granularity calculations and delivery is not known. Hierarchies are identified and stored data is continuously collected. A strategy to identify scope of task in the form of extract transform and load requirements is needed. Model is required to characterize ETL Requirements for situation awareness application. It is necessary to specify the content Schema of an ETL Task. Also a Strategy is required to automatically determine values for parameters given in template. Once ETL parameters are identified, a strategy is also required to encode the resulting task specification. Also the strategy should also define the scheduling strategy of the identified tasks in Situation Awareness system environment.

This specification fills the gap of required information processing task to furnish granularity thereby achieve the next goal of providing user specific data set at the user location.

Instances must be known playing various roles with parameters their determined granularity from them. Outcome of granularity calculation is required as it restrict the required granularity to hierarchy calculated for present situation. Information need specification is required to know record granularity and time intervals at which recorded data will be made available.

Also, generated representations are required as is allows to tell determine if data is ready for processing.

Outcome of the model is a procedure that allows determination of ETL Tasks. This includes specification of source, target, aggregation required time for execution etc.

Information management has discipline dealing with such ETL. The role is ETL designer Role. For information management in complex dynamical system, role has to consistently carry out task of dynamic sources and targets the procedure allows generations of automatically and event driven.

Selection of sources of information about ETL requirement, Query comp for source identification target granularity also determine time procedure is required encode identification generate report.

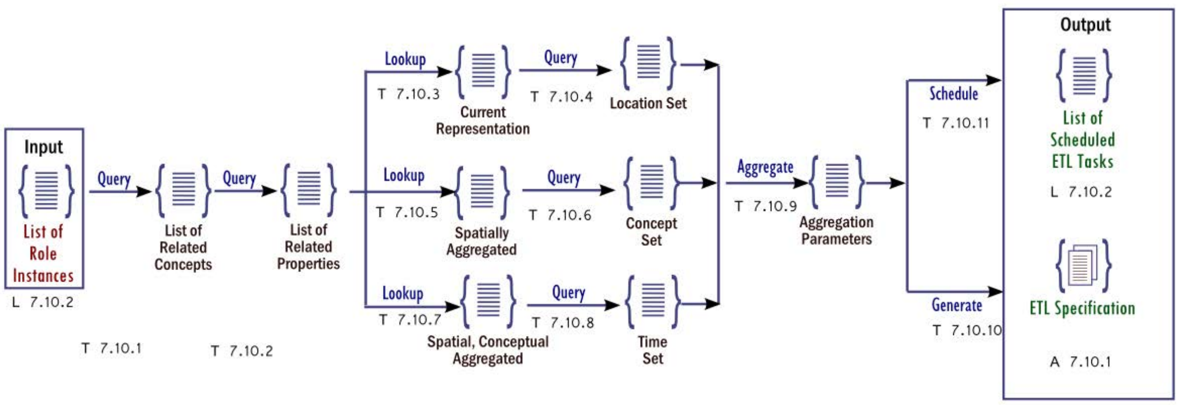

Figure 3.35: ETL Specification Generation

Decision regarding the source of ETL, the time it can be triggered, the selection of target user with there granularity are to be taken by the strategy. Query regarding aggregation, especially type change to suit the need of identified instance is also important in deciding the time interval.

Actor instances with different granularities are generally introduced in the system. It is likely that when information collection is carried out it is done at most granular level. Then it is possible and required to calculate other granularities as seen in OLAP (on line analytical processing) Domain. For given system the instances are now known and their granularity is determined. Now task is to determine which instance will required what data at which granularity and more importantly where. The strategy has to identify source, the operations, the target, and job specification details. This algorithm determines specification of this task.

Algorithm 3.10: ETL SPECIFICATION GENERATION

```
input  : DMRoleInstanceList
output: ETLSpecification and entry in ETLSpecificationList
Load DMRoleInstanceList ;
foreach Entry in the DMRoleInstanceList do
    ConceptList ← Query(Entry RelatedProperty ?a);
    foreach Entry in the ConceptList do
        rqConLev ← QryHrchy( Property ConceptLevel ?b);
        rqSpLev ← QryHrchy( Property SpatialLevel ?c);
        rqTmpLev ← QryHrchy( Property TemporalLevel ?d);
        AvRep ←QueryRepresentation(Property, rqSpLev,rqConLev,rqTmpLev);
        if AvRep =NULL then
            /* ********* Parameter Determination *********                */
            currRepresentation ←QueryRepresentation (Property);
            currSpLev ← QryHrchy(currRepresentation hasSpatialFootprint ?e);
            currConLev ← QryHrchy(currRepresentation hasConceptFootprint ?f);
            currTmpLev ← QryHrchy(currRepresentation hasTemporalFootprint ?g);
            AggregateFn ← QryHrchy( Property AggregateFuntion ?h);
            /* ** Specification and Scheduling of Spatial Aggregate Generation **   */
            Lookup(currRepresentation,rqSpLev);
            LocationSet ← QryHrchy(currSpLev, rqSpLev);
            AggrPara ← Aggregate( currRepresentation, AggregateFn, LocationSet);
            SpAggRepTaskSpec ←
            GenRepTaskSpec(rqSpLev,currConLev,currTmpLev,AggrPara);
            ScheduleRepTask(SpAggRepTaskSpec);
            /* ** Specification and Scheduling of Concept Aggregate Generation **  */
            Lookup(SpAggRep);
            ConceptSet ← QryHrchy(currConLev, rqConLev);
            AggrPara ← Aggregate( SpAggRep, AggregateFn, ConceptSet);
            ConAggRepTaskSpec ← GenRepTaskSpec(rqSpLev,rqConLev,currTmpLev,
            AggrPara);
            ScheduleRepTask(ConAggRepTaskSpec);
            /* ** Specification and Scheduling of Temporal Aggregate Generation **
               */
            Lookup(SpConAggRep);
            TimeSet ← QryHrchy(currTmpLev, rqTmpLev);
            AggrPara ← Aggregate( SpConAggRep, AggregateFn, TimeSet);
            TmpAggRepTaskSpec ← GenRepTaskSpec(rqSpLev,rqConLev,rqTmpLev,
            AggrPara);
            ScheduleRepTask(TmpAggRepTaskSpec);
```

Information processing model ETL Specification Generation is provided as Algorithm 3.10. The model is provided as a strategy to generate ETL Task specification.

For the automated identification of ETL in the configuration, the proposed strategy recommends the available role instance as input. This is recommended because; ETL is required to meet the information need of a specific stakeholder instance available in the present configuration. This restricts the information processing load by only processing the required information as opposed to predefined batch jobs. As discussed earlier, the KB can be search for all stakeholders that requires access to data representations at specific granularities in order to take decisions. These instances are identified as `DMRoleInstanceList`.

For each entry in the list some preparatory steps are required. First for each entry all the decision parameters should be identified. A given instance of role may require representation of various observable properties of UoD. This is identified by querying `RelatedProperty` for all instances. Each property may require unique ETL tasks to be performed. Hence each identified property is handled separately.

Next step requires a strategy to determine the parameters for the task.

Depending upon the unique impact footprint of the assumed role, each individual has specific granularity requirement for the information. This granularity level is identified using `rqSpLev`, `rqConLev` and `rqTmpLev` property for spatial, conceptual and temporal dimension respectively. As described earlier, this may be same level at which they were collected, or they can be less granular. To identify if the same is already calculated, the representation is searched in the database. If a representation satisfying all three granularity criteria is already available, the task needs not be scheduled. Hence, this check prevents the duplication in processing effort.

If required granularity is not available, the representations created at fine record granularity are identified. The representation is treated at `currRepresentation`, and used to retrieve the required levels. Using the available representation, the current spatial granularity is extracted as `currSpLev`. Conceptual and temporal granularity levels are extracted as `currConLev` and `currTmpLev`. Every property has specific aggregation function as captured using Measurement Model. This aggregation model is identified as `AggregateFn`, and will be utilized during all the calculations.

Once the parameters are defined, the aggregations in various dimensions are required. The aggregation can be done in specific order of sequence. The first aggregation needs to be carried out at spatial followed by conceptual dimension. Temporal needs to be calculated last after calculations of the others.

Effort required for aggregation in spatial dimension needs to be identified and scheduled as a task.

Lookup(currRepresentation,rqSpLev) Function is utilized to identify the spatial level. The identified available and required granularity is utilized by a query QryHrchy(currSpLev,rqSpLev) to determine the LocationSet. All the members of the location set at fine spatial granularity are collectively identified a larger unit at coarse spatial granularity. Hence, in order to represent the property attribute at this coarse spatial granularity, the values held by its individual members should be aggregated. Hence AggrPara is derived using Aggregate function that handles current representation, aggregation function and the location set. This AggrPara contains all the necessary information required to carryout spatial aggregation. The resulting new aggregated representation contains the values with spatial interval at required granularity rqSpLev and conceptual and temporal interval at the current levels of granularity specified as currConLev and currTmpLev. This parameters are now ready to be specified as a task and the task specification is generated with GenRepTaskSpec function to create a spatial aggregation task. The task identified as SpAggRepTaskSpec is stored as scheduled task list that needs to be further handled by execution service. Upon execution of this tasks SpAggRep are derived as intermediate representation, aggregated only in spatial dimension.

Next task is the aggregation of all the generatedSpAggRep in conceptual dimension, which can be accomplished in similar manner. Similar to spatial dimension, the desired interval in conceptual dimension is identified as rqConLev and appropriate instance of conceptual elements are identified and stored in a ConceptSet that are to be aggregated to a course level concept. This allows identification of task parameter required to generate data with required spatial granularity rqSpLev and required conceptual granularity rqConLev but at current temporal level currTmpLev. The representation resulting from these scheduled tasks are identified as SpConAggRep.

For all the generated SpConAggRep, the aggregation in temporal dimension needs to be identified and scheduled as a task in similar manner described for spatial and conceptual dimensions. All tasks specifications are stored as ScheduleRepTask.

To understand the functioning of the proposed information processing model, a disaster management scenario can be utilized. In a scenario, emergency medical camps are setup in affected areas. Various aspect of response related information is continuously collected at each camp. The state, national and international levels, the decision makers, volunteers and other stakeholders can utilize the same information but not at collected granularity, but the coarser level detail determined suitable to their footprints.

Considering granularities in spatial, temporal and conceptual dimensions becomes necessary.

In spatial dimensions various levels can be defied as camp level, city level, state level, country level etc. In temporal dimension the levels are minutes, hours, days, weeks, months, and years. In conceptual dimension, describing types of injury to the aggregate concept of victim of the event. These levels and aggregation method are captured according to granularity calculation strategy discussed in Section 3.8.9. Figure 3.36 represents the granularity levels. The source column in the left block contains representation with finer granularity levels including Camp, injured, homeless, minutes etc. Next desired level is depicted for the users at area, victim, hrs in spatial, conceptual and temporal dimensions respectively.

As described in Algorithm 3.10, the first task is the calculation of spatial aggregates. The `LocationSet` is identified as Camp1, Camp2, Camp3... as all these camp are contained in an Area. The data is aggregated with Sum function as number of victims can be added as they are included in the collection. The resulting aggregation is the representation only at desired spatial granularity, as conceptual and temporal is still the same. Next task is collection of conceptual granularity. Injured, Homeless and others are identified as a coarser level attribute in spatial domain with the name Victim in the hierarchy. Aggregation is carried out with summation of values in each member of `ConceptSet`. Similarly aggregation in time dimension is carried out. The identified role instances are defined in Target block with specific granularity requirement. The calculated aggregates are loaded at their respective instances. This explains the Extract, Transformation and load of data. The specification is calculated automatically based on the available instances and hierarchies as captured in KB.

Notations 3.53 and 3.54 identifies factors effecting the total number of ETL specifications required. It is evident that this is a coarse level of estimate, and many intermediate representations are also generated depending upon the instantaneously identified needs.

$$[\underbrace{\underbrace{(Rep_1 + Rep_2 + Rep_3)}_{P_1} + \underbrace{(Rep_1 + Rep_2 + Rep_3)}_{P_2}}_{Role_1} + \underbrace{\underbrace{(Rep_1)}_{P_1} + \underbrace{(Rep_1)}_{P_2}}_{Role_2} \ldots] \tag{3.53}$$

$$\sum_{j=0}^{r}\sum_{k=0}^{s}\sum_{l=0}^{t}\Big(Role_j P_k Rep_l\Big) \begin{cases} r = & \text{Number of Role with Unique Impact Footprint} \\ s = & \text{Number of relevant property} \\ t = & \text{Number of Unique Representations} \end{cases} \tag{3.54}$$

3.8.11 Job Sepcification

Various task specifications are identified that results in creation of templates for appropriate information processing tasks to be carried out at given time interval. Table 3.1 represents

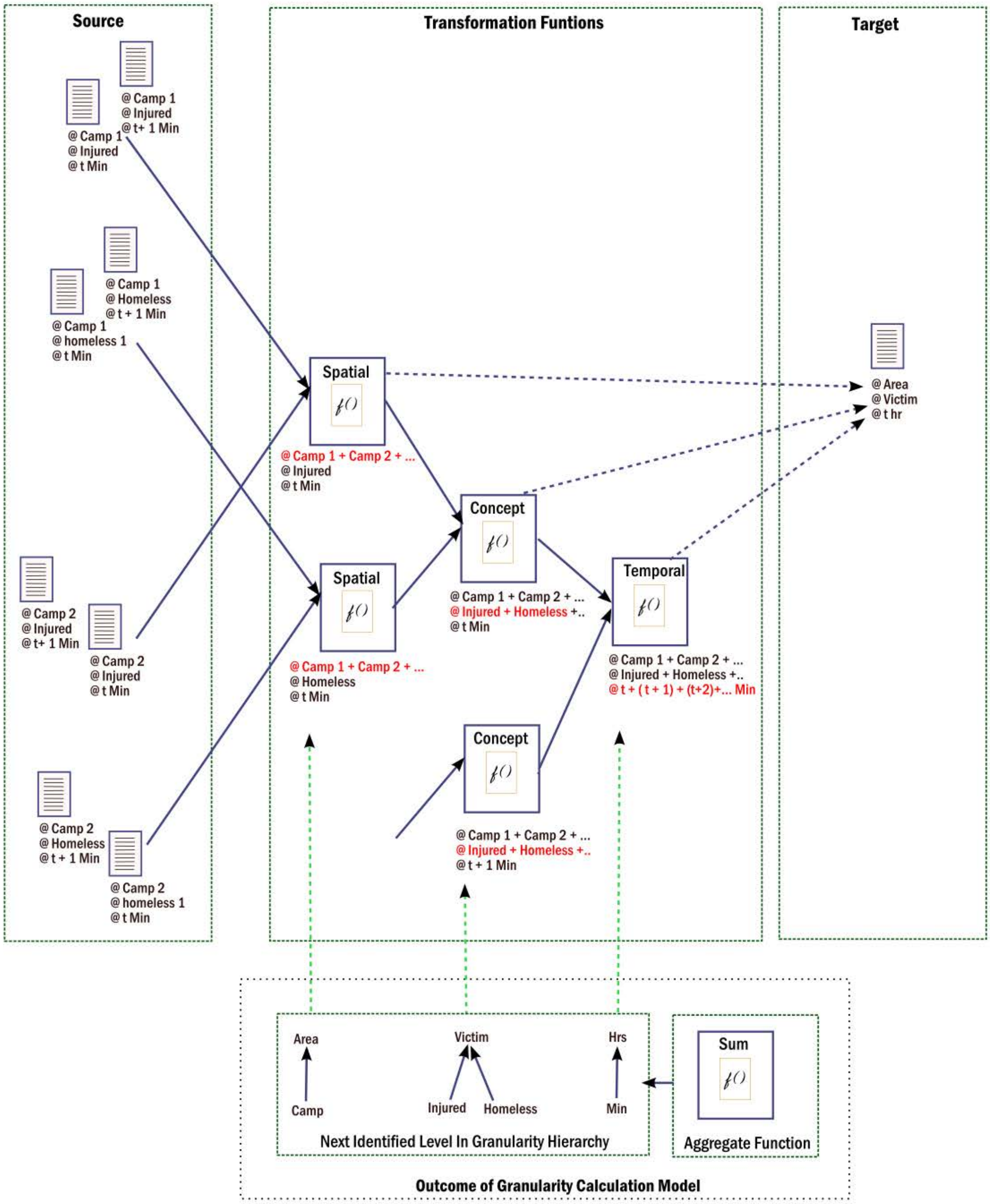

Figure 3.36: ETL Processing Requirement Example

various tasks specifications that are generated with recommended implementation strategy.

Information Processing Model	Algorithm	Task
Assignment of Communication Roles	3.5	Communication Task Specification
Event Detection	3.7	Event Detection specification
ETL Generation	3.10	ETL Specification

Table 3.1: Automated Task Specifications Generated by Implementation Strategy

These tasks must be efficiently carried out on IT infrastructure. The task should be converted to job they can be scheduled, executed, monitored and managed according to IT services.

Identification of appropriate strategy that will provide execution parameters is not addressed by the strategy. The task specification in their current state poses abstract requirements only. The underlying computation framework requires exact information like resource requirements, execution parameters and other execution specific information.

Model is required to characterize a job specification suitable to carry out or execution job. Job specification generated will fulfill the gap of availability of execution parameters this will enable automated management of job, and facilitate identification, specification, submission and monitoring of job.

Prerequisite to job specification strategy is the identification to scheduled tasks generated in the configuration. The task lists are prerequisite as it contains information about task scheduled with required action. Knowledge about application specific parameters is also required that can be encoded in application specific concepts in KB.

The proposal provides a procedure for creation of job specification. The specification is based on application specific execution parameters as identified in KB and hence can directly used for submission.

Information management domain has task of processing information in execution management role is dedicated to use IT infrastructure for optimal use. The automated specification helps appropriate roles responsible for creating job specification in dynamic environment.

Figure 3.37 depicts the elements of the proposed job specification generation strategy.

Decision for selecting candidate tasks as job is a difficult task. Decision to determine tasks are there that can be scheduled and executed. Strategy for determination of list of execution parameters, selection of value for each individual task is also difficult challenge. Strategy to encode the job specification is also critical as it can be taken by execution parameters.

Many functions have been discussed in algorithms that required individual handling. This should be treated as jobs. For example, information specification generation, messaging pattern generation, representation generation and ETL generation as well as execution requires

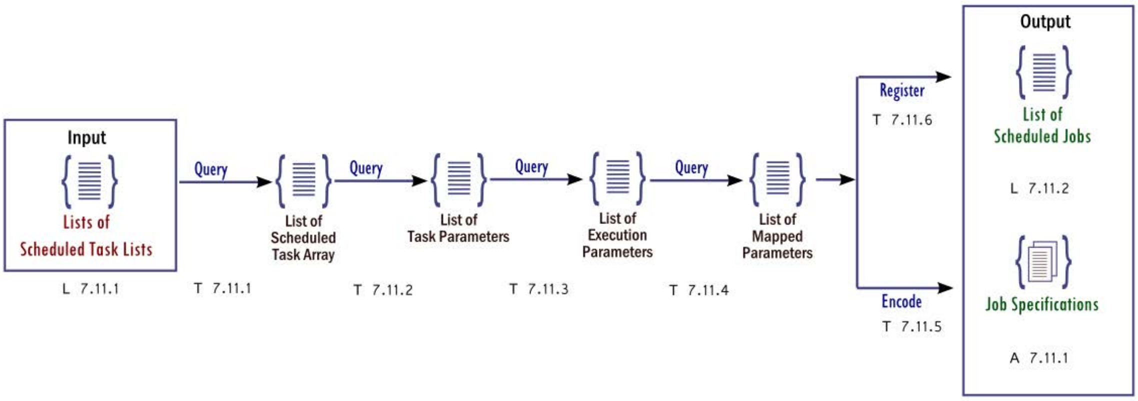

Figure 3.37: Information Processing Model: Job Specification Generation

Algorithm 3.11: Job Specification Generation

input : Scheduled Tasks
output: Job Specification List

```
ScheduledTaskListArray ← Query(ScheduledTaskList isKindOf ?x);
foreach Entry in the ScheduledTaskListArray do
    ScheduledTaskArray ←Query(ScheduledTask has ?y);
    foreach Entry in the ScheduledTaskArray do
        TaskParametersList ←Query(Scheduled TaskPara ?z);
        foreach Entry in the TaskParametersList do
            ExecutionPara ←Query(Parameter ExecutionPara ?a);
            Append ExecutionPara to MappedParameter ;
        Record tempTaskParametersList ;
Load tempTaskParametersList ;
foreach Entry in the tempTaskParametersList do
    JobSepcification ←Encode(Entry);
    Save JobSepcification ;
    Register JobSepcification to JobSepcificationList ;
```

submission of multiple jobs in batch mode. The issue is how job is determined, specified and executed.

Important step is to find out at how many places function call amounts to be a job request. It is done by accessing few lists maintained by the system that enlists the required task to be completed. The task details are available in specification. Middleware libraries supporting the task are also known. What is not known is how to prepare a job specification and submit to a job manager or handler. A scheduled task list for available execution environment is the desired goal.

Any type of specification generated by proposed implementation algorithms that specifies scheduled task is qualified for this. Hence query is to find out all scheduled task specifications. Each specification contains when the task is to be done, what it needs to be done and who will monitor it. These specifications are registered on a list. Hence a procedure is required that will first determine the all specifications that creates scheduled tasks. Then it queries the lists. The lists are loaded in the memory and then one by one job specification is created.

Once a scheduled task is loaded in the memory, it extracts the features like scheduled time, the function, the source, and the target and other input and output requirements. This is done in semantic domain. Now task is to create a job specification that will contain the execution parameters. Hence the knowledge base is again queried that will carryout mapping. The mapping includes the command line tools, arguments, and program specific instructions. Once this is determined, the new execution parameters stored in the memory is then encoded as per the rules in knowledge base. This encoded specification is then registered in the job specification list as well.

Notations 3.55 and 3.56 introduces factors affecting identification of job specifications.

$$[\underbrace{\overbrace{(Para_1 + Para_2)}^{Jsp_1}}_{Stsk_1} + \underbrace{\overbrace{(Para_1 + Para_2)}^{Jsp_1}}_{Stsk_2} \ldots] \tag{3.55}$$

$$\sum_{i=0}^{r} S_i J_i \Big(\sum_{j=0}^{s} Para_j \Big) \begin{cases} r = & \text{Number of Scheduled Data Management Tasks}(Stsk) \\ s = & \text{Number of Job Execution Parameter set}(Para) \\ J = & \text{Number of Job Specification } (Jsp) \end{cases} \tag{3.56}$$

Equation 3.57 provide duration of time $T_{Assertion}$ required to calculate the job specification

$$T_{Assertion} = t_{rep\_fin} - t_{rep\_st} \tag{3.57}$$

3.8.12 Data Provenance Generation

With strategy defined in Algorithm 3.6 Data sets are generated, depicting the observed outcome at predetermined record granularity. With strategy defined in Algorithm 3.10 derived data sets are created from the source at record granularity and furnished to targets.

Distributed nature of the application requires management of the generated data sets to meet various data management strategies [75]. For example, the vast amount of generate data must be managed for archiving, backup and future reuse purpose. This includes the scenario, in which datasets are exposed in catalog service. They are subject to database administration activity.

Numerous data sets generated automatically in each configuration. There can be many systems configured, exposing vast amount of datasets. Identification of a single dataset in unique manner is missing. Identification of characteristics that expose data and identify data for its uniqueness is necessary yet undefined.

Next requirement in information management strategy is therefore to characterize meta data appropriate for situation awareness. The model is required to specify the content for each instance, to provide strategy for collection. A model is required to characterize meta data suitable for situation awareness configuration. The model is expected to specify content of each instance of meta data, capable to sufficiently express the Meta data including, purpose, time, mode and other details of creation, processing and other management task it is subjected to time to time. The model is also expected to provide a strategy that enable collection, record and exposing of meta data for the use in situation awareness configurations.

A data provenance strategy will generate provenance log of the data this will fulfill the gap of meta data, thereby enable users to later discover, identify and handle data accordingly. Knowledge about all process that deals with creation or handling data is prerequisite for this purpose. Outcome of ETL specification provide necessary information about sources, targets and aggregation is carried out. Schedule of tasks are also indicator of time when data was subjected to management task.

The outcome of the proposed strategy is a procedure that will enable recording of the meta data for all scheduled tasks dealing with data records.

In information management, data oriented tasks deal with data administration and management. The role of data administrator is to deal with security, management and other aspect of recorded data. The proposed model will enable the data administrator for situation awareness system to specify parameters for automated data provenance generation.

Figure 3.38 depicts the elements of proposed data provenance generation strategy. List of

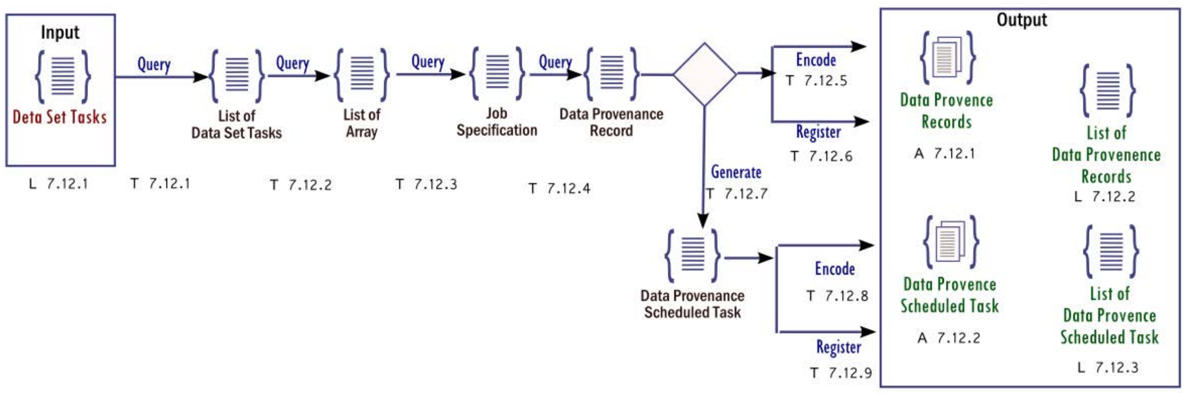

Figure 3.38: Data Provenance Generation Algorithm

data set tasks are taken as input to this strategy. Number of semantic queries is required to determine types of data set tasks and various parameters. Based on the parameters of tasks, the data provenance records `A 7.12.1` are created and updated. Also, data provenance tasks`A 7.12.1` are scheduled to update the data provenance record with the entry of recently completed data management task.

Decision regarding all sources of information that provides data provenance information is a difficult challenge in realization of this strategy. Decision is also critical to determine when data is subjected to change so that the sources of information can be referred for extraction of the data. Strategy is required in encoding the identified information, Decision regarding storage and automated management of data when large amount to data movement is carried out.

The proposal is the Data Provenance Generation strategy in the form of Algorithm 3.12. It is argued that data provenance is required to be generated and updated at any alterations created in a data set. There can be many tasks that may alter existing datasets or generate derived data sets. All these activities must be captured in data provenance records. Hence, instead of manually maintaining the data provenance records, it is possible to integrate data provenance task with all tasks scheduled to alter any dataset. For this, the algorithm employs a query to identity all such jobs and register it as `DataSetTaskJob`. For this job, the next query is employed to identify job specification instances, that are stored in `ListArray`. For each instance in `ListArray`, job specification parameters are identified and stored as `JobSpecification`. For given dataset, the provenance record is looked-up. In absence, data provenance record `DataProRecord` is created and registered. If the data provenance already available, the `DataProSchedTask` is scheduled. Each scheduled data provenance tasks is registered in `DataProSchedTaskList`

Notations 3.58 and 3.59 depicts factors affecting task of data provenance.

Algorithm 3.12: DATA PROVENANCE GENERATION

input : List of Scheduled Job Specifications
output: DataProSchedTask and records in DataProSchedTaskList

```
DataSetTaskJob ← Query(ScheduledJob isDataSetJobType ?x);
ListArray ← Query(DataSetTaskJob ScheduledJobList ?y);
Load List from ListArray ;
foreach Entry in the ListArray do
    JobSpecification ←Extract(Entry);
    DataProRecord ← Query(Entry hasProvenanceRecord ?z);
    if DataProRecord = NULL then
        Encode(DataProRecord);
    DataProSchedTask ←Generate(Entry,JobSpecification,DataProRecord);
    Append DataProSchedTask to tmpDataProSchedTaskList ;
Load tmpDataProSchedTaskList ;
foreach Entry in the tmpDataProSchedTaskList do
    DataProSchedTask ←Encode(Entry);
    Register DataProSchedTask to DataProSchedTaskList ;
```

$$[\underbrace{\overbrace{(Para_1 + Para_2)}^{Prec_1}}_{Stsk_1} + \underbrace{\overbrace{(Para_1 + Para_2)}^{Prec_1}}_{Stsk_2} \ldots] \tag{3.58}$$

$$\sum_{i=0}^{r} S_i P_i \Big(\sum_{j=0}^{s} Para_j \Big) \begin{cases} r = & \text{Number of Scheduled Data Management Tasks}(Stsk) \\ s = & \text{Number of Provenance Parameter set}(Para) \\ P = & \text{Number of Provence Entry } (Prec) \end{cases} \tag{3.59}$$

3.8.13 Distributed Data Management Task Specification

Proposed information processing models in the form of algorithms provided strategy for the generation of representations of UoD depicting some relevant aspects. These representations are realized in physical data sets. The same can be processed and delivered as per the requirements to various users. For this requirement, ETL processes are defined to generate derived data set for different granularities. Automated data provenance is also defined. Overall, the information collection and handling strategy with automated is created.

Various Distributed Data Management (DDM) tasks that are required in VO environment needs to be carried out. Stefano [75] discussed various tasks and provided abstract representations of task specifications. These tasks include archival, purging, synchronization, regionalization and related tasks required for data safety and availability in collaborative environment.

Hence, for the IM Strategy, next challenge is the potential scenario, that requires various data management tasks that are not scheduled, or not written in the database. A typical data management scenario include, requirement for the merger of two configurations, or partition in to smaller configurations. The configuration must be subjected to regular backup, synchronization or purging to suit the requirements. Figure 3.39 represent a data regionalization scenario in a collaborative environment.

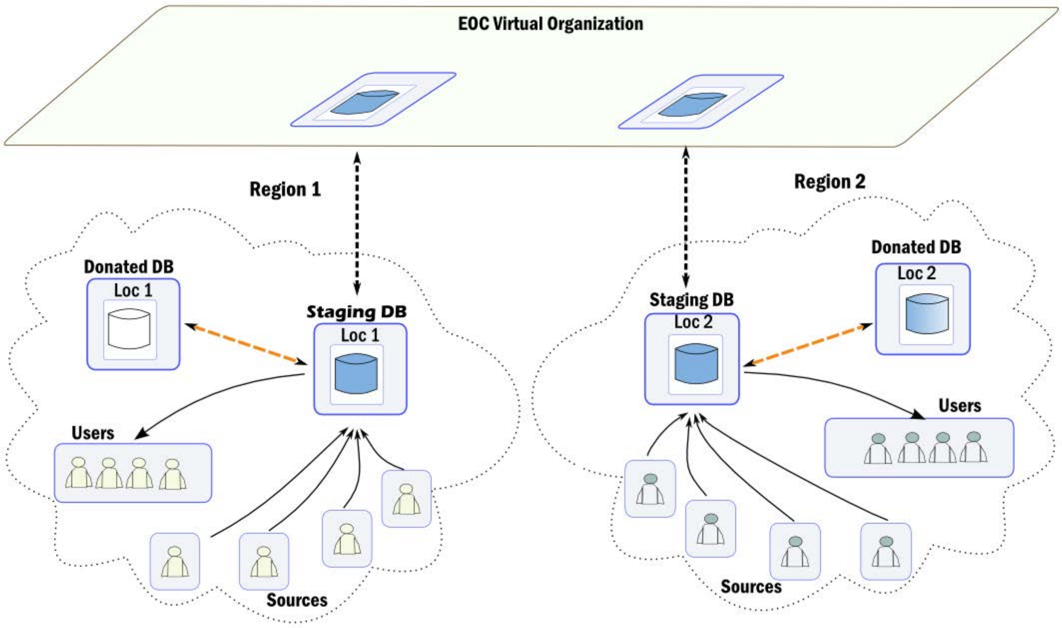

Figure 3.39: Data Regionalization Job

These data management tasks should be carefully identified with relevant parameters. Stefano [75] introduced various data management scenarios that may arise in grid environment and discussed various data management policies that are required to determine the tasks. These policy statements include important parameters that are identified and specified in order to issue a data management task. Table 3.2 enumerate some data management policies with relevant parameters identified as a part of the task specification. As evident from the parameters,

Data Management Task	**Parameters**
Data Distribution Policy	Region, Scope, Pattern
Data Replication Policy	Region, Quantity, Scope
Data Load Policy	Region, Granularity, Adapter
Data Store Policy	Region, Granularity, Operation, Adapter
Event Notification Policy	Region, Scope, Operation

Table 3.2: Distributed Data Management Policy Parameters by Stefano [75]

the nature of required information is specific to situation awareness system configuration in given UoD. All these tasks may not be completely automated, and hence, they require user intervention to interactively build the job parameters. Also, these tasks specification may not be created manually, as user may not have access to instantaneous information about automatically generated representations. As a solution, the strategy includes taking the preference from the user that governs the identification of data management task. The parameters like scope, selection, quantity and operations can be identified with the help of the system.

Model is required for characterization of Distributed Data Management Task specification. Model is required first to characterize the DDM job type specification of various parameters are required strategy for interaction for collecting the values are also required. Lastly decision regarding the encoding is also required.

Job specification generated in this manner fills the gap of missing strategy to define DDM task and thereby allows various distributed data management tasks to suit the instantaneous needs. The prerequisites for this strategy includes the availability of information regarding physical instances, generated representations and data provenance that provides vital input to the task specification process. Outcome of this modeling process is a procedure for interactive building of a job specification providing various job parameters suitable for selected data management task. Information management process has specific role for database management tasks. They are responsibility of database administrator. As it also include instance base, knowledge engineer role is also relevant to some of these tasks. The proposed strategy enables these roles by providing interactive building of task specification that can be difficult without knowledge of instantaneous data sources.

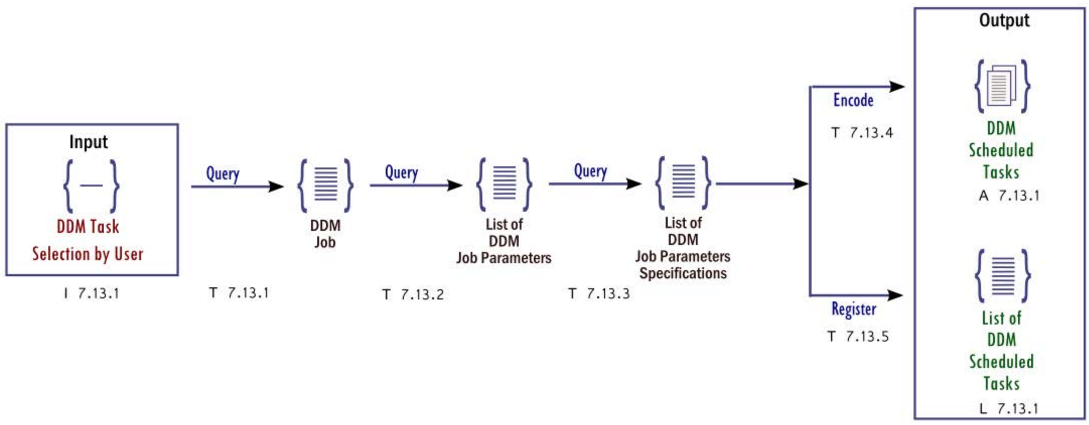

Figure 3.40: Distributed Data Management Task Specification Generation

The basic elements of the process include selection of data management task types, identification of parameters, encoding of tasks parameters as Distributed Data Management Task. Figure 3.40 depicts the elements of information processing model proposed for distributed data management task identification. This is no separate input requirement for this strategy as the parameters should be interactively collected by the user. This is done identification of the task selected by the user as `(I 7.13.1)`. To determine the nature of task Query `(T 7.13.1)` is utilized. For identified task type, the required task parameters are identified using Query `(T 7.13.2)`. For each of the required parameter, the value is determined with the Query `(T 7.13.3)`. The resulting job specification is appended to a temporary job parameter list. The encoding the DDM Task is done with `(T 7.13.4)`. The encoded task specification is registered in a list `(T 7.13.5)`. Output is the Distributed Data Management Scheduled Task List `(L 7.13.1)` and corresponding Distributed Data Management Scheduled Task `(A 7.13.1)`

These abstract requirements are difficult to achieve in complex dynamical environment. The configuration specific values for required job parameters identified with Query `(T 7.13.3)` is a considerably challenging task unless appropriate provisions are made in advance. The task parameter also appropriately determines the scheduled execution of the task. The scheduling parameters should be carefully identified consider the generation of representations as well as the needs of the potential users.

Algorithm 3.13: DISTRIBUTED DATA MANAGEMENT TASK

input : Specified Distributed Data Management Task
output: DDMSchedTask and records in DDMSchedTaskList

```
Accept the task selection from User;
DDMJob ← Query(TaskSelection DDMJobType ?x);
DDMJobParaList ← Query(DDMJob hasParameters ?x);
foreach Entry in the DDMJobParaList do
    DDMJobParaSp ←Query(Entry hasInstance ?x);
    Append DDMJobParaSp to tmpDDMJobParaSpList ;
Load tmpDDMJobParaSpList ;
foreach Entry in the tmpDDMJobParaSpList do
    DDMSchedTask ←Encode(Entry);
    Register DDMSchedTask to DDMSchedTaskList ;
```

The proposal is provided as DISTRIBUTED DATA MANAGEMENT TASK in the form of Algorithm 3.13. With availability of data provenance, ETL tasks and other features about representation generation, it is possible to identify the task specific parameters. A query is utilized to access required values from data provenance records for the involved data sets. Similarly the query is employed to identify other features like footprints, users, their physical location and other parameters as required by the nature of data management task.

With identification of individual values of task parameters, the job can be treated same as ETL task. Similar to ETL specification strategy the task parameter includes the scheduled execution of the task. With scheduling of the task, the update in data provenance record is automatically scheduled as discussed earlier.

3.8.14 Temporal Profile of the Proposed Strategy

The information processing models discussed in his section addresses the processing aspects of information management strategy. While other models allow identification and characterization of system elements relevant to the goals, the information processing model allows handling of instance of such elements. In this regard, the presented information processing models guides the processing on the runtime.

The present discussion identified thirteen issues and presented solutions in the form of algorithms. These information-processing issues are generally faced in the practical situations. Also, the proposed modeling strategy also requires implementation strategy. Hence, the proposed model provide serve both these purpose. Identified information processing models are not the only models that will be sufficient to solve the complete situation awareness problems. There can be many application or scenario specific requirements that must be individually addressed.

The presented are identified as solution to major roadblocks in the modeling process. They are individually treated because they span multiple middleware disciplines. Some functions deal with semantic aspects; some are in spatial or temporal aspects, computation as well as some other domain specific aspect. As the proposed strategy is spanning these underlying domains, the present lists are the primary list of problem that must be defined and solved.

Figure 3.41 represent the sequence and relation of the proposed information processing models. The information processes starts with the initial report from the user from UoD. As assertion is handled by the system to identify the information needs and generate information need specification templates. For the specified information needs, the messaging patterns are identified and appropriate messaging roles are assigned in system as well as the roles available in UoD. For collection of the information, the required communication with roles in UoD is carried out. This two-way communication allows realization of various interaction patterns. Information exchanged over these patterns is extracted to create rich representation of the situation in UoD. These representations are subjected to various event detection profiles. If the system elements are found in undesirable state, the necessary corrective action is identified. The identified actions are provided as alert to appropriate role player in UoD. The representations transformed to meet the information needs of role players at different granularity are generated and supplied accordingly to the targeted users. The feedback control loop is realized with the proposed implementation strategy as depicted in the Figure 3.41.

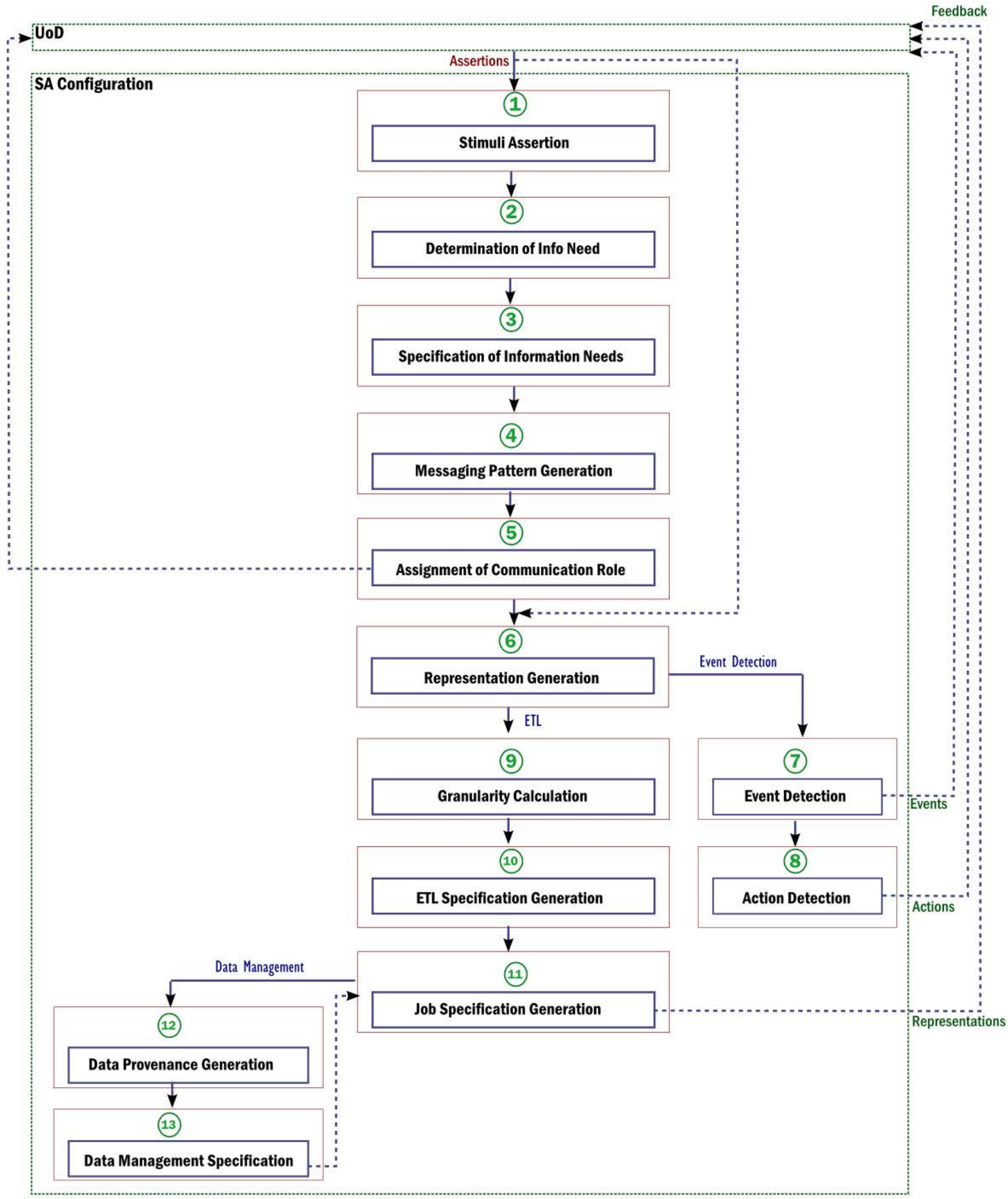

Figure 3.41: The Overview of Information Processing Strategy

Temporal Footprint of Information Processing Models The proposed information processing model provides information processing strategy on the runtime. It allows utilization of modeling strategy proposed as well as the addresses some implementation level issues. From an event detected by on observer up to the role specific message received by appropriate users, the proposed modeling strategy processes the information. The temporal character is to be understood for better clarity regarding the role of the process.

Three types of temporal reference is provided. The instance of time is denoted as t. The duration of information processing time taken by the proposed implementation strategy is denoted as T. Apart from the processing time, the delay taken by the other components, communication and waiting period is identified as τ.

The time when an event take place is denoted as time instance t_e. Event occurred at time instance t_e is first noticed by the observer at time instance t_{s-obs}. τ_{notice} is the duration taken by the observer in noticing the occurrence of the event. The duration is critical in application domains like critical infrastructure protection, disaster management etc. for which the present information management strategy is being proposed. Factors like training, knowledge, experience and human cognition factors of the observer may reduce the delay.

$$\tau_{notice} = t_{s-obs} - t_e \tag{3.60}$$

Once the observation for sufficient comprehension of the event take place at time instance t_{e-obs}, the observer may decide the reporting of the event. Observer may arrive at decision to report the event to appropriate coordinating agency at time instance t_{rpt-dc}. Delay in arriving this decision of reporting the event is known as τ_{cmpr}.

$$\tau_{cmpr} = t_{rpt-dc} - t_{e-obs} \tag{3.61}$$

From time instance at which the observer decide to report t_{rpt-dc} to time instance at which actual reporting of the event start at t_{asrt}, the delay is defined as $\tau_{evt-rsp}$. This signifies the response of the system to alerts reported from the ground. Appropriately designed and publicized systems may have reduced delay for reporting.

$$\tau_{evt-rsp} = t_{s-asrt} - t_{rpt-dc} \tag{3.62}$$

Once the assertion starts at t_{s-asrt} according to the assertion handling strategy, the primary information regarding the events and related instances are collected by the system up to t_{e-asrt}.

From this activity, a `PrimaryInfoNeedList` is identified by the system. This is taken as input by the Information need determination strategy at t_{s-ini}, and all the relevant concepts and their attributes required for the appropriate characterization of event is identified at t_{e-ini}. The delay is defined as T_{ini}.

Similarly, the information need specification takes T_{insp}, messaging pattern generation takes T_{mpgn}, communication role assignment takes T_{cra} as discussed in respective implementation strategies.

