



ActiveEdge® Situational Reasoning Framework Technical Overview

Abstract

In today's rapidly evolving technological and business environments, situation awareness becomes critical to making the right decisions and achieving business goals quickly and most effectively. The key to 'situation aware' applications is not in simply collecting and disseminating data, but it is actually in getting the right information to the right users at the right time, what we call your business 'knowledge.' In your enterprise, various sensors and other data sources gather your data. You need a platform that knows what to do with that data to make it meaningful, to reason about that data in a context that is relevant to your current situation and to the current user, and to update that information continuously as the situation changes. Cougaar Software's powerful ActiveEdge platform is designed to transform your data into relevant, actionable knowledge. Based on a combination of modern game theory techniques and concurrent network concepts, situation management is one of the most innovative components of ActiveEdge. Using our unique situational reasoning framework (SRF), ActiveEdge provides real-time display of the dynamic situation in a rich, virtual world picture, which can be further acted upon by the ActiveEdge services and your custom applications.

<http://www.cougaarsoftware.com>
info@cougaarsoftware.com

Contents

Executive Summary	3
Background	4
What Is Situation Awareness?.....	4
Situation Awareness & Situational Reasoning in Software.....	4
An Intelligent Platform for Situation Management	5
ActiveEdge’s Intelligent Capabilities	5
ActiveEdge Architecture & Components.....	6
How Does It Work?	7
How Does this Apply in the Real World?.....	8
Scenario 1	8
Scenario 2	8
Situation Management in ActiveEdge	9
The ActiveEdge Situational Reasoning Framework.....	9
Domain Controllers.....	11
Resolution Engine.....	12
How Domain Controllers & the Resolution Engine Contribute to the Situational Picture in ActiveEdge	12
Actor Perspectives.....	13
Network Management Components.....	13
Actors.....	14
Situational Object Space	14
Network Object Space.....	14
In Conclusion.....	15
About Cougaar Software	15

Executive Summary

To be successful in today's technological and business climates, it is imperative that business processes be efficient, timely, and accurate. However, in the modern enterprise, business operations are in a constant state of flux evolving from one form to another in fluid and often unpredictable ways. The major problem for the modern enterprise is how to become more aware, more flexible, and more agile. Making employees aware of current trends and the current situation overall and providing timely and accurate business knowledge allow businesses and the users in that business to anticipate and respond to what is happening in an unpredictable and changing environment. Flexibility enables an enterprise to plan for the anticipated changes, while agility allows an enterprise to respond to unanticipated, unpredictable changes, which is key to operating in the modern enterprise.

To achieve the ultimate combination of awareness, flexibility, and agility is to support dynamic and flexible processes that adapt as business situations change. Such systems will need to support growing volumes of information and produce accurate and relevant metrics about past, current, and future situations. And, they will need to provide complete control and visibility into the entire enterprise. These features ultimately require a flexible, distributed architecture that enables secure collaboration, advanced information management, dynamic system updates, and custom, rule-based processes. This allows users to work in a more timely and effective manner, reduce overhead costs on wasted resources or time-consuming processes, and get the information they need when they need it. Information that is shareable inside an organization, with partners, and with customers provides information that is more meaningful and produces overall, shared awareness. Shared awareness, in turn, enables faster operations; and, faster, effective work means greater productivity, greater competitiveness, and greater profit margins for your enterprise.

Intelligent agent technology provides the means to achieve these requirements most effectively. More flexible and sophisticated than peer-to-peer or object-oriented computing, intelligent agent-based systems deal with massive amounts of information to levels of accuracy, timeliness and quality never before possible. Applications that use only object-oriented technology usually fail to achieve the business goals of awareness, flexibility, and agility because processes are hard-coded in the applications. And, customizing business processes tends to be the most costly component of enterprise systems because of this shortfall. When enterprise applications are process-oriented and can enable accurate and current situation awareness, dynamic planning, and rich collaboration capabilities as intelligent agent systems can, these goals can be achieved quickly and cost effectively.

Managing the problems of dynamic change, agile execution, and flexible business processes are the main challenges that Cougaar Software solutions address. With our ActiveEdge platform, which provides an infrastructure for building distributed, intelligent agent-based applications, and our unique situational reasoning framework (SRF), you can achieve a knowledge-enabled enterprise that provides accurate and comprehensive situation awareness for your local and global enterprise operations.

Background

What Is Situation Awareness?

A situation is the combination of people, things, information, and capabilities that, together, have relevance to the user at a certain moment in time.

Situation awareness can be described broadly as a person's state of knowledge of the situation surrounding him or her. Situation awareness involves perceiving critical factors in the environment, understanding what those factors mean in relation to a person's goals, and understanding what will happen in the near future.

Situation Awareness & Situational Reasoning in Software

In the modern enterprise, where situations are in a constant state of flux and people are inundated with vast amounts of information, situation awareness is becoming more and more critical to effective decision-making for current (reactive response) and future situations (proactive response). Situation awareness becomes critical for your competitive edge, your customer's satisfaction, and for more efficient, cost-saving business operations overall.