Once the role are assigned, the tasks are scheduled for specific time instance t_{sc-tsk}. At this scheduled instance, the system is expected to issue the appropriate task message that is received by the specific role player at t_{c-rcv}. Hence, resulting time delay in communication of the message is identified as τ_c. This signifies communication delay among system and the user.

$$\tau_c = t_{c-rcv} - t_{sc-tsk} \tag{3.63}$$

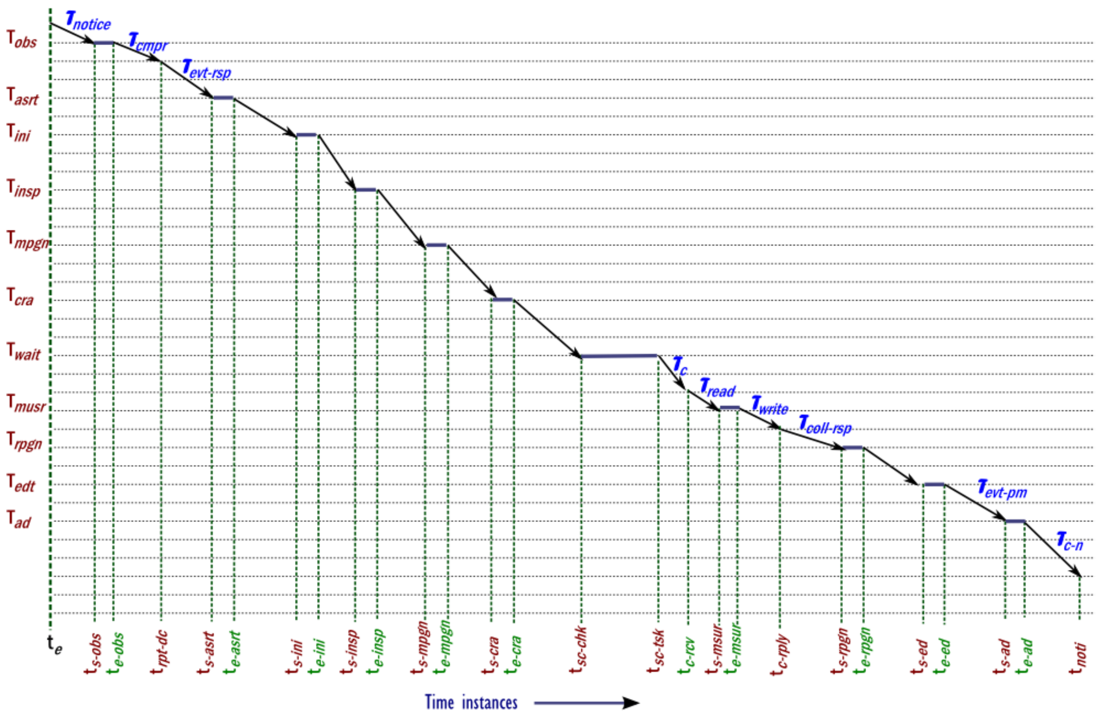

Figure 3.42: Time Difference

Upon arrival of the message at time instance at t_{c-rcv} the observer is expected to carry out the task and / or measure the outcome of the suggested activity. This takes place at t_{s-musr}. This is known as reading gap denoted with τ_{read}.

$$\tau_{read} = t_{s-musr} - t_{c-rcv} \tag{3.64}$$

With end of the required measurement at time instance t_{e-musr}, the user is required to reply in specified manner. This takes place at time instance t_{c-rply}. This results in time delay τ_{write}.

$$\tau_{write} = t_{c-rply} - t_{e-musr} \tag{3.65}$$

Collections of received reports are carried out and the messages are extracted to create the representation. From the instance message was replied by the observer t_{c-rply} up to the starting of the representation generation t_{s-rpgn}, the delay is denoted with $\tau_{coll-rsp}$. This also depends up on the design of communication system and information processing strategy that allows extraction of message and creation of useful representations.

$$\tau_{coll-rsp} = t_{s-rpgn} - t_{c-rply} \tag{3.66}$$

Upon availability of representations, the event detection is employed. Profile evaluation for each possible events takes duration defined as T_{edt} according to event detection strategy. With ending of event detection task at time instance t_{e-ed} up to starting of the action detection at time instance at t_{s-ad}, the delay is identified as τ_{evt-pm}. Various event profiles should be evaluated when appropriate representations are available. Hence, this delay is based on the event detection profile matching tasks.

$$\tau_{evt-pm} = t_{s-ad} - t_{e-ed} \tag{3.67}$$

For identified event, appropriate action detection followed by the identification of appropriate role instance is carried out. The required action identification ended at time instance t_{e-ad}, and notification received by the appropriate role instance at time instance t_{noti} specifies the delay of notification τ_{c-n}.

$$\tau_{c-n} = t_{noti} - t_{e-ad} \tag{3.68}$$

3.9 Summary

This chapter presented a modeling process designed to meet the requirements identified earlier in the discussion. The model-building process allows step-by-step identification of relevant features that characterize complex interdependence and dynamism exhibited by the elements in UoD. The modeling process is not limited to conceptual modeling, and introduced information processing model over distributed computing environment. Considering the scale of the problem, the required information-processing features are identified as task specifications that can be scheduled and executed in appropriate technology infrastructure. By adopting this strategy, the model provided modularized and implementation technology independent solution that can be extended and customized to suit the application needs. In overall goal of situation awareness capability, the proposal introduced the conceptual framework for realization of theoretical principles identified in previous chapter.

In present proposal, the process identifies six conceptual models followed by an information processing model provided as a set of algorithmic strategies. Each of these seven models is targeted to address static, dynamic, logical, empirical, or information processing aspect. With advancement in understanding of the domain, further investigations may reveal requirement of additional models covering specific aspects which are not addressed in present proposal. The presented modeling work therefore, can be considered as work-in-progress and the number of models may change with the progress in research.

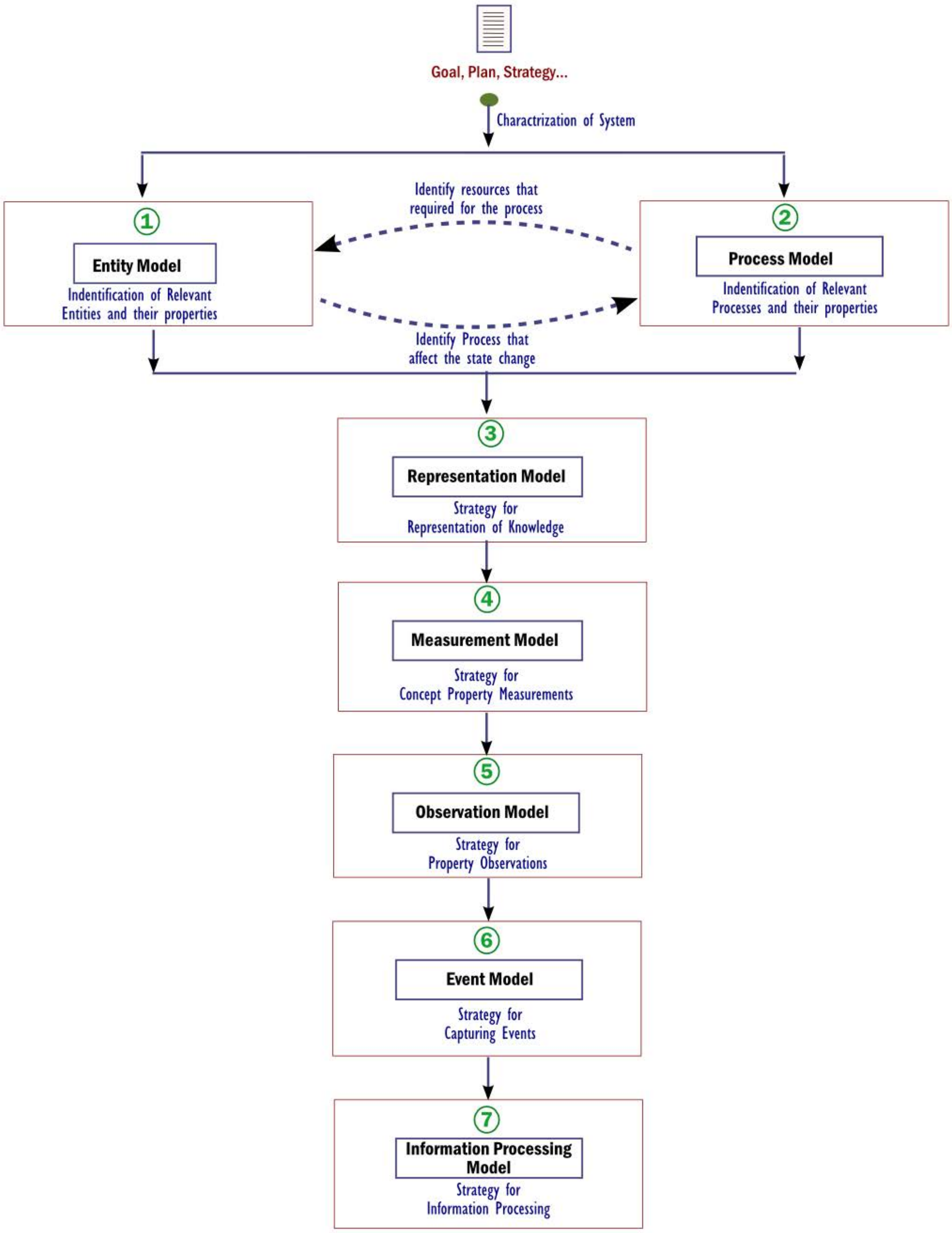

Figure 3.43: The Flow of modeling Process

Chapter 4

Situation Awareness Process

Situation awareness is defined along with the process, configuration and other related details in Chapter 2. A comprehensive modeling strategy appropriate for the situation awareness for complex dynamical system is introduced in Chapter 3. Conceptual models provided step-by-step building process to capture all the required features of UoD. The attributes are carefully selected and captured using empirical methods. This being supported by information and communication technology infrastructure, implementation strategy provides modeling in software environment.

From conceptual modeling to realization, the coordinating agency must cover all the required aspects. With the scale and complexity involved, collaboration from multiple individuals and organizations are required. In this scenario, the consistency and interoperability becomes important. Section 2.4.4 introduce the situation awareness process that provides guidance to large collaborations.

The process content should identify roles, tasks and work products needed to cover complete requirements. These unit elements should be created, organized in to groups and delivered to appropriate actors. Similar to the modeling effort, the task of authoring the process itself becomes complex and requires continuous collaborations from experts. A meta-level solution is required that can guide the collaborative process. These experts should identify workflows for set of role players in realizing a complex task. To support collaboration from users, work breakdown structure of complex activity, work allocation logic, work estimation and other aspects should be recommended. These provisions allow handling of work according to project management discipline.

The meta-level solution must allow identification and handling of task allocation. The monitoring and tracking of activities should also be supported. Due to its nature and complexity,

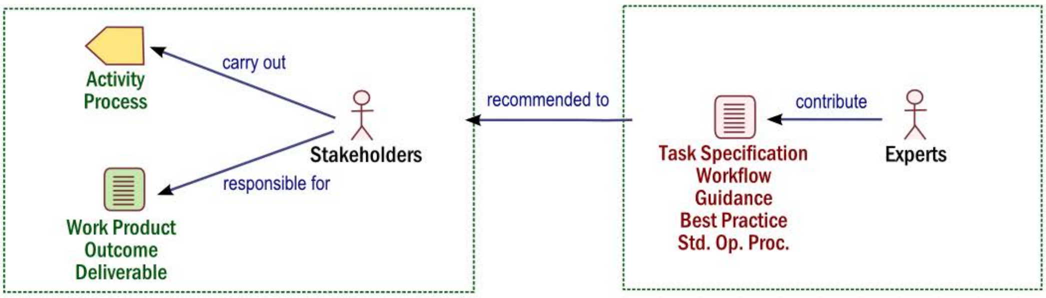

Figure 4.1: Role of Proposed Unified Process

there are multiple disciplines requiring roles with specific process and outcome. While identification of "who is doing what" is important to establish on continuous basis, the roles should be allowed access to content providing proper *separation of concern*. A tool environment must therefore be identified that can support such capability. Computer Aided Software Engineering (CASE) tools are set of applications created to support such tasks. They allow authoring of process with provision for breaking complex activities in to unit tasks and manage in to workflows. It also provides facility to provide role-specific content, along with various types of guidance.

The present chapter introduces the Situation Awareness Unified Process. As the content cannot be provided by single author, it allows identification of approach that will lead towards generation of process content. Using process-authoring tools, multiple experts can collaborate in developing, maintaining and upgrading process content that will lead toward realization of situation awareness goals. The first step towards creation of process content is to identify elements in UoD that must be incorporated in the process.

4.1 Process Elements in Universe of Discourse

The Universe of Discourse (UoD) contains the complex dynamical system of interest. Various relevant aspects of the system should be modeled using the strategy proposed in Chapter 3. The elements of UoD can be modeled from many domain point-of-views. The purpose of information management is to furnish appropriate information to various agents or actors that can bring and keep the system in desired state. This is achieved by appropriate actions. Hence the elements of UoD can be identified as actors and their activities and related elements.

Actors play various roles during their lifetime. In entity modeling process, capturing of roles and various role-specific attributes was discussed. The tasks are the activities or behavior performed by the actors. Based on the performed tasks, role can be identified. A single actor is expected to have one or more roles. The activities those taken by them are also useful. They can be in the form of workflow. They can be separately identified in to specific sub-activities or unit tasks. Some of them require unique resources like consumables or equipments. Some of them have specific work condition requirements. Some of the activities can be carried out in many possible ways, yet due to legal requirements, specific workflows must be followed within specific administrative units. These workflows, related entities and resources are sometimes specific to some domain and have unique requirements. These requirements include skill level, quality and quantity criteria and legal obligations and many other constrains applicable to specific geographical area.

4.2 Stakeholders and Environments

The elements associated with specific role and activity can be collectively handled. The relation and interdependence can also be established. The interrelated roles are called the stakeholders. Stakeholders are active members in UoD. Their contribution is important in elements of UoD. Their collaboration can be implicit or explicit in Chapter 1 stakeholders were introduced. Figure 1.2 indicated complex interdependence among stakeholder activity.

Each stakeholder can be considered as active in specific domain based on type of activities involved, nature of skill set, resource, environment required and the potential use of the work products. Based on this five environments can be identified. Natural and built environment consists of actors and activities in most general sense. The reporting environment those who take preliminary notice and report. Based on the severity of the problem, the organizations take up for further investigation. Technology organizations provide solutions to address the issues. The configuration environment identifies the needs, from actual needs, legal provisions, and technological availability; it provides access to some service. The next sub section briefly introduces each environment.

In activity and process modeling introduced in Chapter 3, the kind of activities possible in given UoD are not defined. Yet, from the outcome of modeling of such diverse activities, it is possible to determine classification of similar activities. Considering only roles and activities relevant to the identified goals, resulting selection include all the stakeholders. For these relevant stakeholder activities in UoD should be handled appropriately to meet the information needs. In order to do this, stakeholder environment boundaries can be established by following criteria:

- Goals
- Type of activity

- Information need for activity
- Impact footprint of the activity
- Work flow and other activity specific requirements

4.2.1 Natural and Built Environment

Natural and built environment consists of all general aspect of the environment. The natural environment includes natural resources and processes on most of which cannot be controlled. Another is the built environment that is the outcome of human activity. The human processes responsible for and related to built environment can be subjected to control by various means.

While natural environment can be identified to include multiple role players, the current discussion will focus more toward human actors and possible roles played by them. The Role can be identified with reference to activity carried out by an individual. Typical roles can be identified with professional activities, voluntary activities and other behavior rendered by individuals during their lifetime. The resource in this environment include all the natural or man made resource that are produced, consumed or involved in the activities. Natural resources, raw material, finished products, wastes are examples.

The processes in natural and built environment can be classified in to natural processes and human activities. Natural processes may have deep impact on individuals and their activities. They are well studied from many branches of science. Their characteristics and impact on human activity is established. These facts can be utilized in determining their role in the process being authored. Various activities carried out by individuals include, professional activity, voluntary activity, routine activity, social activity etc. Their activity is limited to specific domain and local to their surroundings. For natural process, the published articles providing characteristics of the natural processes can be considered as artifacts that can be utilized in authoring the process. In case of human activity, the personal goal statement, communications, memory and related aspects can be considered as artifacts. Personal activity statement is one of the artifacts that can be generated by the individual. The activities can be benefited by establishing professional best practice, work flows to aid professional or voluntary activities.

These artifacts can be in various states like: identified, created, updated and utilized. Artifacts can be found in specific repository. For example who is doing what or who did what type of information in natural in built environment is in the memory of the individuals. The public repositories containing the profiles of individuals also indicate their capabilities for execution of specific tasks. Such repositories can be seen as available pool of resources. The typical artifact

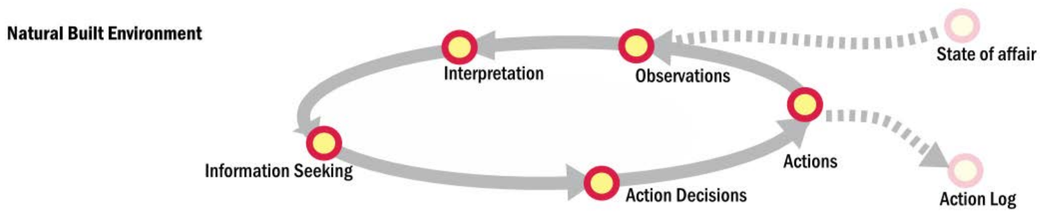

Figure 4.2: Life Cycle in Natural Built Environment

life cycle is depicted in Figure 4.2. For the identification of an activity, individuals are continuously monitoring the natural and built environment. The observed data is been interpreted to determine future course of action. The additional information required for action is collected with information seeking behavior. This includes the decision regarding the activity. The activity is then carried out. The preformed activity stays in the memory of the related individuals. The cycle continues in this manner. According to the punctuated equilibrium theory discussed in Chapter 2, the same process continues with minor modifications and similar activities are getting repeated.

Unless a potential hazard or prospect shown by some other individuals, the perception of individual regarding the activity remains unchanged. The continuous occurrence of routine activity is also not registered until it indicates major incapability, trouble experienced by many. Training is provided that can improve self-efficacy. Knowledge about what is relevant to their personal goals. They engage in social exchange to increase their power by obtaining the best return for their skills and actions. Identification of the most critical action is a plus to them. According to the socio technical theory, more they come in to contact with others, more value attained by them. Hence, for power, they will take up appropriate activity under their skills.

4.2.2 Reporting Environment

Within natural and built environment, there are individuals who are involved in goal-oriented activities mainly in response to events. They monitor various natural and human activities in UoD and investigate and report uncommon events. The purpose of this assumed role is to bring the undesired changes to appropriate role players so that necessary steps can be taken. They themselves are not legally responsible for change, but voluntarily participate in social exchange. The elements in reporting environment include the special events and role players that observe and report them. The resources include equipments, consumables and reporting related items. The processes include survey process, ground investigation and related activities. The environment includes reporting environment with restricted access to certain information.

The purpose of processes in reporting environment is to identify and create awareness about special events that have already taken place or are likely to happen. It is based on data collection, analysis and comprehension about such special events. In order to do this, the roles have to posses and utilize special capabilities that allow them to carry out surveys, investigations and observations. This may span multiple subject or domain point of views. Hence in order to carry out they have to have general knowledge about various disciplines. They can handle many domains and many types of stakeholders and organizations.

Work products of these activities are generally in the form of reports that can be communicated in audio, visual or print form. Some reports may be shorter in length providing observations and interpretation about event. Surveys on the other hand can be based on groundwork and may include data and analysis. News story in video form including interviews with the related individuals may have unique utility.

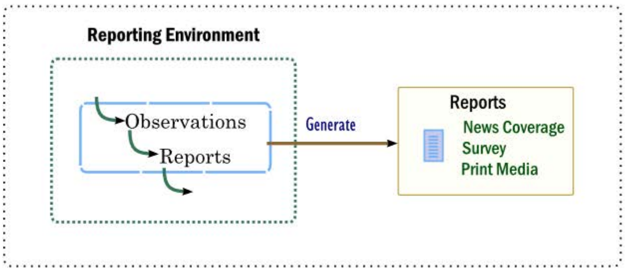

Figure 4.3: Observer Activity in reporting environment

These work products can be in the state of being collected, published, reported, under preparation, and communicated status. These artifacts depending upon their format can reside in various types of repository. News archives in the form of published printed or digital content that is shared in searchable repository.

The process life cycle for a typical news items follows typical life cycle as depicted in Figure 4.4. First an event is identified by the respondent as critical event that must be investigated further. Proper ground work is required and information gathering is carried out to understand the event. The reports are then prepared. They are appropriately published to draw attention from the stakeholders. Tracking of stakeholder activity is also carried out and follow ups are reported as extension updated.

A new event is detected when a major incapability or trouble experienced by many in the natural and built environment. Many respondents will report the same event affirming the importance of event, yet indicating the need for further investigation requiring help from the domain knowledge. Impact footprint of the role specific activity in the reporting environment is unique. The spatial footprint may span globally dealing with global affairs. The subject

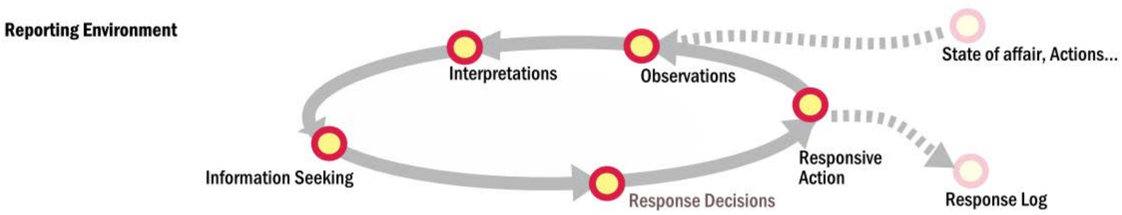

Figure 4.4: Life cycle in reporting environment

or conceptual footprint may span various disciplines like environment, politics, sports, culture, business etc. Temporal footprint is relatively smaller due to the dynamics of the events. Information needs can be quite high spanning various domains from which the event can be evaluated. The sources of information can also be multiple spanning larger geographical areas. Information processing needs may require pattern identification in spatial and temporal dimensions. The domain independent knowledge is required to carry out these tasks and taking decisions regarding the comprehension of the events.

4.2.3 Organizational Environment

Organization is collection of goal oriented individuals devoted to render specific service or product. Here focus is to meet goal in the area under impact footprint. Based on the type of contribution, and goal the organization can be classified in coarsely as:

- Governmental organization
- Non-Governmental organization (NGO)
- Research organization
- Standardization organization
- Business organization

In the context of present discussion, governmental organization, research organization and standardization organizations are considered as basic types and hence elaborated further. The business organizations, NGOs and many other organizations can be understood as derived as special types of these base organizations.

Governmental Organization

In all activities discussed so far, contains reference to an impact footprint. Government is assumed to be legally responsible for certain activities in a specific spatial footprint. This geographical area is considered to be under its jurisdiction, and is expected to provide services according to the policies.

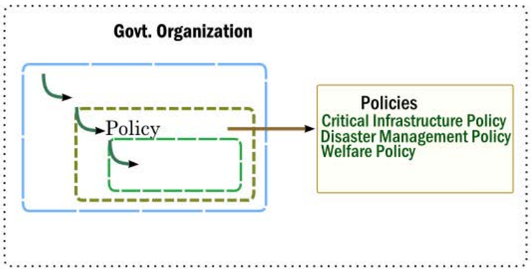

Figure 4.5: Governmental Activity in Organizational Environment

In governmental environment specific elements like roles, resource, processes, activities and environments can be identified. The typical roles can largely be classified in to the administrative roles and the citizens or people who are legally entitled beneficiaries in that jurisdiction. The resources include governmental properties, critical infrastructures, consumables, raw material, goods, finished products, instruments, and other related items. The main activities also include governmental activities like administration, basic services provision, and efforts ensuring peace, welfare and development in the area. The governmental environment consists of the status of relations of the roles, resources within and outside jurisdictions that may affect the processes.

The processes in this environment include making policies, building consensus, providing basic infrastructure services, responding to special events etc. Critical infrastructure services, products are basic work products in governmental environment. They are overall results of various activities carried out by roles in defined hierarchies. Each role have specific work product outcome that can contribute to overall service. The policy, rules, etc. are outcome of the higher level. At detailed level standard operating procedures, guidelines and plans are important artifacts that guide the activities.

These work product outcomes are in various status of availability. They can be in the form of identified, created, published, accessible, discontinued, withdrawn states. Repository is in the form of governmental facility that holds, and shares governmental rules, regulations, policy cases etc. The processes continue to be in the same states of answering the questions raised by the media, public at large or identified by it. Scope and footprint is defined by local jurisdiction that makes the roles legally responsible for certain features. The goal here is to identify potential risks and minimize the damage. Expertise in local goal, decision making, prioritizing and handling in given funds contribute to the tasks.

Research Organization

A research organization has unique role in various stakeholder environments as it is devoted for understanding of a particular discipline. Its outcome can be useful and can bring critical changes and hence treated accordingly.

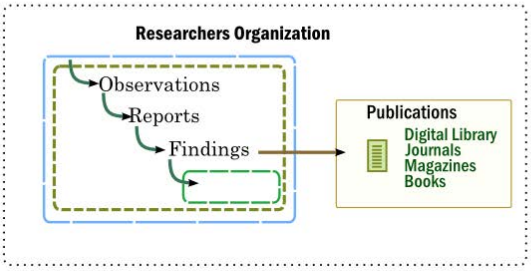

Figure 4.6: Research Activity in Organizational Environment

Basic elements in research environment include the researcher role. Activities include various activities leading to investigations; research and innovation employed in particular discipline or way of problem solving. The resources in these environments are scientific and research equipments, tools, consumables, technologies etc. that facilitate the research and innovation process. The research environment is social exchange of ideas, information, research outcomes, resources among research roles that either facilitate of inhibit the research related activities.

Various activities in research domain include investigation of events in more scientific manner. The assumptions and logic applied in solving problems are based on theoretical framework. Careful data collection and analysis is utilized for decision making.

Work products in the research environment are scientific theories, designs, applicable research prototypes, and findings that can be used for decision making. While most of these work product outcomes are retained by their creators, the descriptions are made available to the community in the form of vision papers, working papers, experience reports, discussion papers, project reports, research papers etc.

A typical work product in research area can be in various states like identified, investigated, designed, developed, tested, published etc. The repository in the research environment is generally the library of research literature. This include on- line repository of research papers that are indexed, cataloged and accessed through search interface. Apart from digital repositories or organizations, such literature is made available in libraries and personal collections.

A typical life cyle in research environment starts with the identification of a research issue. The base data and literature survey is carried out as preparatory step to the research activity. Research issue is noticed in research environment when reports regarding inadequacy and impairments are reported by many, and policies to respond them are identified. When usable results are published in the research repositories, it crosses the boundary to the next level of research problem.

Standardization Organization

Standardization process mostly have global spatial footprint, due to global relevance of the process. It covers special aspects, to improve interoperability and quality of work. It incorporates various business organizations to include outcome of research organizations and using voluntary based participation. Basic elements in the standardization environment are various

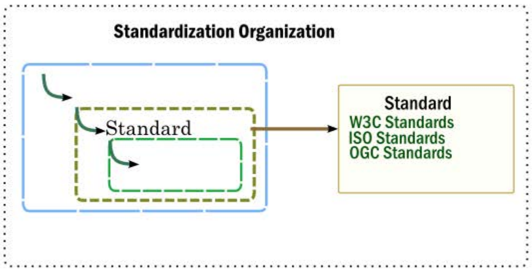

Figure 4.7: Standardization Activity in Organizational Environment

role players spanning governmental organizations, research organizations, business, industry etc. playing role of participant of a consensus building process. They provide case studies, opinions, comments, and experience reports to guide the process of standardization. This environment does not require specific equipments, scientific instruments or items in order to facilitate the process. The consensus building is generally carried out by meetings and conferences. The standardization environment includes regulations, participation of stakeholders and compliance and up gradation of standards that affects the overall standardization process.

The main process in standardization is consensus building upon various aspects of the issues on hand. Major work product in the standardization is a process, method, or guideline in the form of standard. In order to achieve this, various intermediate work products are also contributed by involved stakeholders. This include experience reports, discussion document, specification, request for comments etc.

Standard can be in the state of being identified, investigated, under discussion, specification state, standard form, withdrawn, super seeded stats. Standardization body provides a digital content repository as a library of standards.

A typical life cycle of the standardization process starts with identification of requirement

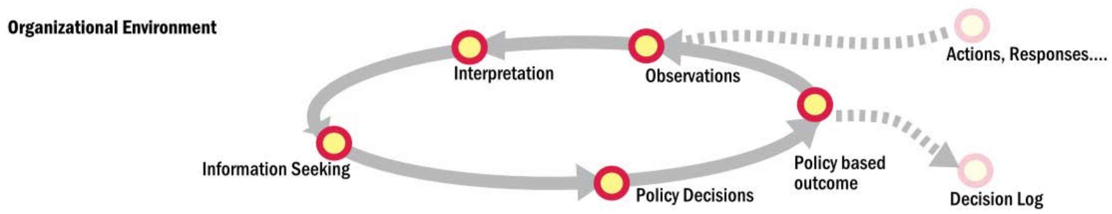

Figure 4.8: Lifecycle in Organizational Environment

for a standard. This is facilitated by first identification of available products or services, a common feature offered by them. At next level it is required to identify various features and aspects that should be addressed in the standardization process determining the scope of the work. Next part can be the publishing the initial draft in the form of candidate specification that undergoes review from the community. In final stage, the comments are incorporated and standard is finalized.

The standardization can first be realized when interoprability issues are faced in practice by using various incompatible products and service providing the same utility. The standardization will continue to iterate the loop discussed above until, a promising implementation confirming to the standard is opened for public use. Considering the applicability of standards, their spatial footprint is global and temporal footprint is for long term where as semantic footprint is specific concept covered in the process.

4.2.4 System Engineering Environment

System engineering environment is the specialized type of environment that is devoted in creation of system instances.

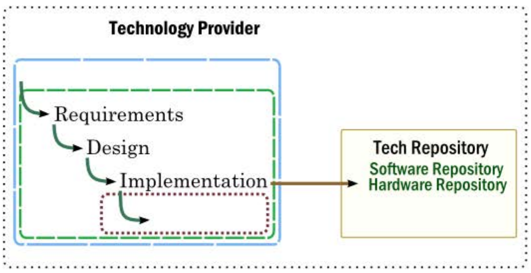

Figure 4.9: Technology Activity in System Engineering Environment

Elements in system engineering environment include system engineering related roles. It includes resources like equipments, devices, machinery, instruments and various types of consumables required in the engineering activity.

Processes include analysis of the requirement, design of the appropriate system that fulfills the identified requirements, creation, testing, delivery deployment and upgradation of the system. Software systems are primarily based on these processes. The rational unified process (RUP) provides most comprehensive guidance for these processes.

Work product in the system engineering environment is primary a system. In order to achieve this various artifacts and work product from numerous role players may be required. The roles specific artifacts defined in the rational unified process can be the deliverables during various phases of system engineering environment.

The work product state can be varying in the form of identified, created, designed, deployed, tested etc. The repository of the designed system can be in the form of product of system catalog. Service or utility offered by various designed systems can be included in this searchable product catalogs. Specific types of products can be found in specific repositories. Software systems, electronic systems, communication systems, engineering systems are appropriately maintained in catalogs. Specifically, software systems can be made easily accessible to users so that it can be immediately brought in utilization. This may include not only the products but, can also be components library that act as building blocks for designing other systems.

A typical life cycle in system engineering environment starts with requirement identification for the product, followed by the analysis and design aspect, construction phase, testing phase, deployment phase etc. For software systems, software development life cycle (SDLC) is well addressed in literature. The waterfall model, the spiral model, the V model are different models for engineering software systems[76].

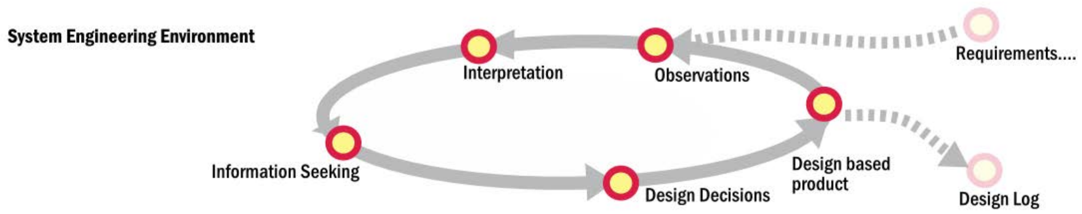

Figure 4.10: Life cycle in System Engineering Environment

Software systems continue to iterate in any of the life cycle approaches discussed above unless major change in requirement or strategy or technology is not changed. Also, the changes and delivery of software system remained unnoticed in the other environments, unless it is solving or providing efficient solution of problem that are faced by larger community.

4.2.5 Configuration Environment

Configuration environment is identified as a provider of a service instance that is utilized by various users.

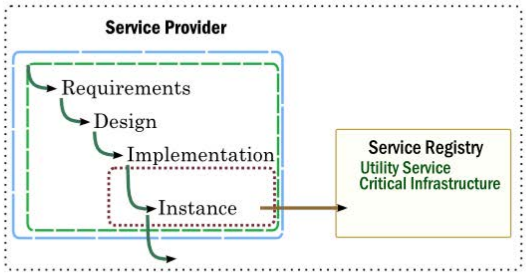

Figure 4.11: Service Provision in Configuration Environment

Configuration environment consists of various elements that are related with the given configuration. The primary role in this environment is service provider. The configurer role is also unique to this environment. Service consumers are also complimentary roles. The services can be in the form of basic infrastructure, or computing or information handling service. Resources in the configuration environment are consumables required for basic infrastructure services. This also includes specific equipments, instruments and devices that are associated with the functioning of the services. Other elements include working environmental conditions. The environmental conditions make it possible for the service provider and service consumer to provide or access services in desired form.

First, it must be determined which services that can be exposed in given area. Once service need is identified, appropriate technology needs to be specified for realization of the service. The technology required for service, needs resources, hence resource identification and acquisition take place. Once resource and technology are acquired, the service is configured to function in desired mode. Configured service is published for general purpose. During the execution, the service maintenance needs to be done. Meeting the changing requirements, availability of new technology and new policies, these services needs to be upgraded.

The work product in this environment is the configured services. The configuration is a result of many activities carried out right from determining the service provision goal, identification, configuration and provisioning of service. Various artifacts are created in the process include: the mission statement of the service providers, the survey about need and market of the service, the survey of the potential services, agreements and contracts with various infrastructure and other basic service providers and specifications of the services and related elements.

These work products and artifacts undergo various state changes during the activities. These states includes artifacts as identified, created, hosted, shared, published and updated.

The work product is made available in the form of configured services. These services are published in service registry. They are also published in specialized service directories, yellow pages etc. The availability is publicized in communication media in improve the uptake.

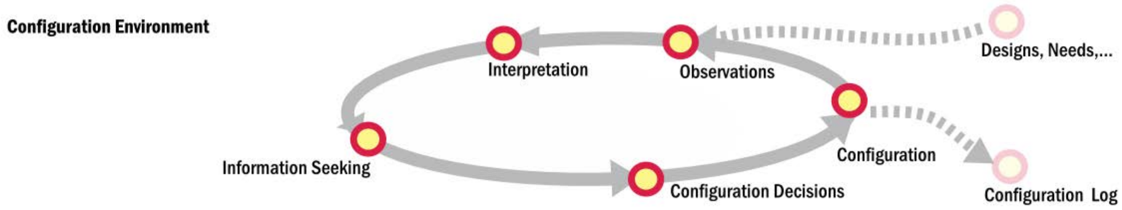

Figure 4.12: Life cycle in Configuration Environment

The process undergoes various phases. The services are in its initial phase identified. Next phase they are surveyed. Appropriate technologies are identified. Contracts and agreements with collaborators are made. The service is then configured and tested. The configured service is exposed for general use. The usage is continuously monitored. And later it is updated, or discontinued or replaced with other services.

The same cycle continues unless new products or technologies enabling the services are introduced to the service providers. If the same services are offered in the overlapping footprint, the services will go to the next level of performance. Configurations are suffered with lack of resources or other constrains. Generally, a configuration is a localized concept. A geographical area may have configuration environment in which various infrastructure service, data services and other services are configured appropriately to furnish the required functionality. In information management, this provides the facility at the runtime, executing and processing of the data and communication platform. A service configuration is an example of configuration environment.

4.3 Implicit Traceability

Activities in various work environments have been discussed. This included the how practice is carried out. It is also stated that major changes are incorporated within such life cycles. This depends up on how information is passed on to the actors. This also confirms to the punctuated equilibrium theory. Any major change communicated will have major changes in the system process.

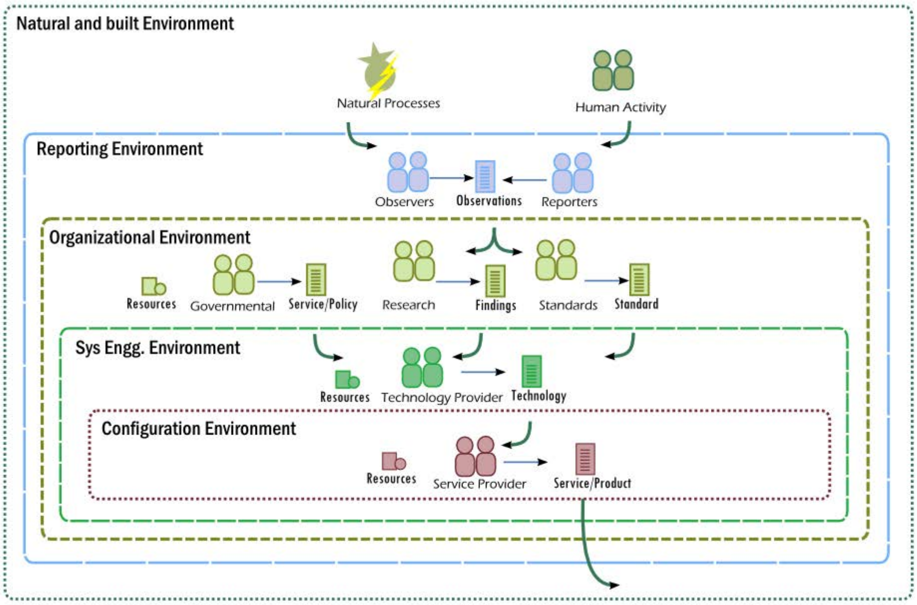

Figure 4.13: Artifacts Traceability Across Environments

4.3.1 Example

This traceability figure tries to depict how any instance (of software, instrument, tool, service) present in given UoD can be trace back to the real life process in Natural or built Environment. The given traceability span across five different environments identified earlier. In each environment, various stakeholders (actors) carryout some tasks (activity) based on the input (stimuli) from the environment and the resulting outcome (work product) is exposed back in to the environment by various means (reports, publication, dissemination, deployment) such that some other stakeholders can take appropriate actions for the same.

It becomes clear that the stakeholders active in different environments possess orthogonal concerns yet they depend up on each other for input and output of their activity. For illustration of this idea, an example of natural process can be considered. Rainfall is a natural process that has considerable influence on many other natural processes and human activities. With this background, a Rain Gauge[1] mounted on a monitoring station designated for a administrative block can be analyzed for cross environmental traceability.

A rain gauge is a scientific instrument utilized for measuring the rainfall over a defined period of time at particular place it is mounted. Following can be identified as elements in various stakeholder environments.

Process Natural process of hydrologic cycle (water cycle[2]) that results in an event of rainfall

[1]Rain Gauge entry on Wikipedia http://en.wikipedia.org/wiki/Rain_gauge
[2]Water Cycle entry on Wikipedia : http://en.wikipedia.org/wiki/Water_cycle

or liquid precipitation. This process is part of natural and built environment as identified earlier.

Events The occurrence of rainfall at particular time and place is considered as event in natural and built environment.

Observations Heavy or scares rain in a geographical area is observed to intervene other natural and human activities.

Reports In one scenario, an incident of heavy rain resulted in water logging and flooding of land; destroying agriculture, transportation network and other built infrastructure. In other scenario, scarce rain resulted in famine affecting agricultural yield of a farming community.

Findings As amount of rainfall is affecting human and natural activities, standard methods for accurately measuring and reporting it are established. Impact of various climatic conditions and environmental parameters that enhance or inhibit chances of rainfall events are identified.

Policy Based on past experience of rainfall, the government resolves to monitor the seasonal precipitation and to take corrective action as per the scientific findings available till date covering all states under its jurisdiction. State Government makes a local department responsible for tracking liquid precipitation as a part of this policy.

Requirements Based on scientific findings and requirements identified by the local departments, requirement of an instrument for monitoring and reporting rainfall in a given area with specific operating range and sensitivity is identified.

Design A rainfall logging system is design by an technology firm.

Implementation Sensor logging application developed based on design.

Instance An instance of instrument mounted at a monitoring station providing representative liquid precipitation data for a local administrative unit.

Hence, with this example, it is identified at how given instance of a rain gauge can be traced to the processes and activities in various stakeholder environments.

4.3.2 Critical Infrastructure Revisited

Critical infrastructure was indicated as one of the challenging application area for situation awareness. UoD consists of a complex network of critical infrastructures and any undesired state impairs specific service rendered directly or indirectly offered by them. A single critical service may require many other basic critical services. If each service is considered as unique network of related elements, several such actor networks can be identified. Commonly used basic critical infrastructures are found in all identified actor networks. Hence, as depicted in Figure 4.14, an area can be visualized to have multiple overlapping actor networks.

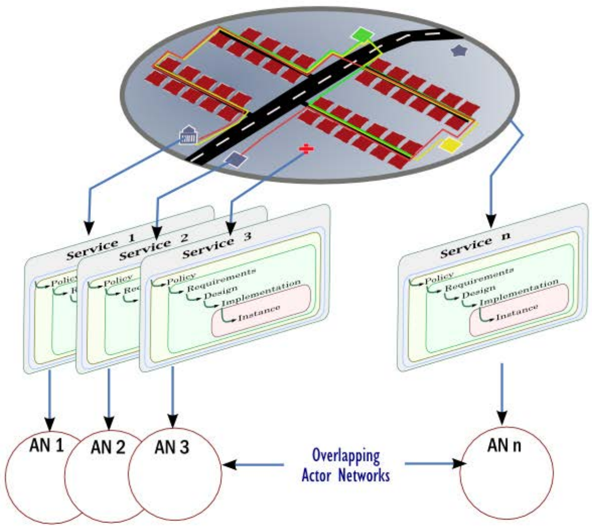

Figure 4.14: Service as Artifact Across Environment

From the example and implicit traceability discussed earlier, it can be visualized that each actor network is outcome of activities being carried out in various stakeholder environments. Hence, any attempt to monitor and manage critical infrastructure must identify and appropriately handle related stakeholders in appropriate environments. This allows identification the relationship of activities resulting in policy, research, standards, designs, technologies, that are not visibly present in the UoD are directly affecting the quality and availability of services in given area.

4.3.3 Traceability Across Environment

Observations done in the reporting environment are based on processes and activities carried out in natural and built environment. These published reports are taken as reference for governmental policy and decision making processes. If reports introduce issues that need to be further investigated in view of some discipline, the research organization environment takes them as reference to the research. Reports of events, governmental policy and research carried out in different environments provide basis to a technology provider to develop and distribute a technology solution in system engineering environment. Multiple solutions offering the similar functionality may create interoperability issues and standardization organization take these product specifications as basis for building standards. The standards set in organizational environment are again referred back in system engineering environment to ensure compliance. Needs identified in reporting environment, the policies, research and standards made available in organizational environment and system built in system engineering environment are tracked by the service providers in configuration environment to provide specific services. The same services are consumed by the end users in natural and built environment.

Hence, it can be concluded that activities and resulting work products are integrated across environment. Impact of one activity or work product can be traced to activities and work products in other environment. Figure 4.15 indicates the though each environment have its own routine work process, they have the implied traceability.

4.4 Situation Awareness Needs

The way the information management is handled, it is not possible to design the supporting system in isolation and later deployed for use. The information management needs to be handled as a critical infrastructure service, and supporting system environment is created, updated managed on the runtime. For required scale of effort, an isolated system development team may have limited staff only fixing bugs and getting extensions. Information management for dynamic environment requires a team with specific roles, and need to maintain a live connection with many other role players in stakeholders. Hence, this task should be created as a service as what role Air Traffic Control plays for the aviation domain. The system related roles must constantly identify and address emergent needs and provide their contribution in successful operation of the service. In other words, all roles must be provided task related situation awareness.

In Figure 4.15, the work product usage is found to go across the environment. Once shared in respective repositories, these work products are identified by the stakeholders in other environment. Generally this takes place in response to two events. In planning phase, when plans are made, the survey is made about basic reference documents, and relevant work product outcomes are searched and retrieved from their repositories. In punctuation scenario, they are accessed the similar manner, but in response to a punctuation. In any of the cases, availability of all the work products that are required to sufficiently meet the information need of the stakeholder should be shared in the repository. On the other hand, from all the available work product relevant to the present activity should be identified and accessed by the stakeholder. In both the cases, for complete coverage the stakeholder must have to keep the track of all work products being shared. In other words, those sharing and the others who are searching and utilizing these work products should be aware of all the repositories for complete coverage. Or, there must be some mechanisms defined that can allow sharing and searching of work product outcome is various repositories that stakeholders may use in different environments. This calls for a unifying approach across environments that enable consistent sharing and using of artifacts across environment.

Apart from using shared work products as input to the process, the stakeholder activity can have specific information needs. The human activity modeling approach introduced in Section 3.3, indicated information that is required to complete the task. This includes information about task allocation, status of related inputs, work flow, state transitions, process outcome, guidance required for activity. All these contribute to the situation awareness of the specific user planning for the activity.

4.5 Method Engineering Approach

System engineering domain provides solution for the identified needs. It recommends identification of elements like process, methods, tools and environment to support the system development task[76]. A logical sequence of activities employed to achieve particular objective is identified as process. While process defines the "what" part of the solution by identifying the required activities, detail steps and required guidance is defined as "methodology' to provide the "how' part of the solution. In order to execute the process according to the method content, set of systems may be required to support them. The systems supporting the method part are known as tools.

The management strategies applied for general project management or software project management are examples of such strategy. The domain addressing issues related to development of method content is known as Method Engineering [77]. Method engineering targeted for systems that is continuously faced with dynamism may require event and scenario specific method content. Situation Method Engineering [78] is one such technique employed in the domain. In software development environment, Rational unified process (RUP) is widely accepted method content among the practitioners [79]. Some domain or application specific approaches

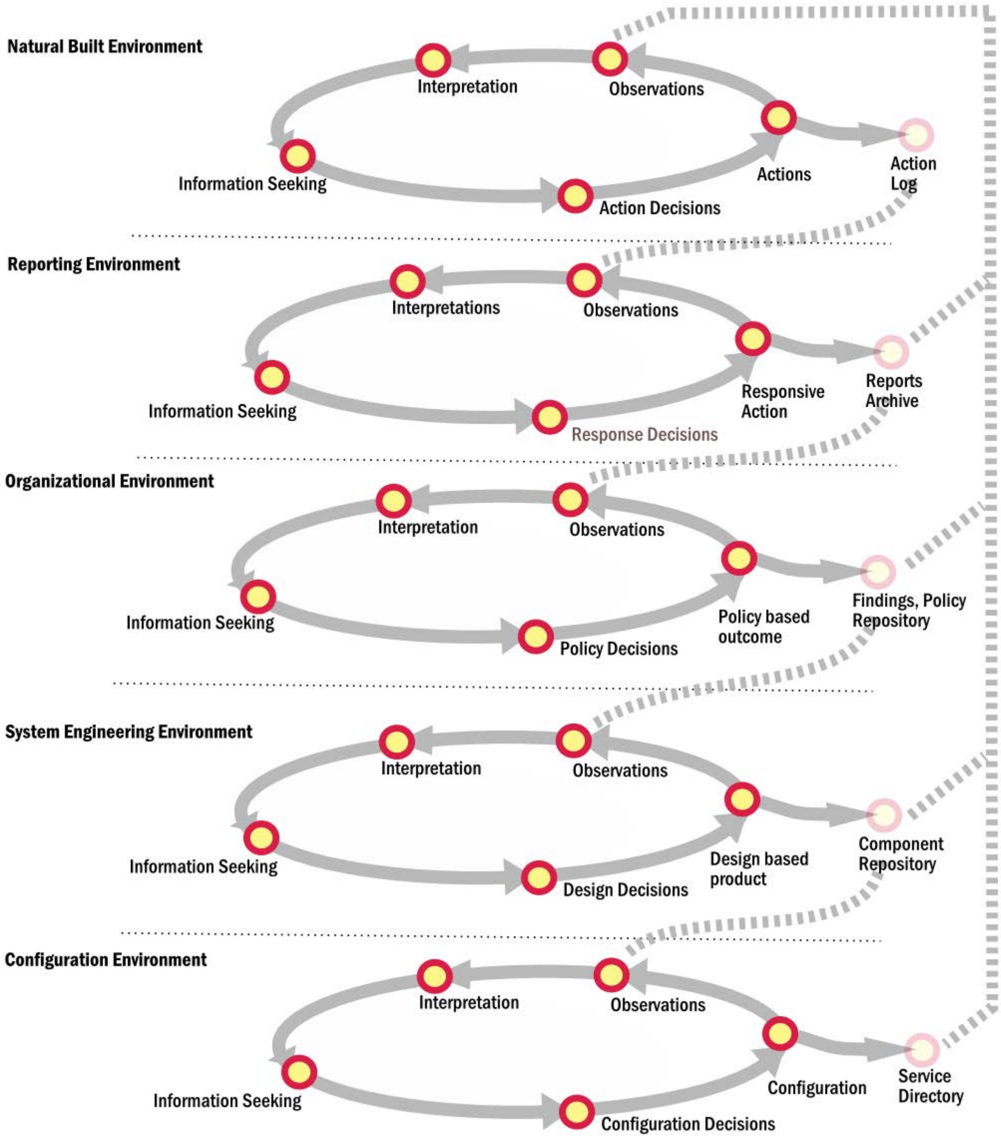

Figure 4.15: Implicit Traceability across Environments

may involve special consideration in system design process. For example, utilization of product specific knowledge can be employed to enhance the performance of the development process [80]. Project management discipline provides comprehensive guidance in the form of Project Management Body of Knowledge (PMBOK).

During the development of systems and work products, commitment to method content proves beneficial. During the requirement gathering stage of an enterprise information system many non-functional requirements that are essential for collaboration in dynamic environment can be missed. Efforts toward team integration later become difficult task. The basic principle of the unified process is that, organizations will define their commitment toward methods instead, and then the developers will develop and test according to the given reference model. Thus, rapid adoption of missing components in the system could be made possible.

4.6 Situation Awareness Unified Process

The roles, activities, work product outcome, repository workflow, life cycle, discussed needs to be arranged based on following criteria:

- Use of models (Chapter 3) to assemble activity related aspects
- Use of situation awareness theory to determine information needs of the roles
- Use of Method Engineering Strategy to create content

Rational unified process provides comprehensive method content and tool support for large-scale collaborative efforts. The feature of authoring and customizing the process to suit the emergent need is also included in the process. Hence, the content that can be reusable can be directly adopted from RUP and extensions provided by the collaborators. Content for situation awareness suitable for targeted stakeholders can be managed in similar manner.

Situation Awareness unified process is defined considering these aspects, following is the brief summary introducing the content of the Situation Awareness Unified Process (SAUP) [81].

4.6.1 Role and Role Sets

Roles defined in the process represents the specific responsibility assumed by the actors. This responsibility includes generation of specific artifact or work products by engaging in various tasks as prescribed in the method. The method content regarding a role typically include general information regarding the role, detail description of the duty, staffing information and

Elements	Content Type	Description
Basic Elements	Role	Actor responsible for specific tasks
	Task	Activities performed by the actors
	Work product	Outcome of activities
	Tool	Tool support
Aggregates	Disciplines	Collection of Tasks
	Domains	Collection of work products
	Process	Logical Sequence of Activities
	Role Sets	Collection of similar roles
	Work Product Kind	Type of work products
Reference	Guidance	suggested Reference Material by experts

Table 4.1: Content of Suggested Situation Awareness Process

other information regarding the date and versions of the role specific content. The roles are associated with specific work products. They are also prescribed specific guidance. Depending upon the type of work and skill-set possessed by the role, it can be assigned specific category. This information is handled by conventional method engineering approaches.

From situation awareness point of view additional information is required. The SAUP recommends these additional features to be identified for the roles. The need of the role is explicitly identified so that appropriate stakeholders can be allocated for the task. Typical use cases of the roles are represented stating the role in the collaboration. Common challenges faced by the role are required to provide guidance and make provisions to solve the difficult issues. The input to situation awareness and output to situation awareness are two important elements defined for each role. The frequency of accessing the method content is to be defined. The matrix created for the role to establish cross environment traceability is also identified. Set of all possible events regarding the role is also enumerated. Duration of typical activity is identified. Type of system requirement in accessing the situation awareness artifacts are also specified for each roles.

Figure 4.16 represents major role sets identified by the Situation Awareness Unified Process (SAUP). The role sets are identified is respective stakeholder environments.

4.6.2 Tasks and Disciplines

The tasks are the activities identified by the process that can be executed by related role players. Typical task related content in conventional method engineering approach include purpose of task, description, key considerations and alternatives. The content variability type is also defined for given tasks. Elements providing task related guidance include steps, associated roles and additional performers, related work products in form of mandatory and optional inputs and

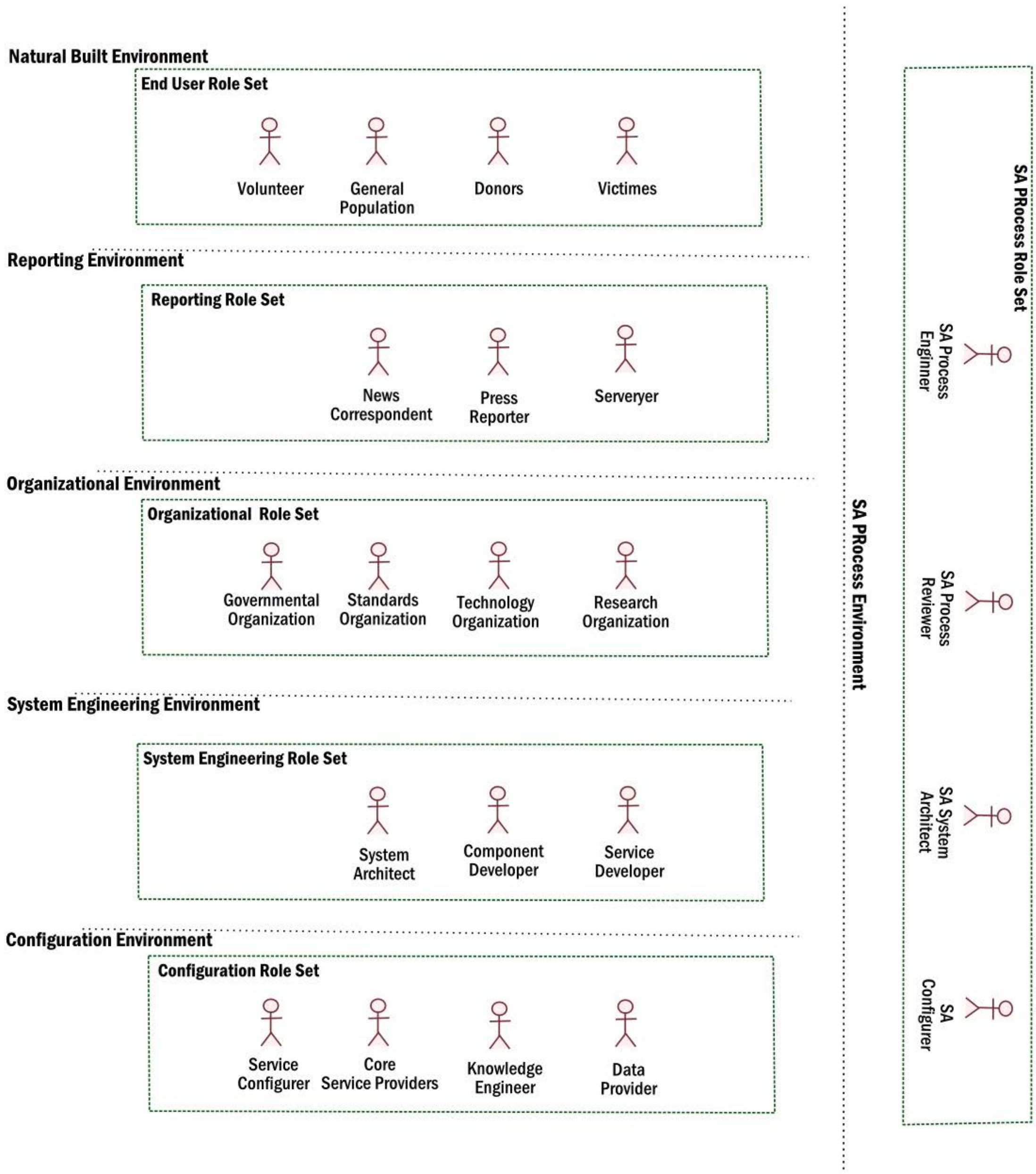

Figure 4.16: Unified Process Roles Sets in Stakeholder Environments

outputs, and guidance. Each activity known in stakeholder environment is converted as task and appropriate content is developed.

From generated content regarding tasks, the related tasks are aggregated to build a discipline. Each identified disciplines therefore contains set of tasks, and workflows that involves execution of tasks in logical sequence. Conventional disciples are analysis and design, business modeling etc., recommended by the RUP. The SAUP deals with five stakeholder environments and specific disciplines are introduced for each environment. Table 4.2 indicates the disciplines introduced in SAUP.

Discipline	**Description**
Professional Activities	Activities of specific profession
Reporting	Activities related to reporting of events
Standardization	Activities related to standardization
Research	Research activities
Administration	Administrative activities
System Engineering	System engineering activities
Configuration Activity	Service configuration and provisioning activities
SA Process	Building situation awareness process content
SA Configuration	Configuration of SA system

Table 4.2: Disciplines in SA Unified Process

4.6.3 Work Products and Domains

During content development of roles and tasks, work products are identified as one of the element. Work product is the outcome of task or activity carried out by the role. Typical content in work product include, description, key considerations, outline, representation options, impact of not having the work product, reason for not having the work product, version information, content variability type information and suitable guidance.

Similar to role and tasks, related work products can be aggregated to form a Domain. Some known domains in RUP are analysis and design, business modeling, configuration and change management, deployment, project management etc. Analysis and design domain is logical collection of analysis and design related artifacts like analysis model, data model, design model, architectural proof of concept, software architecture model etc.

Situation awareness unified process recommends domains that logically collect products in different environment. Table 4.3 enumerates domains proposed by SAUP.

Domain	Description
Professional Activities	Work products related to professional activity
Reporting	Work Product related to reporting
Standardization	Standards, specification and related documents
Research	Research documentation and work products
Administration	Administrative documentation and work products
System Engineering	System work products
Configuration Activity	Configuration work products
SA Process	SA process and related content
SA Configuration Specifications	SA configuration related work product

Table 4.3: Domains in SA Unified Process

4.6.4 Processes

The tools for building method content support two forms of the process, namely "Capability Pattern" and "Delivery Process". Table 4.4 represents the process content with examples. The

Process	Description	Example
Capability Pattern	Fragments of process	
– Discipline Work flow	– focusing a discipline	Analysis and design
– Template for Delivery Process	– focusing a discipline	Construction iteration
– Typical Pattern	– focusing a goal	Create product to release
Delivery Process	Complete life cycle	Classic RUP

Table 4.4: Process Components in SA Unified Process

process made available as complete life cycle or a fragment as indicated in table 4.4, it is having specific features. Capability pattern and process package are managed in RUP.

A typical delivery process has general information, name, brief description, external ID, purpose, etc., detail Information like main description, scope, usage note, alternatives, staffing, key considerations, scale, project characteristics, risk level, estimating techniques, project member expertise and type of contract, A delivery process may include content from other existing method plug-ins. A proper configuration is required that allows incorporation of existing content.

Work breakdown structure breaks the large process into breakdown like phase, iteration, activity, and task. Phase signifies considerable period of a project resulting in important project outcome or deliverable. Iteration indicates the recurring activity or asks with repetitive nature. Activity is an important building block for a process. Team allocation indicates the roles and role sets are allocated the task. Work product usage Indicates links to the identified work product. SAUP have three types of Delivery Process namely: *SA Global Configuration*, and *SA Local Configuration* separated based upon configuration requirement. Local requires minimal setup whereas international requires maximal setup, connecting and monitoring to globally relevant activities, work products and roles.

4.6.5 Guidance

The purpose of the process is to provide guidance to the development process. The roles, tasks and work products discussed are the concepts that are characterized by various attributes in the context of the process. Mere identification and enumeration is not sufficient in realization of process. To aid this process, material in various forms is introduced to guide the process.

For example, representation model recommends the knowledge engineer to carry out conceptual modeling in description logic. A useful step-by-step guide for building KB using this manner is discussed by Borgida and Brachman [68, page 359] providing the guidance required in knowledge representation. Similarly, converting multidimensional conceptual models in to physical design, Malinowsk and Zimanyi [69] provide comprehensive reference. Current method composing technologies supports following types of guidance.

Checklists Checklists are common mechanisms to verify the status of a list of important items. In verifying operating conditions, availability of inputs, generated outputs, processing steps etc, these checklists can be useful.

Concepts The process content have to incorporate business specific, information processing and handling specific, application specific terms. For explanation of critical vocabulary entries, the conpets are defined. RUP supplied an Actor Checklist for identification of various actors in the system.

Estimation Considerations Considerations required in estimating the effort in completing the task.

Example Examples are sample worked out solution of a work product created according to various aspects of the rational unified process.

Practice Practice is a proven approaches to successfully complete the activity. Some examples are Risk management, continuous testing or component based development demonstrated by an organization.

Report Report is a documentation of work product outcome resulting from the recommended process. Outcomes and results are collected in represented in predefined template and may be automatically generated by the tool.

Reusable Asset Assets are solution to context specific problems. Therefore they can be utilized in similar situations, enabling the reuse. Such reusable assets are supplied in the form of rules for its usage.

Roadmap Roadmap provide simplified representation of the process. They are useful in depicting how tasks and activity may lead to specific outcomes.

Template Documents, conceptual models or physical data models are generated as a part of the process. Template provides guidance for content of these artifacts. The content guidance include various sections, diagrams, packages or snapshots that must be arranged in specific sequence to improve readability and comprehension of the work product outcome.

Team Definition Uncommon terms, domain specific vocabulary or concepts mentioned in process is defined using Term Definition.

Tool Mentor Tools provide important support in managing various work products and outcomes from collaborating roles. The utilization of tool in managing various roles is provided as tool mentors.

White Paper General introductory document prepared to cover specific aspect of process, is published as white paper. It can be referred independent of the other component of the process.

Guideline Best practices and various ways to perform the activities or group of activities to achieve desired outcome is captured and provided as guideline.

Supporting Material Supporting material is any other type of guidance that are not falling into various guidance categories discussed above.

4.6.6 Authoring the Process

Process engineering is currently supported with method composition tools both in open-source and commercial domain. The Eclipse Process framework (EPF) have provided tooling with open method library like OpenUP. A well-established commercial counterpart is Rational Method Composer (RMC). RMC is packaged with Rational Unified Process plug-in with various extensions for business and enterprise development with a number of software engineering paradigms. RMC also provides additional method plug-ins like SOMA and DoDAF. As of now, the first version of SA Unified Process is authored as RMC Plug-in extending these available required method parts. Benefits of composing tools is that community contributed methods can be incorporated to suit the requirement. Also, reusable components can be directly utilized to suit the needs. In EPF, OpenUP and others are available. Upon availability, there can be may others from community contribution. The method plug-ins authored in Rational Method Composer product, contains rich library addressing various issues. RUP SOA plug in etc provide extension to RUP to develop service-oriented product. Apart from this, they also provide Asset based development, business modeling, system engineering and others. These method fragments can be utilized by specifying configuration and appropriately integrating with the custom made processes.

4.7 Summary

For identified “Situation Awareness” capability, a suitable framework was proposed that can appropriately enable the actors to collaborate and fulfill the required information needs. It is proposed that situation awareness can be realized in a process-like form. Rational Unified Process is one commonly used effective process that enhances the software development capability in team environment. The proposed theory can also be delivered in similar form, as all the required features for situation awareness are achievable in process form. Various stakeholders like: observers, researchers, standards organization, service organization etc. can share their outcomes in form of artifacts or work products. The theory proposed here requires incorporating the local business rules and operating procedures. This requirements demand for the feature for extending or customizing the template process. Stakeholders and members of “active publics” assume specific roles, and engage in input output dependence with other actors. Monitoring of the status of various activities carried out by roles is also important feature of the proposal. Similar to the software development scenario the checklists, best practices, white papers and other kind of guidance is equally useful to various stakeholders like respondents, government organizations, research organizations, service providers etc. active in respective stakeholder environments.

Chapter 5

Situation Awareness Architecture Framework

The discussion started with the objective of meeting information need in complex dynamical system so far introduced three dimensions. First being theoretical underpinning of situation awareness, followed by modeling process that lead toward characterization of complex situation and provided strategy for information processing. With recognition of wide scope and considerable amount of collaboration to carry handled the proposed modeling approach, a proper management strategy was required. The situation awareness unified process is introduced for management of this ongoing process. This process allowed creation, handling and sharing of work products created in various stakeholder environments.

Next steps for the stakeholder are to make utilization of the guidance for constant delivery of their work products. They can also identify appropriate tasks, activities suitable to their skills and utilize the available guidance. They can also track the process of complex activities, by tracking work breakdown structures, team allocations and work product usage.

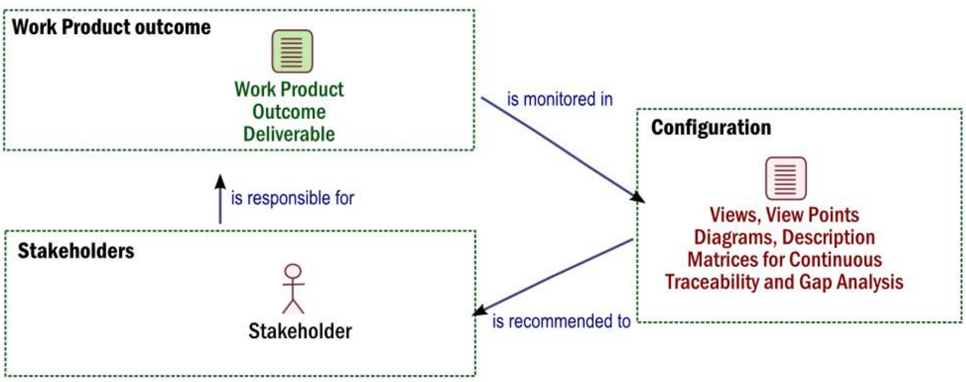

Figure 5.1: Role of Proposed Architecture Framework

Even with existence of delivery process spanning stakeholder environments, each environment has unique delivery mechanism. Situation Awareness Unified Process provides appropriate guidance by means of method content and templates for consistent delivery of work products. Though they have implied traceability among products, mechanisms that can specify the link is missing.

Appropriate mechanisms must be identified that can hold the information about work product status and usage as depicted in Figure 5.1. The strategy should introduce new mechanism that can support visualization of these implied links. This should not only be limited for identification, creation and delivery of work products, but it should be maintained as discoverable reusable components. When work products are reused, or utilized in other stakeholder environments, it should contain explicit links. This allows two-way benefits to those who share and those who use the product. The user, who shares the work product, would come to know, how their work products are utilized. They can track issues in utilizations that can incorporate in next versions. On the other hand, the person reusing the component will also have link to the source or reference. In case of any updates, they can appropriately get the benefits of updated products.

5.1 Need for Architectural Framework

In complex dynamical systems, requirements are continuously changing. In response to the changing requirements, various role sets are providing components services or artifacts. For them it is necessary to track how their contributions are contributing. Some of these are reusable and can be utilized by others in different setting. They also want to be alerted when a reusable component is available. Once they are utilized by others, those who contributed may also be interested in knowing the usage details. In complex dynamical systems, answering these questions is difficult.

The situation awareness unified process introduces novel process role set, configuration Role set, and component developer role set. All these role set are connected to software components. The task of these role set is to provide service, component or specific artifact. The roles are also required to track the usage of their outcomes, and find out gap that will identify requirement of for further architectural work.

Architectural frameworks are blueprint of the architecture and they reveal how they are built using components. They provide various views that reveal different aspects of the system. Hence, architectural framework provides answer to the question.

5.1.1 Architectural Products for Situation Awareness

The conventional architectural frameworks are provided as static view of single configuration. Architectural products are predefined and created as models which may not useful in dynamic environment. For identified situation awareness theory the collaboration from stakeholders from multiple environments are required. The outcome should be available in information system. To provide complete coverage, the repository of all reusable outcomes should be configurable. Various architectural products can be utilized to reveal instantaneous scenario in the configuration.

5.1.2 Architectural Frameworks

As a conventional practice, the business logic and system related aspects are represented with UML diagrams. This include creation of set of UML diagrams that allows capturing structural, behavioral and interaction properties as indicated in Table 5.1.

Diagram	**Content Type**	**Description**
Structure	Class Diagram	Class attributes and relation to others
	Object Diagram	Object instances
	Component Diagram	Components and inter-relations
	Deployment Diagram	Deployment overview
Behavior	Use Case Diagram	Representing business actors and functions
	Activity Diagram	Representing flow of activities
	State Chart Diagram	Representing possible states and transitions
Interaction	Sequence	Sequence of interaction among objects
	Collaboration	Interactions among objects

Table 5.1: Diagrams in UML

Based on the set of UML diagrams, the conventional architectural framework known as 4+1 View Model provided views like logical view, process view, deployment view, physical view and scenarios. The Reference Model for Open Distributed Processing (RM-ODP) is later adopted as architectural framework strategy for depicting distributed system architectures. It recommended creation of enterprise viewpoint, information viewpoint, computational viewpoint, engineering viewpoint and technology viewpoint. Rich set of architectural description created under these categories are widely accepted in practice by industry and standards organizations. These approaches generally provide static views of the system components and the stakeholders. For meeting the information needs depicted in Figure 5.1, these UML based approach are not useful due to expected dynamism and complex interdependence.

The Open Group Architecture Framework (TOGAF) [1]Classification of products [11] pro-

[1] TOGAF Version 8.1.1: http://www.opengroup.org/architecture/togaf8-doc/arch/

posed by the Open Group, provides architectural product development method, introduces enterprise continuum and other relevant aspects. Considering the requirements of defense and military domain, the Department of Defense Architectural Framework (DoDAF) [82, 83] is defined. This proposes comprehensive set of views that revels details of various components. They are targeted to reveal how various system elements are built and interact with other elements to realize the operation needs.

Such comprehensive representations allows monitoring of dynamism and complex interdependence of system and related elements in given UoD. For example, Figure 5.2 indicates a DoDAF architectural product identified as High Level Operational Concept Graphic (OV-1). The content of the product are dynamic set of system elements standing in relation for emergent needs. The architectural product is useful as it provides exact amount of operational node working after the event. As the operation may span across the administrative and geographical boundaries, it is necessary to incorporate all the participating entities. Also, role of EOCs are important in mobilizing the resource. The donations and inflow of volunteers should be appropriately managed by the central coordinating agency. These elements are not available at design time, but only created on the runtime after occurrence of events. DoDAF products allows representation of dynamic situation, hence all DoDAF products are adopted in proposed strategy.

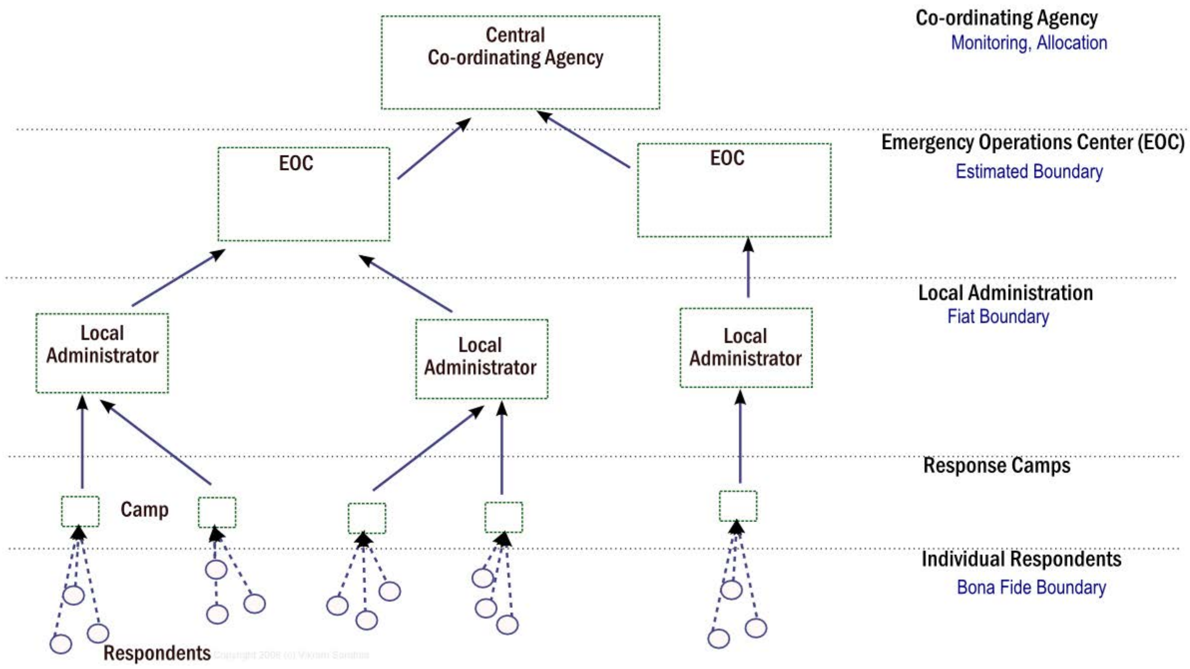

Figure 5.2: DoDAF - High Level Operational Concept Graphic

Federal Enterprise Architectural Frameworkb(FEAF), Zachman Framework[2] (ZF) and may other are introduced to address requirement of architectural frameworks[84]. Among these, Zachman Framework provide comprehensive matrix for addressing architectural issue of information system. This comprehensive approach starts at system level up to business goals and covers concerns of various system related stakeholders. The current focus in domain[85] put emphasis on graphical representations and visualization capability [86]. Architectural framework is one such mechanism providing capability [87].

5.2 Architectural Framework for Situation Awareness

A properly desired architecture framework can provide situation awareness capability to the information management strategy. As architectural framework allows creation of rich views of the current state of affair, they allow mechanism for visualization of important system components.

5.2.1 Requirements of Situation Awareness Architecture Framework (SAAF)

For architectural framework strategy, In order to be useful for situation awareness capability in complex dynamical system, following requirements must be met.

Reflecting Changes Instantaneously They involve complex dynamical entity with changing states. Hence should be uploaded frequently. With capability to establish state changes to represent reality.

Continuous Gap Analysis They should be designed in a manner such that it can be able to establish gaps instantaneously. This gap analysis provides formal reference to pending task thereby allows identification of the scope of management activity.

Collaborative Architectural framework can not be prepared in isolation, and requires properly managed collaboration of multiple individuals and organizations to meet the full coverage of the requirements.

Extensible Product Repository The architectural products should be available in repository with search capability. Hence the enterprise continuum can be extensible allowing inclusion of other repositories.

Live Traceability The roles take action according to gap analysis. The traceability therefore should be established as live artifact such that tracking of action role guidance and other artifact should establish live traceability.

[2]Zachman Framework http://www.zifa.com/

Resuablity The architectural products should be reusable in various configuration environment meeting unplanned reuse requirement.

Configurable As it is expected to provide live reference and continuum, many contributing systems should be configurable. It should be possible to change the configuration parameters dynamically.

Generated on Demand Many potential users, varying scopes, can not be created static as bath. It should be created only on demand of the user, or driven by rules.

Matrices Many architectural products are created as $m \times n$ matrices representing link among two sets of entities. In larger scales anticipated in complex dynamical system, conventional approach may not be useful with larger count of m and n. Appropriate strategy should be employed for the matrix based products for easier handling and visualization.

5.2.2 Role of SAAF Product

Implicit traceability was found and established among the work products of various stakeholder environments. To establish traceability and to allow coverage, specific mechanisms are required. Figure 5.3 indicates that instead of assumed traceability, specific encoded information can be utilized by the stakeholder as source. Similarly, they can continue to produce the work products in appropriate repositories, yet they can submit information about the availability. This is also part of task closure mechanism discussed in the activity modeling in Chapter 3. In Figure 5.3, it the required mechanisms are indicated in the form of architectural products that can provide these facilities.

5.2.3 Template for SAAF Product

The template of a typical SA architectural Framework is as follows:

Architectural Product Name : Name of the architectural product with abbreviated short name.

Related Role(s): Role in SAUP that can be benefited or associated for this architectural product.

Responsibility: Actions, tasks, or work products that that the role is made responsible in Unified process.

*States:*Status of action, tasks, artifacts, that are relevant to the role.

Creation Logic: Architectural product can be generated on demand of the user meeting unique information needs or it can provide a static view to all users. The query type states how artifact is generated. Query prepared, executed and displayed to the user as architectural product. The instances are individual concepts, properties, and entities in KB that are retrieved by the query.

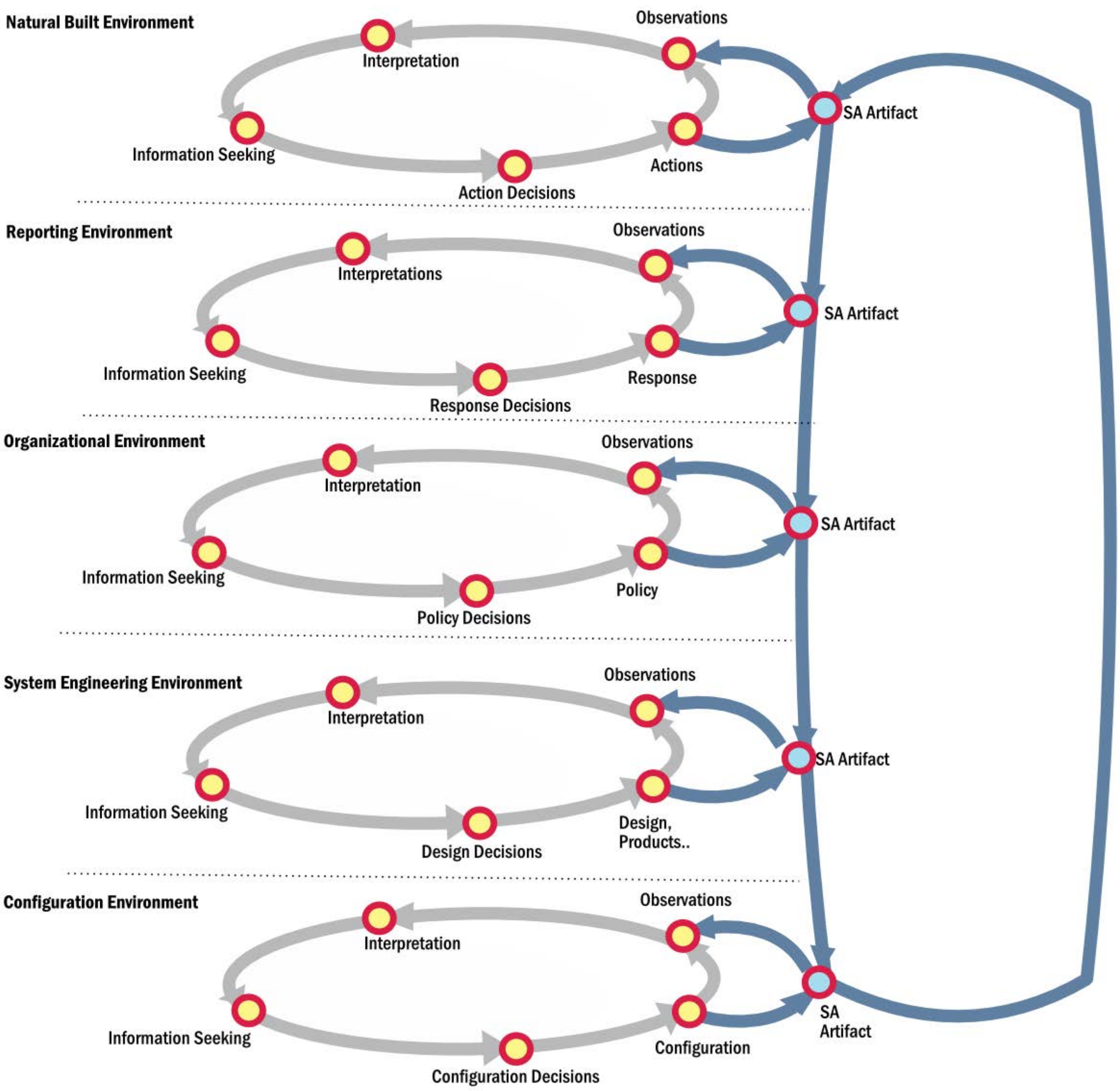

Figure 5.3: Cross Environmental Traceability with SAUP

Update Frequency: Update frequency provides estimate of scale in time when the architecture product is updated and how states are reflected in the update frequency.

Visualization Strategy: Visualization strategy is useful in selection process of the user, when interactive tree based exploration is provided to the user to build the query, provide inputs, or select instances. Visualization also supports user in display of the results. Rendering of the result set can be in the list form, map form or any other for determined suitable by the given rules.

Impact Foot Print: Impact footprint is the scope defined using three dimensions. Temporal dimension is useful in specifying temporal footprint of the artifact- the time period up to which the artifact is going to be useful. Spatial dimension is useful in specifying spatial footprint stating the estimated area served by the product. Conceptual dimension specifies the level of granularity in conceptual hierarchy used in creation of artifact from available concept tree in KB.

5.2.4 DoDAF Artifacts for SA

DoDAF is a viewpoint oriented architectural framework enabling rich representation. Identification and explanation of data elements, attributes and explanation is provided that allows creation of comprehensive views of the system. A typical DoDAF architectural product consists of Product Definition, Product Purpose, Product Detailed Description, UML Representation and Data Elements definitions. Among these common elements, the Data Element definition provides comprehensive guidance of various elements and attributes handled by the architectural products. The data element is identified for various attributes and explanation of each of them. The data elements in a typical architectural product is classified a Non-Graphical Types, Graphical arrow types, referred types and relationships. This detailed characterization of concepts improves visualization of situation. For example, depicting the information dependence of one operational unit to another is a common requirement. To represent such information flow DoDAF introduces *Needline* -a data element. Needline is depicted as graphical arrow type data element. Similarly, various provisions of DoDAF is useful in depicting information flow, information needs, and information processing blocks that contribute to situation awareness requirement of stakeholders.

DoDAF architectural products are classified in Overview products, Operational Views, System Views and Technology View. All the four products are found useful for Situation Awareness. Yet, they needs to be appropriately altered to meet the special requirements identified above.

DoDAF Overview Products are Useful in SA. DoDAF recommends two overview products defined in Table 5.2 namely Overview and Summary and Integrated Dictionary. The Overview and summary provide introduction to the purpose of architectural framework that is required for overview and starting point for stakeholders. Integrated Dictionary is difficult to manage in conventional terms as the large-scale collaboration spanning various domains may result in large domain specific concepts that are contributed and updated continuously. Hence, it is recommended to be in the form of mapped Ontology. DoDAF Operational Views (OV) provides

Name	Instance
Overview and Summary	No
Integrated Dictionary	concept: All

Table 5.2: DoDAF All View

useful information towards situation awareness. The operational view architectural products recommended in DoDAF provide detail about how operation is carried out among functional units of the organization. Table 5.3 includes the Operational View products defined by DoDAF. Operational Views provide important representation about role played by individual nodes,

Architectural Product	Instance
High Level Operational Graphic	isActiveNode(?x)
Operational Node Connectivity Description	hasNeedline(?x,?y)
Operational Information Exchange Matrix	exchgInfo(?x,?y)
Organizational Relationship Chart	coordinates(?x,?y)
Op. Activity Model	hasProcess(?x)
Op. Activities Sequences and Threads	-
Op. Rule Model	Rules
Op. State Transition Description	Transitedfrom(one, two)
Op. Event Sate Description	Event(concept, time)
Logical Data Model	SchemaProp(Org Entity)

Table 5.3: DoDAF Operational view

operational units and systems. This is important in situation awareness to establish complex interdependence among instances. Operational Rules Model provides rules that must be followed by various operational nodes. Various state transitions experienced by the operational nodes and characteristic of events are recommended. These facts are part of activity model as discussed in Chapter 3.

It is evident that number of nodes that can participate in typical domain application scenario can be many. The state transitions and event detection that are required to be defined for each of them should be done by collaborators. Similarly, the rules should be appropriately encoded by collaborators. In these tasks, consistent representation is required to make it useful, reusable and interoperable. Textual capturing is not suitable. Hence, the proposed approach allows capturing as extension to modeling process described in Chapter 3. The attributes of SA architectural product as introduced in 5.2.3 allows required representation. The rule model can be defined in KB were concepts and other rules are already defined. The query that generate the architectural views are also indicated in Table 5.3.

Name	**Instance**
Systems Interface Description	deployedIn(System, Node)
Sys. Communication Description	Comm System, Link, Network
Sys. Systems Matrix	hasInterface(?s, ?y)
Sys. Functionality Description	provideFunc(?s)
Oper. Activity to Sys. Function T. M.	hasSysFunc(?a, ?sf)
Sys. Data Exchange Matrix	ExchgInfo(Sys1, Sys2)
Sys. Performance Parameter Matrix	System Performance Log
Sys. Evolution Description	Version(System)
Sys. Technology Forecast	TechForecast(System,Time)
Sys. Func. Sequence and threads	No
Sys. Rules Model	rules: All
Sys. State Transition Description	State(System,Time)
Sys. Event Trace Description	Event(System,Time)
Physical Schema	SchemaProp(System)

Table 5.4: DoDAF System view

The operational view covered organizational rules, operating procedures and roles, how they can be realized in the system environment is independent. DoDAF recommends System View architectural product to capture this system specific aspects. Table 5.4 enumerates system views recommended. Addressing software component and information exchange among them, the system view is more technical version of the same views that are abstractly discussed in OV. System resulting from collaboration is expected to continuously evolve. As a part

Name	**Instance**
Tech. Standards Profile	Standard(ServiceArea)
Tech. Standards Forecast	StandardRelease(ServiceArea,Time)

Table 5.5: DoDAF Technology View

of this evolution, system components are upgraded to meet newly identified requirements, to incorporate newly available technology or research outcome, to incorporate new standards or other such reasons. As availability of new requirements, standards, research or technologies are recognized; their incorporation needs to be planned. The Technology View of DoDAF addresses this aspect by proposing two architectural products. Similar to other cases, creation of such product can not be done by single person at single node but created as continual contribution of multiple stakeholders. These contributors keep track of the relevant advancements, and contribute their observations. Hence, situation awareness strategy for building architectural framework recommends this as a result of multiple assertions made in KB. Creation of this view can be done on demand by queries defined in Table 5.5.

5.2.5 Service Artifacts for SA

Services are important aspects of the architecture. They expose data or functionality is standards form. Creation of services and governance thereof provides critical input to the process. Architectural frameworks should appropriately include service. Service Oriented Modeling and Analysis (SOMA) is one such approach [88].

Name	**Instance**
Service Specification Diagram	hasSpec(Service)
Service Consumer Specification Diagram	hasSrvConSp(Service)
Service Provider Specification Diagram	hasSrvProSp(Service)
Information Type Diagram	hasInfoType(Service)
SOA Structure Diagram	hasSOAStru(Service)
Service Interation Diagram	hasSrvInt(Service)

Table 5.6: SOMA View

5.2.6 ToGAF Artifacts for SA

ToGAF architectural products are descriptive in nature. Some of them can be mapped as identical to DoDAF artifacts. Business architecture in ToGAF is comparable to Operational View in DoDAF. Information System Architecture is comparable to System View, and similarly, Standards view of the technology architecture in ToGAF is comparable to Technical view of DoDAF.

Yet, ToGAF provide unique features that are also useful in proposed situation awareness process. The concept of *Enterprise Continuum* introduced by ToGAF for realizing the repository of reusable architectural products, software components, assets, solutions and related artifacts is one such example. ToGAF provide comprehensive guidance for building architectural framework as opposed to comprehensive views recommended by DoDAF. Hence, As DoDAF architectural products are selected for SA, the guidance from the ToGAF is recommended to be utilized along with the framework development process. A typical ToGAF architectural product is recommended with Elements like Enterprise Continuum, Gap Analysis, Inputs, Steps and Outputs. ToGAF suggests architectural board as important role and suggest detail procedures. To address architectural issues it provides comprehensive collection of checklists to facilitate the process. These architecture-related checklists are incorporated as guidance in SAUP. The ToGAF recommends following architectural checklists:

Component	Description	Proposed Use
Enterprise Continuum	Availability of resources	Provisioning of work products
Gap Analysis	Strategy to find shortfalls	Creation of Matrices
Inputs	Required artifacts	Establish Dependence
Steps	Steps for building	Suggested Guidance
Outputs	Generated artifacts	Establish Dependence

Table 5.7: ToGAF Description Template

	Recommended Architectural Checklists in ToGAF
1	Hardware and operating system
2	Software service and middleware
3	Applications
4	Information management
5	Security
6	System management
7	System engineering / Overall architecture
8	System engineering / Methods and tools

Table 5.8: ToGAF Checklists

5.2.7 Novel Architectural Products

Architectural Products mentioned in the previous section are based on the conventional approach and are not sufficiently equipped to support situational method engineering. For the task of building situation aware process, a few novel products are added. A brief description of each is provided below:

Organizational Knowledge base URL View (SAV-1)

The situation awareness is achieved with proper representation of KB including content regarding organizations, their goals, and domain knowledge from relevant domain, technology, standards and heuristics that affect the decisions during the organizational process. Once the knowledge representation and mapping is over, the organization can publish a knowledge URL where all concepts and rules are made available consistently for further information processing tasks.

Architectural Product Name: Organizational KB URL View

Related Role(s): Organizations that announces new URL configuration. Configures the required enterprise continuum and others as per the requirements.

Responsibility: Organization is responsible for information management to be carried out in jurisdiction

States: Planned, Created, Announced, Updated, Expired, Suspended, Merged.

Creation Logic: This architectural product is generated on demand. A query to the configuration reveals the available and configured all URL from the repository. The instance can be created with a semantic query `SAURL(Organization,URL)`

Update Frequency: The organization URL creation in possible in the scale of months, overlapping events can be handled in the same URL.

Visualization Strategy: Single organization may have created multiple configuration and hence, for each of them, unique URLs may be published. There can be many such configurations available in the given area. Hence, visualization strategy should support rendering all URLs published by the organizations in given UoD.

Impact Foot Print: In spatial dimension, it is the estimated boundary of the collaborative agency for the response work. A large-scale collaboration may have many such URLs.

SA Role Product Matrix (SAV-2)

This is an important matrix that determines various SA Roles that are being introduced in each stakeholder environments. The details regarding the role can be retrieved from the knowledge base URL product. The prime purpose of this matrix is to define association of every SA architecture product with one or more users defined in the SA namespace.

Architectural Product: Role product matrix to make sure every possible role is having product for situation awareness.

Related Role(s): SA Process engineer is the role responsible for creating process components

Responsibility: Responsible for creating provisions for every role identified in SAUP Process Configuration

States: Product is identified, created, Under Utilization, Updated, Purged etc. depending up on the status of the product.

Creation Logic: It can be generated statically. This can also be done by semantic query provided interactively. Instances are retrieved a query `Creates(Role, Product)` for given selection.

Update Frequency: The update frequency is very low. The products are changed only when process is updated. It can be included in next version of the process. In product content is also static.

Visualization Strategy: Selection from process to role set to individual role. Display a few products per role only. No map view possible as there are no spatial or temporal attributes for the given entities.

Impact Foot Print: This product has global footprint where process is useful. Regarding the temporal dimension, the version is useful in the timescale of few months. Conceptual is fixed depending upon the context used in process.

Other Architectural Products

Apart from the two SAAF architecture product described in previous section, other architectural products provide important feature for brining situation awareness capability. These products are targeted at various role set identified in SAUP, and provide traceability to their work products. Table 5.10 indicates the Name and abbreviation of SAAF products. Table 5.10 represents the query associated with each product that allows creation of the product on the runtime.

Name	ID
Organizational KB URL	SAV-1
SA Role Product Matrix	SAV-2
Information Need Component Traceability Matrix	SAV-3
Information Need Service Traceability Matrix	SAV-4
Information Need Data Traceability Matrix	SAV-5
Information Need ETL Traceability Matrix	SAV-6
Information Need Pattern Matrix	SAV-7
Information Need s VO Matrix	SAV-8
Information Need Coverage Matrix	SAV-9
Need Response Matrix	SAV-10
Artifact/WP/Design Decision To Standard Matrix	SAV-11
Information Need Research Matrix	SAV-12
Policy Resource Action Utilization Matrix	SAV-13
Information Need to system Matrix	SAV-14
Organizational Goals to SA Artifact Matrix	SAV-15
Organizational Need to SA Artifact Matrix	SAV-16
Situation Awareness Ontology View	SAK-1
Situation Awareness Ontology Mapping View	SAK-2
Situation Awareness Report	SR-1
Situation Awareness Process	SP-1

Table 5.9: SAUP Products

SA Information Need-Component Matrix (SAV-3) The proposed unified process employs the method of determining the instantaneous information need based on the mapping. Thus for each inferred information need, a component must be identified that can achieve the required task.

SA Information Need-Service Matrix (SAV-4) The information need can be fulfilled with each functionality exposed as a service. This matrix reveals how various services are traced to identify information need. This architectural product is also helpful in determining the coverage of the services. It may reveal the information need for which services are not defined. This product is mainly useful for SA Service Provider Role.

SA Information Need-Data Matrix (SAV-5) The service provides mechanism for accessing, processing, handling and displaying the data, but availability of data is an important matter. This architectural product allows identification of need vs. availability of sources that provide data at given spatial and temporal granularity.

SA Information Need-ETL Matrix (SAV-6) The raw data at the most granular level is collected on field. But users at different level of an execution center requires different level of granularity in data. The aggregation of miniature data can be achieved with specific function. For example some values can be directly added for achieving aggregates whereas to aggregate properties like climatic conditions; a kind of average can be used. Thus depending upon the property being observed, the rule will define specification for aggregation at next level. This architectural product defines information need for a given role at specific granularity to an ETL specification.

SA Information Need-MoM Pattern Matrix (SAV-7) The data is collected in the forms of attributes of an entity. The Message Oriented Middleware (MoM) used for collaborative environment is capable of defining data type channel according to specific type of attributes inferred in information need document. Creation and destruction of messaging endpoints will be based on inferred time period. Hence, this architectural product defines mapping among information need and the instances of relevant MoM Patterns.

SA Information Need-VO Matrix (SAV-8) The messaging related patterns considered are not limited only to MoM, but they must also be mapped to actual nodes where a particular type of user will produce and consume the message. These nodes are in effect members of the VO. They are constantly polled for availability. They are provided appropriate security certificates and they all use required services of the grid. This matrix maps the nodes onto the type of data they consume or generate.

SA Information Need-Coverage Matrix (SAV-9) This architectural product provides coverage analysis to the SA Configurator Role, stating the traceability among Information Need vs. Service and Component instances.

SA Need-Response Matrix (SAV-10) The need is determined by the knowledge base in the form of a set of actions, resources, environment that should be carried out or facilitates in response to the current situation. This architectural product provides traceability amongst a need to the actual response in answer to the identified need. This product is important in avoiding duplication of effort, idling of resources and other allocation related issues which are generally faced in dynamic environments.

Artifact-Standard Matrix (SAV-11) The Technical Standards View Products of DoDAF provides snapshot of standards adhered to while developing a system architecture. Yet this is not made available in real time to the standardization related stakeholders. Artifact to standard metric reveals the specific purpose for which the standard is utilized. The standardization authorities can consistently update and review the standards in the knowledge base and monitor the performance of the standards. This architectural product is useful in a scenario if during a configuration phase; use of certain standards creates additional issues that need to be communicated to the standardization authorities. Since the participation in standardization process is voluntary, the organizations identifying issues in this product will be encouraged to participate in standardization process.

Information Need-Research Matrix (SAV-12) The recent incidents like avian flu, bioterrorism and other such uncommon situations revel that organization as well as research community may have very little understanding or formal knowledge about handling such events. This architectural product defines the gap in scientific understanding. And as soon as the gap is filled; it can be instantaneously reflected in the knowledge base.

Policy/Resource-Action/Utilization Matrix (SAV-13) Each organization has some set policies for allocation and utilization or resources. This matrix reveals if for a given policy, the organization has enough resources available. Thus the resources gaps, or updates required in the policy itself can be determined.

Needline-System Matrix (SAV-14) The concept of a needline as defined in DoDAF as a static specification of information exchange among entities participating in information system. For SA systems, where information need dynamically changes among the members of the VO, static representation is not sufficient. Determination of needlines on a given instance in dynamic environment can be made possible with inference in knowledge base. The matrix provides traceability and visualization of available grid nodes, role being played by them and corresponding the needline supported by them.

Organization Goals-SA Artifact Matrix (SAV-15) The SA Manager is the interface between organization and the SA Process engineer. The dynamically changing organization structures, policies, and decisions for response are communicated to SA process engineer through the SA manager. Hence, this architectural product is used for the manager to track how the changes are reflected in the present configuration.

Organization Need-SA Artifact (SAV-16) This architectural product is for a reviewer who can do coverage analysis regarding the implied organization needs and how that is matched by the SA Artifacts.

ID	**Query**
SAV-1	SAUrl(Organization, URL)
SAV-2	Creates(Role, Product)
SAV-3	Providedby(InformationNeed, Component)
SAV-4	Providedby(InformationNeed, Service)
SAV-5	Providedby(InformationNeed, DataSource)
SAV-6	Providedby(InformationNeed, ETL)
SAV-7	Providedby(InformationNeed, MoM Pattern)
SAV-8	Providedby(InformationNeed, VOResource)
SAV-9	Providedby(Need, Coverage)
SAV-10	Providedby(Need, Responce)
SAV-11	Providedby(InformationNeed, Component)
SAV-12	Providedby(Interpretation, Research)
SAV-13	IdentifiedBy(ActionRequirement, Policy)
SAV-14	SatisfiedBy(Needline, System)
SAV-15	SatisfiedBy(Goals, System)
SAV-16	SatisfiedBy(OrgNeeds, System)
SAK-1	All Concepts
SAK-2	Mapping Template
SR-1	Report Template
SP-1	-

Table 5.10: SAUP Products: With Creation Logic

5.3 Situation Awareness Features

The proposed situation awareness architectural framework introduced novel strategy for building architectural products. Architecture products adopted from existing frameworks and those introduced by the proposal were briefly discussed. They are expected to enhance situation awareness capability with following features.

- Establishes interdependence by breaking down one work product as collection of individual outcomes.
- Establishes dynamism by breaking and tracking individual components
- Gap Analysis based work products
- Individuals select desired tasks, utilize guidance, and deliver appropriately
- Detection of architectural components on the run-time as opposed to single site architectural documents
- Possible to integrate with change management, bug tracking, portfolio matching and other aspects.

Department of Defense Architecture Framework (DoDAF) provides comprehensive set of architectural products for building information systems. The conventional approach for building DoDAF product and the proposed approach using SAAF are compared in Table 5.11. It must be noticed that SAAF proposal includes DoDAF architectural products, as they provide comprehensive reference for creation of the same. The features are compared for the creation strategy for DoDAF and not the content.

Provisions/Approach	**DoDAF**	**SA AF**
Scope	Generic	Situation Awareness
Focus	Building	Configuration over Core
Style	Generic	SCSS
Roles	Generic	Specific Roles
Process Guidance	DoDAF Plug-in	SA Plug-in
Matrix	Architecture level	Configuration Level
Matrix Strategy	Table	Ontology Based
Terminology Strategy	Text Based	Ontology Based
Event Engine	Not Defined	Implements MoM Patterns
Architecture	Monolithic	Service Based
SA Capability	Not Defined	SA Core
Separation of Concern	Not Defined	Well Defined
Granularity	Not Defined	Well Defined

Table 5.11: DoDAF and SAAF Comparison

5.4 Summary

This chapter introduced *Architecture Framework* as an effective mechanism in bringing situation awareness to various roles in different stakeholder environments. The philosophy of architecture framework was introduced along with introduction to existing proposals. These proposals provided comprehensive views of the system architecture, thereby providing important inputs to the architects. Here, in case of situation awareness domain, the architectural framework is accepted as one of the important artifacts instrumental in meeting information needs of the stakeholders. As, the requirement identification in complex dynamical system is complicated to capture, the role-specific situation awareness is established in the form of traceability and gap analysis. Gap analysis provides continuous identification of required vs. available work-products, thereby providing scope of work to be done. Similarly, once the work is done and component is delivered, the same mechanism is useful in tracking its usage in different stakeholder environments. Hence, the proposed architectural products provide various views in order to establish traceability across environments. Comprehensive architecture framework like DoDAF was found completely useful in this regard. In addition to that, novel architectural products were proposed to meet situation awareness requirements. The dynamic generation and access of these architectural products was discussed in reference to situation awareness requirement of various stakeholders in complex dynamical systems.

Chapter 6

Situation Awareness System Architecture

Situation awareness and related concepts defined in Chapter 2 provided characterization of information need. Chapter 3 introduced modeling strategy that is required to meet the information need. Modeling entities and process along with their characteristics with specific rules for defining attributes results in creation of a knowledge base. These comprehensive concepts are taken as basis for creating instances in given UoD. Hosting of knowledge base can provide single point of reference. From large number of concepts and instances created based on them, query mechanism is needed to retrieve required instances. For newly created instances, inferencing is employed to identify its membership to other named classes.

Among the attributes, spatial footprint is introduced as absolute spatial coordinates or an area bounded by such coordinates that occupy space in UoD. Based on the spatial footprint, the instance can be identified to stand in spatial relationship with other instances. This may have implications in information management strategy. Similar to this, the temporal footprint establishes the temporal coordinates of the instance, and in the same manner it can be established to stand in relation with itself or others. Based on identified spatial, temporal and conceptual relations of given entities, it is possible to depict situation and determine events in given UoD. Events are identified from asserted facts, as critical state change in given situation that requires some action or response from appropriate actors. The processing of assertions, detection of events, detection of actions and notification to appropriate actors requires series of information processing tasks. This task being distributed in nature, it cannot be realized with single system. Depending up on the scale of the UoD and context, multiple collaborative systems are required.

6.1 Requirements

From the discussion so far, the required solution emerges to be a system that supports modeling, instantiation, processing, and handling of information. This is done in collaborative manner where various users are contributing. This is based on existing functionality yet; present system cannot provide end user application. Hence, it is emerging as domain specific middleware that allows domain specific common functionality to the users. The awareness is provided in order to trigger tasks in various stakeholder environments.

6.1.1 Communication Requirements

In view of the above, following communications requirements are identified for situation awareness systems[89].

Communicating Entities The task is to identify the participants in communication. The entity can be a human observer, a sensor or a system. In case of sensor, a sensor can report occurrence of event to an agent. The agent may react by sending command messages to actuators. From disaster management perspective, a person will communicate to another person reporting the occurrence. The receiver will consult to an agent to determine potential outcome of the event. To assess the effects of an event, the agent may notify the actors who can report status of various observables. This will again come back to the system and also trigger a job that will keep on probing the identified actors, and continue analysis and report. The storing of continuous data is performed on storage devices. The jobs and the resource also will act as source of event, informing their status. Thus both, the human actors and computer agents can be the communicating entities.

Communication Patterns Communication patterns are to be realized as discussed in information processing model.

User-System User-System communication takes place when user first starts the communication with the system. The determination of users' credential and allocation of resources are performed by the system on the basis of the message content. Once user has become a member of an EOC with a specific role, he then receives control messages or Information Request messages generated by the system for further response.

System-System System-System communication takes place when a new resource is joining the EOC. System then determines its membership to EOC and polls at a desired duration to check the availability. The system may also be set-up to receive notification when specific state is achieved.

User-User User-User communication takes place when the explicit suggestion for taking necessary action is required. A user can be notified by the higher authorities for alteration in structure of the EOC. This User-User communication will trigger processes that will result in logical partitioning of EOC resources. Similarly direct commands issued by higher authorities can be communicated without any processing to the specific actors.

Subscriptions The state change may take place at the actor level, resources level or process level. When such state-change occurs, appropriate actors should be notified. The subscriber can be a person or a process. With dynamically altering situation, such alternation can be countably many. Hence, subscriptions should be appropriately managed.

Interaction Patterns Interaction patterns can be synchronous or asynchronous, depending upon the information requirement and type of messaging entities. Subscription to specific topics or topic tree of relevant events can be suitable in a Publish-Subscribe scenario. The control messages being exchanged amongst services may utilize message queues. The identification of appropriate interaction patterns is based on specific communication requirements of messaging entities[13].

6.1.2 Middleware Requirements

Middleware is defined [90] as a class of software that are employed to address complexity and heterogeneity that is characteristic to the distributed systems. It is realized as a software component providing a layer of abstraction in the middle of operating system and the client application. A particular middleware provide specific functionality commonly required by applications. In this manner, it can be seen as reusable software components that can further be utilized by application developers. With the advancement in technology the layers providing common programming abstractions have increased and middleware can be found to have many layers. Schmidt and Buschmann [91] suggested Host infrastructure Middleware, Distribution middleware, Common middleware services and domain specific middleware services as distinct layers among middleware. Such dynamic nature of the resulting system poses additional requirements on communication middleware.

Identification of Messaging Endpoints As situation unfolds, availability of some of the planned entities and resources may possibly be affected. In response, the volunteers and donated resources join the response work. According to the role of the voluntarily joining actors, they should be mapped either as information sources or sink. Upon detection of information requirement, these new sources can be approached with information request, as well as with other notification to adjust their response action. Likewise, newly joining computing and communication resources need to be registered in the resource pool, and should be used for collecting, processing, managing and delivering the collected information.

The challenge is to determine the limited time period for which these entities are actively involved in communication. A disastrous event may be followed by secondary events. It is also likely that while the effects of such complex events are continued, new events may take place. Hence the lifespan of each involved entities in response-work must be determined.

Lifespan of Messaging Endpoints The EOC may be responding to multiple co-occurring or separate disasters overlapping in time. The time period of response to such event may overlap. In this scenario, the life-cycle of *Event Sources* and *Event Listeners* become a critical design issue.

Figure 1.3 provides a comparative view of lifespan of various concepts related to response work. During a lifespan of EOC, it is legally responsible to respond to any kind of events that may take place in the given region. For this the EOC acquires human and material resources targeted for specific type of response-work. During the lifespan of each individual planned resource, they may have to respond to multiple events. As the situation unfolds, an actual response is expected to contain dynamic set of planned and unplanned entities. The design of the communication middleware therefore must integrate life-cycle management of various messaging endpoints.

Rules Rules play a vital role in determination of communicating entities. Rules are required for:

1. ***Handling Event Detection :*** An event can take place in various domains. Hence strategy for detection is also targeting specific features of the domain. The rules provide logical framework to determine the event detection process.

2. ***Handling Actors/Resources :*** The unplanned actors and resources join the EOC with no predefined agenda. In some cases, it is possible that actors/resource starts playing a specific role in the situation and that is later reported to the system. The appropriate

role with explicit or implicit interest in the instantaneous situation should be determined using rules.

3. ***Handling Subscriptions :*** Identification of roles in EOC also facilitates determination of new entities as messaging endpoints. The rules can govern the management of their subscriptions in Communication Middleware.

4. ***Handling Storage and Data Management :*** The unit information communicated among endpoints is to be utilized to build EOC level snap-shot of the situation. A rule governed strategy to maintain the snapshot in persistent storage is needed.

5. ***Managing Life Cycle of Entities :*** Lifespan of messaging endpoints and related sub components should also be managed by the communication middleware with changing membership to the EOC.

Handling User Users are considered both- source and sink- of the EOC messaging system. Users or actors can be of two types. A person, who is assigned a specific responsibility during planning process, is generally trained and made aware of standard operating procedures with access to various resources. Thus user can be expected to be available consistently during the tenure of service. Another type of user/actor are those who volunteer their services to EOC. These volunteers may or may not be having access to resources. Their availability is generally expected to be short-lived, and hence they cannot be expected to be consistently available throughout the event response.

In such case, the EOC can have multiple programmatic abstractions of user entities as described in Table 6.1. The EOC member with specific responsibility is defined as an instance of appropriate class in the EOC Ontology. All such instances cannot assumed to be available as the discussions above. Hence the Message Oriented Middleware (MoM) should poll those instances that are absent, and those unplanned that have joined.

Domain	**Role of a User**
Virtual Organization	Member of VO to access available resources and services and everything else that is useful in carrying out assumed role
Knowledge Base	Instance of specific concept(s)
Communication	Message Source or Sink

Table 6.1: User Forms

Handling Data Sources Sensors and human observations can act as data source, providing relevant information regarding the event. The programmatic abstraction of data source must be guided by the events that are reported by the sources.

Handling Data Resources Data Resources provide means to retain the information regarding the situation and events in various stages of their life cycle. The data life cycle include raw information directly collected from the data sources, the cleaned state, processed state or analyzed state. What particular state during the data life cycle is to be used by eventing application is a critical design issue. The raw data are the actual facts reported by human or sensor sources, but their count and size of multiple snapshots can be difficult to handle.

Shared Knowledgebase The data resources can be used to store event related data in various state of their life cycle. The EOC is expected to handle all type of events that are continuously taking place. The EOC banks upon the same pool of resources and entities for in responding to various events. Thus achieving the all-hazard approach is quite difficult to capture in database schema.

Interaction Patterns Following issues are to be addressed to strategize the interaction among the entities:

Managing Life Cycle of Entities Interaction patterns that manage life cycle of messaging endpoints should support reliable messaging among services.

Managing Processes The appropriate pattern for communicating control messages among system should be determined by application rules. Depending upon the case FIFO or LIFO property must be appropriately selected in communication channels.

Information Collection Pattern The Information Collection Pattern is based on the *Request-Reply* pattern[13]. The modification is required in the request and response message as it is expected to handle human observation.

6.1.3 CASE Requirements

Chapter 4 introduced a unified process created to support information needs of the actors in carrying out assumed roles in different execution environments. Chapter 5 introduced architectural products to meet information needs of the users in software development environment by establishing coverage and gap analysis. Such requirements are generally met by class of application known as Computer Aided Software Engineering (CASE). While many popular CASE tools provide common provisions, certain additional features must be achieved for complex dynamical systems[92].

Supporting Separation of Concern As discussed earlier, the actors associated in the process may have completely orthogonal concerns and hence, tooling must provide support for customized experience of each actor playing one or more roles in a given instance. This need becomes more relevant when actors just provide their skill-set and the CASE tool thereafter should be able to infer what possible roles they can play.

Event Driven Targeted systems are reactive in nature; and the case tool must support event-based triggers. The RUP may specify if a particular activity is event triggered or not, but the mechanism of detecting the event is not known. The events are also considered to be delivered not to the explicit subscriptions but the general roles. For example, set of trained volunteers are capable of providing many ICT related services, when specific type of event happens how they are informed for the requirement of the job. As the RUP further allows specification of skills, this information can be used for identification of the proper recipient of the event notification. Hence the CASE tool must be able to collect profiles of the potential members, and the event detection mechanism must be able to identify appropriate recipient and be able to deliver message to them over collaborative environment.

Dynamism in Organization In situations where organizations are responding to emergency situations or crisis, the decision for setting up an Emergency Operations Center (EOC), and decisions related to allocation of resources keep on changing as the event unfolds. As exact boundary of emergency is discovered, organizations may have to come together at federal or international level resulting in change often in drastic manners. The resources in terms of skilled manpower and ICT infrastructure may change with volunteers and donated resources that need to be incorporated instantaneously. The CASE tool must support the required level of dynamism experienced by the organizations during the events.

Estimation of Efforts For any organization, the volume of work should be determined. There can be some systems already in place, which can be integrated with the planned information system. The knowledge representation needs to be incorporated also in the estimation that allows the agency to identify the items they are supposed to make provision for.

Knowledge Representation Consistent flow of information in situation awareness system is only possible if consistent representation of knowledge is adhered to through out the process life-cycle. It also allows reusability of the domain knowledge. The spatial, temporal and semantic reasoning is very critical for the success of process, so the knowledge representation that can allow such reasoning is a basic necessity.
While knowledge representation and reasoning allows support for spatial and temporal relationship, the architectural products should be able to use this aspect. For example various grid nodes, which are part of VO, can be scattered across a larger area; so the decisions regarding data regionalization and other such provision requires spatial nature to be considered. The meta data is also having spatial elements in the schema, hence architectural product should be able to render instances and allow query based on spatial/temporal attributes.

From Monolithic to VO The collaborating teams should be considered as members of a virtual organization (VO). Like successful applications as bug tracking is handled in collaborative mode, the process can also be handled in similar manner.

Semantics based Traceability Traceability among architectural products provide basis for tracking the coverage of the effort. The software teams involved in SA is provided the tasks during the entire life cycle, and hence each and every part is interpreted as required due to some precursor. Thus traceability can actually be based on semantics and should be able to cover the entire set of the tasks.

Task Allotment Task allotment to volunteers or the team members can be done based on evaluating the skills. Once some volunteer defines that they will be continuing with the task, system should be able to take notice of the same. The status of ongoing task should be identifiable at any intermediate interval.

Visualization Complex level architectural approaches demand high degree of technical expertise for the user. How users will take up the task depends on how effectively it is provided with proper visualization. Monitoring of allotted work, overview of the process status; search of architectural components, the hierarchical view coverage and errors should be rendered to improve the quality of the development.

Standardization The standardization related concepts in the ontology suggest their applicability. Each artifact or work product should be traced to appropriate standards. Standardization traceability matrix not only supports the developers to consider standards, it also provides the specification that must be considered while developing given application.

Artifact Impact and Reusability A true test of proposed SME in SACore can be possible at global level adoption of such methodology for responding a real world crisis situation. There are already existing global initiatives like GDACS that allows alerts at global scale, but with limited amount of flexibility for customization and to suite the individual needs of organizations.

The SA life cycle demands the knowledge of when and how reusable artifacts can be published, discovered and utilized. There can be some artifacts that are relevant for some time only for very specific region and cannot be reused. The data collected the execution environment, system level tasks, organizational decisions and reviews fall in to this category. Some artifacts are local but valid for longer period of time and do not require frequent updates. For example, SA Configuration, organizational knowledge, mapping, transformation and organizational policies. SA review and SA management can be considered to be globally relevant but short term and needs to be verified by multiple implementations. The fourth quadruple contains the set of artifacts that can be reused. For example SA Process, Services, Components Standardizations and domain knowledge, can be created or incorporated by any SA Configuration and can be reused globally for long period of time.

6.2 Building Blocks

Requirements identified for realizing situation awareness system indicate few recurring themes that lead to the identification of basic building blocks for the system. One such functionality is information handling in conceptual dimension. This is identified as knowledge management. Representation, processing and reasoning of the knowledge being captured by the various roles and their actions. Physical and impact footprint are two important features for entities in the UoD. Apart from conceptual dimension, the attributes in temporal and spatial dimension allow additional reasoning capability for capturing complexity and dynamism. Information handling in spatial dimension by capturing, representation, query and analysis of spatial attributes of the entity is important building block. Handling temporal dimension of information is supported by both conceptual and spatial building blocks.

Apart from processing the information, transfer of information is important recurring requirement. They are identified as various information exchange patterns. This leads to the identification of communication building block. Vast amount information handled by the system requires appropriate database management functionality that can be achieved by multiple databases. Distributed database management is one building block that allows handling of underlying database at multiple physical instances. The handling and processing also includes features that allows analysis and reporting of continuously generated data.

Dynamic set of users, databases, required functionalities and systems that support information processing implies the requirement of computation. The computing system that allows processing of information over multiple underlying systems connected with network is required. Collectively, these dynamic sets of entities can be visualized to form a virtual organization. Handling and management of virtual organization by providing basic infrastructural services over heterogeneous components is therefore another important building block.

Figure 6.1 depicts identified building blocks. The required functionality in the discussion is depicted in the figure. Corresponding to each functionality information about realization of this functionality is achieved is also depicted as corresponding middleware. Figure 6.1 indicates appropriate middleware that fulfills the identified needs. Message Oriented Middleware (MOM)[13, 93, 12], Semantic Middleware[64], Grid Middleware[94], Data Middleware[75], and GI middleware[65, 66] are examples of popular and available middleware technologies exposing various required functionalities.

Figure 6.1: Building Blocks

6.3 Design Decisions

The identification of building blocks is an important advancement, yet it does not provide complete answer, as they must be appropriately integrated. With the integration of lower level enabling services is to be utilized to achieve higher-level domain functionalities. Hence, the integration should be based on architecture that can seamlessly interact with various services and provide the functionalities. This is achieved in the form of domain middleware components. While building these components, some design decisions are followed. Following discussion introduces some important design decision taken along with rationale for the same.

6.3.1 Semantic Web Technology for Knowledge Management

The requirement of knowledge management in introduced in the beginning of Chapter 3. Entity and processes identified along with their attributes must be represented and utilized for further reference in handling situation during the runtime. The concepts may be specific to the entities given in UoD. Depending upon the requirements, entities can be modeled from various domain viewpoints. Terminology of the domain therefore also be included in the representation. The rules, standard operating procedures and other guidelines that support decision-making process are also required for determining and monitoring actions in ongoing situation. Depending upon the involvement in various activities visible in the UoD, the users can be identified to play specific roles. Various environments in which users can take actions is discussed in Chapter 4. The outcome of their actions has impact on the action of others. Hence, the entities, interactions among them and their outcomes along with processes and conditions in UoD all contribute to the requirement of handling knowledge.

Topic	**Need**	**Shown as**
Representation Model (Sec 3.4)	Create KB	-
Representation Model (Sec 3.4)	host KB	-
Assertion Management (Sec 3.8.1)	`Query`	`T 7.1.1` in Figure 3.21

Table 6.2: Collaborative Knowledge Management Requirement

Section 3.8 introduced implementation algorithm that identifies how reference to the captured knowledge based is going to play important role in information processing. Table 6.2 enumerate a few example requirements identifying need of knowledge management activity required in proposed information management strategy.

Semantics for Knowledge Management Among other possible approaches, Semantic Web technology have proved useful for knowledge management [95]. From repository centric management of information to smaller information managing communities followed by recent trends in social computing, the knowledge content and its representation is growing at larger scales requiring dynamic view. Since it inception, the development of Semantic web technology have successfully demonstrated [96] its capabilities in meeting identified goals. This is supported by collaboration among related communities for developing theories, standards, tools and technologies that are instrumental in achieving the objectives.

Basic framework for handling semantics was already in place. Yet considering the growing information and its management, large-scale collaborations among participating communities are envisaged. One important enabling factor to support this effort is identified to standardize the handling of the knowledge being created. The Internet Engineering Task Force[1] and The World Wide Web Consortium[2] (W3C) standardization activities resulted in standards and specifications that enable consistent generation and handling of knowledge. Resource Description Framework (RDF) enabled triple based representation of resources exposed on the web. The growing uptake of standard resulted in repositories storing the knowledge captured in the form of triplets. This initiated technology and tooling support in handling repository. With growing needs and application scenarios, more expressive standard was required and hence the Web Ontology Language[97] (OWL) is introduced. Moving from meta data handling capability of RDF, the OWL allowed creation of Ontologies meeting knowledge handling requirement of various domains. The standardization process also addresses queries. As rules are characteristic to domain knowledge provision of rule is covered in Rule Interchange Format (RIF). These standards allow knowledge representation and reasoning capability. With available formal framework of theory, standards and technology support, the collaborative effort resulted in creation and sharing of Ontologies supporting reuse of captured knowledge.

Requirements to be met by Ontology The development of semantic web technology can meet the identified information management requirements. The required granularity of representation can be handled by appropriate level of conceptual hierarchies captured in the ontology. Various functions requiring relation among concepts and instances is realized with semantic queries. For given instance, membership to specific class based on the predefined rules can meet the need of inference support during the runtime. All these features allow building of application that is based on ontology. Event detection for rule is employed to realized distributed event based system [98]. Application logic is also successfully captured using this technology. It is also employed for realizing event driven systems [99].

Alterations required in Handling Ontologies While the practice of knowledge management is reported to be meeting the knowledge management need of real life applications, the complexity and dynamism still provide a difficult challenge. In present scenario, there can be many distinct domain of scientific and technological inquiry. Terminology, rules and other aspect of domain knowledge must be consistently captured. The allocation of task of capturing the domain knowledge and its continuous update is a difficult task allocation problem. One such approach is handling of smaller chunks of domain specific knowledge is restricted to individual ontology. And upon the requirements, they required Ontologies are integrated to meet the

[1] IETF Web Page: http://www.ietf.org/
[2] W3C Web Page: http://www.w3.org/

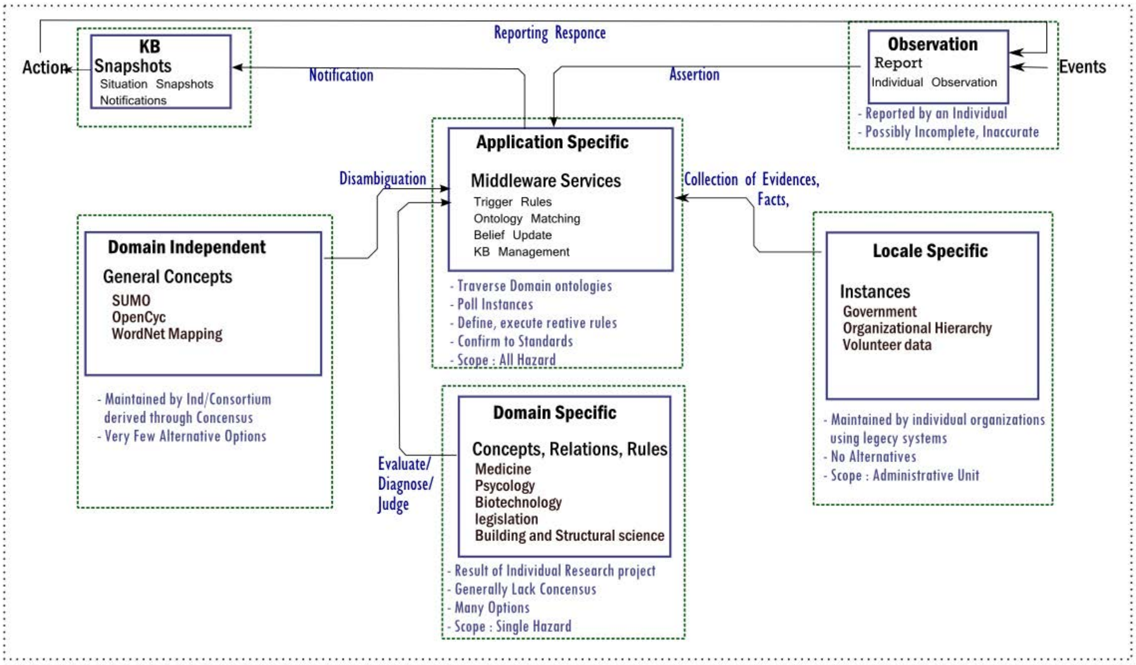

Figure 6.2: Ontology Mapping

coverage of knowledge representation. This integration is done using mapping. Some times, the concepts and not completely compatible and certain conversions are required before utilization of Ontologies. Mappings can be created and shared to support these features. Various levels of Ontologies can be identified. Based on the content and the nature of expertise in domain expert, four distinct type of ontology concepts identified [100]. Domain independent, domain specific, local specific and application specific as indicated in Figure 6.2.

The domain knowledge representation carried out by collaborating team of domain experts and knowledge engineers use desktop applications providing workbench for ontology representation. This knowledge based must be exposed in a manner that users can access it over open standards. This requires a semantic repository and that is exposed within a server based execution environment, such that the client can access it. Sesame[3] [101] provides an this functionality of publishing the Ontologies and allowing access through standard semantic web query languages. SPARQL [102] is utilized for querying the published ontology.

6.3.2 Human Observation as Source of Information

Information management methodology requires instantiation, monitoring and communication of various domain concepts, resources and process as captured by the modeling process defined in Chapter 3.

[3] Sesame Project: http://www.openrdf.org/

Topic	Need	Shown in
Assertion Management (Sec 3.8.1)	Input for Algo. 3.1	Figure 3.21

Table 6.3: Information Collection Requirement

Limitations of Sensor-Only Approaches Generally, approaches towards situation awareness mostly rely on the sensors technology [103] for their information requirements. Various sensors deployed in the area of interest plays the role of *data sources* that are frequently polled for tracking the change in observables. Other than the possibility to control location and adjustment in sensing frequency, these approaches do not provide the required flexibility to meet the information needs as described below.

Information Content The information from a deployed sensor network is almost fixed. For example, a sensor network deployed for tracking the climatic condition will provide information in form of reading of particular observables like temperature, pressure, wind velocity, wind direction etc in a predefined range. Information requirements beyond this will require to set-up another set of sensors.

Information Flow In case of sensor network, the information flow is in predefined pull or push modes. It is possible to send control messages that will change the behavior of the sensors to control the information flow. The information flow in case of EOC may vary to suit the role of actors. The human observer not only act as sink of control messages but also can act as subscriber to specific events published by other sources on the network.

Control Messages Implementation of control message for human subscribers helps achieving the rule driven information collection that is not possible in case of sensor assemblies/sensor networks. The rules defined in ontology suggest information need for a specific situation, at a given space and time granularity. It is also possible to generate message requesting the information update that is consistent with the ontology and the rules.

Topology The topology of a sensor network remains unchanged during the operations. The adjustments of node locations and the hierarchy are almost static. In case of command and control mechanisms, the topology and hierarchy of the information sources will be generally changed in the duration the operation persists.

In reference to the issues represented above changes required in sources of information. Human observations are appropriately handled to act as sources of information. The uncontrolled reporting from human observation may lead to semantic, syntactic heterogeneity, disambiguation and many other issues that is not suitable for the information management requirement. Hence the strategy must be defined that allows handling of human observation to provide reporting free from these errors.

6.3.3 Message Queue for Processing Task Lists

Automation in the process of information management result in generation of task lists. In Section 3.8 information management strategy is introduced that employed rule driven generation of task specification for various information processing requirements. As there can be many possible information processing requirements that can be identified on the run time, these algorithms allows identification of task parameters. The identification of task and execution of task requires separate handling strategy and have unique requirements. Identification of task is triggered by some events, and is involves reference to the application rules in detecting the task parameters. Where as execution of the determined task requires conversion of an abstract task specification in to a job specification that can be submitted to job execution environment. Job specification needs to be executed at determined time along with other required inputs and arguments.

In both cases, the continuously generated specifications need to be handled appropriately. Table 6.4 indicates various task lists generated in information management strategy recommended in Chapter 3.

Topic	**Form**	**As shown in**
Assertion Mgt. (Sec 3.8.1)	`PrimaryInfoNeedList` in Algo. (3.1)	`L 7.1.2` in Fig. 3.21
In. Need Deter. (Sec 3.8.2)	`CharConceptPropList` in Algo (3.2)	`L 7.2.2` in Fig. 3.23
Mssg Pat Gen (Sec 3.8.4)	`PatternMapList` in Algo. (3.4)	`L 7.4.1` in Fig. 3.26

Table 6.4: Queue Management Requirement

Message Queue Message queue is a distributed computing concept employed to efficiently process the jobs. Communication among collaborating components within distributed system is a common requirement like interaction among programs. This can be request of the task that can be responded back. There can be two possible ways of communication. When application requirement is such that calling procedure cannot execute further until is get the request served by the other component. For this it has to wait for reply thereby halts the execution. This is known as synchronous communication pattern. In many other scenarios, the requesting procedure may not require to wait and stop execution, as there is no input required from the receiver. This is known as Asynchronous communication. Message queue is example of asynchronous communication[104].

Alterations required in Message Queue Message queues are created generally with administration tool provided by the message queue technology. Some of them are created and handled programmatically. In each case, the administrator or the programmer creates and monitor the queue individually. In information management scenario in complex dynamical system, creation, monitoring and utilization of message queue should be handled automatically. Various parameters for generation of queue is determine on the runtime. This is driven by accessing the application specific rules for handling messaging patterns.

Utilization of Message Queue All lists in Information Processing Model is considered as queue. With the help of middleware services, their administration should be appropriately handled. Also, rule driven handling of patterns should be supported at higher-level domain middleware service.

6.3.4 GML as Application Schema

Application schema is an artifact established and utilized by application developers that provides conceptual schema of data handled by the application. In information management applications where communication of data is frequently carried out among vast application user base, the XML based approach for application schema is a common practice. All the XML based interchanges are done in reference to this shared and published application schema; therefore provide consistent handling and processing of information in distributed environment.

The information management strategy recommended for situation awareness must also subscribe to appropriate schema so that consistent processing and handling of XML based exchange can be consistently handled. The requirement of application schema can be traced to the information processing models discussed in Section 3.8. Table 6.5 enumerate few instances where use of application schema is required. `Encode` function specified in information need specification algorithm requires the encoding of the identified information need to be done according to a suitable application schema.

Topic	**Form**	**Shown in**
Info.Need Specification (Sec 3.8.3)	`Encode` in Algo (3.3)	`T 7.3.7` in Fig. 3.24
Message Pattern Gen. (Sec 3.8.4)	`Instantiate` in Algo. (3.4)	`T 7.4.3` in Fig. 3.26
ETL Sp. Gen (Sec 3.8.10)	`GenRepTaskSpec` in Algo (3.10)	Fig. 3.36

Table 6.5: Application Schema Requirement

Geographic Markup Language Geographic Markup Language (GML) is a comprehensive specification[105] recommended to build application schema for handling spatial and non spatial data. The GI applications typically require modeling, capturing, representing, processing, communication, publishing and storage of application specific features in given UoD. Many commercial and open source products handle such information in proprietary format making distributed information processing difficult challenge. The GML specification is initially devised to handle spatial and non spatial attributes of such features within vendor neutral open source framework to address interoperability issues.

GML schema specification introduces set of XML schema that provides standard vocabulary for handling geographical objects. The base schema provides the common minimum requirement of for handling features, and other optional schema can be incorporated to suit the application need. Table 6.6 enumerates few schema introduced in GML Specification. The utilization and example is given along the name of the provided schema. With adding application specific entities along with recommended GML schema, application schema can be derived for consistent use.

Schema	**Use**	**Specifying**
gmlBase	base which other GML objects are derived	objects, properties
feature	representing an object from UoD	objects, properties
coverage and grid	map spatiotemporal domain to attribute values	spatial data structure
dataQuality	Reporting accuracy or precision	Positional Accuracy
datums	relation of coordinate system to a reference	geodetic, temporal
direction	Describing Directions	Compass Point like SW
dynamicFeature	Representing Time varying properties	time slice, history, track
geometryPrimitives	Primitive geomatric objects	point, curve, surface
coordinateSystems	Mapping of coordiate with invarient properties	Cartesian, temporal, polar
observation	Modeling observation activity	Sensor, Target, result
measures	Specifying value of quantity with units	measurement
temporal	absolute, duration or ordinal temporal reference	Calendar, Clock
topology	representing spatial relationship	Node, Edge, Face
units	Units of Measure	Base Unit, Derived Unit

Table 6.6: Some GML Schema

Requirements met by GML As established by the list of schema and their utilization, the GML provide comprehensive standard based framework for handling entities in UoD. The entity model introduced in Section 3.2 required handling of footprints. These footprints are spatial, temporal and conceptual coverage of the given entity. GML is schema supports handling of spatial and temporal attributes. Apart from this, observation and measurement of instance are important aspects. Observation and measurement is also covered in the specification. Capturing dynamic features like event, tracking of object state, maintaining history etc. is also supported by the proposed schema.

Alterations required in Standard GML Handling While, GML specification comprehensively addresses the modeling concerns of potential applications, generally the utilization is limited to the basic elements. Generally in practice, the spatial data is created in proprietary form and later exported to GML format or hosted on standard services where it is handled in GML format. The alternation is required in practice to incorporate all applicable attributes offered by the GML specification. Secondly, GML creation is to be carried out not as export format but should be included in information processing strategies. For example, in case like `Encode` function defined in Algorithm 3.3 should be encoded in reference to GML based application schema. Similarly GML is also appropriate for creating `Data Type Channel` patterns that deals with spatio-temporal attribute collection from human and sensor observations. For handling aggregation in ETL, GML is useful means for storing intermediate and final representations.

Utilization of GML The proposed information management strategy relies on human observation as source of information. In this scenario, various inputs and observations are required from users, that are further processed and provided to appropriate users to aid their decision making process. Thus, collection of observed information must be handled consistently. The entities and processes holds attributes in spatial, temporal and semantic domain. Appropriate values should be supplied while creating a new instance or reporting monitoring status. These attributes are further processes for analysis hence should be consistently carried out. The Time coordinate should be collected with conformance to GML schema. Similarly reference to spatial location should be done appropriately. The observation and measurement also must be carried out as identified by the system. Apart from information collection, GML schema is utilized in handling messages and messaging patterns among collaborating services.

The GML specification is useful in creating information collection templates. GML `BasicTypes` schema introduces `gml:NullType`. It introduces a content model to indicated an absent value with one of the explanation like: inapplicable, missing, template, unknown and withheld. A `template` indicate value that will appear later. The information need specification indicates missing value of required object property. This should be observed and reported back to the system. From schema listed in Table 6.6 indicates possible candidates that can be useful in this case. The specification that represent part of a UoD is created as a *Coverage*. `gml:rangeSet` and `gml:domainSet` enable assignment of possible value range to specific domain point corre-

Domain Range Representation in GML

```
<RptClimateTemp>
<gml:domainSet>
<gml:MultiPoint srsName=...>
<gml:PointMember>
<gml:Point>
<gml:pos>12.64034 92.76855</gml:pos>
</gml:Point>
...
</gml:PointMember>
</gml:MultiPoint> </gml:domainSet>
<gml:rangeSet>
<gml:ValueArray>
<gml:ValueComponents>
...
<Temperature uom="urn:...">27</Temperature>
<Temperature uom="urn:...">28</Temperature>
<Temperature uom="urn:...">29</Temperature>
...
</gml:ValueComponents>
</gml:ValueArray>
</gml:rangeSet>
</RptClimateTemp>
```

Figure 6.3: Domain Range Template in GML

sponding to the UoD. The attribute value is also annotated with unit of measure. Thus, GML have sufficient provisions to enable information collection from human or sensor sources. Figure 6.3 represent an abstract representation of such template.

6.3.5 SOA as the distributed computing architecture

The nature of collaboration involves multiple systems spread across larger area. Each system may have unique specification, and hence, appropriate distributed computing strategy and information processing paradigm needs to be adopted to realize the required functionality over them. Sharing of data, functionality and capabilities are possible to achieve among systems.

SOA Increasing complexity in handling information in organization makes it a complex problem. It is recognized that conversion of large problem in to smaller manageable units separating the concern of the related entities provides enhanced capability of solving problem. Each unit identified corresponds to a smaller problem, is addressed with a logic easily defined, realized, executed and managed. From an organization's perspective, organizations try to decompose and expose these small executable components addressing unit functionality. From a single users perspective, user can utilize these units to use fulfill the required needs. When this principle is followed in architecting the system, the approach is known as Service Oriented Architecture (SOA).

These composable units are known as service [106] and exhibit following features. They demonstrate loose coupling, service contract, anatomy, abstraction, reusability, composability, discoverability and statelessness.

SOA introduces the concept of web service [107], [108]. The web service framework is characterized. Core building blocks of the architecture are Web services, service descriptions, and messages. The functionalities exposed by web services is discovered and utilized based on service descriptions provided in the standard format.

Standards and protocol are central to the web service architecture. The XML technology provides data representation and management features. The first generation of web service architecture was build using three core standards matching core requirements identified by the building blocks. For description of developed web service, Web Service Description language (WSDL) is defined. Simple Object Access Protocol (SOAP) is developed to support message exchange pattern required in accessing the services. Universal Discovery Description and Integration (UDDI) is required as registry of available services. In its second generation, more specification were developed meeting the specific needs. The consensus driven Standardization process carried out by consortia involved participation of vendors, technology provides, research and governmental organization. Organization for the Advancement of Structured Information Standards[4] (OASIS). These standards are based on web services, and hence they are benefiting from The Internet Engineering Task Force[5] and The World Wide Web Consortium[6] (W3C) hosted standardization process that have resulted multiple standards and specification. Each are based on the core standards and addresses specific aspect of web services upper level of the business logic. Arrangement of the standards organized according to their level and functionality is known as web service stack, representing available specifications [109, 110].

Data exposed with web service is particularly useful in disaster management scenario. The disaster alert services are capable to pin point the exact location of disasters along with the magnitude. This information must be utilized along with the domain specific rules that allow identification of impact footprint of the event. Based on the estimated spatial impact footprint the corresponding administrative blocks are identifies, that can further lead to identification or governmental organizations and officials that are legally responsible for response. The explanation provided a rule driven composition of web services to meet the user information requirements. Use of sensor web enablement is demonstrated in [111]. Orchestration of more web services is done with Business related standards in Web Service stack[112].

[4]OASIS Web Page: http://www.oasis-open.org
[5]IETF Web Page: http://www.ietf.org/
[6]W3C Web Page: http://www.w3.org/

Alterations required in using SOA As discussed in the principle, SOA recommends decomposing larger problems into smaller one and exposing the functionalities over standard, vendor neutral platform such that it can be composed and consumed to suit the requirement. For organization, exposing identified and known service is relatively easy task as technology and tooling support along with organizational knowledge helps realizing this task. Yet, from user perspective, the discovery of appropriate service and utilization to meet the processing need can be a difficult challenge based on quality and quantity of candidate services available. A more difficult problem arises to the organization whose goal is in ensuring that user needs are appropriately met by services exposed by organizations. Hence Analysis and design in SOA for them must include the dynamic set of users and their standing and potential future requirements. Addressing this issue, also help increase the unplanned reuse of service. The Situation Awareness Architectural Framework recommended in Chapter 5, address this issue and provide creation of architectural products that establishes coverage and gap analysis for service.

6.3.6 Grid for Building Virtual organization

Complex dynamical system is characterized with a dynamic set of entities playing various roles in given UoD. Their goal-oriented nature allows them to be considered for an organizational structure. Information management strategy is focused on this organization that is targeted at provision of information. This is to be supported by various systems used by the members. Information management strategy must therefore handle information over these collaborating systems. This nature of organization is identified as Virtual Organization.

Resource sharing and information processing introduces specific challenges in VO environment. Foster et al. [113] introduced specific issues regarding resource sharing. The availability of shared resources is subject to constrain identified deliberately or due the nature and characteristic of the resource. The sharing policies may result in dynamic changes in availability of shared resource, permissions and those who can access them. This requires continuous identification, characterization and monitoring of shared resource on the run time. Depending up on nature of functionality required by a VO member, it can be supported by multiple resources or services each may be hosted by unique providers having unique sharing policies. Hence, appropriate mechanism for coordinating required operations served by multiple providers must be realized. The same resource can be utilized for multiple operations within the purview of the sharing policies that govern the access.

Topic	Form	Shown in
ETL Sp. Gen (Sec 3.8.10)	`GenRepTaskSpec` in Algo (3.10)	Fig. 3.36
Data Prov. Gen (Sec 3.8.12)	`Generate` in Algo (3.12)	Fig. 3.38
DDM Gen (Sec 3.8.13)	`Encode` in Algo (3.13)	Fig. 3.40

Table 6.7: VO Resource Management Requirement

Grid Computing Grid computing have evolved as distributed computing platform suitable for meeting computational need of virtual organizations (VO)[110, 114]. Characteristics of grid [115] include collaboration, aggregation, virtualization, service orientation, heterogeneity, decentralized control, interoperability achieved by standardization, access transparency, scalability, reconfigurability and security. The definitions and characteristics introduced here reveals basic feature of a typical grid implementation. To aid the process of determining weather the given implementation qualifies as a grid, a three point checklist[115] is suggested by Foster. A grid coordinates resources that are not subjected to centralized control. A grid utilizes standard, open, general purpose protocols and interfaces to realize functionalities. And, a grid delivers non-trivial quality of service within service-oriented architecture.

Layer	Role	Example
Application	Applications and Portals	Scientific, Engineering, Collaboration
User Middleware	Development Env./Tools	language, library, debugger, monitor, webtools
Core middleware	Dist. resource coupling	Security, data, process, trading, QoS
Fabric	Local resource manager Networked resources	OS, queuing systems, Libraries, protocols Computers, Storage systems, Networks, Data sources, Scientific instruments, Sensors

Table 6.8: Grid Architecture Model [114]

The Open Grid Service Architecture [116] provides set of grid middleware services that are specified as standard set of services to be supported by planned grid middleware technologies [94]. The Open Grid Forum (OGF) is currently developing grid related standards created as a merger of the Global Grid Forum (GGF) and the Enterprise Grid Alliance (EGA) that independently involved in the advancement process.

Alterations required in Managing Grid Resources As indicated in The Grid Architecture, the grid service stack exposes various functionalities that can be consumed as service to address computing and information management needs. Rules play a vital role in expressing and utilizing complex business, application or domain logic. Hence, utilization and handling of grid resources should be driven by rules as supported by the semantic web technology. For instance, grid exposes computation power to the users. In order to tap this power appropriate job specifications must be created and submitted to job management component of the grid. Through command line tools, desktop clients or portals, job is submitted. Computational requirements identified in the form of task specifications should be automatically converted in to job specification and it should be appropriately monitored and controlled using various grid features.

Utilization of Grid Capability Dynamic set of users and resources identified characteristic to the complex dynamical systems are considered to form a Virtual Organization. Hence, Grid architecture proposed to meet distributed computing requirements of VO is utilized to set up VO for collaborating organizations. Each available member is created and handled as a member of VO. Shared services and resources following the policies are appropriately described while incorporating in the grid. The grid security is setup to handle VO level security. Computing power of grid is exposed by allowing job execution environment. Grid container is set up for hosting grid services that can be dynamically composed and consumed as application workflows[117]. Grid applications are utilized to aid specific information processing requirements.

6.3.7 OGSA-DAI for ETL

For information management across distributed users, Extract Transform and Load (ETL) is an important feature in controlling the flow of information among them.

Topic	**Form**	**Shown in**
ETL Sp. Gen (Sec 3.8.10)	`GenRepTaskSpec` in Algo (3.10)	Fig. 3.36
Data Prov. Gen (Sec 3.8.12)	`Generate` in Algo (3.12)	Fig. 3.38
DDM Gen (Sec 3.8.13)	`Encode` in Algo (3.13)	Fig. 3.40

Table 6.9: ETL Management Requirement

OGSA-DAI The OGSA-DAI project [118] is involved in realizing data services as identified in OGSA Specification. It enables a grid application targeted at distributed data management. Management of distributed data in grid environment introduces special requirements. It is deployed as a service in grid execution environment. The present OGSA-DAI can be deployed on Globus container as well as the OMII container introduced in earlier. Figure 6.4 indicates the same.

RDF-OGSA-DAI [7] is the project developed to handle Ontologies[119]. The present ver-

[7]OGSA-DAI-RDF: http://wiki.dbgrid.org/index.php?OGSA-DAI-RDF

sion is supporting ontology exposed with Jena2 and Sesame technology. Figure 6.4 represents

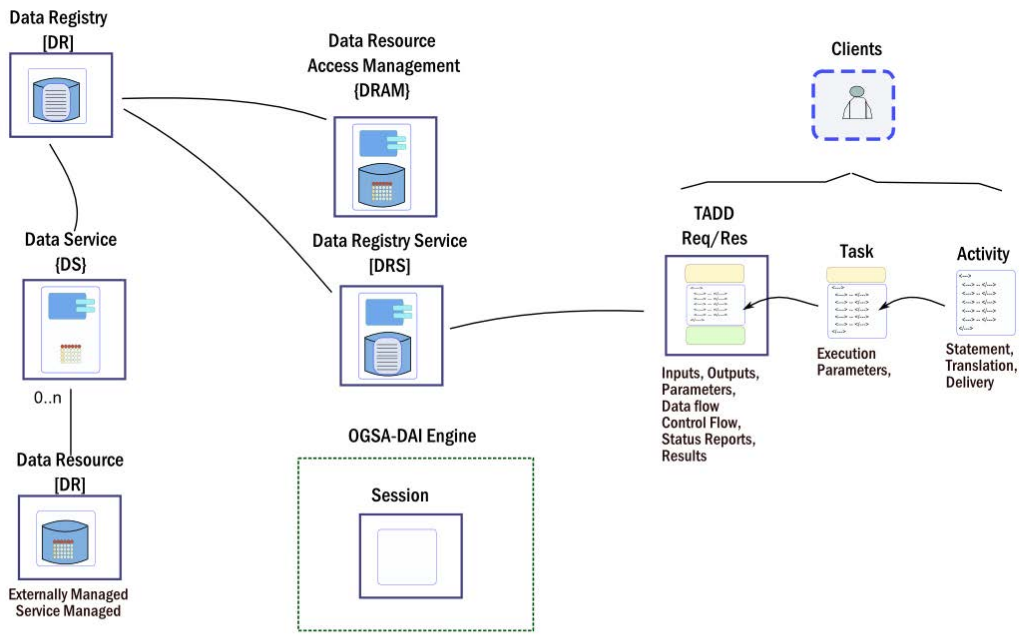

Figure 6.4: OGSA-DAI Overview

components of OGSA-DAI. The following is the brief overview of main components.

Data Resources Data Resources can be any externally managed database system. OGSA-DAI supports, major RDBMS, XML databases and file storages.

Data Sevice Data Service provides interface to access zero or many data resources.

Activity Activity is unit of work. There are three type of activities supported namely: Statement Activity, Translation Activity and Delivery Activity. Statement Activities are set of SQL or XQuery commands that are supported for common functions. Translation activities are set of activities related to changing the normal data to compressed formats. Delivery Activity is set of activities that allow unique features that support delivery of the data to third party location. These locations may be a URL, Mail, FTP, GridFTP or any other valid location in Grid.

Task Task is set of activities with execution parameter.

Task and Data Document TADD contains data flow, control flow various input, output parameters, and other related information to execute the task in required format.

Session Management of session and transaction are two important features provided by OGSA-DAI exeuction engine.

Transaction WS-AT is supported as standard to achieve transaction in Web Service environment.

Data Registry Service Data registries hold DS related information and allows discovery of DS.

Requirements met by OGSA-DAI One of the basic utility of OGSA-DAI project is to include new data resources and expose it using the web service. This is useful in making the newly acquired, allocated or donated resources to be added to the service so that it can utilized in data management tasks. Present version supports various relational and XML based database technologies. With RDF-OGSA-DAI, utilized on top of OGSA-DAI instance, the shared Ontologies can also be exposed for various semantic operations.

Type of Activity	**Example**	**Project**
Relational Database XML Database	sqlQueryStatement, sqlUpdateStatement,sqlBulkLoadRowSet sqlStoredProcedure, relationalResourceManagement, xPathStatement, xUpdateStatement xmlCollectionManagement, xmlResourceManagement	ogsa-dai ogsa-dai
Transform	gzipCompression, gzipArchive, xslTransform stringTokenizer, blockAggregator	ogsa-dai
Delivery	deliverToResponse, deliverToURL, deliverFromGFTP deliverToStream, deliverToGDT, deliverFromGDT inputStream, outputStream	ogsa-dai
Utility	dataStore	ogsa-dai
Semantic	DataSetManagementActivity , GraphManagementActivity sparqlQueryStatementActivity , OntologyReasonerActivity	rdf-ogsa-dai

Table 6.10: OGSA-DAI Activity List

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- (c) International Business Machines Corporation, 2002 - 2005.-->
<!-- (c) University of Edinburgh, 2002 - 2005.-->
<!-- See OGSA-DAI-Licence.txt for licencing information.-->

<perform
         xmlns="http://ogsadai.org.uk/namespaces/2005/10/types">
    <documentation>
SPARQL Desribe Query
    </documentation>

    <sparqlQueryStatement name="SPARQLDesribeQueryActivity">
      <query-request>
        <query>
PREFIX rdf:&lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;
PREFIX rdfs:&lt;http://www.w3.org/2000/01/rdf-schema#&gt;
PREFIX sa: &lt;http://www.da-iict.org/research/processes.owl#&gt;

DESCRIBE *
WHERE {?x ?y "Basic-phone-service"@en}
```

```
</query>
      </query-request>
      <query-result name="DescribeResult"/>
    </sparqlQueryStatement>
</perform>
```

Alteration required in employing OGSA-DAI The example perform statement is represented in the code snippet is created manually. Due to the dynamic nature of the resources, static representations are not useful to act as ETL. This resource includes the source and target of the information, the granularity of the events, the availability of intermediate data sets and the logic for transformation may change during the runtime. Hence it should be generated automatically. In Chapter 3, Algorithm 3.10 represents one strategy for dynamic rule driven generation of the ETL specification that are programmatically transformed in to OGSA-DAI perform documents and executed with the service client. Similarly data provenance[120] is also required to be handled automatically with every state change experienced by the datasets.

6.3.8 Eclipse as a Tool Integration Platform

Software development platform plays important role by providing basic services to aid the software development life cycle. It provides Integrated Development Environment (IDE) for development deployment and testing of the applications using specific technologies. This includes basic programming and productivity related features like editor, debugger, compiler and deployer. Apart from this, it can handle other feature of CASE tools like requirement management, configuration management, versioning, and test management. It provides access to help and support. To support collaboration, it provides communication features. The platform must expose middleware level services at lower levels. It should support creation of rich client application.

Eclipse Project Eclipse in its initial phase was introduced as an IDE. It is following plug-in architecture based on the standard specification[121]. In its core, eclipse has a small footprint kernel that is acting as a plug-in loader. All the functionality planned by eclipse product is realized in form of plug-ins. At the time of loading, required plug-in are identified and loaded to serve the functionality. In the plug-in based approach, a small functionality is realized and provided as a plug-in. Apart from the normal use by loading the plug-in and accessing the functionality, the declarative approach of Plug-in architecture allows description of plug-in extensions and extension points [122]. These extension points tell how the functionality rendered

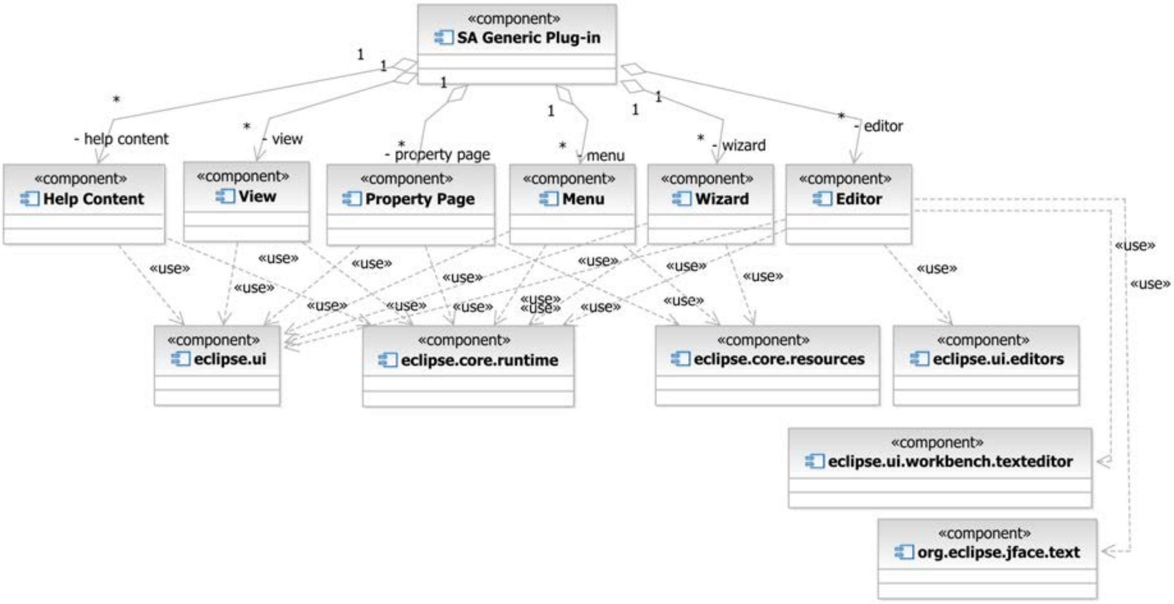

Figure 6.5: Generic Eclipse Plug-in Template

by the plug in can be extended to provide more complex, extended features to meet the user need. In this manner, the vast community effort resulted in developing large stack of plug-ins each extending the basic capability offered by the lower level plug-ins. In this scenario, every developed plug-in is considered as contribution to the community, enhancing the functionality of the existing state. The licensing of products also enhanced it's uptake in academia and open source community. This approach is argued to be a better solution as compared to XML based approach, Net beans or IntelliJ [123]. Eclipse is emerging as dominant IDE of choice among developer community [124]. Figure indicates a generic plug-in indicating the building blocks.

Requirements to be met by Eclipse In its current stage, Eclipse has evolved as a stack of technology projects from which required blocks are to be selected and utilized to meet the user need. Following table indicates how all the requirements are available as eclipse project. A comprehensive list of project is available on eclipse foundation website[8].

6.3.9 Semantic Web Technology for Method Engineering

Method engineering is introduced in Chapter 4 as appropriate approach for orchestrating the collaborative effort[79].

[8]Eclipse Project http://www.eclipse.org/projects/listofprojects.php

Chapter	Section List/Function	Figure
4	4.6.1 Role Sets	4.16
5	5.1.2 Enterprise Continuum	-

Table 6.11: Unified Process Management Requirement

Semantic Web Technology Semantic web technology is discussed earlier in this section provides support for knowledge management.

Semantics in Method Engineering Method Engineering consists of methodology, processes, environment, artifacts and related guidance. The method should provide support for the activities of all the role players in different environment discussed in Chapter 4. Creation of content for domain specific activity requires capturing information about related role, processes, resource dependence, standard operating procedures etc. This in turn becomes exercise of capturing domain knowledge. As the method content cannot be created in isolation, it acts as a live repository continuously being updated. Other static aspects like creation of Method plug-ins in Rational Method Composer or Eclipse Communication Framework may not support this dynamic feature. As identified in the requirements for CASE tool in Section 6.1, Event driven, rule based access in virtualized environment is required.

Alterations required in Process Tooling To establish traceability involving the identified instances, creation of architectural products, providing rule-based access to the method content provides required support of appropriately configured middleware services. The major change is required in identifying the coverage and accessing the shared content. Instead of stand-alone instance of method plug-in, a service is required that can access to other configuration providing the missing components identified. Also it is necessary to handle method content related events like, when it is created, published, shared and accessed.

6.4 SA Core Architecture

Requirements are identified in Section 6.1. Building blocks are identified in Section 6.2. Design decisions are introduced in Section 6.3. Next step is to propose architecture based on them. Based on the design decision each building block is revisited to suit the newly identified requirements.

6.4.1 Message Oriented Middleware

Messaging is identified as one of the core requirements of situation awareness middleware. Based on its role, the message oriented middleware that provides the required services is identified as one of the building block. One of the design decisions that recommend handling vast amount of identified batch job in the form of queue, the responsibility of message oriented middleware not only limits to the communication but also becomes core enabler in information processing strategy.

In realizing various communication scenarios, specific messaging patterns are established. A comprehensive list of various messaging pattern is discussed by Hohpe and Woolf[13]. This includes specific messaging patterns that addressed case specific realization of messaging system components. These patterns can be used by establishing rules for characterization of requirement and identification of appropriate messaging patterns that can support them. The MoM supporting creation and handling of messaging components is utilized by this rule driven system. Hence the resulting system acts as a Messaging Pattern Factory that manages handling of messaging component on the runtime. This includes creation of new patterns, configuration, monitoring and administration to suit the need. This in turn becomes the life cycle management of messaging pattern.

Considering the need of specific usage scenario, various messaging patterns are suggested. The patterns include various system components like message, message channel, pipes and filters, message routers, message translator and message end-point. Patterns also recommended various communication strategies that can be realized within the system by employing these building blocks. As an example, the channel can be realized with patterns. These features are

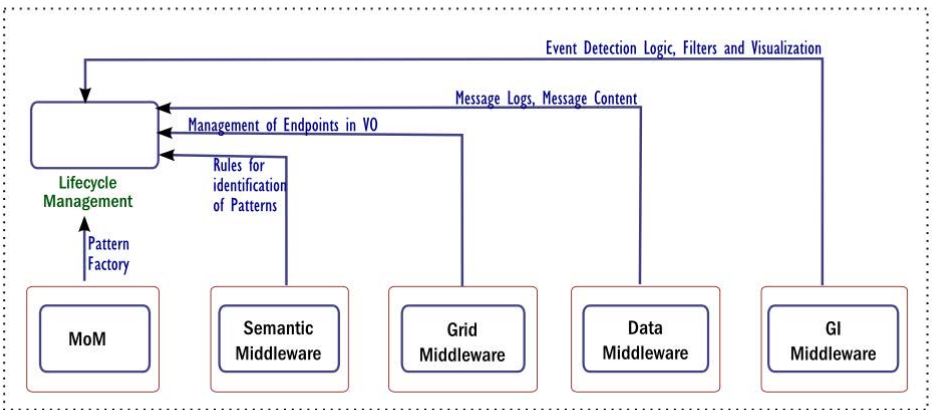

Figure 6.6: Improvements in MOM

supported by popular platform architectures. Java platform[9] and Microsoft .Net Framework[10] are competing technologies. Java community process has resulted in multiple specifications addressing the messaging services. Java Message Service (JMS) is one such specification that provides creation, administration and handling of messaging patterns. JMS[93] supports realization of various messaging related patterns may be identified with different names. Other vendors in MoM domain may support building of other patterns. It handles message channel pattern as *Destination*. Point to Point channel is supported as a *Queue* and Publish Subscribe channel is supported as *Topic*. Message producer and message consumer are the endpoints supported in JMS. Other vendor technologies like Microsoft MSMQ, WebSphere MQ, TIBCO, WebMethods, SeeBeyond and Vitria provide similar capabilities. Sun JMS is selected as preferred technology as it is a proven open source implementation in Java technology. Java message queue JMQ [125] is the implementation of JMS made available. Glassfish[11] release is an open source server technology that provides various implementation of Java specification.

Individual pattern creation and handling can be done using web based client or administrator tool. In application scenario, individual creation of pattern cannot help. Due to requirement of handling many patterns, it should be handled by program following the specified rules. Java Management Extension (JMX) [126] provides MBeans that allows creation, monitoring and handling of the messaging components from program.

Existing MoM is capable in providing creation, monitoring and handling of messaging patterns with important features. Automation the management of life cycle is missing that is identified as important feature of SA middleware. Figure 6.6 represents how other building block can bring benefits to MoM in achieving the objectives.

Rules for identification of patterns There can be many possible patterns. Selection of appropriate pattern should be done automatically on the run time. There are specific criteria for selection. They are encoded in rules form. Hence service from semantic middleware is required.

Management of Endpoints in VO The pool of systems and resources that utilize them are seen as forming the virtual organization. Hence their membership to virtual organization needs to be handled as grid resources.

Message Logs and Message Content The automated strategy is expected to create many

[9] Java technology: http://www.sun.com/java/
[10] http://www.microsoft.com/net/
[11] Glassfish: https://glassfish.dev.java.net/

individual information interactions in the form of identified information exchange patterns. Status is monitored and recorded during the lifetime of these patterns. Hence they are expected to generate considerable amount of data in form of logs and event traces. Data middleware services must be used to handle this generated information.

Event detection logic and filters Within rules for generating and messaging patterns, geographical attributes provide important basis for decision-making. The required query, filtering and rendering functionality is supported by GI middleware.

In realization of the discussed functionality, the required solution must be able to invoke operation to other middleware services. Hence a control class is designed that can handle cross middleware service invocation and handling is carried out by them. All the lists of identified task specifications are maintained as a list of scheduled batch jobs. These lists are implemented in the form of queue.

Transient resources are identified first as messaging endpoints. The members of VO may act as source or sink of the information. The determination of appropriate channels, message format and strategies for handling the message is achieved with corresponding instances of the patterns.

6.4.2 Semantic Middleware

Semantic middleware is important building block providing the required support for creation and handling of knowledge base and fact base. The logic of unified process is handled by the semantic services in order to determine roles, providing situation awareness in the form of required task allocation, guidance, alerts and other information determined suitable for them. Information processing logic is also captured in the form of rules. This includes determination of appropriate granularities and identifying suitable task specifications for them. Application logic that governs the processing and handling of information is also handled with semantic middleware service providing important functionality. Capturing and using knowledge in all the related domains also requires support of semantic middleware service. Local knowledge and facts captured to depict ongoing situation is also realized over the same service. All these are handled in reference to the ontology created for the same.

This functionality is offered by multiple products available in semantic web technology domain. For creation of ontology according to RDF and OWL standard is supported by Jena. Hosting of ontology is realized with Sesame service. This also support query according to SPARQL. A reasoning service is provided by Fact++, pallet and Racer products. They provide interface for reasoning service.

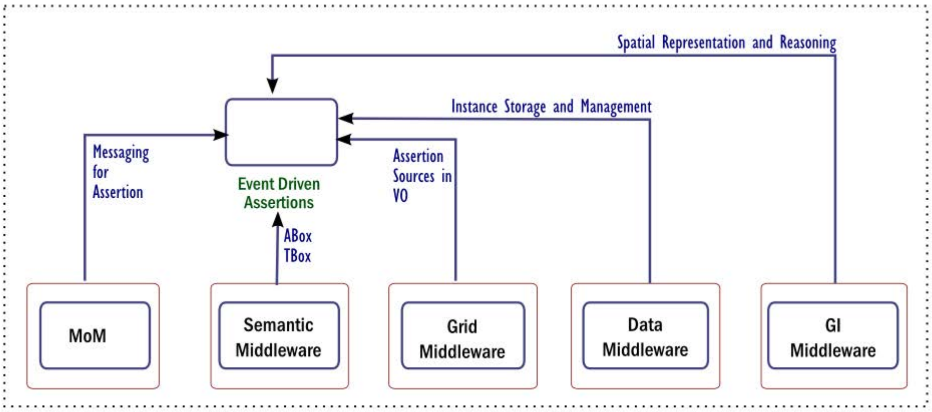

Figure 6.7: Improvements in Semantic Middleware

Various resources need to be handled to meet the requirement. First the knowledge captured in the form of Ontology involves local, domain independent, domain specific and application specific concepts. They are captured and used in the form of multiple shared Ontologies continuously being upgraded. This vast pool of ontology is important resources that must be handled by the planned middleware service. The handling also include pool of providers in the from of users who enable creation of concepts, rules and provide status of instances. The pool of users who access the knowledge should also be handled by the system.

Existing semantic middleware is capable in providing creation, handling, sharing utilization of knowledge with important features. Automation in them is missing that is identified as important feature of SA middleware. Figure 6.7 represents how other building block can bring benefits to semantic middleware in achieving the objectives.

Messaging for Assertion Assertions made on the runtime are done by multiple users. The users must be appropriately informed about the nature, frequency and content specification of required assertion that is suitable in given configuration. This requires specific information exchange pattern. Hence, it can be said that in order to handle consistent knowledge management in collaboration environment, messaging plays an important role by realizing required communication pattern.

Assertion sources in VO Attributes of instances, roles assumed and the information accessed by them in the configuration is done being part of a VO member. This allows application of security framework, thereby providing features of grid into handling of knowledge base. Intersection of semantic web and grid technology brings benefits to knowledge management process in collaborative environment.

Instance storage and management According to the design, multiple assertions are handled in automated manner. This results in huge instance base. The storage, management and handling of such larger instance base must be done with the help of data management services.

Handling of spatial attributes Semantic middleware is equipped with handling conceptual relationship among concepts. Some extensions also allows query and reasoning with temporal properties. Yet, spatial query and reasoning requires additional GI processing capabilities. These capabilities required GI middleware support. Queries for filtering, selection, operations like aggregation, event detection and snapshot generation requires the functionality of GI middleware. This is extended by appropriate integration of GI functionality in handling knowledge base.

Control classes must be defined that can provide access to services offered by the middleware building blocks. A control class created for management of assertion requires access to middleware service.

Transient entities include instances being asserted and handled by the system. The inferencing is also critical service provided by the semantic middleware. The assertion of basic instances and continuously updated information regarding their status allows identification of their membership to other classes. Various relationships they stand with other instances in form of roles and events also becomes transient entities handled by the system. In case of mapping the instances to meet the transformations various ontology mappings identified among instances, rules for transformation etc are also identified as transient resources.

6.4.3 Grid Middleware

Virtualization of resources and execution environment are two important function provided by the grid middleware. Open Grid Services Architecture (OGSA) is the standard. It addresses the requirements like interoperability and support for dynamic and heterogeneous environment, resource sharing across organizations, quality of service, job execution, distributed data management, Security Globus Alliance [12] is a community dedicated in developing grid technology.

[12] http://www.globus.org/alliance/

Globus Toolkit[13]. Open Middleware Infrastructure Institute[14] (OMII) UNiform Interface to COmputing REsources[15] (UNICORE) are popular implementation providing grid services [127].

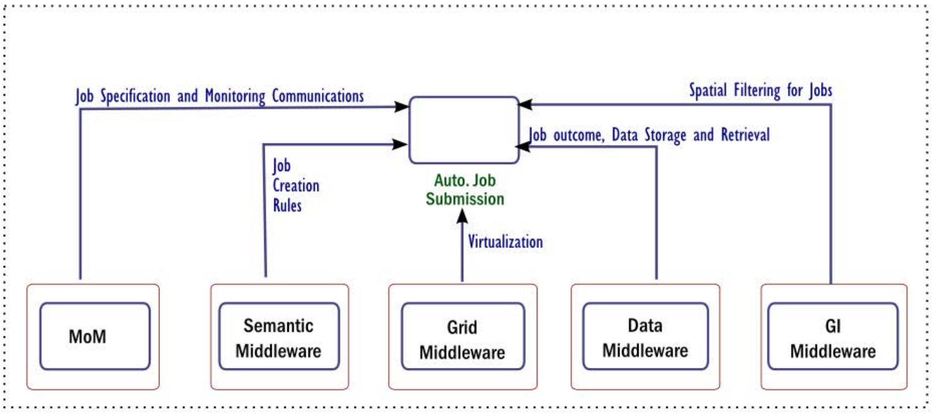

Figure 6.8: Improvements in Grid Middleware

Within the VO formed by dynamic set of collaborating entities, resources to handled by the grid middleware includes VO members, computing resources, data resources, data sources, services and other important feature providing data, functionality or grid service. Existing Grid Middleware is capable in providing handling dynamic set of resources with various grid features. Automation in handling resources in utilization of computing capability and in other grid specific task is missing that is identified as important feature of SA middleware. Figure 6.8 represents how other building block can bring benefits to grid middleware in achieving the objectives.

Job Specifications and monitoring communication Various components and members in virtual organization is continuously required to communicate in order to utilize various features of grid. For exchanging job specification, event and alert notifications, monitoring and control messages, appropriate interaction patterns must be realized in the VO. MoM features in creating and handling appropriate messaging patterns bring important benefit to the grid middleware.

Rules for generation of Task specifications Grid middleware provide job execution environment. The jobs can be submitted using command line clients, portlets of desktop applications. In complex dynamical system, identification of individual job submission becomes difficult. Application logic captured in rules form can be utilized for automated

[13]Globus Toolkit Web Page: http://www.globus.org/toolkit/
[14]http://www.omii.ac.uk/
[15]http://www.unicore.eu/

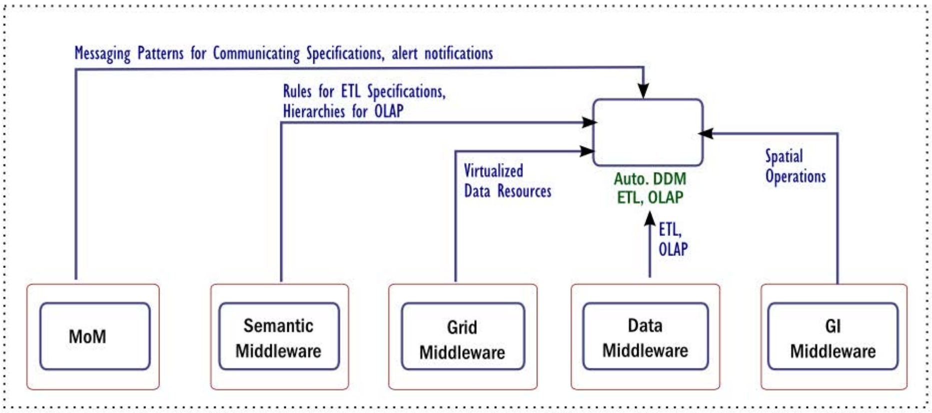

Figure 6.9: Improvements in Data Middleware

generation of job specifications. Hence semantic middleware can bring the benefit to grid middleware.

Storage and handling of Job Outcomes Storage and handling of information exchange patterns and job outcomes are required. The data middleware can allow handling of data sources and data resources that capture and handle data generated on the runtime.

Spatial reasoning in managing VO resources In managing VO resources, spatial attributes of the resources provide important criteria for decision making.

6.4.4 Data Middleware

Data middleware provide data handling and management capability. Products for storage and management available in the form of database management systems are useful to act as data resources in the grid environment. In order to access the shared grid resources middleware in form of libraries and products are useful. OGSA-DAI is one example of data middleware that provides this facility. It allows basic database operations, monitoring and administration in grid environment. Existing data management middleware is capable in providing distributed data management, ETL, OLAP and other important features. Automation in them is missing that is identified as important feature of SA middleware. Figure 6.9 represents how other building block can bring benefits to Data middleware in achieving the objectives.

Communication of Data Management Tasks Data management in distributed environment requires interaction patterns among system and users that manage them. This includes communication of data management tasks, status monitoring and their output. Some of them may be created in response to the observed events. Hence incorporation of communication pattern brings required reactiveness to the data management tasks to suit the ongoing requirements.

Creation of Data management task specification In complex dynamical systems where considerable amount of data is created and handled among distributed data resources, the individual attention to database management task becomes difficult. Data management task specification can be generated automatically by the system. Semantic web technology can help utilization of rules that can automate identification of required tasks.

Virtualization of Resources Multiple data resources made available in a configuration. Handling them individually for appropriate utilization is not possible where availability of such resources vary with time. By making it a virtualized resource in grid environment, and handling it as a grid resource brings benefits of grid middleware services to data management.

Spatial query and processing In processing of data, spatial attributes play important role as spatial footprint of concept may have implications. Available database management systems have varying capability in handling geographical information. Hence provision of consistent middleware services in generating and handling spatial data can bring benefits to database middleware. Also, distributed data management tasks also requires spatial attributes of data sources and data resources and therefore it can also bring benefits to distributed data management tasks.

A control class created for Snapshot generation provides implementation of Algorithm 3.6. Distributed Data management provides implementation of Algorithm 3.13. ETL Specification class provides implementation of Algorithm 3.10. Data Provenance manager class provides implementation of Algorithm 3.12.

The data middleware supports handling of various transient resources with the help of other middleware functionalities. The data sources that are sources of information are dynamic set of entity that may join the VO as sensor, human observer or other information service. The data management at lower level requires data resources, the physical instances database server that host the required information during the runtime. The availability of data resource is also dynamically changing and hence each one is treated as a transient resource. The data sets identified created handled, utilized and purged according to ETL and distributed data management policies, are also transient resources handled by the middleware. Along with the data sets, the corresponding provenance records are also treated as a transient entity.

6.4.5 Geographic Information Processing Middleware

Requirement of handling geographical information is identified since the discussion of spatial footprint in Chapter 3. This includes spatial data representation, spatial data storage, spatial query and other related tasks.

There are several commercial and open source libraries, tools and technologies available that support the required GI processing functionality at varying levels. GeoTools provide a set of libraries in handling spatial data. The Web services exposing GI functionality as per the standards by OGC is important to development of GI Infrastructure. The Deegree and Geoserver provide implementation of various OGC service for handling map, spatial features, catalogs and other aspects to access information over web service framework. From client point of view, user Friendly Desktop Internet GIS (uDig) is an eclipse based client for accessing data as a client of OGC web services. This product provides with extension point that can be utilized to develop required functionality.

Existing GI middleware is capable in providing creation, handling, sharing and utilization

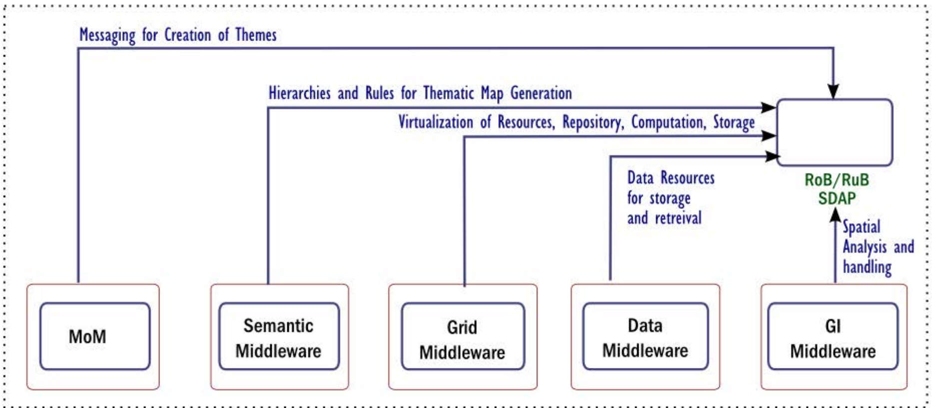

Figure 6.10: Improvements in GI Middleware

of geospatial information with important features. Automation in them in order to realize spatial data access patterns are missing that is identified as important feature of SA middleware. Figure 6.10 represents how other building block can bring benefits to GI middleware in achieving the objectives.

Messaging for Spatial data generation and Access Realization of spatial data access pattern in creating and providing spatial content in response to an event requires utilization of a rule based system for hosting required communication pattern among systems and users. Appropriate rules captured and utilized using messaging middleware can bring benefits to GI middleware.

Rules for spatial data generation and access Realization of spatial data access pattern in creating and providing spatial content in response to an event requires utilization of a rule based system. Appropriate rules captured and utilized using semantic middleware can bring benefits to GI middleware.

Virtualization of content and service Sources of GI information, GI services and other related resources involved in providing and handling spatial content may dynamically be available to the configuration. Handing them as members of the virtual organization can bring benefits of grid features to GI middleware.

Data sources for spatial data storage and handling Within rules for generating and messaging patterns, geographical attributes provide important basis for decision making. The required query, filtering and rendering functionality is supported by GI middleware.

A control class that exposes functionality of libraries that allows creation and handling of GML documents. `GML Manager` is suggested that provides encoding function in creation and handling of GML documents. It utilizes functionalities of other middleware services in listening and writing to task specification queues. This along with other components in GI component in turn provides domain middleware service. Transient resources in GI environment are handled by the configuration. Representation of situation at specific granularity is created, processed and published by the GI middleware. They can be considered as coverage providing thematic data derived from monitoring of real life features prevailing in UoD. In order to create such themes of representation at varying scale, appropriate human or sensor observations are required. The tasks of observations, messages that hold the observed values, communication patterns that allow realization of information flow and templates are all the transient entities that are handled by the system to create the desired geographical representations. Apart from these themes many Architectural Products are also rendered as map, hence they are also one of the important transient entities handled by the GI environment in realizing Situation Awareness services.

6.4.6 Runtime Middleware

The proposed system is designed to provide domain middleware service targeted at situation awareness. This is realized with reactive nature. Various information processing tasks are triggered by the occurrence and detection of event. The availability of dynamic set of participating resources is continuously checked and handled appropriately. In this scenario, the runtime services provide basic features required for configuration of underlying services and components.

Configuration of the instances of basic building blocks providing core services is primary requirement of the runtime middleware. Any state change may affect the overall services, hence continuous monitoring needs to be carried out. As many types of resources are configured within the configuration, specific roles can be identified and made responsible for those resources in the configuration. Hence, a dashboard view of the system providing configured resources is also required. The administration of various configuration services is also required functionality. For appropriate control of the runtime, the functions like starting and stopping of service is required. To support maintenance, include newly available updates and other administration scenarios should also be supported.

Various control classes can be identified that are responsible for providing middleware services. `SA Configuration Manager` is a class designed to handle configuration of core services. Link to the service is established by this class. `SA monitor` is the control class that provides monitoring features of the configured resources. The basic monitoring mechanism of lower level service is utilized to provide overall monitoring of resources in domain middleware. `SA Process Manager` is responsible for exposing the process content of Situation Awareness Unified Process. This is responsible for creation, handling and sharing of various process artifacts, architectural products and related content. `SA Runtime` provide runtime features of the domain middleware.

Deployment view of a SA configuration is depicted in Figure 6.11. As indicated, instances of various building blocks are required to fulfill the configuration requirement. MoM service instance is a link to messaging service that enables creation and monitoring of messaging patterns. Monitoring facility exposed by the MoM is utilized by the runtime. The link to service hosting knowledge base and providing query, inference and instance support is required as one of the core service. Access to GI services can be exposed with connection to many possible services providing query, data access and other GI processing functionality. Hence, there can be links to multiple GI services required to meet the needs. Link to Grid Middleware determines the membership to a virtual organization. The link provides access to various grid services and resources. Multiple links to data management services and instance may also be handled by the system.

The runtime view includes handling of various transient entities. One class of transient entities supported by the runtime is configured entities. These entities are the resources like computational resources, storage devices, sensors, services etc. Monitoring and handling of such configured entities providing infrastructure for domain middleware services are handled

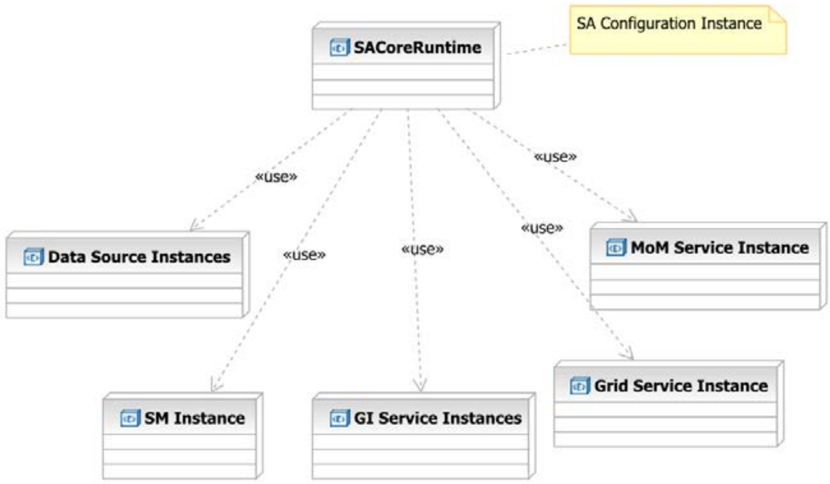

Figure 6.11: Deployment view of the SA Runtime

separately by the middleware. Another class of transient resources are the programming or logical entities that is handled by the configuration. They are handled according to the rules defined in the KB. These runtime entities are in the form of identified tasks, job, role, events or other abstract entities. As they provide the information processing functionality; appropriate monitoring and handling is supported by the runtime.

6.4.7 Proposed Architecture

Based on the core components and interactions discussed among them, an eclipse platform based architecture is proposed. Figure 6.12 represents the plug-in architecture revealing important components. The set of plug-ins are classified based on the functionality provided by them. The logical collection of core set of plug-ins and related resources are identified as *SACore*. SACore includes Core Plug-ins, Runtime Plug-ins, Utility plug-ins and resource pool.

Core plug-ins are set of plug-ins that realizes the control class that provided extended functionality of middleware as discussed in the earlier part of this section. Assertion Manager, Information Need Manager, MOM Pattern Manager, Event detector, ETL Manager, Grid Manager etc .are the core plug-ins that provide core functionality of the SA Middleware. Their functionalities are realized as higher level domain services that are built over middleware services provided by the building blocks in lower level of abstraction.

Utility plug-ins are set of plug-ins that expose services of middleware building blocks such that they can be consumed by the core plug-ins. The utility plug-ins includes: MoM plug-in, KM plug-in, Data Plug-in, VO Plug-in and GI Plug-in.

Runtime Plug-ins as the name suggests, provide functionality for configuration and execution management. These plug-ins provide support for configuring, monitoring and handling of required computing resource. They also provide point of access that exposes the core services to the various clients. The situation awareness configuration is discussed throughout the discussion is made accessible using these runtime plug-in components.

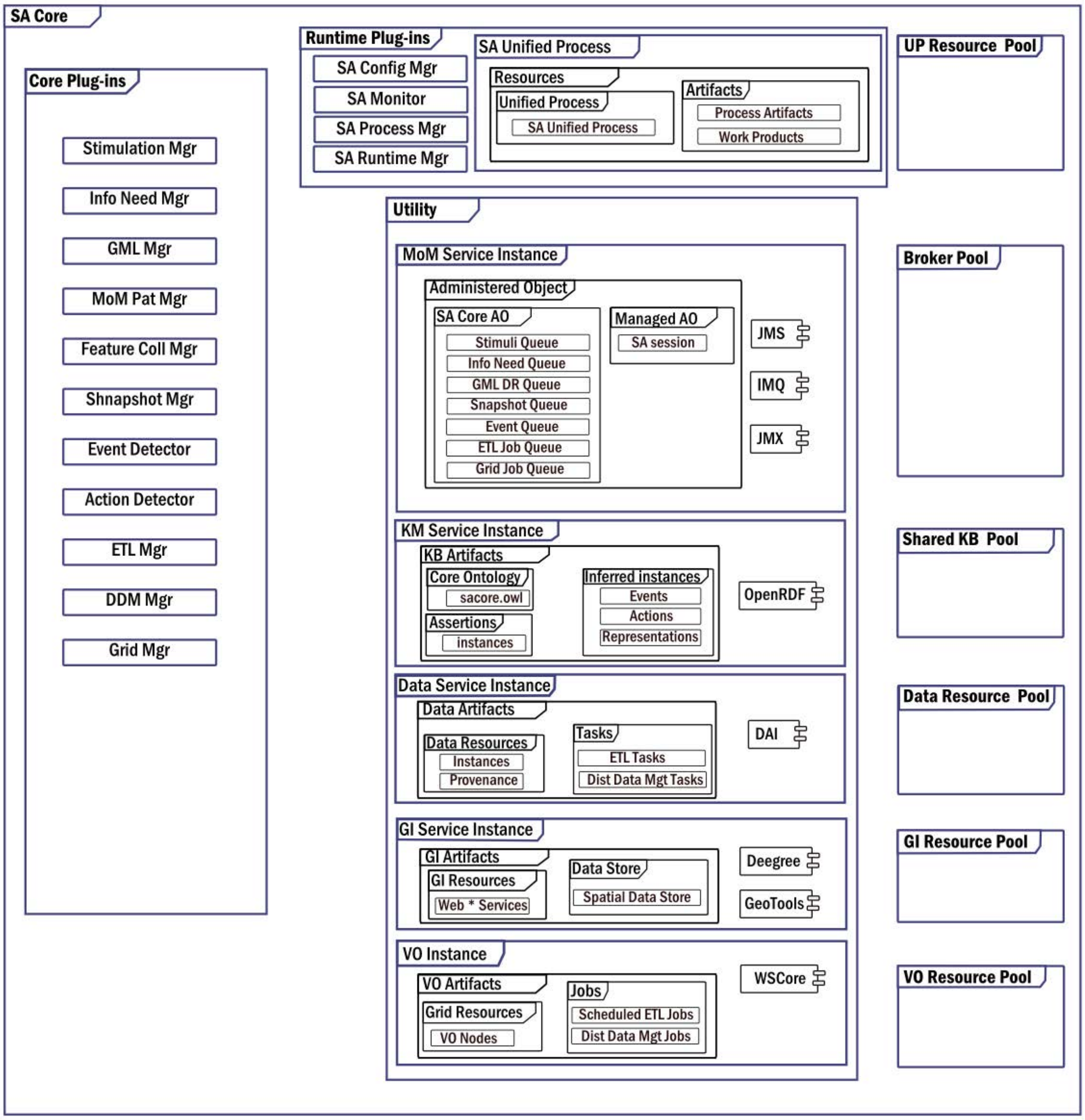

Figure 6.12: SA core Architectural Components

Apart from the static content and the specific functionality rendered by the plug-ins in the SACore, there are special features represented in the system. The utility plug-ins are not represented as serving the requests issued by the control plug-ins in the above layers, they also provide, Situation Awareness configuration specific content. For example, the MoM utility plug-in exposes various functionalities provided by JMS, JMX and JMQ libraries. It also represents core queues required to handle automated tasks identified on the run time. Apart from this configuration aspect, the SA Middleware manages the life cycle of messaging patterns. These managed objects are depicted to be part of MoM. Also it is possible that multiple MoM services are required to be bridged to meet the multiple configuration requirements. In the resource pool various message brokers and other messaging system components are depicted.

Similarly, the KM service instance represents the Situation Awareness Core Ontology that consists of KB. All the facts and assertions created in references to this KB. It also contains the rules for various application specific information-processing tasks. Inferred instances in form of identified events, actions and generated representations are indicated as transient resources handled by the service.

Data service instance, GI service instance and VO services instance are similarly represented as handling appropriate artifacts and transient resources associated with them. Required and available resource pools that support these services are also represented to establish the responsibility and capability of the system to meet the needs. The content hosted and served by the runtime plug-ins include Situation Awareness Unified Process and various artifacts, work products and components exposed in the enterprise continuum.

6.5 System Views

Architecture discussed earlier can be utilized to meet the requirement. The system should be able to demonstrate its capability in meeting the user needs of complex dynamical system. There are the abstract representations created for demonstrating system capabilities. Once they are configured and created, deployed according to the process.

The architectural products provide view of the individual component features. Whereas, a separate level of viewpoint is required to evaluate and monitor the overall system. For the visualization of salient features of the proposal, various views are proposed. They demonstrate specific capabilities show how system design is meeting the desired requirement. The views are identified as follows:

6.5.1 Event View

Primary focus of the event view is to demonstrate all possible events that are handled by the system. According to the definition of event proposed in Chapter 3, event is attached to entity or process, has detection mechanism in reference of a configuration and appropriate role that is responsible for suitable response. This leads to involvement of various environments discussed in unified process.

Hence, it can be said that event view is focused on depicting event space along with related entities. Figure 6.13 provides a part of event view mainly covering the configuration level events. In similar manner, events can be depicted with related concepts across all environments.

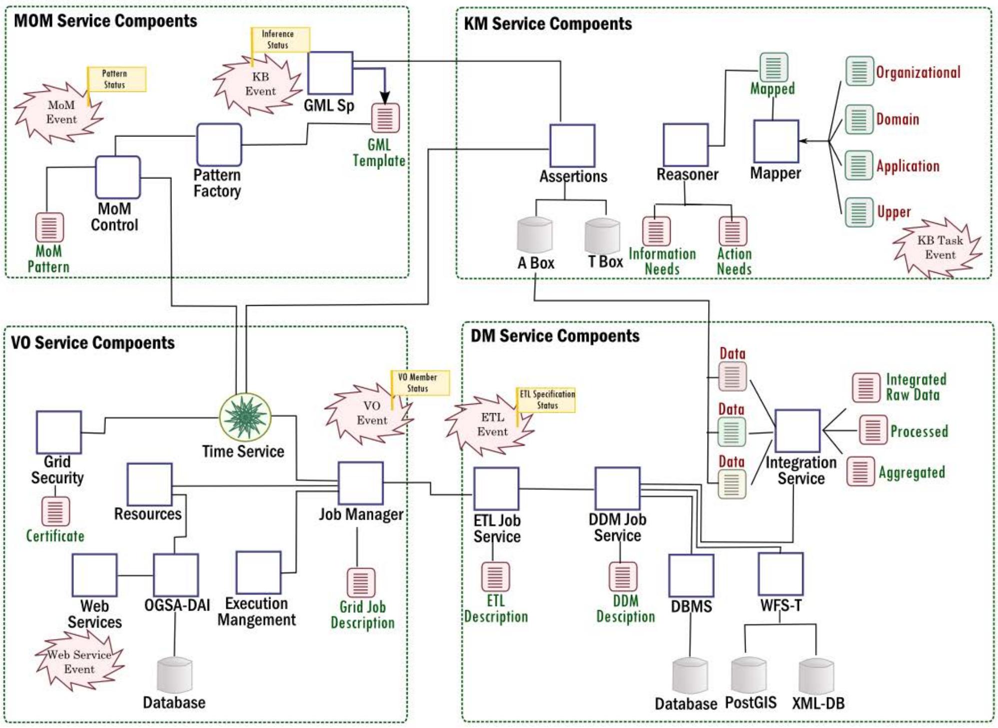

Figure 6.13: Event View of the Situation Awareness System

The event view indicates the interactions among related components that are triggered due to events. Prerequisites for such interactions are specific for handling events. Event sources are identified a priori in the knowledge base and their occurrence is verified with the help of event profiles. Based on this knowledge, available instances of event sources must be configured and monitored appropriately. The profile-matching task needs to be carried out at defined interval. In case of detected event, the identification of action, selection of user and notification should be created. This view amounts to be a considerably comprehensive view of situation awareness middleware capability that is responsible for desired reactivity. Realization of such view demonstrates that the designed system is an example of Distributed Event Based System. Also, it establishes the event space - a set of all possible events that can be handled by the system.

Capturing Events Any state change in an entity can be considered as event. The goal is to identify those changes that are relevant to an actor or a process, and to notify appropriately so

Entity	Domain	Type of Event
Organization	Real-world	Adopting response
Actors	Real-world	Taking Actions
Concept	Ontology	Inferred membership
Situation	Ontology	Inferred Info. Need
Resource	Grid	Joining/leaving VO
Job	Grid	Execution State Change
Res. Prop	Web Service	Change in Res. Prop
Subscription	MoM	Change in subscription
Event	MoM	Detection of Event

Table 6.12: Event Types

that necessary action can be taken. Here *entity* can be a job, a resource, a person, equipment or any other real life object. The change of state is very specific to the type of entity. For example a job can be in *Started*, *Suspended*, *Failed*, *Aborted* or *Finished* state. These state changes are very different from the changes that can be experienced by other entities. The types of events (state change) that can take place in a given environment are identified are shown in Table 6.12.

Services as Source of Event Services are the smallest important functional units in the proposed SOA based system. Change in their state is an important source of events. In its simplest form Web Services can be developed based on publish-find-bind paradigm. By following the standard-based approach, the service description in form of Web Service Description Language (WSDL), the discovery supported by Universal Discovery Description and Integration (UDDI) specification and subsequent interaction achieved using Simple Object Access Protocol (SOAP).

Beyond this minimalist approach, appropriate implementation with other specifications in *Web Service Specification Stack*, more specific functionality can also be incorporated. Notification design pattern can be achieved in web service by realizing the WS-Notification Specification[128]. WS Notification is a set of three specifications namely: WS BaseNotification, WS Brokered-Notification and WS Topic. WS-Topic deals with subjects or items that can be of interest for notification. A Web Service can publish a set of topics that a client can choose to subscribe. In case of a change in subscribed topic, the subscriber is sent notification of the event. The specification also supports the hierarchical or tree based structure of topics. Subscriber can directly subscribe to desired topics or can use the broker service to manage their subscription. In case of brokered notification, publishers register the published topics to the broker and similarly subscribers manage their subscription to published topics through a broker. Regarding the possible topics in Web Services, any state change in defined resource property can be published.

Grid as Source of Events The Open Grid Services Architecture (OGSA) Specification offers a framework for building standards-based grid services [128]. The security services provided by Grid Security Infrastructure (GSI) that is responsible for the membership to the VO, the monitoring and discovery service (MDS)[129], the job scheduling and execution monitoring services[130] are some important services that the middleware communicates with. The status of submitted job, status of grid resource, or the membership to the VO are some sources of events that other services may track for change. Hence, mechanism to detect such grid service specific events[131] is of critical importance to the communication middleware. For facilitating the required monitoring of grid resources [132], access to these services can be provided to the end-user using appropriate portlet technology.

Data Sources as Source of Event The data sources like a sensor, a network of sensors or human observers are considered as grid resources. They are uniquely identified across EOC VO, and can create and consume various control and information messages from the middleware. The status of the data source is also a relevant source of events as it allows tracking the availability of source in a dynamic environment.

Inference as Source of Event The events discussed till now, forms a set of event that can be monitored, verified and accessed programmatically. These state-changes are explicit and easy to capture. There can be another set of events that are not identified explicitly but can be inferred based on certain rules. The rules that govern inference of implicit events can be realized by hosting an inference service in the Grid. The inference engine is referred to with new assertion for detection of any implicit events. The reasoner is capable for calculating the membership of given fact to specific classes that are defined as event. In summary, event view of the situation awareness system is helpful in bringing out event driven nature of the system. It also demonstrates event space of the system.

6.5.2 Middleware View

The proposed system is realized as domain specific middleware services. The domain here is the situation awareness. The requirement of situation awareness capability is established in earlier chapters. The middleware view represents how requirements are met. The basic elements in the view are therefore the features of a situation awareness system. As these features can be realized with existing tools and technology support, the same are included as basic building blocks. As the system is proposed as a middleware, it is exposing services to clients and applications in the upper layer.

The entities in the middleware view interact with each other in form of procedure calls and responses. In the lower layer, that consists of distribution middleware that include SOAP based message exchanges. The domain specific middleware services at higher level utilize such lower level services in response to events detected and rules identified for the same. In this manner it solves the problem of Wild card capabilities of the lower level middleware components.

Prerequisites for the realization of middleware view are the configuration of the various building blocks. This configuration is guided by the rules defined for achieving the domain specific functionalities. These functionalities include realization of information processing models discussed in Chapter 3.

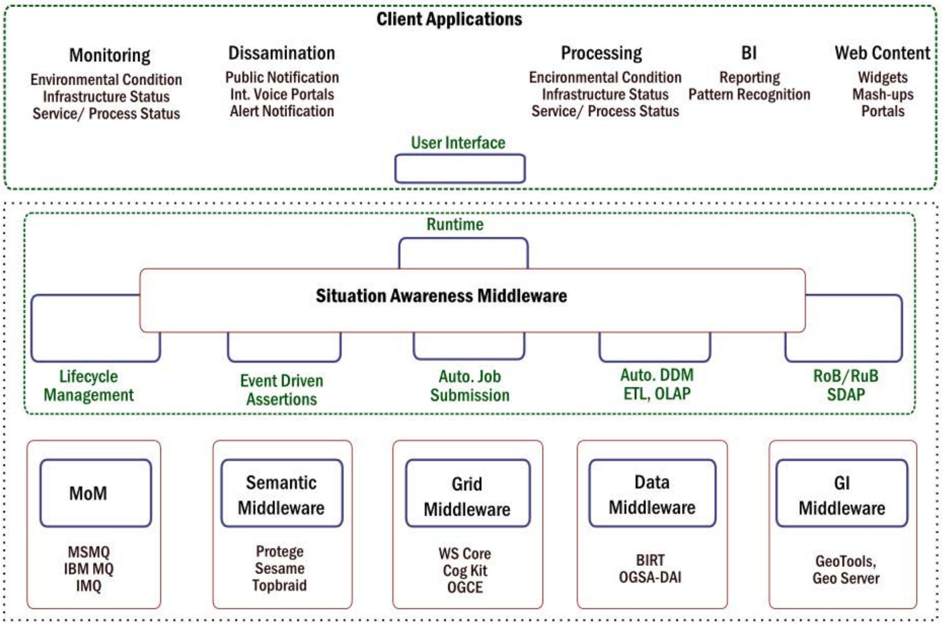

Figure 6.14: Middleware View of the Situation Awareness System

With this view, the situation awareness capability of the system is demonstrated. The functionalities like life cycle management of transient resources, event driven assertions, automated ETL and job specification generation, automated GI theme generation are the capabilities that are built over lower level middleware services. This is realized using various queries, generation, manipulation and representation of information. Figure 6.14 depicts the layers of middleware. MoM, Semantic Middleware, Grid Middleware, Data Middleware and GI Middleware are depicted as building blocks. Various situation awareness middleware capabilities enumerated above are realized by appropriate interactions among these blocks. These facilities can directly be consumed by clients with the help of GUI. Alternatively, these functionalities can be exposed to client applications. Dynamic web content realized in the form of Widgets, Mash-ups, portals or feeds could act as clients. Similarly, in accessing large amount of data, the business intelligence and reporting applications can be utilized. Monitoring, dissemination and processing applications can also be configured to utilize information processed by the middleware.

This capabilities are fulfilling the needs identified for situation awareness of individuals and organizations as defined in Chapter 2. According to the definition, individual requires information at specific granularities. Apart from basic monitoring data, individuals may require task specific actions, event notifications, alerts, guidance and other related information. Middleware services in proposed view depicts how these requirements are fulfilled with the help of middleware at different level of abstractions and exposed to client applications.

6.5.3 Runtime View

Views discussed so far provided representation from the internal part of the system. Event View depicted event space handled by the system, whereas the middleware view depicted how the components are utilized in realization of various middleware services. Yet, these views do not reveal how configured system components are serving the purpose. The runtime view is proposed as a solution to this problem.

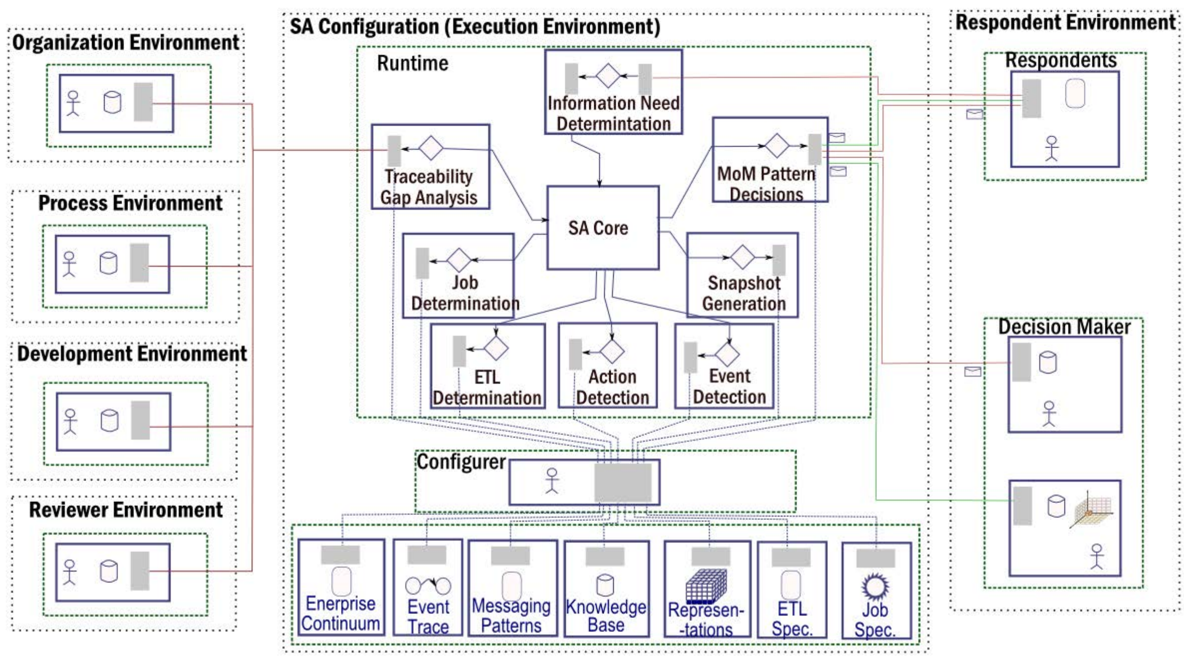

Figure 6.15: Runtime View of the Situation Awareness System

As runtime view is being proposed to reveal the configuration of various entities at runtime; it consists of the users, their interfaces to the system, system components, and various transient entities that are managed by the system. Actions, events and reaction to the identified events by users and system provide the basis of interactions. Prerequisite for this view is the system configured appropriately. This includes configuration of building blocks, various resource pools and various role sets.

This view demonstrated capability of handling life cycle of transient entities. Various queues realized to perform scheduled jobs according to rules. Continuous gap analysis provided to the users. Alerts provided to the users. Access to representations and guidance provided to the users. These capabilities fulfilled the need of systems in meeting situation awareness. The management of transient resources fulfilled the need of automation in services. The rules handling resource pool in the runtime addressed the need of handling available wild card capabilities of the resources.

6.5.4 Configuration View

In runtime view, many transient resources are created and handled by the system. Apart from this, work products from the users are collected, architectural products and various components are stored in enterprise continuum. Various representations are also created in the runtime. Many of these are potentially useful in other scenarios. Hence, it is necessary to establish reusability of the discussed entities. The configuration view is proposed as a solution.

Determination of reusable components followed by accessing the required one and sharing the created resources are next logical step. This is achieved by appropriate configuration. Hence, these entities are the primary focus of the view. As discussed earlier, such entities include transient entities, representations, events, alerts, notification, architecture products, work products, guidance and other items generated during the runtime or created in the configuration. This also includes knowledge base consisting of rules, mapping and concept definitions.

Based on reusability and relevance these products can be separately identified in to four classes. Figure 6.16 represents quadruple of generated artifacts. They are classified on the basis of spatial and temporal relevance. The generated artifacts are valid either for long term or short term. Similarly, in spatial dimension, the generated artifact is applicable locally or globally. Based on such classification, it is easy to determine how configuration will be required to share and reuse the available and required artifacts respectively.

These work-products are outcome of various users active in distinct work environments. As

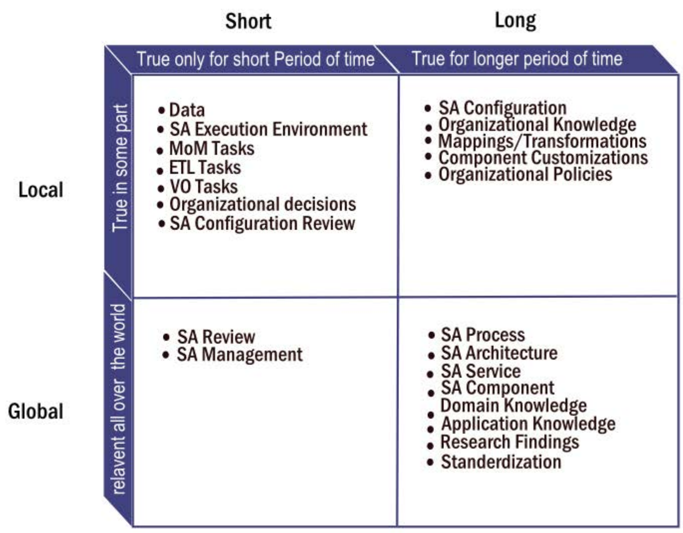

Figure 6.16: Qudaraple of Artifact Generation

a prerequisite, all such actors must be connected to the configuration. They must be sharing the work-product outcomes that are created as per the guidance provided by the configuration. These outcomes are created for specific purpose that can be attributed to some rule and the rule also can be traced to occurrence of event. There can be multiple such events that trigger various rules resulting in numerous work products. Hence appropriate Meta data clearly identifying the purpose and nature of artifact is required. The appropriate strategy for determining the suitability of reuse in spatial and temporal dimensions is required to be established.

By fulfilling these prerequisites, it is possible to define configuration of a system. Figure 6.17 depicts the type of configuration required to access or share four type of artifacts depicted in Figure 6.16. The artifact that are relevant for short period of time and that are valid locally requires single configuration. The enterprise continuum is not required to share and import any other artifacts from external continuum. A scenario in which local but long-term configuration is required, enterprise continuum created (and currently archived) for short term can be utilized to cover the longer period. Hence multiple instance of enterprise continuum for the same location is configured to meet the need. In case an enterprise continuum contains globally relevant components, and may require additional relevant components that are created and hosted by other enterprise continuum, then required configuration as depicted in quadrants named global short and global long term.

This view demonstrated configuration capability of the system. This indicated sharing of generated artifacts and utilization of globally archived and shared artifacts that can be useful. This demonstrates the capability of enterprise continuum. Also, it provides mechanism of deter-

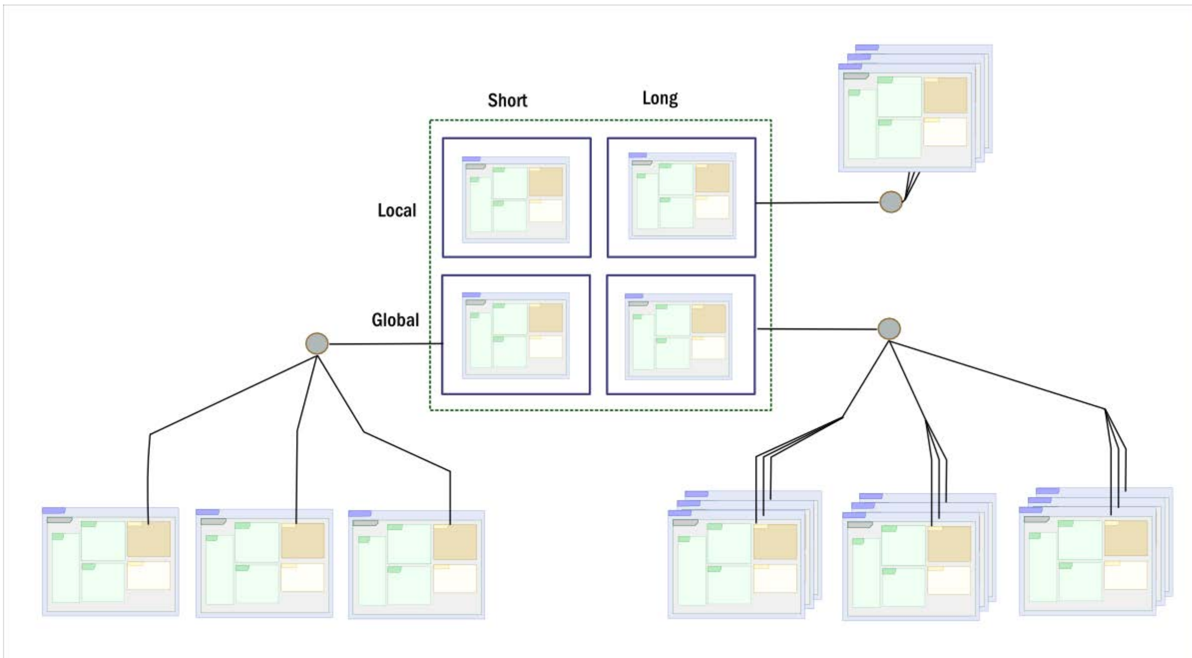

Figure 6.17: Configuration View of the Situation Awareness System

mining coverage and identifies gaps. As the shared resources are utilized in other configurations, it also provides mechanism of keeping track of artifact reusability. This capability fulfills the requirement of gap analysis. This also meets the need of traceability across configuration. Task allocation, reducing duplication of efforts is also demonstrated.

6.5.5 Role View

Role sets are introduced in Chapter 4. This included various actors performing some actions and creating work products. To aid the action, situation awareness is provided in the form of guidance, alerts, notifications and other relevant information. Architectural products are important means of providing situation awareness as they provide traceability and gap analysis. How various role sets are utilizing the situation awareness is important. This is captured in Role View. Basic elements in role views are role sets, artifacts produced by them and the architectural products utilized. Interaction in role view is depicted in the form of user-system communication. System provides consistent representation of architectural products, establishing traceability and gap analysis. The users on the other hand, provide task related information about status of the generation of work products. Prerequisite to the role view is configuration in which all role sets engaged in producing work products are connected to the Situation Awareness configuration. They access the architectural products identified suitable for the role. Also they provide information about the status.

This view demonstrated capability of SA Configuration in linking role sets active in various

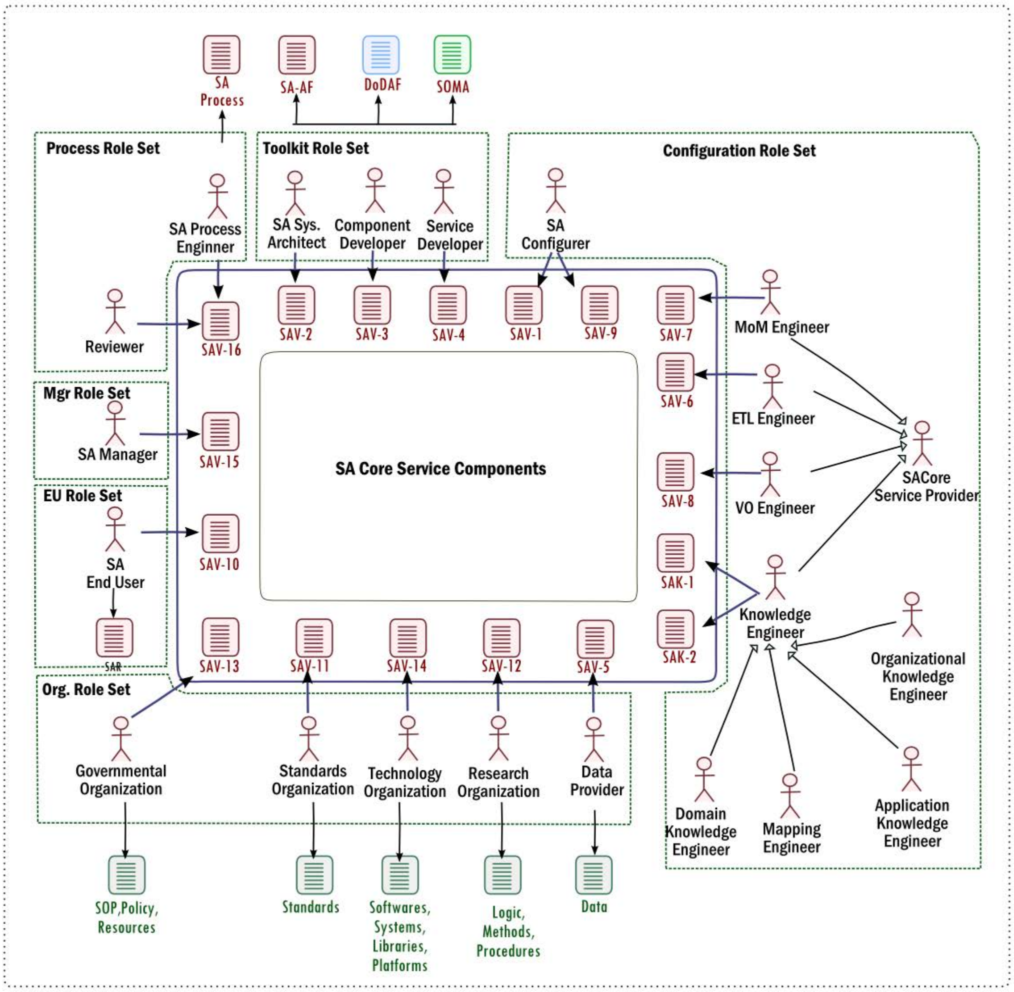

Figure 6.18: Role View of the Situation Awareness System

work environments to be connected to single unified configuration. For example organizational process, configuration and end user role sets are in active in different environment yet, they are depicted to be connected uniformly in SA configuration. Even though various users are connected to the same configuration, they have separate architecture products. This demonstrates the *Separation of Concern*. This is allowing the user to access and use only required information without manually filtering off all the information handled by the system. This fulfills the requirement of establishing cross environment traceability. For example, the standard created by standardization organization in organizational role set can be traced to the product that complies with the same. Hence, though in different environment, the organizational actor can trace the use and check the performance and related detail of the work product outcome.

6.5.6 Service View

Services are important building block of the system. Many domain-specific middleware functionalities are exposed as service or consume service. Service, Service component, execution environment and related role are the basic elements of the service view. All the stakeholders related to the given service may not interact directly. Some of them are having implied links among them while others are directly connected. Other can interact by having work product configured and monitoring the performance or just by fact of being utilized.

Prerequisite for service view is the prior knowledge of how service is composed. How various building blocks of the service are available as a outcome of some activity of a role.

This view demonstrates the capability of SA architecture for being able to dissect a service to all possible unit contributions as outcome of user activities. It shows how a single service can be monitored and characterized beyond WS * protocol stack to reveal the complexity. By providing links of each aspect of a web service to associated user role sets, the view established the capability of providing traceability and separation of concern. With establishment of web service characteristic various component level, the basis for appropriate service governance is also created.

By considering an instance of a Web Service instance as an SA entity, various attributes can be established along with the impact footprint as defined in Chapter 3.5. A footprint of a web service, indicating its service covering a specific geographical area, for specific time providing a specified functionality or information as defined in SA Configuration. In case of service unavailability, the exact gap analysis and traceability can identify precise action requirement from appropriate role sets, in order to fulfill the service gap. Also, the contributing role sets can identify how the unplanned reuse of their contribution is taking place in realizing unique

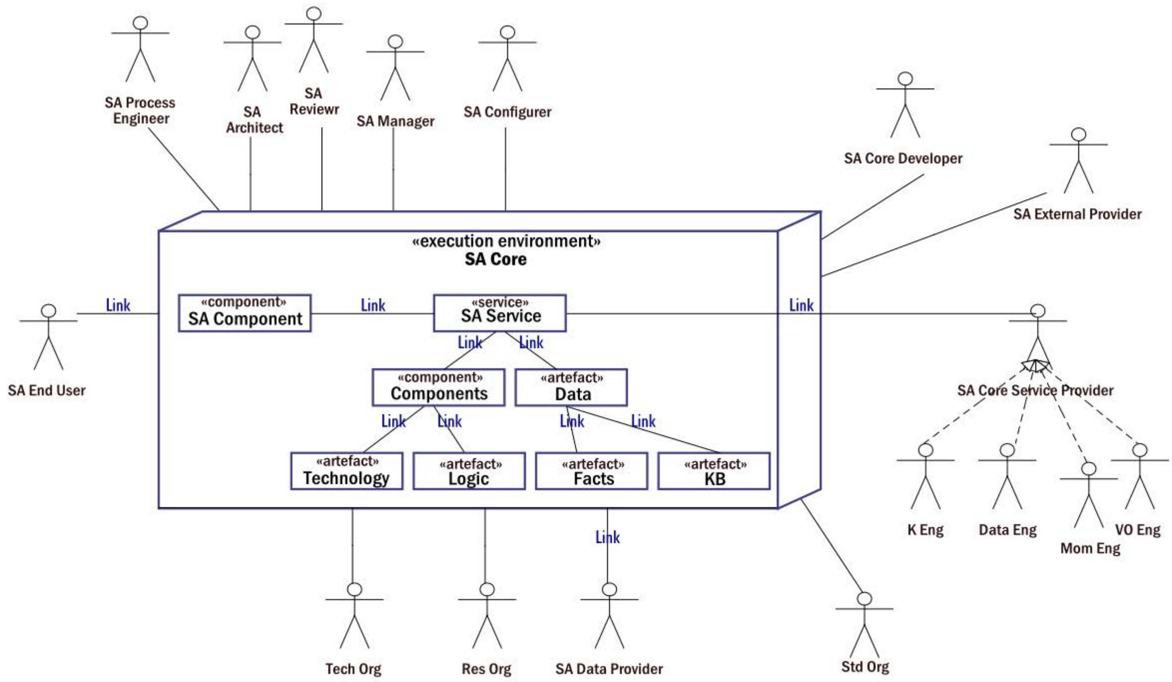

Figure 6.19: Service View of the Situation Awareness System

user objectives. In this regard, the service view offered by the SA Configuration provide unique capability to SOA.

6.6 Summary

This chapter presented views on realization of situation awareness capability in distributed computing environment for the requirements identified from initial chapters of the thesis. The proposal is made as extensible system that is built over existing components. These components have matured with research and development effort of vast community and compliance with standards. The extensibility of the proposal ensured that the proposed system can be carried out as community contributed extensible plug-in based project. In its primitive version, the system is currently identified with basic minimum functionality of situation awareness domain. Collaborating contributors can extend the functionality to meet the newly identified needs. In summary, the chapter discussed middleware, patterns and framework as described by Schmidt and Buschmann [91] in reference to situation awareness capability. The unique features and capabilities of the proposal are depicted with the help of various views.

Chapter 7

Research Experiment

The contribution of the thesis is an information management methodology; carefully engineered to meet the information needs in complex dynamical environment. The theory to determine access to information in the form of situation awareness, the characterization of situation with the help of modeling and subsequent information processing in collaborative environment is central to the proposal. The method content, and artifacts and system proposed that realize the required functionality is provided as support to the main proposal.

Any research experiment providing proof of concept must establish the claimed functionality in specific scenarios for which it is proposed. For given proposal, the target application domain requires collaborative effort by role player in various stakeholder environments. The complexity is observed only when contributions from various users depend upon outcome of others. The dynamism can be experienced only when it is observed over a period of time. Such aspects are difficult to establish in lab environment. Yet, a considerable amount of proposal deals with information processing in particular manner that also amounts to be important component of the contribution.

Long period of gradual improvement in the systems are known as stasis that indicates the planning phase according to punctuated equilibrium theory discussed in Chapter 2. The disturbance created, as event that triggers short term response to the event is known as punctuation. The information processing is important component in punctuation period, as it realizes information flow required for situation awareness. The punctuation scenario is therefore useful in establishing the information management capability of the proposed approach.

Figure 7.1 depicts a feed back control loop realized with the information flow among various entities in UoD. The scenario indicates the post-disaster event scenario with relevant entities. Relief camps are setup as response to the event. They are scattered spatially in various admin-

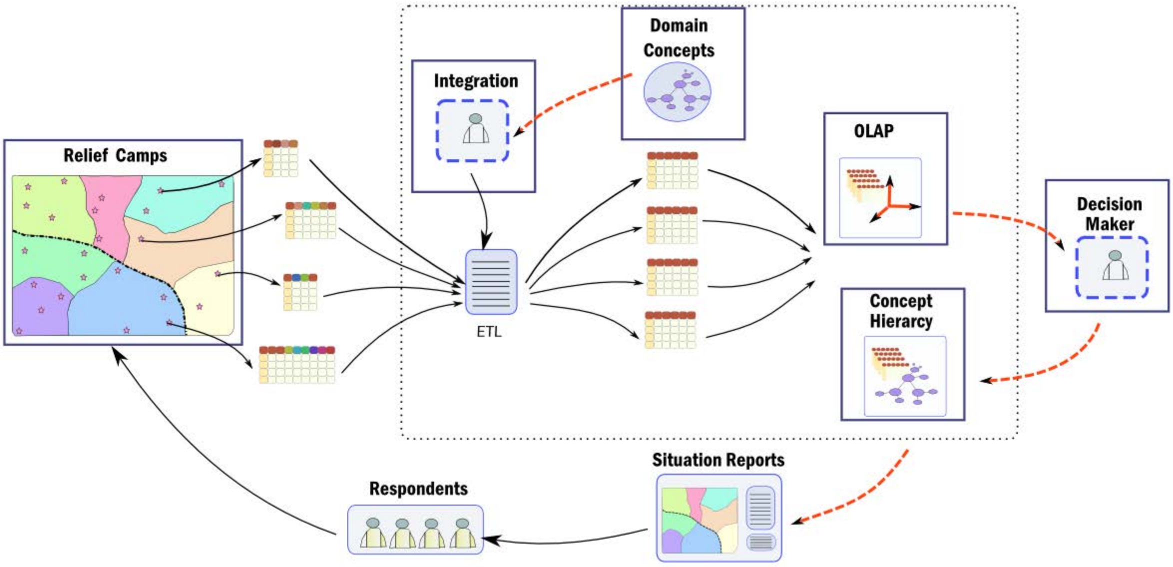

Figure 7.1: Situation Awareness Use Case Scenario

istrative units in a given area. The information-processing unit is indicated as a set of elements delimited with dotted lines. This includes extraction, transformation and loading (ETL) of raw data collected from camps to make it useful for analysis and decision support. To guide the integration and transformation, the domain concepts are utilized as reference. Multidimensional information generated as outcome of the process provide information at specific granularity to the decision makers and the end users. The information product prepared as situation report includes the rich representation of situation on ground, the decision regarding required action and other related information to keep the situation in desired status. The stakeholders refers to the situation reports and carry out the activity in UoD at camp levels, that alerts the ongoing situation. These changes are detected by the information processing strategy and situation reports are appropriately updated. In this manner the feedback control loop is realized in complex dynamical system.

Apart from realization of feedback control loop, the proposal requires support for soft system methodology that involved rich representation of problem and set of actions required from various stakeholders which is included in the situation awareness reports. This scenario also indicates stakeholder activity in natural and built environment, reporting environment and configuration environment.

7.1 Punctuation Scenario

The Emergency Operation Center (EOC) is the central coordinating agency that monitors and controls resource allocation and overall response work carried out by various service providers and organizations. Emergency Medical Service is one of the critical services that is rendered by a team of actors serving every camp setup in response to a disaster. Based on the inflow of victims and medical conditions obtained by them, the team is required to communicate with EOC for the decisions necessary for allocation of resources. Actors, who assume the responsibility of such communication, therefore must have provision to connect to EOC Service and be able to engage in reporting the status as well as seeking required assistance.

For the purpose of testing the capability of the proposed framework, the following functionalities are identified:

- Upload and report camp specific information to the EOC,
- Determine entities, resources and actions required to respond according to instantaneous plan of the EOC,
- Determine information needs and collect data accordingly,
- Handle and process collected data,
- Detect events, and determine actions,
- Notify role specific messages to available instances,
- Aggregate and load data on other nodes.

7.1.1 Experimental Set Up

Basic requirement towards the experimentation is to setup the SA Middleware components. Ontology is created to cover concepts, relations and rules that are required to support the representation of requirements in the given scenario. The MoM service is created by configuring MoM features provided by the recent Glassfish stable release. Grid infrastructure is setup using the Globus toolkit realizing time synchronization, certification, service execution environment, job management and data services. For exposing the data resources, the OGSA-DAI component is configured on Grid Instance. The GI service is deployed and configured as provided with GIS server product. The runtime is in the mode of listening to the requests made by the clients.

Knowledge Base

As discussed in Section 3.4 KB holds domain-independent, domain-specific, local-specific and application-specific concepts. Among these, the domain independent content include the general concepts and properties like natural and man made entities, resources, along with their common properties related to space, time, classification physical objects that are required by any application.. The domain specific concepts include vocabulary, rules, facts, measurement methods etc. related to specific domain. For given scenario, domain specific concepts include the specific provisions required for emergency health services, captured in the form of concepts, properties and rules. The local specific concepts include the name and orientation of administrative units, organizational hierarchy and operating procedures identified by them. Application specific concepts include messaging and information processing related concepts like event, profiles, subscription, notification etc. Application specific knowledge representation also contains the description of SA Process concepts that lead toward traceability and gap analysis. Figure 7.2 indicates one such concept of event in Protégé knowledge engineering environment. As a

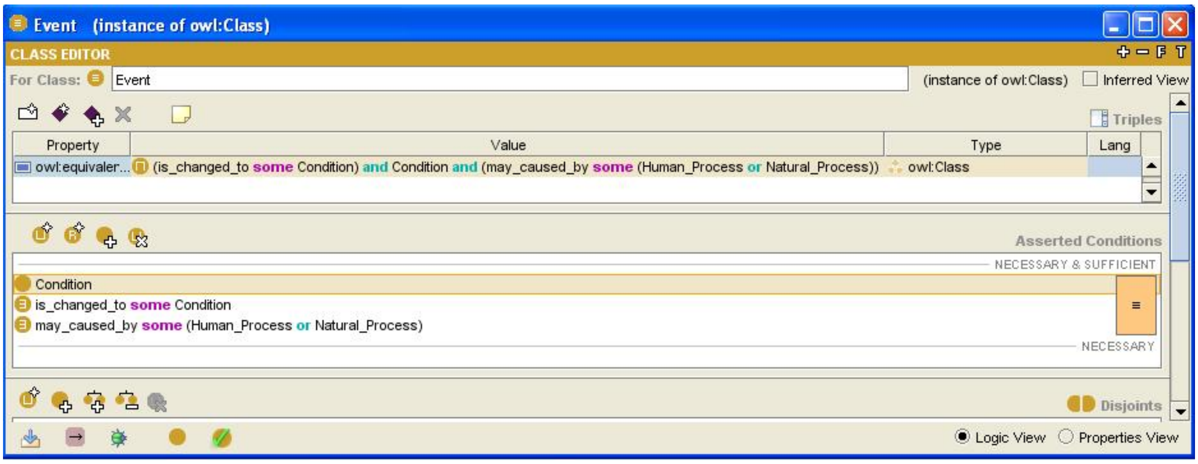

Figure 7.2: Event Concept in ontology

result of comprehensive knowledge representation process, a knowledge base with complex concepts, properties, relations, and rules is generated. Visualization capability of the tool reveals the captured complexities. Ontoviz provides visualization of complex interdependence of the concepts as indicated in 7.3.

Middleware Services

Middleware was introduced in chapter 6 as providing reusable building blocks. Semantic Middleware provides service provides important functionality as it exposes the KB to various components and services of the system. Message oriented Middleware service provided required

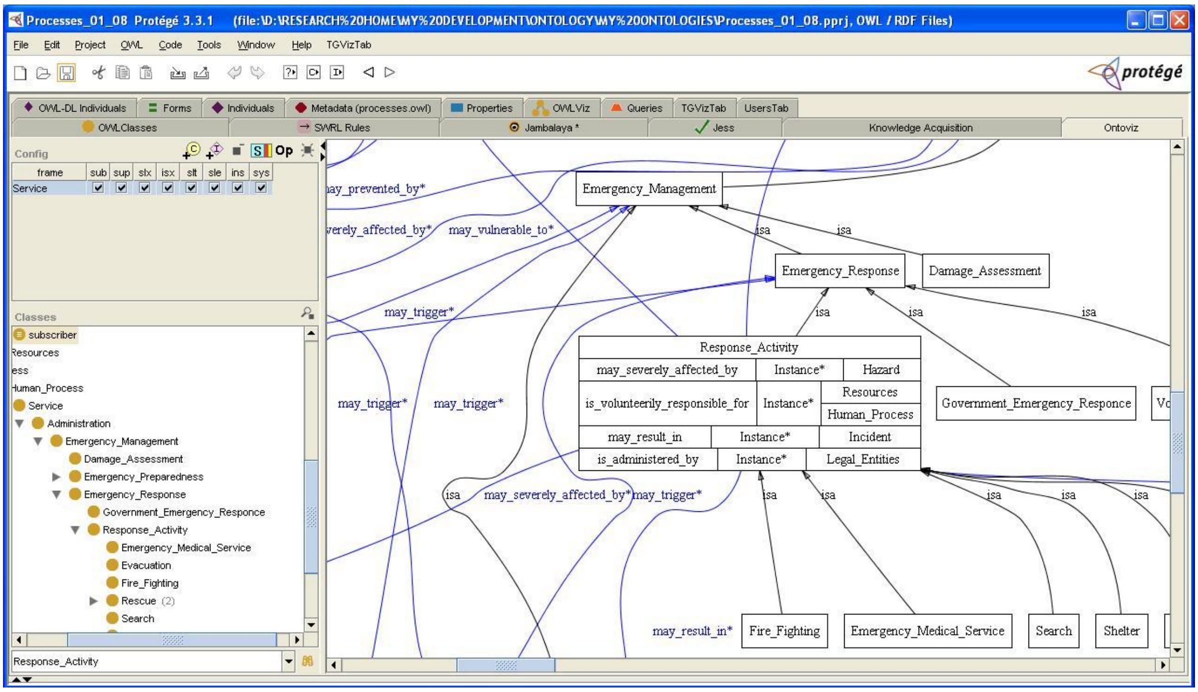

Figure 7.3: Relation of conceptes

communication facility. The Grid Middleware provide required execution environment. Data middleware exposes various types of data resources and enable access and administration of them. As these services are critical building blocks in rendering situation awareness services, the configuration and testing is discussed in this section.

Semantic Middleware Services Set of services are required that can expose the KB to the clients. Semantic middleware have important responsibility of handling KB that is one of the core building blocks of the situation awareness service. The KB is an outcome of knowledge representation task from various collaborating knowledge engineers, representing the domain independent, domain specific, local specific and application specific concepts. Among many alternative ways of knowledge engineering, the method for building KB with Protégé has been discussed in previous section. This knowledge base in available in desktop environment of the knowledge engineers and should be exposed as a service in client-server environment. This enable access to KB by various web based clients. This requires a semantic repository that can expose the database and allows various options. The OpenRDF product Sesame provide semantic web repository. It enables creation and handling of KB encoded in popular formats like rdf, owl, triples etc and allow exposing in server environment. This is stored in memory or persistent repository to ensue fast access. The Storage And Inference Layer (SAIL) API exposes semantic data stored in physical files over standard web protocols in web server environment. This can later be explored and managed with web based client application OpenRDF- Workbench or by custom made programs. Figure 7.4 provides a snapshot of openrdf workbench web based client. It indicates that KB encoded in the form of `SA.owl` is exposed in a default repository environment. The client allows exploration of concepts.

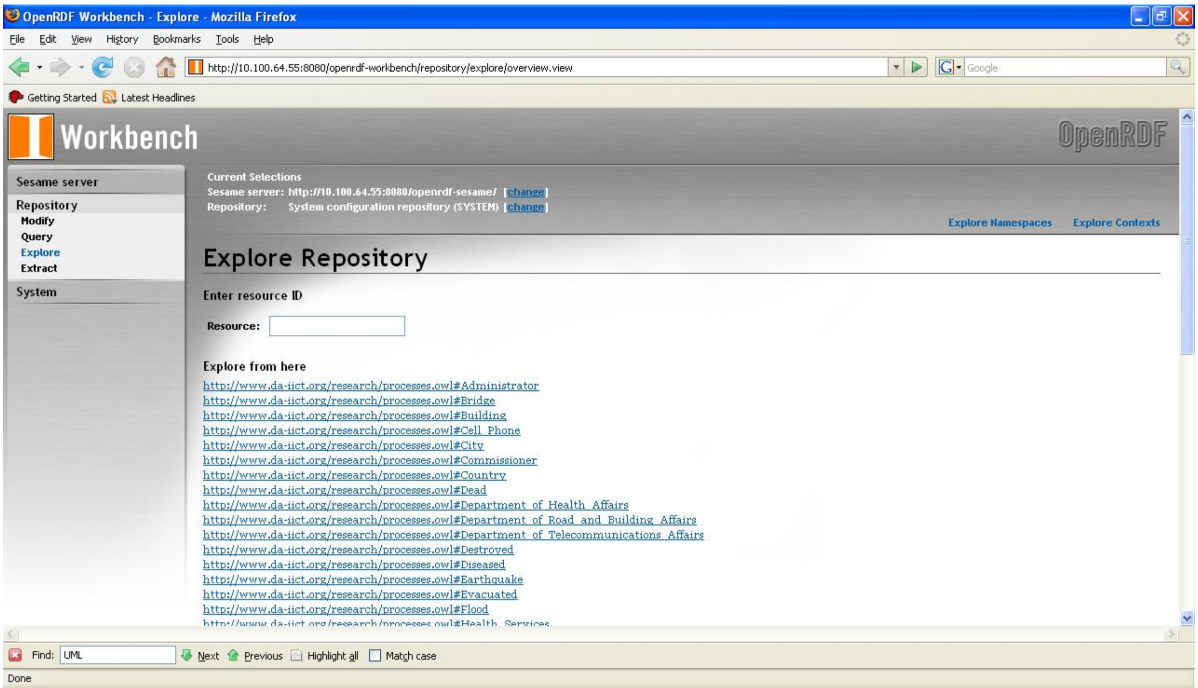

Figure 7.4: KB Repository

The openrdf workbench is not useful as a middleware service as, it requires manual intervention from the users. In application scenario, the KB should be access by other services programmatically. The programmatic access to the KB is ensured via query facility. The URL of the sesame server and credentials of the repository work in the fashion similar to database server and specific instance of database. As in case of database, connection is established to the server and query statements are prepared on the runtime, the similar approach is allowed by the sesame service.

The access is allowed in the form of providing resolutions to semantic queries made to the KB. Various types of usage scenarios are possible that employees semantic queries. It provides support for SeRQL (Sesame RDF Query Language) and SPARQL (Simple Protocol And RDF Query Language). The SPARQL is the W3C standard language for querying semantic repositories. It supports three types of query namely Select Query, Construct Query and Boolean Query, each useful in the realization of information processing tasks.

An *Ask Query* is particularly useful feature in information processing requirements discussed in information processing model. When an instance of a classed represented in KB is to be created or it's status is to be modified, it should first be looked up in the KB. The information about an instance is provided by the client in the form of a string. By supplying client supplied string allowing with it's language, ASK query can support determination weather the instance is available or not. For example is a report regarding basic phone service in local area needs to be asserted, the `"Basic-phone-service"@en` is used with ask query to determine its instance in present KB. The query is represented as follows:

```
PREFIX sa:<http://www.da-iict.org/research/processes.owl\#>
ASK
WHERE {?x ?y "Basic-phone-service"@en}
```

The ASK query is a boolean type of query, and the result is either TRUE or FALSE. If it finds an instance for the entered string `"Basic-phone-service"@en` in the repository, query response is as follows:

```
TRUE
```

The ASK query provided simple, yet important look up capability to the KB, but in application, more complex queries are required. For example, upon availability of an instance for the user string `"Basic-phone-service"@en` requires identification of more features. As discussed in information need determination, the concepts related to the given concepts along with important attributes must be identified. In other words, it must be determined that how this given concept is related to other concepts in KB. This is supported with a Describe Query. Following usage of a describe query allows identification of all concepts related to `"Basic-phone-service"@en` .

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns\#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema\#>
PREFIX sa:<http://www.da-iict.org/research/processes.owl\#>

DESCRIBE *
WHERE {?x ?y "Basic-phone-service"@en}
```

As the describe query is employed to determine relation of given concept with others, the outcome is a collection of statement each representing relation with other concepts. This relation is created in the form of triplets. The following is the result of describe query.


```
 <?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:sa="http://www.da-iict.org/research/processes.owl#">
```

```
<rdf:Description rdf:about="http://www.da-iict.org/research/processes.owl#HasBrandName">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<rdfs:domain rdf:resource="http://www.da-iict.org/research/processes.owl#Service"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</rdf:Description>
<rdf:Description
rdf:about="http://www.da-iict.org/research/processes.owl#Dot_Guj_Ahmedabad_Telecom_District">
<sa:is_legally_responsible_for
rdf:resource="http://www.da-iict.org/research/processes.owl#B_Fone_Gujarat"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.da-iict.org/research/processes.owl#B_Fone_Gujarat">
<rdf:type rdf:resource="http://www.da-iict.org/research/processes.owl#PSTN"/>
<sa:HasBrandName xml:lang="en">Basic-phone-service</sa:HasBrandName>
</rdf:Description>

<rdf:Description
rdf:about="http://www.da-iict.org/research/processes.owl#Navrangpura_Telephone_Exchange_Building">
<sa:is_mendatory_requirement_for
rdf:resource="http://www.da-iict.org/research/processes.owl#B_Fone_Gujarat"/>
</rdf:Description>

</rdf:RDF>
```

While describe query simply reproduces relations available in KB, sometimes in application scenario, this feature in not sufficient to meet the information processing needs. The outcome of the query should be constructed to meet the further information processing needs. This requires construction of customized graph from query result. This feature is supported by a construct query.

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX sa:<http://www.da-iict.org/research/processes.owl#>
PREFIX saappont:<http://www.daiict.ac.in/SAapp.owl#>

CONSTRUCT {?x saappont:eventSubscriber saappont:B_Fone_Gujarat }
WHERE {?x sa:is_legally_responsible_for sa:B_Fone_Gujarat}
```

Following is the outcome of construct query. As specified, the resulting concept is annotated with `saappont` namespace that is determined in `SAapp.owl` that contains application specific concepts.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:sa="http://www.da-iict.org/research/processes.owl#"
xmlns:saappont="http://www.daiict.ac.in/SAapp.owl#">

<rdf:Description
rdf:about="http://www.da-iict.org/research/processes.owl#Dot_Guj_Ahmedabad_Telecom_District">
<saappont:eventSubscriber
rdf:resource="http://www.daiict.ac.in/SAapp.owl#B_Fone_Gujarat"/>
</rdf:Description>
</rdf:RDF>
```

The few examples of semantic query introduced in discussion above, provide overview of how various semantic queries discussed time to time in this proposal is realized in execution environment.

Message oriented Middleware Services Communication among physical entities, various interaction patterns must be supported. For this task reusable building blocks are available. Section on architecture introduced MoM, The basic mom services are accessed using JMS, JMX and IMQ features provided by existing middleware technology. Java Message Service (JMS) API provides framework for realizing destinations (topic or queue), connection factory, connection, session, message producer, message consumers that realizes the required communication patterns identified in modeling section. Message queue MBeans provides mechanism for monitoring and administration of various instances supporting the MoM Patterns. It supports handling of message brokers, connection services, connections, destinations, message producers, message consumers, transactions, cluster of brokers, logging and other related aspects. For example, `Imqbroker` provides utility to monitor the message broker deployed in the configuration. Figure 7.5 represents important utilities like `ListDestinations` that enables monitoring of destinations. In the background, an instance of a broker monitoring feature is visible indicating the `ETLJobQueue` holding some messages.

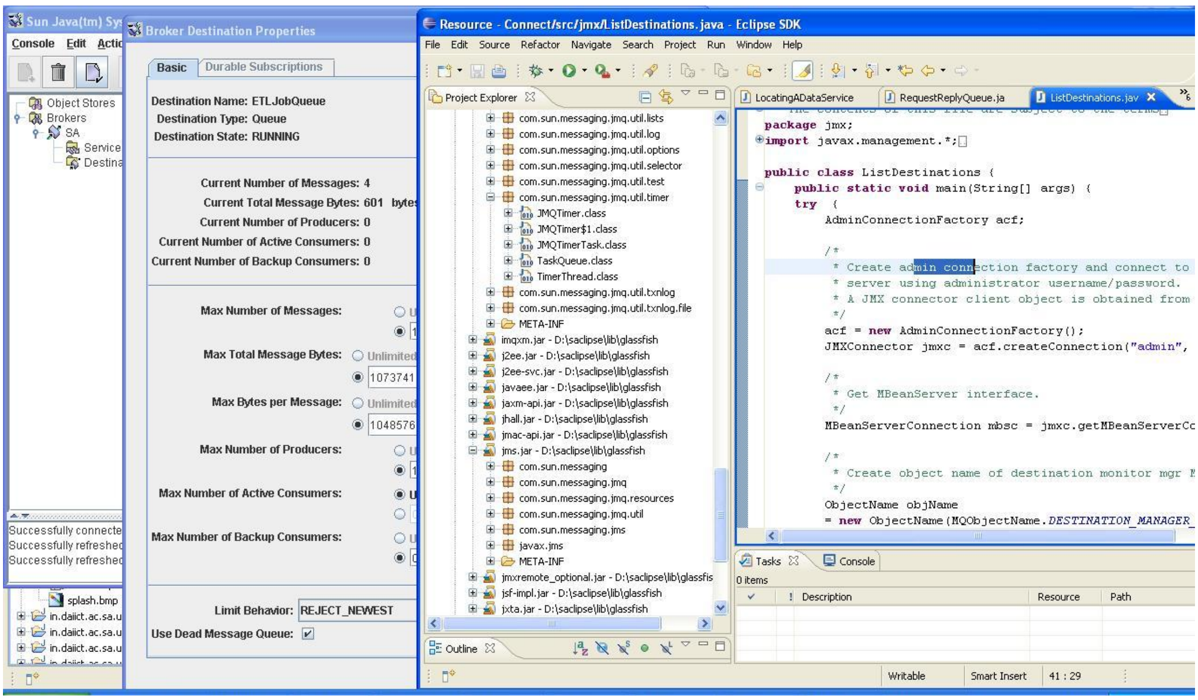

Figure 7.5: JMX and Configuration Queue

Grid Middleware Services Open Grid Service Architecture (OGSA) introduced set of services that are useful in bringing security, execution environment, registry, data and many other aspects of distributed computing in grid environment. These services providing important building blocks for realizing situation awareness services.

Building of a Virtual Organization(VO) for all the collaborating individuals and organization is the basic requirement identified in Chapter 6. Various technology solutions are available that allows building of virtual organization. For setting up grid environment, the Globus toolkit is utilized that include configuring building VO and other components of grid [133].

Various features of grid are implemented on available set of computing nodes. As a part of installation, some nodes are setup to play specific roles. The temporal reference is a recurrent requirement through out the information-processing task. In distributed computing environment, consistent temporal reference becomes necessary. This is realized by assigning a node to provide time reference over Network Time Protocol (NTP) service. The creation of VO is subjected to grid security service that allows assumption of various roles in Virtual Organization. Certification Authority (CA) is established the role of assigning membership rights. The membership to VO is assigned with certificate mechanism. The following is the content of one certificate.

```
Certificate:
   Data:
       Version: 3 (0x2)
       Serial Number: 2 (0x2)
       Signature Algorithm: md5WithRSAEncryption
       Issuer: O=Grid, OU=GlobusTest, OU=simpleCA-ddg36.egrid.daiict.ac.in,
               CN=Globus Simple CA
       Validity
           Not Before: Sep 27 06:05:43 2006 GMT
           Not After : Sep 27 06:05:43 2007 GMT
       Subject: O=Grid, OU=GlobusTest, OU=simpleCA-ddg36.egrid.daiict.ac.in,
                OU=egrid.daiict.ac.in, CN=vikram sorathia
       Subject Public Key Info:
           Public Key Algorithm: rsaEncryption
           RSA Public Key: (1024 bit)
               Modulus (1024 bit):
                   00:ac:a0:d7:12:52:c3:af:55:bb:3c:29:68:8e:df:
                   b9:1b:97:57:b6:c3:39:46:1d:c0:28:95:25:da:4a:
```

The certificate is issued by a certification authority setup in lab environment. The globus toolkit utilized for setting up of grid environment allows creation of Simple CA - a simplified version of certification authority sufficient for use in lab environment. For real life implementation, community level CA is required to be setup. Specific provisions for issuance, storage and retrieval of security certificates are also required. In given example a computing node with hostname `ddg36` of domain `egrid.daiict.ac.in` is issued a certificate for specified period.

The globus toolkit installation provides installation of web container. The successful installation allows installation of `wsrf` based services. 51 services are installed as a part of globus installation. The monitoring and discovery service, job execution service, index service and various other services are installed. Figure 7.6 indicates these services. The installation so far

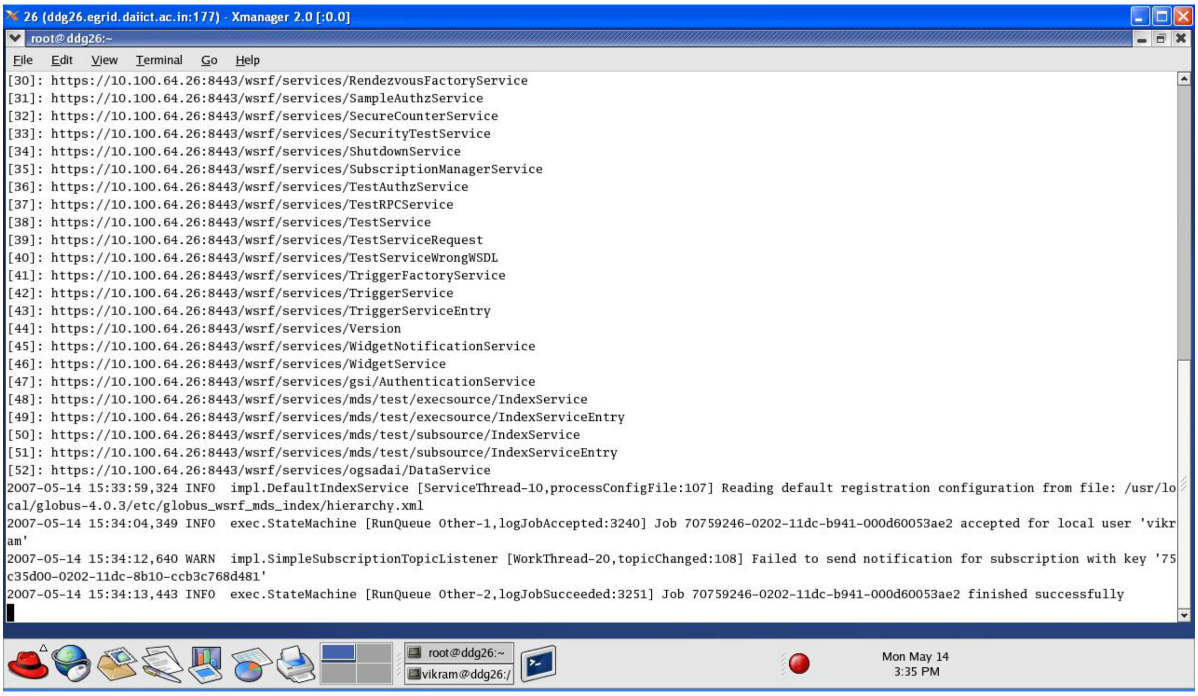

Figure 7.6: Globus Container

provides consistent temporal reference, security service and other aspects of building VO. One of the important features required is support for job execution. The `globusrun-ws` is utility provided by the globus toolkit. This service is utilized to execute job by appropriately specifying job details. As an example to test the installation, the following sample job request and response can be identified for the given installation.

```
[vikram@ddg26 ogsadai-wsrf-2.2]$ globusrun-ws -submit -s -c /bin/hostname
Delegating user credentials...Done.
Submitting job...Done.
Job ID: uuid:3f79448e-01db-11dc-84ab-000d60053ae2
Termination time: 05/15/2007 05:23 GMT
Current job state: Active
Current job state: CleanUp-Hold
ddg26.egrid.daiict.ac.in
Current job state: CleanUp
Current job state: Done
Destroying job...Done.
Cleaning up any delegated credentials...Done.
```

In the sample job request test above, the job execution in grid environment is specified. First, the security credentials are checked for given user issuing the job request. A unique job ID is generated by the execution environment. During the job execution, various job states are monitored and reported. Various job specifications, automatically generated by SA middleware are submitted in the grid environment in similar fashion. Grid portlets are another mechanisms for submission of job in web browser environment instead of console.

Distributed Data Management Service The jobs include handling of various types of computing resources, data sources and resources. Database is one important type of resource that is to be handled. For the nature of application domain, multiple types of data resources need to be handled. As discussed in Section 6, OGSA-DAI is employed for this purpose. Installation of OGSA-DAI is carried out over given instance of grid middleware. Once the installation is done properly, the data access service is reflected in available services list at the start of the container. `DataService` created for local instance allows distributed data management functionality to clients. This is further utilized for managing various data resources. Among supported types of resource, relational data is common requirement. Following is the configuration of an instance of `postgres` data resource.

```
dai.resource.id=PostgresResourceSADBon47
dai.data.resource.type=Relational
dai.product.name=Postgres
dai.product.vendor=Postgres
dai.product.version=
dai.data.resource.uri=jdbc:postgresql://10.100.64.47:5432/sadb
dai.driver.class=org.postgresql.Driver
dai.credential=
dai.user.name=sadb
dai.password=sadb
dai.driver.jars=/usr/local/GridSoft/pg74.216.jdbc3.jar,
dai.driver.jars.0=/usr/local/GridSoft/pg74.216.jdbc3.jar
```

Middleware projects extending the capability of existing data access middleware can be appropriately employed to meet the identified need. Handling of semantic repository is important feature required by SA middleware services. OGSA-DAI-RDF project discussed earlier supports exposing of semantic repository as `ogsadai` resources. Similar to SQL activities supported over relational data resources, the semantic query is supported over semantic resource exposed using OGSA-DAI-RDF middleware.

Once the instance is exposed through OGSA-DAI data service, clients are appropriately utilized with performed documents discussed in Chapter 6. ETL specification prepared in the form of performed document is executed as data service client in OGSA-DAI web service. The jobs are scheduled and executed with appropriate monitoring mechanism. Following is a sample delivery activity performed as job.

```
ant dataServiceClient
-Ddai.url=http://10.100.64.26:8080/wsrf/services/ogsadai/DataService
-Ddai.resource.id=PostgresResourceSADBon48
-Ddai.action=sa/SAPerform/Delivery/delRainfall.xml
```

The activity performed by client is handled appropriately by the OGSA-DAI service instance configured as `DataService` on a grid container. This holds data resource installed on other node of the VO. Figure 7.7 represents execution identified by the SA Middleware services in a client environment, that invoke service on a DataService installed over a grid container, executing an activity accessing a postgres instance on a seperate node. Various computing resources and

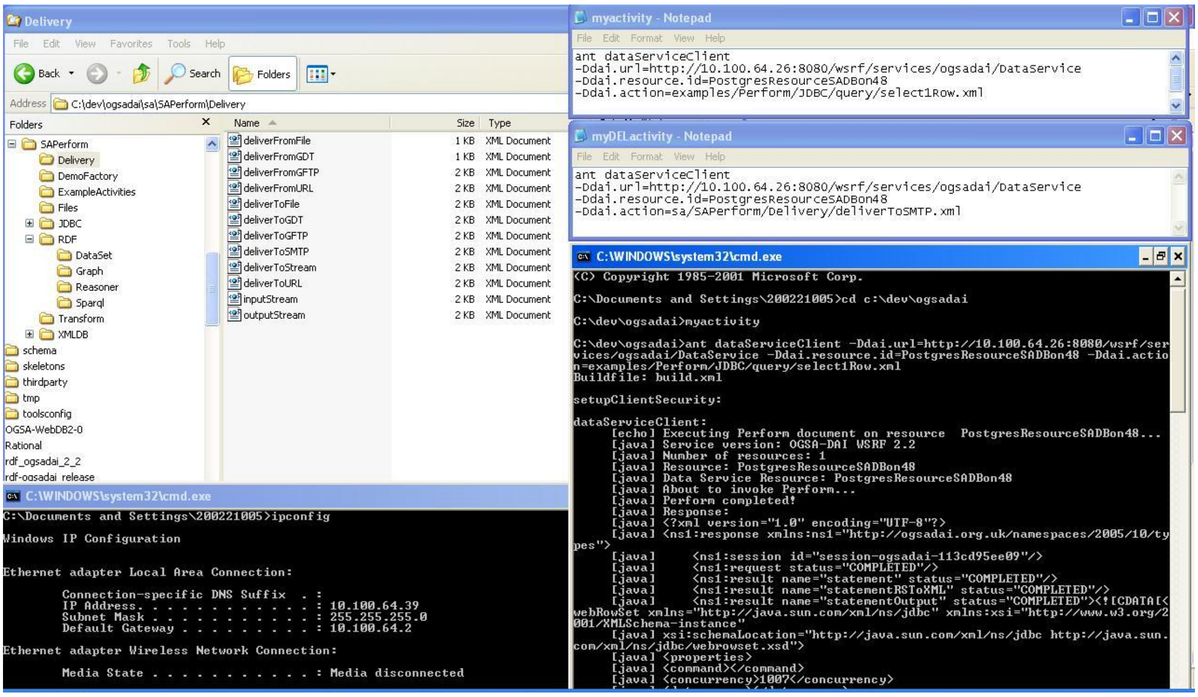

Figure 7.7: OGSA DAI Execution

data resources are configured to suit the need identified earlier. The resulting configuration represents setup of a Virtual organization. Figure 7.8 indicates VO created in lab environment. For execution environment it discussed a grid service container exposing various wsrf-based services. Certification authority and time server roles are assigned among available nodes. The VO involves database instances and semantic repository resources exposed by the DAI service. Resulting configuration is depicted in Figure 7.8.

7.1.2 Information Processing Steps

The information processing taking place during runtime is discussed here with the help of test scenario. A disastrous event takes place at time instance T_e. In response to this event a camp is set up to carry out emergency response activity. The respondents in camp are aware of a Situation Awareness configuration hosted by the coordinating agency; hence, they utilize the capability of the configuration to meet the information management needs. The occurrence of events is to be reported by a camp manager to the configuration in order to meet the

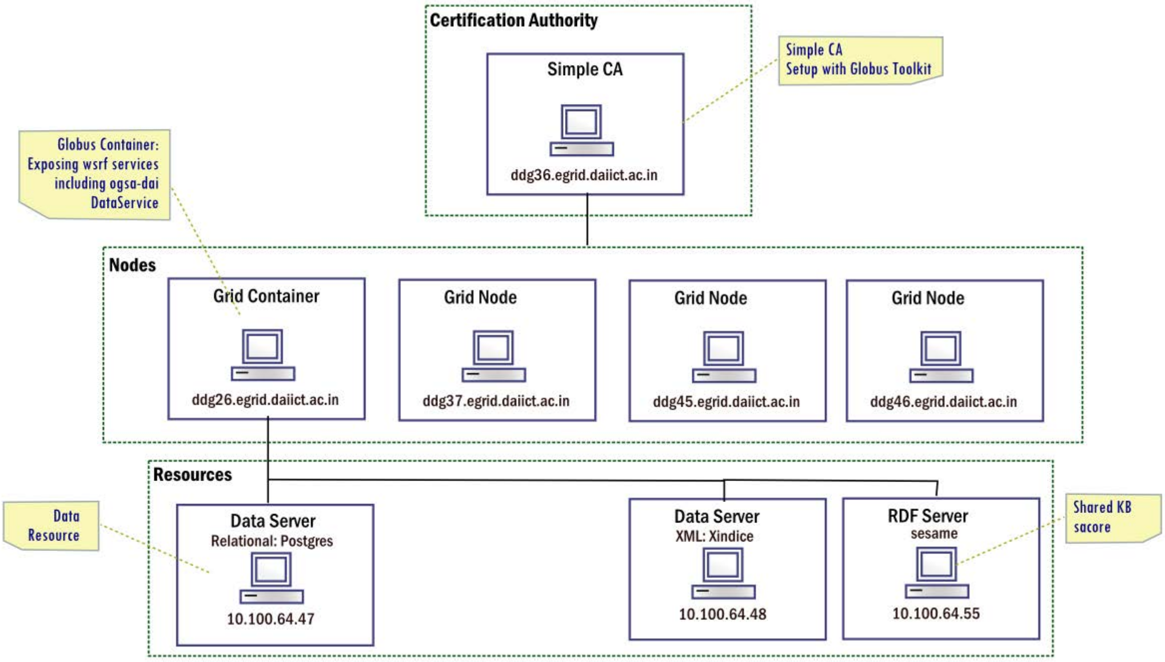

Figure 7.8: Experimental Set up

information needs of respondents, volunteers, resource providers and victims. Hence, the camp manger instance has to indicate the event and camp related information to the configuration.

```
Event Time: Te
Reporting Time: Te+Tr
Reported by: Camp Manager
```

Assertion

The event is observed by the respondent and needs to be reported in the system. The assertion strategy requires to be called upon. From the known KB URL of the emergency operations center, the respondent search for appropriate mechanism for reporting. Assertion box is provided by the configuration. According to the algorithm primary information need is determined and hence following information is collected from the reporter, which is essential for the required representation of the camp.

```
Subject: Occurance of an Event
       : Instance of a Camp
       : Instance of Entities in the Camp
```

According to the details of assertions above, the information regarding type of response, available entities and respondent instances are created.

Information Need Determination

According to the Algorithm 2, the information need is to be determined to fully characterize the situation. Hence, more than one views of the systems are required. Each view requires one or more properties of the entities or processes. Based on the footprint, each can be appropriately captured. The rules defined in the knowledge base provides following requirements.

```
Number of Patients in Camp
Collect from Camp Manager Every 5 min until response time
Number of Doctors in Camp
Collect from Camp Manager Every 3 hour until response time
Amount of Medicine in Camp
Collect from Camp Pharmacist Every 1 hour until response time
Amount of Food in Camp
Collect from Camp Pharmacist Every 6 hour until response time
Amount of Rainfall Recorded in the Camp Area
Collect from Camp Manager Every 2 hour until response time
Amount of Temperature Recorded in the Camp Area
Collect from Camp Manager Every 2 hour until response time
```

From the assertion it is known that camp is created as response to an event. The `Camp manager` and other role instances are available during the response. The response is planned for a period defined in the rules. This period is response time $T-r$. All the attributes are collected up to the planned response time. Spatial, temporal and impact footprints of all the required attributes are separate. Hence, each attributes needs to be collected from appropriate users at determined time interval and the rules given in observation and measurement models. Victim inflow, stock of medicine etc. are kind of information required more frequently than other information listed in the representation.

Information Specification Generation

Information needs established above are identified with details about what is needed at what frequency and who will provide the current values. This information is in textual format. For subjecting it to automated information processing and handling strategy, these needs are to be converted in to machine processable specifications.

```
Who: Camp Manager  instance
What: Victim inflow in camp instance
What format: number
When: Te+ 1.Tf
```

Specification of information need above is derived from the identified information need. Here, as it needs to be collected from the user instance, it have to be in the form of an infon introduced in Chapter 2. The attributes should be collected up to response time Tr. The required update frequency for attribute is T_f. The specification is created for time instance $Te + 1(T_f)$, $Te + 2(T_f)$, $Te + 3(T_f)$... up to all times instances that are occurring before the completion of the response time T_r.

The representation created in information need specification is in the form of a utterance situation as introduced in Section 2.2.3.

$$
\begin{aligned}
u \models \langle utters, speaker, \Phi, l, t, 1\rangle \wedge \\
\langle refers - to, speaker, camp, C, l, t, 1\rangle
\end{aligned} \tag{7.1}
$$

Messaging Pattern Generation

The utterance situation is determined by initiating utterance action from the observer. The instance of an individual who may provide this utterance may not know frequency and detail of utterances required. This is known and specified by the system. Hence, system should engage the instances in information interactions that are required to meet all the utterance required. This requires utility of MoM components by creation of appropriate MoM patterns. From the specification determined as above, the roles in message interaction needs to be defined along with the messaging system related other aspects. Following representation provide messaging patterns identified.

```
Pattern: Request-Reply
Requester: MoM Manager Instance
Replier: Camp Manager Instance
Channel: GML Data type channel
Message: GML Template
```

The interaction pattern is identified as request-reply pattern, where system sends request at particular time interval, which response is expected from the user. The requester is role is assumed by the system, and replier role is extracted from the information need specification. The message is in the form of GML template, and the channel required is the data type channel required to handle GML data. The lifespan of these patterns are determined to cover the response time T_r.

Data Collection Request

The messaging patterns created in last step, are to be utilized by sending and receiving of message according to determined interaction pattern. Time extracted from the information need specification is appropriately utilized to trigger the messages.

```
Channel: Request-Reply GML Data Type channel instance
To: Camp Manager instance
```

```
Content: GML message Request
Requirement: furnish appropriate value for Victim inflow in <template> from given Range
```

As this message is sent, the utterance situation is conveyed on the messaging middleware. Though, it appears to be similar to the utterance situation as depicted in Equation 7.1, it is extension of the classical feature. Typically, various types of situation described in theory components are of Boolean types, confirming if situation holds or not. The present form allows collection of situation with assignment of values to specific attributes describing the situation. In this sense, it provides extension by handling concrete domain property of logical representation.

Representation Generation

As utterance situation capability extended to handle the attribute values in specified namespaces, they are extracted and subjected to further processing. The information need specifications hold the template for which appropriate values is assigned by the individual instances. These individual situations are integrated to create a comprehensive view of the situation covering larger footprint. In order to achieve this, the attributes values should be extracted from individual situation assertions to build larger representations. The resulting representations provide snapshot of the situation from specific attribute point of view. All the attributes identified in information needs and collected in the form of utterance situation are processed to created various representation depicted as follows:

```
 Time: te+tl
Camp #1
Property: Count of Patients : value 20
Camp #2
Property: Count of Patients : value 30
Camp #3
Property: Count of Patients : value 45
```

This representation provides theme-based overview of the persisting situation in part of UoD covered. Such representation is known as coverage in geographical information system domain. When it is rendered with mapping application, it generates a thematic map. Each representation generates a thematic map that can manually or automatically processed further for identification of events, patterns or developments.

Event Detection

The purpose of creating representation is to create rich pictures of the ongoing situations that can lead to decision-making as proposed in soft systems methodology. The representations are subjected to processing to identify developments in ongoing situation. Detection of event is one such goal. Various event profiles are identified with event modeling process discussed in 3.7.

```
-------------- Representation -----------------
Camp #1
Property serving capacity : value 25
Camp #2
Property serving capacity : value 25
Camp #3
Property serving capacity : value 35
-------------- Event Detection Rule -----------------
Event profile #1 for Camp capacity
If
Number of victims arriving in camp is grater than the numbered Capacity of the Camp,
the camp is in critical status.
-------------- Representation -----------------
Victim #1
Property Treatment Status : Treated
Victim #2
Property Treatment Status : Not Treated
Victim #3
Property Treatment Status : Not Treated
-------------- Event Detection Rule -----------------
Event profile #2 for Vitim Response
If
Patients  Property is  treatment status
and value Not treated belong to Critical Range of treatment status
```

In the context of camp scenario, the inflow of victims received by camp should be rendered various emergency response services. The status of individual instances of victims therefore is also monitored in similar fashion. The event profiles contain critical range of values that determine occurrence of event. These representations are subjected to the event profiles for determination of event.

Action Detection

To complete the feedback control loop, upon identification of a critical state change in the form of event, appropriate corrective actions should be identified. As identified from process modeling activity, the next desired status and activities required for that transition is captured in KB. The next step is therefore to determine appropriate next state and activity for given instance.

```
-------------- Representation -----------------
Doctor #1
Property : Respondent status : Active
Doctor #2
Property : Respondent status : Inactive
Ambulance Driver #1
Property : Respondent status : Active
Pharmacist #1
Property : Respondent status : Active
-------------- Action Rule -----------------
If
Current state of entity is ‘‘Not treated ’’;
```

```
Next goal state is ''Treated'' ;
''Doctor'' is responsible for transition from ''Not Treated'' to ''Treated'';
Doctor#1  status is  ''Active'' that can render Treatment
```

Actions detection logic is applied to available instances in camp. The process modeling allows requirements regarding roles, resources and information that should be appropriately handed by the coordinating authority. The representation above allows identification of instance of a medical professional role `Doctor#1` that can render the required action.

Notification

As discussed in the case of utterance situation, the action requirement is identified in system environment. The instance of individual is active in natural and built environment. Hence, appropriate interaction pattern should be utilized to initiate action required.

```
To: Doctor#1
Pattern: Publish-Subscribe
Message Content: Action Request render Treatment
Reply: Response Status
```

The representation above indicates the use of messaging patterns, this time for the purpose of notification. By realizing this notification, the control part of the feedback control loop is realized.

Extract Transform and Load (ETL)

As the information processing is being carried out in a system environment, an instance of physical system is utilized to support these operations. As discussed in stakeholder environments, various stakeholders utilize the same information recorded by the method above to take decisions about appropriate action in respective stakeholder environment. These stakeholders are expected to utilize their own systems. In order to complete the information flow, the information processed and handled in the originating instance to the instances of all the stakeholders. During this transfer, appropriate processing must be carried out to meet their information need. This may involve change in granularity as discussed in information processing model. This calls for Extract Transform and Load (ETL) requirement to be supported by the system. In dynamic environment, determination of available instances of stakeholders, their required granularity and physical instances of the system should be identified. By employing logic for automated identification ETL specification as discussed in Section 3.8.10, ETL specification including available instances can be identified as follows:

```
Time of Event: Te
```

```
Time of Report: Te + Tm
Source Role: Camp Manager
Source Granularity: (Spatial=camp, Temporal=min, conceptual=patient)
Transformation: Aggregation with sum
Target Role: Administrator
Target Granularity: (Spatial=adminblock, Temporal=hrs, conceptual=patient)
```

The representation above indicates various parameters that are identified as input to the specification process. This includes the granularity levels in spatial, temporal and semantic dimension. Also from measurement model, the aggregation function for specific attributes being processed. The role instance and physical system at which aggregated information needs to be loaded. The footprint having temporal relevance, the aggregates are to be made available with specific update frequency.

Job Execution

The representations discussed above include specification of information need, required messaging patterns, representations, event detection and ETL tasks. These specifications indicate requirement of information processing that should be carried out in computing environment. As these task specifications are taken as jobs to be executed. The specifications also identify physical instances, processing requirements and execution at specific time instances, the job management is also carried out in automated environment.

```
Time: te+tm
ant dataServiceClient
 -Ddai.url=http://10.100.64.26:8080/wsrf/services/ogsadai/DataService
 -Ddai.resource.id=PostgresResourceSADBon48
 -Ddai.action=sa/SAPerform/Delivery/PatientInflow#1.xml
```

The ETL is proposed to handle as automatically generated OGSA-DAI performed documents. These generated specification are taken as job and need to be executed at specific time interval.

Geographical Information Layer

The Geographical Information (GI) available in the form of coverage provides a useful cognitive reference to decision-makers. Each coverage contains specific aspect of the UoD, by representing thematic information. Multiple such coverages are layered one on top of the other for visualization of all the parameters necessary for the decision. These coverages are the outcome of the ETL process. The spatial coordinates and the recorded values assigned to create thematic representation of a geographical area. The indicated values of the attributes may not be at the same granularity as it was at the time of recording. The aggregation is to be carried out according to the rules specified in measurement model.

```
Time: te+tm
Admin Block #1
Patients : value 60
Admin Block #2
Patients : value 90
Admin Block #3
Patients : value 76
```

The example indicates an abstract representation of a GI theme representing number of patients recorded in each administrative block. In this representation, domain is indicated as various administrative blocks in given UoD. The information is originally collected at camp level. The number of patients is a kind of attribute that can be aggregated with summation function. Hence, all the camp following in to respective administrative blocks can be aggregated by summation of the recorded values. This is done according to the aggregation rule defined in the measurement model in Section 3.5.

7.1.3 Observations

The presented experimentation demonstrated how assertions about the states of entities and process could be identified and collected from dynamic pool of actors. In achieving so, the specific user instances are periodically presented the information collection template, once filled and replied, are processed to create representations as thematic maps. This demonstrated the intended rule and role based handling of spatial data[134]. In issuing role specific messages and collecting data, it is required to manage the life cycle of messaging endpoints. The representations are based on atomic assertions about state of the entity. Decision makers are interested in aggregate information. The aggregation in space, time and semantic domain, followed by loading of data is determined by the automated ETL capability. Resulting layers created at target locations are also asserted in the knowledge base with the metadata that holds information about why and how it was identified, collected, processed and loaded. This lineage information provides the data provenance capability.

The research started with the objective to meet the information needs in complex dynamical systems. As indicated in Figure 7.9, the Feedback control loop has to be implemented for supporting various information exchanged patterns in identified domain applications. In order to achieve Feedback Control Loop, the entities in the system are continuously monitored in the given UoD. The processes, actions and surrounding environment that affect the states of the entity are known to the system. These critical elements are appropriately employed and

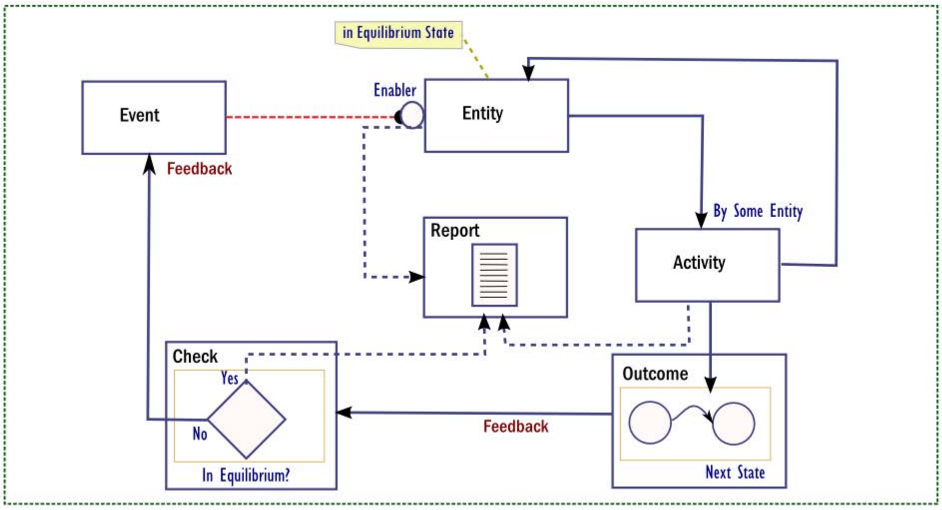

Figure 7.9: Feedback Control Loop Required

maintained to keep the entities in equilibrium states. Any event signifying critical state change in required elements results in undesired state of entity. The system therefore must determine, corrective actions needed and should apply the same to bring the entity in the equilibrium.

This simple requirement is challenging to achieve in case of complex dynamical system. The dynamism and complex interdependence among entities makes it difficult to estimate exact amount of entities and corresponding critical elements that need to be handled by the single system. In order to address these problems appropriate solution must be suggested that can allow identification, characterization, instantiation, monitoring and handling of various elements in given UoD. As this task is expected to cross multiple disciplines, a methodology is required for coordinating the large scale collaborative effort required to realize the system.

Coverage The term *Coverage* indicates various aspects considered in addressing the given problem. In the beginning of the thesis, the modeling was discussed on four founding aspects. The *Logical* Aspects addresses in capturing the domain knowledge. The knowledge representation of concrete domain and the rules that handle attributes is required. SWRL Built-ins supports the temporal and arithmetic operations for this requirement. The evaluation of current state of affairs, determination of future course of action is based on various measured quantities. This needs to be done consistently over larger area. Hence, *empirical* aspects are appropriately incorporated in the strategy. Chapter 3.5 introduced measurement model that incorporated standards about measurements, units of measure, observable properties etc. These provided basis for instance creation, representation generation, and subsequent calculations. In capturing, processing, handling and communicating information the *computational* aspects become critical. Isolated systems deployed on monolithic computing resources cannot provide suitable computing infrastructure. SOA and Grid Computing based architecture is therefore discussed introducing loosely coupled, interoperable components providing distributed computing infrastructure suitable at required scales. Based on this infrastructure, the automated job specification queue indicated utilization of computational support. ETL and Dynamic Data Management jobs indicated management storage and retrieval and management of data in multiple resources. The *Organizational Behavior* aspects in determining information needs, information exchange patterns and behavioral patterns of individuals that are important for desired objectives..

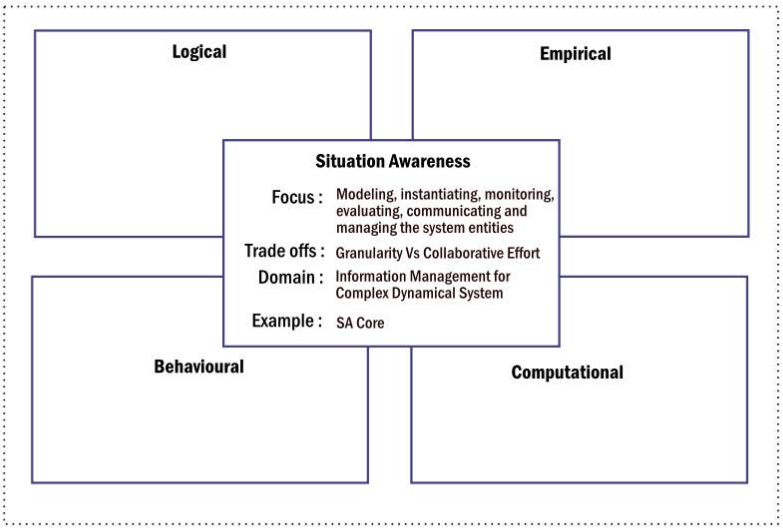

Figure 7.10: Domain View Revisited

All these aspects were integrated with proposed SA framework. This framework indicated seamless integration of underlying disciplines. The resulting functionality provides domain specific objectives. Resulting situation awareness solution is realized in the form of domain specific middleware service. The trade off in this domain is level of detail vs. collaboration that is required to achieve that. The detail is defined as footprint containing spatial, temporal and conceptual dimensions.

> "Information management strategy should benefit from research and development in related disciplines to cover related aspects."

Data Access Patterns Access to situation is provided in the form of data determined suitable for specific roles. The attributes not observable by sensors are collected from human reporters acting as source and sink of information. The users are provided only relevant information for executing their assumed roles, thereby providing *separation of concern*. The read and write access to data is ensured with rules. This feature is realized as *Data Access Patterns*. Rule and Role driven data Access pattern is realized with the desired configuration. The data is not only in the form of attribute values, it can also be in the form of artifacts, work products and other details that can be useful in setting up the configuration. Hence, various data access patterns are expected in the system.

"Rule and role driven data access patterns should be realized for each roles"

Cross Environment Traceability Natural and built environment in the UoD contains entities that are to be managed. Information management strategy should be employed to fulfill these needs. From natural and built environment to system component that provide processed information to entities that again manifest the impact in natural and built environment, thus realize a feedback control loop. Specific work environments can be identified in between namely the reporting environment that observers and reports important fact. The organizational environment identifies and responds to reported issues according to the jurisdiction. While they provide policy, research outcomes, or standards, etc. the design environment provide the usable artifacts. These artifacts can be taken and configured properly to ensure interaction patterns. This is ensured in the configuration environment. The information exchange patterns hosted by the system link to users in natural and built environment completing the information flow.

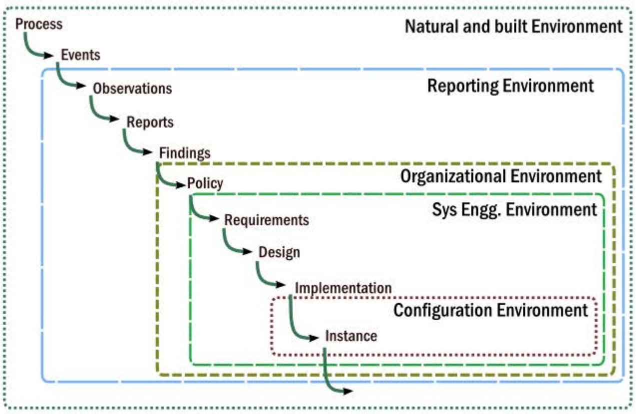

Figure 7.11: Traceability

As indicated in Figure 7.11, each of these environment are implicitly connected and have specific dependence. Without any mechanism for explication of the dependence, the slow delivery and slow uptake is next environment results. The explicit identification and provision of traceability product can improve the uptake. Hence, traceability generally used in software development life cycle can be incorporated to other work environments to improve the speed of uptake and traceability.

> "All identified requirements and provisions made for them should be linked with appropriate traceability"

Life cycle management of Transient Resources Complex dynamical system emphasized dynamism and complexity as two important source that indicated challenges in information management strategy. It was realized that there could be many concrete and abstract entities identified in the given UoD. The dynamism indicates the effect of change that must be addressed. Depending up on the complexity, these changes may trigger changes in the state of many other dependent entities. Hence, these short-lived and complex entities must be monitored and managed. Corresponding to such entities, many programmatic entities are required that can help processing, modeling, communicating, processing and handling. These are the transient resources, automatically created, managed and destroyed according to the observed dynamism in UoD. This includes instances of computing resources, actors, actions and events in KB. In order to communicate to them, various messaging patterns, snapshots, jobs, and other resources handled by the system. These are the transient resources. Figure 7.12 indicates a life cycle of a typical transient resource in the SA Configuration.

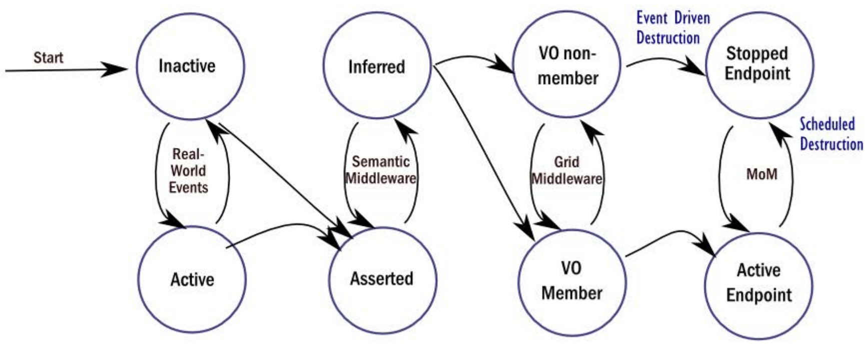

Figure 7.12: Object life cycle states handled by the system

> "Programmatic equivalents of various concrete and abstract elements in UoD should be maintained by the system and life cycle of these entities should be handled by the information management strategy."

Granularity Granularity is the level of detail at which the information is handled. The situation awareness requires information at specific granularities appropriate for the users. The granularity also is characterized in the form of footprint. As indicated in Chapter 3, footprints have tree dimensions namely spatial, temporal and semantic. Information sources have unique footprints when designed and commissioned. To meet the user needs, more information sources are integrated to fulfill the spatial, temporal and semantic coverage of the requirement as indicated in Figure 7.13.

In complex dynamical system, the footprint of required information may change dynamically. Also, the relevance for the reusability of information source can also be identified in spatial and temporal dimensions. Hence, a proper approach is required to meet the information need as identified by the ongoing event. The configuration approach proposed in 6 represented similar capability of the proposed solution.

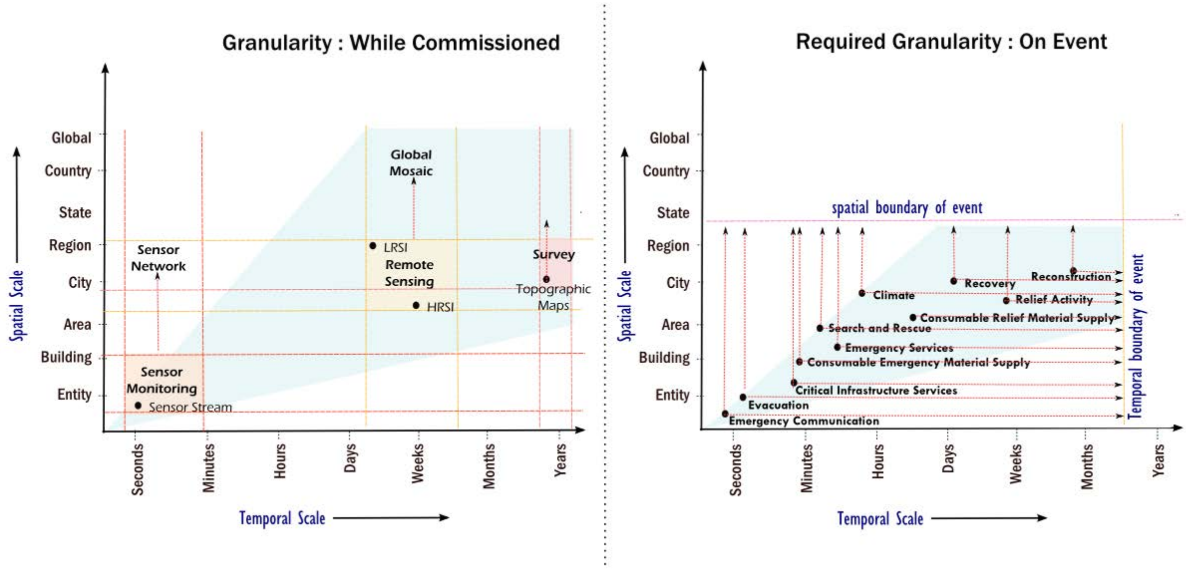

Figure 7.13: Granularity handled by the system

> "Granularity of information being handled should be accurately determined and information management should follow it strictly in collecting, handling, archiving and destroying the content."

Configuration The proposed information management process is recommended to be handled as a service. It is designed to support all the roles in various stakeholder environments identified in Chapter 4. For appropriately handling individuals playing various roles, unique architectural products are introduced to meet the information needs still providing the Separation of Concern. This approach recognizes the increasing requirement more specialized contribution provided by actors in the form of various roles assumed in the system. With increasing specificity of the outcome of role, it is difficult to share and re-use. Appropriate repository structure is required that can handle sharing and discovery of specialized artifacts and work products. Ensuring the coverage of training is difficult in complex dynamical system. The configuration approach explicates training requirements for executing role specific activities using available actors. *Quadruple of artifact* discussed in Chapter 6 indicated capability of configuration approach to enable access and sharing of archived artifacts.

In summary, configuration approach provide multiple features like performance tracking, feedback, task management, reduction in duplication of work, improved productivity, fast dissemination, adherence to standards, gap analysis, estimation and preparation for multiple skills. Important finding is:

> "Information management should be handled as a critical infrastructure service with explicit provision of continuous monitoring and upgradation of all the underlying components."

7.2 Related Work

For comparative evaluation of the proposal, various research issues needs to be revisited with solution approaches available in literature.

Overall Information Management Strategy Comfort [2] discussed information management in complex dynamical system, identifying general issues and basic requirements. The proposal recommended a five-steps namely, assessment of initial conditions, information acquisition and transformation, representation, information search-exchange-feedback and reasoning-learning and action. This contribution is accepted as one of the basic reference for the proposed project. The complex dynamical system characteristics and N-K system based representation of the system are adopted in design principles. The methodology proposed is similar to the soft system methodology recommended by Checkland [18]. This also accepted as one of the guidance and incorporated as design principle.

While these closely related contributions are incorporated in the proposal, the present work provided information management strategy with comprehensive guidance in modeling and realization of such systems. The present approach is based on realizing a configuration that can allow tracking of complexity and dynamism in given system. Dynamism captured in the form of events, and its ramification along with required actions identified based on complex interdependence of system entities are two basic component of the proposal. Goals of individuals and organizations are taken as selection criteria for determining the flow of information. Resulting information management strategy designed based on these components allows various type of information content delivered via appropriate information exchange patterns. The proposal allows access to information as needed by the user or determined suitable by the configuration. This type of access identified as Situation Awareness, is dynamically handled as a form of critical infrastructure service.

Information management practice is identified as providing right information at right time to the right person in the right format. In recommended strategy, the determination of what is right is decided by the strategist. In present approach, it is said that what is right is to be determined both by the individual or organization and all the stakeholders. Considering this fact, the proposed strategy provides mechanism for continuous identification of "what is right".

Situation Awareness Systems Situation awareness is introduced in in-flight automation [21], as a solution to meet information need of the pilot. The same approach is found useful in industrial and organization environment and addressed by the research community. Among important proposals, Matheus et al. discussed higher level fusion [135], and application of semantic web technology [136] for situation awareness. Mirhaji et al. [137] discussed situation awareness for public health domain using knowledge based approach. Using similar knowledge based approach, Smart [138] discussed the fusion of information sources for situation awareness. Yau et al. [139] discussed situation awareness in ubiquitous computing environment. Baker et al. [140] discussed collaboration management infrastrucutre using customised process and situation awareness.

These approaches towards situation awareness are discussed with closed world assumption. In complex dynamical systems, the situation can be characterized based on the context of the users. This requirement goes beyond the scanning of the environment. In order to help goal directed actors in meeting their information needs to carry out the assigned tasks, the task allocation and execution specific information should also be considered as part of situation awareness.

Process/ Process Framework Large scale collaborative efforts are one of the basic features for given application domain. Method engineering is a class of solution. Rational Unified Process [79] and its components is proposed as potential solution. Method engineering and its special aspect known as situational method engineering is discussed [77] for its utility in dynamic environment.

Yet, the process work flow is considered as static outcome of few experts at global scale. The access to guidance is made available as reference, and presented on demand by special effort by the user. The proposed situation awareness unified process span across various stakeholder environment integrating all of them according to the domain knowledge. The activities and workflows are captured as part of process modeling recommended for situation awareness. The content creation is available as ongoing process, where knowledge representation and work product outcome continuously upgrade domain and local specific content. The content is semantic based, and made available as part of situation awareness requirement.

Architectural Framework Architectural frameworks are important in representing the current dynamic scenario of collaborating systems. The organizations have specific complex interdependence among various collaborating entities. This problem is addressed with proposals like DoDAF [82], [83], ZF [141] and ToGAF [11]. These architectural frameworks addressed the problem by proposing comprehensive guidance based strategy for creating views for the system. Winter et al. [142] discussed the fitness of enterprise architectures to meet the business needs by introducing enterprise architecture governance.

The problem of representing the complexity and dynamism still continues because the proposed approaches are done in monolithic desktop environment. The representations are made available to the architects as part of specialized tool and design environment. Availability of resources are not monitored and appropriately reflected on the architectural products on the runtime.

The proposed architectural framework approach recommended use of semantic web technology. Each collaborating individual organizations and related resources are handled and monitored as separate instances. Any change are handled as events and appropriately updated in KB. For Architectural Views, various products are created on the runtime according to the need and footprint of the users. Hence users are provided access to appropriate elements on the runtime, providing ongoing situation of resources and their dependence.

Distributed Data Management in Grid Management of data in grid environment is a challenging task, as members of virtual organization and data resources in VO may be a dynamic set. The data management tasks like synchronization, partition, backup, regionalization, purging etc. needs to be carried out appropriately to ensure availability of data to its users. To support these features Stefano [75], recommended various strategies in the form of mathematical representations. The parameters are geographical, temporal and other factors affecting the use of data. This strategy is useful in carrying out tasks in proper tooling environment.

Yet the complexity and dynamism is not appropriately supported. Also the abstract representation does not specify how various parameters affecting the decision regarding the data management tasks are monitored in grid environment. Also, the task specifications are manually generated to meet the needs. As a solution to this problem, the information processing model represented automated generation of data management task.

Distributed Event Based System Distributed event based system is critical in realizing reactiveness of the system. In business scenarios, various situations may qualify as events. Adi et al. [143] introduced AMIT- situation based approach can be employed to trap business events with the help of comprehensive representation of profiles. Event detection is basic component of distributed event based system. The occurrence of event is generally based on explicit representation of event profiles. Practically this approach is not useful, as explication of all event profiles are not possible. The identification and capturing of implied event can be achieved by semantic approach as discussed by Qian et al.[144]. Apart from events targeted for business application users, mobile agent based system have specific requirement. The event publish subscribe mechanism in mobile can be realized to meet specific needs [145]. The detection of event is based on information collected from the sources. These sources may be in the form of sensors spread across wide geographical region. Joshi et al.[146] proposed a rule based system to control eventing in such systems. Event detection and notification requires flow of information among various components of the middleware. Banavar et al.[147] discussed information flow based event distribution system. In automated subscriptions, the pruning of unrequited subscriptions is addressed with dimensional pruning approach [148].

ETL Strategy ETL provide important feature in distributed data management[149]. In dynamical environment ETL parameters must be identified using an automated procedure.

Proposed approach provides mechanism for monitoring various elements that are parameters of ETL process. Based on the monitored parameters, specification of required ETL tasks are proposed.

Data Cube Strategy Calculation of multidimensional data cube is common requirement in business. The granularity levels at which the calculations are carried out are generally static. Hierarchies in spatial and temporal dimension along with appropriate functions are introduced by Malinowsk [69]. Karakasidis et al. [150] discussed ETL queues for active data warehouse. Priebe et al. [151] discussed Ontology based integration of OLAP cubes. Based on the potential users available and the data captured at specific granularities, the decision should be taken. The granularity calculation discussed in the proposed modeling approach allows identification of required hierarchies. This reduces unnecessary calculations and improves the system performance.

Spatial Data Access Pattern Access to the spatial information has evolved. Handling of spatial data by information strategy is critical aspect. Initial years, the data access pattern is monolithic, requiring the data to be transferred on secondary storage devices. Multiple copies sources and user derived data are retained by the individual users. The limitation is realized and solved with server based GIS that allows single instance of data shared by server access by clients allowed fast propagation. With requirement to access and handle data from multiple sources, the standardization in GIS is required [152], and standard based spatial data repository in the form of spatial data infrastructures [153] came in to existence. While standards allowed syntactic heterogeneity, the semantic one is still the problem, addressed by semantics in GIS [154]. The static links to server data soon introduced problem of discovery of new and relevant data. This is provided by location-based access to information. Syndication based access also, the event-based access to information solved problem. Incorporation of grid [155], Web services[156] and rules [157] demonstrated novel approaches in handling geographical data.

In complex dynamical systems, the GI should be generated, handled and provided to the appropriate users to meet the instantaneous information need. Identification of GI, Planning of information sources, collection mechanism and delivery should be made rule driven. This results in realization of spatial data access pattern. Proposed approach provide rule and role based spatial data access pattern to meet the situation awareness requirement.

Disaster Management System Disaster management information systems are widely required application and effective implementations are already under utilization. Some of the important disaster management projects include SHANA[158], OASIS, WINE, ORCHESTRA [159], SHARE[160] and DERMIS [161]. These approaches are based on comprehensive requirements collected and appropriate decision decisions arrived for system design.

In such cases, complexity and dynamism is ignored. The system should be made open to identify new risks incorporate new resources, application components, work flows and users requirements. The present proposal addresses the limitations by making appropriate mechanism for evolution under large scale collaborative environment.

Modeling Infrastructure Infrastructure services provided by government and privately owned organizations are useful in many human activities. Due to heavy dependence, some of these services are identified as critical infrastructure service. A properly functioning critical infrastructure service is outcome of successful collaboration of many service and resource provisioning activities. The modeling and monitoring of critical infrastructure is therefore important aspects. Modeling of complex interdependence among critical infrastructure services are addressed by Dudenhoeffer [162], Pederson [163], Rinaldi [164] and Hollman[165].

While most of the infrastructure modeling approach is based on resource dependence, the complete ramification is not supported. The hierarchical semantic based process modeling allows identification of complete ramification as described in the conceptual modeling and information processing section of this work.

7.3 Discussion

The experimentation steps and observations demonstrated that it is possible to realize a feedback-control-loop for goal directed dynamic complex systems by employing a situation awareness method proposed here. Various models created for representation of entities and subsequent implementation demonstrated that it is possible to use human beings as sources of information, automated granularity calculation and ETL can be realized to generate aggregate data targeted at various level of users. It is also demonstrated that based on the knowledge represented it is possible to determine the set of all possible events and subsequent identification of the set of all possible actions, resources and conditions required to restore and maintain desired state even in complex systems. It also demonstrated the usability of a configuration approach that utilizes rules in configuration and management of various transient computing objects in form of messaging components, GML templates, ETL Specifications, Job Specifications and GI themes. The same features also displayed that the same strategy is useful in handling dynamic set of resources in various resource pools.

Chapter 8

Conclusion

This thesis proposed *Situation Awareness* as important capability imparted to an individual or an organization in achieving desired goals in a system. Domain application scenario selected for this research work involved dealing with complex dynamical systems. The information management strategy is appropriately altered to meet the changing information needs of all the potential roles. The proposal is made in form of a process and required domain middleware services in meeting the information needs.

What has been achieved is a mechanism that facilitates the creation of information management applications for processing complex dynamic information. The designed templates allow information management services to be linked together to process dynamic and complex tasks. Dynamic instantiation and run-time configuration enable these templates to be adapted and tailored to suit specific analysis tasks and execution environments. This multi-level coordination mechanism supports both the design-time phase and run-time phase of a software lifecycle.

8.1 Research Issues Addressed

Chapter 1, discussed issues in meeting information needs of various users in complex dynamical system. In Section 1.3.1 specific research questions have been identified. This section revisits all the research questions and provides answers to them.

How to characterize system and its boundary? This research question is addressed by classifying the system as ***complex dynamical system***. The nature of problem domain is discussed to reveal salient features. Hence, this characterization allowed selection of methodology for information management.

How to identify the management strategy? This research question is addressed by evaluating various methodologies suitable for the given kind of the system. It was established that ***Soft Systems Methodology*** is suitable approach for complex dynamical systems.

How situations can be defined and possible worlds can be determined? The conventional ***definition*** of situation is updated with emphasis on information granularity. It is also stated that possible worlds are only restricted with known domains that are relevant for given goal statement. Hence, from very generalized definitions of situations and possible worlds, they are enhanced for utilization in complex dynamical systems. The utilization of domain knowledge is also identified as a basic requirement, towards modeling of situation.

How situation awareness can be defined for complex dynamical systems? This research question is solved by ***Situation Awareness Theory*** explained in Chapter 2. Based on the comprehensive definition of the situation, the situation awareness is defined for complex dynamical environment. First realization is that it should be separate for individuals and organizations. As individual and group behavior are separately governed by principles discussed in Chapter 2. Based on these principles, situation awareness is defined for actor, and organization point-of-view. These new definitions entrusted much information handling responsibility on system, and hence new definitions are required to characterize system, configuration and processes.

What modeling strategy will be useful to capture required roles, properties and rules? This research question is addressed by proposing a modeling strategy explained in Chapter 3. To capture all the required entities and their salient features important to establish complex interdependence and dynamism, the step-by-step conceptual modeling is introduced. Among the proposed sequence, first six conceptual models provide identification and characterization of various entities by providing templates and rules. The last model is information processing strategy, provided as a set of thirteen algorithms provided guidance for handling information in the runtime.

How complex event processing can be achieved in complex dynamical environment? This research question is addressed by ***Eventing model***. By addressing the event processing need in complex dynamical system, new definition of event is proposed along with concept of *Event Space*. Templates for expressing various event profiles are also discussed. The evaluation of event profiles on the runtime is also a critical decision. The event detection strategy is provided as one of the information processing strategy that is recommended for the runtime.

How coordinating agency can identify the setup and resource requirements? This research question is solved by Situation Awareness Unified Process ***SAUP*** explained in Chapter 4. The delivery process provided estimates and project management details in order to meet the information need for stakeholders of the system in given UoD.

How stakeholders can identify the work? This research question is solved by ***Gap Analysis Matrices*** explained in Chapter 5. These products provide traceability to work products of activities in various stakeholder environments. Each element in work breakdown structure specifies resource and other requirements for successful completion of the task.

How stakeholders can share the work products? This research question is solved by ***Enterprise Continuum*** explained in Chapter 6. This enterprise continuum acts as a repository of work product outcomes. Also the traceability is established with various architecture products created for specific roles. Hence, after assuming these roles, the stakeholders can access the enterprise continuum to share the work and can explore the traceability matrices to identify the reuse of the shared products.

How multiple systems and sources are integrated? This research question is solved by ***Configuration Approach***. It demonstrates that it is possible to connect to multiple situation awareness configurations. The identification of configurations is done by the types of artifacts that are planned to be accessed.

How stakeholders can track the work and get feedback and still get separation of concern? This research question is solved by ***Cross Environment Traceability*** explained in Chapter 4 . Different architectural products allow traceability of their work product in individual stakeholder environments. Traceability matrix allows identification of gaps by providing coverage of current efforts.

How information requirements of the stakeholders can be identified? This research question is solved by ***Information Need Identification*** explained in Chapter 2. It is established that information need is determined not only by goal of individuals, but also by the complementary goal of the stockholders. Exhibited activities of stakeholders, dynamic behavior of other related entities and their impact of resources and the environment - all contribute to dynamic change in the information needs.

Which new interaction patterns are required? This research question is solved by ***Spatial Data Access Pattern*** explained in Chapter 6 . Data is accessed according to various interaction patterns. It is either accessed or provided to the seeking entity. The system in dynamic environment must quantify which part must be collected from the users, and how it should be managed to meet individual information needs.

How dynamic set of stakeholders and their work products can be handled? This research question is solved by strategy of corresponding ***Transient Resources*** explained in Chapter 6. Some examples of transient resources include: job specifications, representations and messaging patterns. These are the configuration environment artifacts and they are best managed with rule-based life cycle handling strategy of transient resources.

How dynamic set of resources can be employed on the runtime? This research question is solved by ***Configured Pool of Resources*** explained in Chapter 6. According to this strategy, configuration strategy is such designed that every time a resource noticed in UoD, a corresponding instance is created in the knowledge base. The monitoring and handling of this instance is then appropriately carried out in different functions. The corresponding transient resources are destroyed when resource disappear or cease to exist.

How information specificity can determined for different stakeholder hierarchies? This research question is solved by ***Information Granularity*** explained in Chapter 2 The concept of footprint is introduced in Chapter 3, allowing identification and handling of specific information granularity.

8.2 Conclusion

This research concentrated on information management methodology for complex dynamical systems like disaster management, environmental management and critical infrastructure protection that are now reaching global scales. A collection of entities and their behavior in retrospect with the available physical resources are visualized to form such a system. Extraneous forces, both positive and negative, act continuously and independently in either healing or harming this physical system. The instantaneous state of the system is controlled by the dominating force. Occurrence of sudden unpredictable events may further deviate the system from the desired state. A viable information management strategy was called for to involve multiple disconnected autonomous actors, their activities, resources and operating conditions in the real life decision making process for such systems.

It was observed that during the evolution and in response to the events, the system receives help from within and outside of its boundaries. Any attempt for management, thus must build

Research Question	Complex Dynamical System	Soft Systems Methodology	Theory and Novel Definitions	Conceptual Models	SAUP	Gap Analysis Matrices	Enterprise Continuum	Configuration Approach	Cross Environment Traceability	Information Need Identification	SA Architectural Products	Spatial Data Access Pattern	Transient Resources	Configured Pool of Resources	Information Granularity
Characterization of system	●														
Management strategy		●													
Characterize situation			●												
Characterize situation awareness			●												
Modeling strategy				●											
Complex Event processing				●									●		
Requirement Identification					●										
Task Identification						●									
Sharing of work products							●	●							
Integration strategy								●							
Separation of concern									●		●				
Information requirement										●					
Interaction patterns							●					●			
Dynamic set of entities								●					●		
Utilization of Resource Pools														●	
Information Specificity															●

Table 8.1: Summary of Research Issues Addressed as Requirements Vs. Design Matrix

capability in handling activities of existing and potential stakeholders of the system. Access to information is a factor that governs the stakeholder behavior. The information need was identified for the stakeholders and access to it was characterized in the form of a feedback control loop. This approach, generally known as situation awareness when employed in a closed system environment, was extended further to make it applicable to complex dynamical systems. New definition of situation awareness was offered from actor, system and organization point of views.

The development, deployment and usage of an information system that can model, instantiate, monitor, evaluate and communicate system elements were required by the identified situation awareness. It is also known that systems with overlapping boundaries have overlapping scope of underlying information systems. Coordination of these systems at global scale therefore, must incorporate capabilities of all relevant existing information systems. With this fact, it was argued that information management could not be carried out in isolation but must be set up with a configuration approach by reusing the available systems. For achieving this, a method engineering approach termed as Situation Awareness Unified Process (SAUP) was proposed. The proposal introduced seven models to help representation and evaluation of system status. Along with modeling, the method provided process guidance to other stakeholders in creating their work-products. Once followed in the prescribed manner, the process allows creation of shared enterprise continuum in a virtual collaborative team environment.

Traceability, granularity, observability and adaptability were established as important features that must be supported in hosting the enterprise continuum. A Meta CASE tool called Situation Awareness System, was provided to support these capabilities. To aid consistent development of interoperable software components, Situation Awareness Architectural Framework (SAAF) was designed. The framework reveals both static and dynamic views of the system. These proposals were founded on existing practice, standards and technology. As a proof of concept, an emergency response scenario was introduced and the difficulties that are typically faced due to lack of required information flow are enumerated. An indicative setup was discussed, with details on the internal behavior of the middleware corresponding to use cases being identified.

8.3 Directions for Future Work

This thesis addressed the information management issues of complex dynamical systems that are increasingly being handled in fields like disaster management, environmental management and critical infrastructure protection. It is recognized that these domains span geographical and political boundaries and any attempt towards management requires coordinated collaborative efforts. At international levels, commitment to build systems at such scale is evident in forms of intergovernmental agencies setting up policies followed by systems and processes that support.

By addressing possible issues, the thesis introduced a suitable information management strategy targeted for such collaboration. Duly recognizing the scale of effort required in realizing the information management, the limitation of the proposed work is accepted in the section on scope of work of this thesis. The proposed solution is designed for extensibility to facilitate contribution for the collaborators. Hence, the proposed methodology can be enriched by contribution of collaborators as they utilize the methodology in respective areas. While the components like method content can be enriched by individual contribution of collaborators, there are certain aspects that require contributions at research levels. Following issues requires special attention from research point of view:

Extension of Situation Theory and Semantics to cover concrete domains: Situation theory is identified as basis to handle basic unit of information during interaction patterns and its use in building complex situation prevailing in the given UoD. It is found that mere Boolean relations handled is not sufficient to meet as situation. Also situation in real world requires assignment of attributes. Hence, appropriate theoretical framework needs to be established with proper discussion of theorems and proofs as discussed by Barwise and Zalta. This will enable formal handling and representation of rich situations. Considerable level of effort from researchers is required in extending this aspect of the theory.

Design Principles for Information Management Strategy: Fifty seven design principles are introduced to guide the information management process. These principles are based on the theories establish specific ways that facilitate information generation, utilization and handling of information to meet the requirements in complex dynamical systems. The new theories will continue to establish more aspects that should be integrated as design principles in future version of strategy.

Knowledge Representation Process: Four types of concepts are handled by the proposed methodology namely: Domain independent, domain specific, location specific and application specific concepts. The knowledge engineering task involved in this creating and maintaining consistent and updated representation is identified with appropriate roles. Architectural product to identify the gaps in available knowledge representation and availability of guidance in creating the knowledge based is also discussed. Ontology mapping approach that allows management of the contribution of involved knowledge engineers introduces a difficult research challenge. Proper systematic approach is to be designed and tested to work in dynamic collaborative environment. Designing such system involves identification of knowledge representation gaps is not limited only to concepts, but also covers representation of rules regarding observations, measurements, type conversion, operating procedures, event detection, event notifications to suit the local and application specific needs.

Upgrade SAUP method content: In its current version, SAUP have single author content. As organizations adapt to SAUP, their lessons learned, some domain specific expertise and other guidance contributed by the SAUP users, need to be incorporated in SAUP. Hence future versions of the process method plug-in will have added guidance for existing roles. For instance, the seven models proposed for the conceptual modeling and information processing in complex dynamical systems can be further extended with additional models to cover other relevant aspects of such systems.

Build and improve Visualization Capability: Separation of concern is achieved by handling role-specific messages, matrices and architectural products. These are the information content derived as a part of situation awareness. These chunks of information exist in middleware domain that should be accessed by the end users. Based on the prevailing situation and the assumed roles; the information chunks containing the events, alerts, representations and guidance can be many. Users have cognitive capability that limits the comprehension and utilization of information at given time. As a unique feature of this strategy, user involvement is also expected for features like utterance, task allocation, task closure and establishment of traceability. All these features require design considerations from Human Computer Interaction domain.

Introduction of new domain specific role sets and method content : In its current version, SAUP consist content for domain independent roles. There are many domain specific roles, artifacts, delivery patterns required to meet the real-world complexities and therefore appropriate guidance is required for building domain specific SA Configurations.

Establishment of Situation Awareness Maturity Model for individual Application Domains: Planning phase has not been discussed in this work. In planning phase, coordinating agencies are expected to setup SA Configurations so that local, national and international level coverage is achieved. This is essential requirement for coordination of collaboration at global scale, yet it can achieve only gradually. For example, realizing provisions of Hyogo Framework for Action in disaster management domain requires global level coordination from involved organization, that can be achieved gradual developments in configuration. A Situation Awareness Maturity Model (SAMM) should be established to identify and establish various stages in gradual advancement towards the desired coverage of the configuration.

Upgradation of SA System Architecture to incorporate new standards, services and technology components: Standardization processes are continuously resulting in new standards and specifications. They are absorbed by related stakeholders in different environments resulting in standard-based products. SA system architecture must appropriately be updated incorporating these components. For instance, various Web Services related standards are continuously contributed by the community by bringing every relevant aspect of business and practical needs. Governance of Web Services therefore is a critical challenge that can be addressed in future research efforts related to architecture frameworks.

Bibliography

[1] C. Schlögl, "Information and knowledge management: dimensions and approaches," *Information Research*, vol. 10, no. 4, 2005. [Online]. Available: http://InformationR.net/ir/10-4/paper235.html

[2] L. Comfort, "Information management for complex dynamic systems," in *Health Monitoring and Management of Civil Infrastructure Systems, Proceedings of SPIE Vol. 4337*, S. B. Chase and A. E. Aktan, Eds. SPIE, 2001. [Online]. Available: http://spie.org/x648.xml?product_id=435583

[3] J. Kirk, "Information in organisations:directions for information management," *Information Research*, vol. 4, no. 3, 1999. [Online]. Available: http://informationr.net/ir/4-3/paper57.html

[4] L. V. Bertalanffy, "The history and status of general systems theory," *The Academy of Management Journal, General Systems Theory.*, vol. 15, no. 4, pp. 407–426, 1972.

[5] N. Boccara, *Modeling Complex Systems.* Springer-Verlag New York, Inc., 2004.

[6] E. Romanelli and M. Tushman, "Organizational transformation as punctuated equilibrium: An empirical test," *The Academy of Management Journal*, vol. 37, no. 5, pp. 1141–1166, 1994.

[7] J. Law, "Notes on the theory of the actor network: Ordering, strategy and heterogeneity," the Centre for Science Studies, Lancaster University, Lancaster LA1 4YN, Tech. Rep., 1992.

[8] S. Gregor, "The nature of theory in information systems," *MIS Quarterly*, vol. 30, no. 3, pp. 611–642, 2006.

[9] P. Dargan, *Open Systems and Standards for Software Product Development.* Artech House, Inc. 685 Canton Street Norwood, MA 02062, 2005.

[10] I. Mirbel and J. Ralyte, "Situational method engineering: combining assembly-based and roadmap-driven approaches," *Requir. Eng.*, vol. 11, no. 1, pp. 58–78, 2005.

[11] The Open Group, "The open group architecture framework version 8.1, enterprise edition." The Open Group, San Fransisco, USA., Tech. Rep., 2003.

[12] Q. H. Mahmoud, Ed., *Middleware for Communications.* John Wiley & Sons, 2004.

[13] G. Hohpe and B. Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.

[14] P. Dyson and A. Longshaw, *Architecting Enterprise Solutions: Patterns for High-Capability Internet-based Systems.* John Wiley & Sons, 2004.

[15] C. Nock, *Data Access Patterns: Database Interactions in Object-Oriented Applications.* Addison Wesley, 2003.

[16] D. Guha-Sapir, D. Hargitt, P. Hoyois, and R. B. andDavid Brchet, “Thirty years of natural disasters 1974-2003: The numbers,” Centre for Research on the Epidemiology of Disasters, Tech. Rep., 2004.

[17] T. Homer-Dixon, “The rise of complex terrorism,” *Foreign Policy*, no. 128, pp. 52–62, 2002.

[18] P. Checkland, “Achieving ’desirable and feasible’ change: An application of soft systems methodology,” *The Journal of the Operational Research*, vol. 36, no. 9, pp. 821–831, 1985.

[19] K. Eisenhardt and J. Martin, “Dynamic capabilities: What are they?” *Strategic Management Journal, Special Issue: The Evolution of Firm Capabilities.*, vol. 21, no. 10/11, pp. 1105–1121, 2000.

[20] J. Grunig, “A new measure of public opinions on corporate social responsibility,” *The Academy of Management Journal*, vol. 22, no. 4, pp. 738–764, 1979.

[21] M. R. Endsley, B. Bolte, and D. G. Jones, *Designing for situation awareness: An approach to human-centered design.* London: Taylor & Francis., 2003.

[22] H.-J. Schellnhuber and J. Kropp, “Geocybernetics: Controlling a complex dynamical system under uncertainty,” *Naturwissenschaften*, vol. 85, no. 9, pp. 411–425, Sept. 1998. [Online]. Available: http://dx.doi.org/10.1007/s001140050525

[23] A. Ankolekar, M. Krötzsch, T. Tran, and D. Vrandecic, “The two cultures: mashing up web 2.0 and the semantic web,” in *WWW ’07: Proceedings of the 16th international conference on World Wide Web.* New York, NY, USA: ACM, 2007, pp. 825–834.

[24] UN/ISDR, “Report of the world conference on disaster reduction, kobe, hyogo, japan.” United Nations, Tech. Rep., 2005.

[25] L. von Bertalanffy, “An outline of general system theory,” *The British Journal for the Philosophy of Science*, vol. 1, no. 2, pp. 134–165, 1950.

[26] Longman, *Longman Dictionary of Contemporary English.* Pearson Education Limited, England, 2003.

[27] C. T. Meadow and W. Yuan, “Measuring the impact of information: Defining the concepts,” *Information Processing & Management*, vol. 33, no. 6, pp. 697–714, Nov. 1997. [Online]. Available: http://www.sciencedirect.com/science/article/B6VC8-3SX2J1V-1/2/3c22b6a9bf7c9402420d752b3c62b661

[28] C. W. Choo, “Towards an information model of organizations,” *The Canadian Journal of Information Science*, vol. 16, no. 3, pp. 32–62, 1991.

[29] ——, “Organizational disasters: why they happen and how they may be prevented,” *Management Decision*, vol. 46, no. 1, pp. 32 – 45, 2008. [Online]. Available: http://www.emeraldinsight.com/10.1108/00251740810846725

[30] M.-L. Huotari and T. Wilson, “Determining organizational information needs: the critical success factors approach.” *Information Research*, vol. 6, no. 3, 2001.

[31] N. Ahituv, “A meta model of information flow: A tool to support information systems theory,” *Communications of the ACM*, vol. 30, no. 9, pp. 781–791, 1987.

[32] E. Macevciute and T. Wilson, "The development of the information management research area," *Information Research,*, vol. 7, no. 3, 2002. [Online]. Available: http://InformationR.net/ir/7-3/paper133.html

[33] C. W. Choo, "Environmental scanning as information seeking and organizational learning," *Information Research*, vol. 7, no. 1, 2001.

[34] H. Bruce, "Personal, anticipated information need," *Information Research*, vol. 10, no. 3, 2005. [Online]. Available: http://InformationR.net/ir/10-3/paper232.html

[35] E. Maceviciute, "Information needs research in russia and lithuania, 1965-2003," *Information Research*, vol. 11, no. 3, 2006. [Online]. Available: http://informationr.net/ir/11-3/paper256.html

[36] C. Shannon and W. Weaver, *The Mathematical Theory of Communication.* University of Illinois Press, 1949.

[37] W. Brown, "Systems, boundaries, and information flow," *The Academy of Management Journal*, vol. 9, no. 4, pp. 318–327, 1966.

[38] R. Daft and R. Lengel, "Organizational information requirements, media richness and structural design," *Management Science, Organization Design*, vol. 32, no. 5, pp. 554–571, 1986.

[39] G. Miller, "The magical number seven, plus or minus two some limits on our capacity for processing information," *Psychological Review*, vol. 101, no. 2, pp. 343–352, 1955.

[40] J. Galbraith, "Organization design: An information processing view." *Interfaces*, vol. 4, no. 3, pp. 28–36, 1974.

[41] V. Choudhury and J. Sampler, "Information specificity and environmental scanning: An economic perspective," *MIS Quarterly*, vol. 21, no. 1, pp. 25–53, 1997.

[42] A. P. Sinha and I. Vessey, "Cognitive fit: An empirical study of recursion and iteration," *IEEE Trans. Softw. Eng.*, vol. 18, no. 5, pp. 368–379, 1992.

[43] D. Goodhue and R. Thompson, "Task-technology fit and individual performance," *MIS Quarterly*, vol. 19, no. 2, pp. 213–236, 1995.

[44] D. Compeau and C. Higgins, "Computer self-efficacy: Development of a measure and initial test." *MIS Quarterly*, vol. 19, no. 2, 1995.

[45] D. Straub and E. Karahanna, "Knowledge worker communications and recipient availability: Toward a task closure explanation of media choice," *Organization Science*, vol. 9, no. 2, pp. 160–175, 1998.

[46] G. DeSanctis and M. Poole, "Capturing the complexity in advanced technology use: Adaptive structuration theory," *Organization Science*, vol. 5, no. 2, pp. 121–147, 1994.

[47] T. Donaldson and L. Preston, "The stakeholder theory of the corporation: Concepts, evidence, and implications," *The Academy of Management Review*, vol. 20, no. 1, pp. 65–91, 1995.

[48] B. Wernerfelt, "A resource-based view of the firm," *Strategic Management Journal*, vol. 5, no. 2, pp. 171–180, 1984.

[49] D. Ulrich and J. Barney, "Perspectives in organizations: Resource dependence, efficiency, and population," *The Academy of Management Review*, vol. 9, no. 3, pp. 471–481, 1984.

[50] R. Emerson, "Social exchange theory," *Annual Review of Sociology*, vol. 2, pp. 335–362, 1976.

[51] R. Grant, "Toward a knowledge-based theory of the firm," *Strategic Management Journal, Special Issue: Knowledge and the Firm.*, vol. 17, pp. 109–122, 1996.

[52] I. Nonaka, "A dynamic theory of organizational knowledge creation," *Organization Science*, vol. 5, no. 1, pp. 14–37, 1994.

[53] W. Cohen and D. Levinthal, "Absorptive capacity: A new perspective on learning and innovation," *Administrative Science Quarterly*, vol. 35, no. 1, pp. 128–152, 1990, special Issue: Technology, Organizations, and Innovation.

[54] T. Robertson, "The process of innovation and the diffusion of innovation," *Journal of Marketing*, vol. 31, no. 1, pp. 14–19, 1967.

[55] J. Barwise and J. Perry, "Situations and attitudes," *The Journal of Philosophy, Seventy-Eighth Annual Meeting of the American Philosophical Association Eastern Division.*, vol. 78, no. 11, pp. 668–691, 1981.

[56] J. Barwise, *The Situation in Logic.* Center for the Study of Language and Information, 1989.

[57] K. Devlin, *Handbook of the History of Logic, Volume 7.* Elsevier, 2006, ch. Situation theory and situation semantics, pp. 601–664.

[58] E. Zalta, "Twenty-five basic theorems in situation and world theory," *Journal of Philosophical Logic*, vol. 12, pp. 385–428, 1993.

[59] C. Parent, S. Spaccapietra, and E. Zimnyi, *Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach.* Springer-Verlag Berlin Heidelberg, 2006.

[60] D. D. Dudenhoeffer, M. R. Permann, S. Woolsey, R. Timpany, C. Miller, A. McDermott, and M. Manic, "Interdependency modeling and emergency response," in *SCSC: Proceedings of the 2007 summer computer simulation conference.* San Diego, CA, USA: Society for Computer Simulation International, 2007, pp. 1230–1237.

[61] J. Raper, *Multidimensional geographic information science.* Taylor and Francis, London., 2000.

[62] R. Brachman and H. Levesque, *Knowledge Representation and Reasoning.* Morgan Kaufmann, 2004.

[63] C. Imhoff, J. G. Geiger, and N. Galemmo, *Relational Modeling and Data Warehouse Design.* New York, NY, USA: John Wiley & Sons, Inc., 2003.

[64] S. Staab and R. Studer, Eds., *Handbook on Ontologies*, ser. International Handbooks on Information Systems. Springer, 2004.

[65] M. Egenhofer and R. Golledge, Eds., *Spatial and Temporal Reasoning in Geographic Information Systems.* Oxford University Press., 1998.

[66] P. Longley, M. Goodchild, D. Maguire, and D. Rhind, *Geographic Information Systems and Science*. John Wiley and Sons, 2001.

[67] T. Hadrich, R. Maier, and R. Peinl, *Enterprise Knowledge Infrastructures.* Springer-Verlag New York Inc, 2005.

[68] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, Eds., *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.

[69] E. Malinowsk and E. Zimanyi, *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications.* Springer, 2008.

[70] M. Lambalgen and F. Hamm, *The proper treatment of events.* Blackwell Publishing Limited, 2004.

[71] D. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems.* Addison-Wesley Professional, 2002.

[72] G. Mhl, L. Fiege, and P. Pietzuch, *Distributed Event-Based Systems.* Springer, 2006.

[73] T. Faison, *Event-Based Programming: Taking Events to the Limit.* Apress, 2006.

[74] E. Malinowski and E. Zimányi, "Olap hierarchies: A conceptual perspective," in *CAiSE*, ser. Lecture Notes in Computer Science, A. Persson and J. Stirna, Eds., vol. 3084. Springer, 2004, pp. 477–491.

[75] M. D. Stefano, *Distributed Data Management for Grid Computing.* Hoboken, New Jersey.: John Wiley & Sons Inc., 2005.

[76] J. Estefan, "Survey of model based systems engineering (mbse) methodologies," Jet Propulsion Laboratory, California Institute of Technology, Tech. Rep., May 2007.

[77] C. Rolland, "A primer for method engineering," in *Proceedings of the INFormatique des ORganisations et Systmes d'Information et de Dcision (INFORSID'97)', Toulouse, France.*, 1997.

[78] P. J. Ågerfalk and J. Ralyté, "Situational requirements engineering processes: reflecting on method engineering and requirements practice." *Software Process: Improvement and Practice*, vol. 11, no. 5, pp. 447–450, 2006.

[79] P. Kruchten, *The Rational Unified Process: An Introduction, Second Edition.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2000.

[80] C. Ebert and J. D. Man, "Effectively utilizing project, product and process knowledge," *Information and Software Technology*, vol. 50, no. 6, pp. 579–594, 2008.

[81] V. Sorathia and A. Maitra, "Situation awareness unified process," in *International Conference on Software Engineering Advancements, French Riviera, France.*, 2007.

[82] DoDAF Working Group, "Dod architecture framework version 1.0, "volume 1 : Definitions and guidelines"," DoDAF Working Group, Tech. Rep., 2004.

[83] ——, "Dod architecture framework version 1.0, "volume 2 : Product desciptions"," DoDAF Working Group, Tech. Rep., 2004.

[84] A. Tang, J. Han, and P. Chen, "A comparative analysis of architecture frameworks," in *APSEC '04: Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC'04)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 640–647.

[85] M. Lankhorst, *Enterprise Architecture at Work: Modelling, Communication and Analysis.* Springer, 2005.

[86] K. Gallagher, A. Hatch, and M. Munro, "Software architecture visualization: An evaluation framework and its application," *Software Engineering, IEEE Transactions on*, vol. 34, no. 2, pp. 260–270, March-April 2008.

[87] S. Leist and G. Zellner, "Evaluation of current architecture frameworks," in *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2006, pp. 1546–1553.

[88] U. Wahli, L. Ackerman, A. DiBari, G. Hodgkinson, A. Kesterton, L. Olson, and B. Portier, *Building SOA Solutions Using the Rational SDP*, ser. IBM Redbook. International Business Machines Corporation. USA., 2007.

[89] V. Sorathia and A. Maitra, "Communications middleware requirements for situation awareness systems," in *International Conference on Software, Knowledge, Information Management and Application (SKIMA) , Nepal*, 2008.

[90] D. Bakken, *Encyclopedia of Distributed Computing*. Kluwer Academic Press, 2001, ch. Middleware.

[91] D. Schmidt and F. Buschmann, "Patterns, frameworks, and middleware: their synergistic relationships," *Software Engineering, 2003. Proceedings. 25th International Conference on*, pp. 694–704, 2003.

[92] V. Sorathia and A. Maitra, "Web based requirements management approach for organizational situation awareness," in *Proceedings of AWSOR, Italy.*, 2007.

[93] T. Modi, *Practical Java Message Service*. Manning Publications Co., 2002.

[94] B. Sotomayor and L. Childers, *Globus Toolkit 4 Programming Java Services*. Morgan Kaufmann, 2006.

[95] J. Davies, M. Lytras, and A. P. Sheth, "Guest editors' introduction: Semantic-web-based knowledge management," *IEEE Internet Computing*, vol. 11, no. 5, pp. 14–16, 2007.

[96] N. Shadbolt, T. Berners-Lee, and W. Hall, "The semantic web revisited." *IEEE Intelligent Systems*, vol. 21, no. 3, pp. 96–101, 2006.

[97] D. L. McGuinness and F. van Harmelen, "Owl web ontology language overview," World Wide Web Consortium, Recommendation REC-owl-features-20040210, February 2004. [Online]. Available: http://www.w3.org/TR/2004/REC-owl-features-20040210

[98] Z. Laliwala, V. Sorathia, and S. Chaudhary, "Semantic and rule based event-driven services-oriented agricultural recommendation system." in *ICDCS Workshops*. IEEE Computer Society, 2006, p. 24.

[99] ——, "Semantics based event-driven publish/subscribe service-oriented architecture," in *A workshop on Emerging Services delivery platforms and software models for Next Generation Network Services SOFTPLATFORM2006 in The First International Conference on Communication Systems Software and Middleware (COMSWARE)*, 2006.

[100] V. Sorathia and A. Maitra, *Encyclopedia of Artificial Intelligence*. Idea Group, Inc., 2008, ch. Ontology Mapping, p. 8.

[101] J. Broekstra, A. Kampman, and F. van Harmelen, "Sesame: A generic architecture for storing and querying rdf and rdf schema," *The Semantic Web ISWC 2002*, pp. 54–68, 2002. [Online]. Available: http://dx.doi.org/10.1007/3-540-48005-6_7

[102] K. G. Clark, L. Feigenbaum, and E. Torres, "Sparql protocol for rdf," World Wide Web Consortium, Recommendation REC-rdf-sparql-protocol-20080115, January 2008. [Online]. Available: http://www.w3.org/TR/2008/REC-rdf-sparql-protocol-20080115

[103] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," *Commun. ACM*, vol. 47, no. 6, pp. 34–40, 2004.

[104] J. Myerson, *The Complete Book of Middleware*. Auerbach Publications, 2002.

[105] Open Geospatial Consortium, "Opengis geography markup language (gml) encoding specification," OGC 03-105r1, Version 3.1.1, February 2004. [Online]. Available: http://portal.opengeospatial.org/files/?artifact_id=4700

[106] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, 2005.

[107] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, *Web Services: Concepts, Architectures and Applications*. Europe: Springer Verlag, 2003.

[108] S. Staab, W. van der Aalst, V. R. Benjamins, A. Sheth, J. Miller, C. Bussler, A. Maed, D. Fensel, and D. Gannon, "Web services: been there, done that?" *Intelligent Systems, IEEE*, vol. 18, no. 1, pp. 72–85, Jan-Feb 2003.

[109] L. Wilkes, "The web services protocol stack," 2005. [Online]. Available: http://roadmap.cbdiforum.com/reports/protocols/

[110] J. Joseph, M. Ernest, and C. Fellenstein, "Evolution of grid computing architecture and grid adoption models," *IBM Systems Journal*, vol. 43, no. 4, pp. 624–645, 2004. [Online]. Available: http://www.research.ibm.com/journal/sj/434/joseph.html

[111] S. Chaudhary, V. Sorathia, and Z. Laliwala, "Architecture of sensor based agricultural information system for effective planning of farm activities," in *IEEE SCC*. IEEE Computer Society, 2004, pp. 93–100.

[112] V. Sorathia, Z. Laliwala, and S. Chaudhary, "Towards agricultural marketing reforms: Web services orchestration approach," in *IEEE SCC*. IEEE Computer Society, 2005, pp. 260–270.

[113] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The physiology of the grid: An open grid services architecture for distributed systems integration," Open Grid Service Infrastructure Working Group, June 2002. [Online]. Available: http://www.globus.org/alliance/publications/papers/ogsa.pdf

[114] R. Buyya and S. Venugopal, "A gentle introduction to grid computing and technologies," *Computer Science of India*, vol. 9, 2005.

[115] H. Stockinger, "Defining the grid: a snapshot on the current view," *The Journal of Supercomputing*, vol. 42, no. 1, pp. 3–17, 2007.

[116] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, *et al.*, "The open grid services architecture, version 1.5," Open Grid Forum, 2006. [Online]. Available: http://www.ogf.org/documents/GFD.80.pdf

[117] Z. Laliwala, V. Sorathia, S. Chaudhary, and P. Jain, "Integrating stateful services in workflow," in *APSEC*. IEEE Computer Society, 2006, pp. 131–138.

[118] M. P. Atkinson, V. Dialani, L. Guy, I. Narang, N. W. Paton, D. Pearson, T. Storey, and P. Watson, "Grid database access and integration: Requirements and functionalities." DAIS-WG, Global Grid Forum, Tech. Rep., 2003.

[119] I. Kojima, "Design and implementation of ogsa-dai-rdf," GGF16 Semantic Grid Workshop. Athens, Greece: Global Grid Forum, Tech. Rep., 2006. [Online]. Available: http://www.semanticgrid.org/ggf/ggf16/papers/kojima34.pdf

[120] V. Sorathia and A. Maitra, *Encyclopedia of Data Warehousing and Mining (Second Edition)*. Idea Group, Inc., 2008, ch. Data Provenance.

[121] P. Kriens, "How osgi changed my life," *Queue*, vol. 6, no. 1, pp. 44–51, 2008.

[122] D. Rubel, "The heart of eclipse," *Queue*, vol. 4, no. 8, pp. 36–44, 2006.

[123] Z. Yang and M. Jiang, "Using eclipse as a tool-integration platform for software development," *Software, IEEE*, vol. 24, no. 2, pp. 87–89, March-April 2007.

[124] D. Geer, "Eclipse becomes the dominant java ide," *Computer*, vol. 38, no. 7, pp. 16–18, July 2005.

[125] Sun, *Sun Java SystemMessageQueue 4.1TechnicalOverview*. SunMicrosystems, Inc., 2007.

[126] B. Sullins and M. Whipple, *JMX in Action*. Manning Publications Co., 2003.

[127] D. Kurniawan and D. Abramson, "An integrated grid development environment in eclipse," *e-Science and Grid Computing, IEEE International Conference on*, pp. 491–498, 10-13 Dec. 2007.

[128] M. Baker, A. Apon, C. Ferner, and J. Brown, "Emerging grid standards," *Computer*, vol. 38, no. 4, pp. 43–50, 2005.

[129] A. Bradley, K. Curran, and G. Parr, "Discovering resources in computational grid environments," *J. Supercomput.*, vol. 35, no. 1, pp. 27–49, 2006.

[130] I. Foster, K. Czajkowski, D. Ferguson, J. Frey, S. Graham, T. Maguire, D. Snelling, and S. Tuecke, "Modeling and managing state in distributed systems: the role of ogsi and wsrf," *Proceedings of the IEEE*, vol. 93, no. 3, pp. 604–612, 2005.

[131] S. Zanikolas and R. Sakellariou, "A taxonomy of grid monitoring systems," *Future Gener. Comput. Syst.*, vol. 21, no. 1, pp. 163–188, 2005.

[132] H. Gibbins and R. Buyya, "Gridscape ii: A customisable and pluggable grid monitoring portal," in *Proceedings of the 5th International Conference on Grid and Cooperative Computing (GCC 2006), Oct. 21-23, 2006, Changsha, China.*, 2006, in press.

[133] B. Jacob, M. Brown, K. Fukui, and N. Trivedi, *Introduction to Grid Computing*. International Business Machines Corporation., 2005.

[134] V. Sorathia and A. Maitra, "Spatial data access patterns in semantic grid environment," in *OTM Workshops (2)*, ser. Lecture Notes in Computer Science, R. Meersman, Z. Tari, and P. Herrero, Eds., vol. 4278. Springer, 2006, pp. 1586–1595.

[135] C. J. Matheus, M. M. Kokar, K. Baclawski, J. A. Letkowski, C. Call, M. L. Hinman, J. J. Salerno, and D. M. Boulware, "Sawa: an assistant for higher-level fusion and situation awareness," in *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005. Edited by Dasarathy, Belur V. Proceedings of the SPIE, Volume 5813, pp. 75-85 (2005).*, B. V. Dasarathy, Ed., Mar. 2005, pp. 75–85.

[136] C. Matheus, M. Kokar, K. Baclawski, J. Letkowski, and C. Call, "The practical application of semantic web technologies for situation awareness," in *ICWS*, 2004.

[137] P. Mirhaji, J. Zhang, A. Srinivasan, R. L. Richesson, and J. W. Smith, "Knowledge-based public health situation awareness," in *Proceedings of the SPIE, Volume 5403, pp. 198-209 (2004).*, E. M. Carapezza, Ed., Sept. 2004, pp. 198–209.

[138] P. R. Smart, "Knowledge-intensive fusion for situational awareness: Scenario specification," Department of Electronics and Computer Science, University of Southampton., Tech. Rep., 2004.

[139] S. S. Yau, Y. Wang, and F. Karim, "Development of situation-aware application software for ubiquitous computing environment." in *COMPSAC*, 2002, pp. 233–238.

[140] D. Baker, D. Georgakopoulos, H. Schuster, A. R. Cassandra, and A. Cichocki, "Providing customized process and situation awareness in the collaboration management infrastructure." in *CoopIS*, 1999, pp. 79–91.

[141] J. A. Zachman, "A framework for information systems architecture," *IBM Syst. J.*, vol. 26, no. 3, pp. 276–292, 1987.

[142] R. Winter and J. Schelp, "Enterprise architecture governance: the need for a business-to-it approach," in *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing.* New York, NY, USA: ACM, 2008, pp. 548–552.

[143] A. Adi and O. Etzion, "Amit - the situation manager," *The VLDB Journal*, vol. 13, no. 2, pp. 177–203, 2004.

[144] R. J. Qian, N. C. Haering, and M. I. Sezan, "A computational approach to semantic event detection," *cvpr*, vol. 01, p. 1200, 1999.

[145] A. Padovitz, A. Zaslavsky, and S. W. Loke, "Awareness and agility for autonomic distributed systems: Platform-independent publish-subscribe event-based communication for mobile agents," *dexa*, vol. 00, p. 669, 2003.

[146] A. Joshi and A. Wytzisk, "Exploiting an event-based communication infrastructure for rule based alerting in sensor webs," *dexa*, vol. 00, pp. 485–489, 2005.

[147] G. Banavar, M. Kaplan, R. E. Strom, D. C. Sturman, K. Shaw, and W. Tao, "Information flow based event distribution middleware," *icdcs*, vol. 00, p. 0114, 1999.

[148] S. Bittner and A. Hinze, "Dimension-based subscription pruning for publish/subscribe systems." in *ICDCS Workshops.* IEEE Computer Society, 2006, p. 25.

[149] P. Vassiliadis, Z. Vagena, S. Skiadopoulos, N. Karayannidis, and T. K. Sellis, "Arktos: towards the modeling, design, control and execution of etl processes." *Inf. Syst.*, vol. 26, no. 8, pp. 537–561, 2001.

[150] A. Karakasidis, P. Vassiliadis, and E. Pitoura, "Etl queues for active data warehousing," in *IQIS '05: Proceedings of the 2nd international workshop on Information quality in information systems*. New York, NY, USA: ACM Press, 2005, pp. 28–39.

[151] T. Priebe and G. Pernul, "Ontology-based integration of olap and information retrieval," *dexa*, vol. 00, p. 610, 2003.

[152] G. Percivall, "Overview of geographic information standards development," in *Geoscience and Remote Sensing Symposium*, vol. 5, 2000, pp. 2096–2098.

[153] D. R. Ghimire, I. Simonis, and A. Wytzisk, "Integration of grid approaches into the geographic web service domain," in *From Pharaohs to Geoinformatics, FIG Working Week 2005 and GSDI-8,Cairo, Egypt. April 16-21,*, 2005.

[154] Y. Wei, L. Di, B. Zhao, G. Liao, A. Chen, Y. Bai, and Y. Liu, "The design and implementation of a grid-enabled catalogue service," in *Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International*, vol. 6, 2005, pp. 4224–4227.

[155] L. Di, "Customizable virtual geospatial products at web/grid service environment," in *Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International*, vol. 6, 2005, pp. 4215–4218.

[156] M. N. K. Boulos, "Web gis in practice iii: Creating a simple interactive map of england's strategic health authorities using google maps api, google earth kml, and msn virtual earth," *Int. Journal of Health Geographics*, vol. 4, p. 22, 2005.

[157] H. Chen, S. Fellah, and Y. A. Bishr, "Rules for geospatial semantic web applications." in *Rule Languages for Interoperability*. W3C, 2005.

[158] P. Currion, C. de Silva, and B. V. de Walle, "Open source software for disaster management," *Commun. ACM*, vol. 50, no. 3, pp. 61–65, 2007.

[159] R. Iannella and K. Robinson, "Tsunami warning markup language (twml). a standards-based language for tsunami bulletins. version 1.0. 52," National ICT Australia - NICTA, Tech. Rep., 2006.

[160] D. King, "Structured humanitarian assistance reporting (share)," in *Global Disaster Information Network (GDIN) Conference 2001, Canberra, Australia*, 2001.

[161] M. Turoff, M. Chumer, B. V. de Walle, and X. Yao, "The design of a dynamic emergency response management information system (dermis)." *Journal of Information Technology Theory and Application*, vol. 5, no. 4, p. 136, 2004.

[162] D. D. Dudenhoeffer, M. R. Permann, and M. Manic, "Cims: A framework for infrastructure interdependency modeling and analysis," *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, pp. 478–485, Dec. 2006.

[163] P. Pederson, D. Dudenhoeffer, S. Hartley, and M. Permann, "Critical infrastructure interdependency modeling: A survey of u.s. and international research," Idaho National Laboratory, Tech. Rep., 2006.

[164] S. Rinaldi, J. Peerenboom, and T. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *Control Systems Magazine, IEEE*, vol. 21, no. 6, pp. 11–25, Dec 2001.

[165] J. Hollman, J. Marti, J. Jatskevich, and K. Srivastava, “Dynamic islanding of critical infrastructures: a suitable strategy to survive and mitigate extreme events,” *International Journal of Emergency Management (IJEM) Special Issue on Critical Infrastructure Protection*, vol. 4, no. 1, 2007.

[166] *26th International Conference on Distributed Computing Systems Workshops (ICDCS 2006 Workshops), 4-7 July 2006, Lisboa, Portugal.* IEEE Computer Society, 2006.