Situation awareness is critical for effective decision-making and performance in many modern business domains and systems including logistics, supply chain, retail, health care, military operations, and various other environments with widely dispersed locations and diverse data sources such as wireless or mobile devices/content, sensors, legacy systems, and traditional data bases. In these complex and dynamic environments, human decision-making is highly dependent on situation awareness — the constantly evolving picture of the state of the environment. To provide the most accurate and complete situational picture for modern business systems, the following areas must be considered:

- **Diverse Data Sources** - Business environments create a plethora of data through Automatic identification technologies (AIT) such as Radio Frequency Identification (RFID) tags, bar codes, smart cards, physical sensors (e.g., climate control), actuators (e.g., security systems and network), and global positioning systems. All of this data creates an integration nightmare and produces a data volume that either threatens to drown traditional backend systems or is too cumbersome to manage.
- **Information Relevance: Domain Knowledge & Ontology** – Translating raw data into information that is relevant to the current situation, that is, domain-specific knowledge, is required. Semantic concepts and data ontologies are essential because semantics provide an understanding of the overall concept of a situation versus a one-dimensional understanding of individual objects. For example, the concept of a secretary who works in an office environment is more meaningful than the individual objects of a file or a desk. Or, in a healthcare scenario, if a physician using a mobile device enters a patient's room, but the electronic situation that is assembled for him has no semantic concept of the information that is being shown, it will likely not be very useful. However, if the device knows what an EKG or an X-RAY is, and that the patient had a stroke while being on a particular medication, it can represent that situation much better for the physician. Semantics makes situational software more useful because they consider knowledge of the application domain.
- **Dynamic Re-planning** – A situation dynamically changes as conditions in the situation change. Mechanisms must be available to monitor and address any unforeseen changes, allowing custom and dynamic actions during execution.

- **Data Mediation and Situational Reasoning** – You need a platform that can reason about and manipulate the massive amounts of disparate data to provide relevant information to the appropriate users while maintaining a global picture of the situation.

The key to ‘situation aware’ applications is not simply in collecting and disseminating data, but is actually in getting the right information to the right users at the right time, what we call your business ‘knowledge.’ In your enterprise, various sensors and other data sources gather your data. You need a platform that knows what to do with that data to make it meaningful, to reason about that data in a context that is relevant to your current situation and to the current user, and to update that information continuously as the situation changes. ActiveEdge leverages the dynamic characteristics and cognitive reasoning capabilities of intelligent agent technology to provide advanced, accurate, and comprehensive situation management.

An Intelligent Platform for Situation Management

Based on the preceding definitions of situation awareness and the associated business needs, Cougaar Software’s goal is to provide a representation of a situation that focuses on the current relevance of information to an end-user and that enables secure collaboration, advanced data management, dynamic updates, and custom, rule-based processes.

Cougaar Software’s ActiveEdge is an intelligent agent-based middleware platform for building complex, distributed, intelligent applications. ActiveEdge is built on the Cognitive Agent Architecture (Cougaar)—an open source, distributed agent architecture. ActiveEdge provides all the power of Cougaar and includes key extensions to simplify application development, increase agent functionality, and provide enhanced system capabilities. ActiveEdge is designed to transform data into usable knowledge and provides advanced situational awareness, creating an understandable real-time picture of current enterprise operations. In addition, ActiveEdge provides advanced execution monitoring and collaborative decision support.

ActiveEdge’s Intelligent Capabilities

ActiveEdge uses a distributed intelligent agent architecture based on the human cognitive model of reasoning and planning. This advanced software model captures the way humans observe, reason, plan, and act. It allows you to automate the complex processes humans do everyday more realistically and robustly than other traditional technologies.

Intelligent agents are software entities that operate continuously and autonomously in a particular environment. An agent’s responsibilities are defined by the behaviors that have been built into it, and agents carry out their activities in a flexible and intelligent manner that is responsive to changes in the environment. For example, in a supply chain scenario, a supply agent could predict and address the problem of an imminent stock-out before it affects operations. Agents can adjust behavior dynamically to fit the current situation, determining how their actions and behaviors should change as events change. Agent-based systems represent the next major advancement in network computing and leverage the strengths of object-oriented, peer-to-peer and service-oriented architectures while providing a process-centric design. The real value is that intelligent reasoning occurs at each level of the system to reduce overall system load and increase quality, control, and responsiveness. The key benefits of agent technology come in these areas:

- **Dynamic Re-planning** – The ability to develop and modify distributed workflows using rules and domain knowledge that is appropriate to the current situation. This benefit allows

enterprises to create more accurate and appropriate plans and to react more quickly and appropriately when conditions change.

- **Advanced Data Mediation** – The ability to gather and process data from multiple diverse sources into a single environment so that it is appropriate for the current situation.
- **Situational Awareness** – The ability to build and maintain a virtual world representation of the current situation on which intelligent reasoning can occur.
- **Collaborative Information Management** – The ability to easily share information and coordinate changes across your enterprise.
- **Intelligent Reasoning** – The ability to emulate the way humans observe, reason, plan, act, and monitor at computer speeds.
- **Scalable, Distributed Computing** – The ability to handle massive amounts of data across the enterprise while providing more efficient processing.
- **Business Process Adaptation/Evolution** – The ability to allow significant business changes to be implemented quickly and dynamically by actual users who can easily manage adjustments to the business rules or policies—without engaging consultants to significantly alter their systems. This benefit allows enterprises to be agile and adaptive as conditions change, thus saving valuable costs in process re-engineering.

ActiveEdge Architecture & Components

ActiveEdge offers a full framework for building intelligent applications and provides a comprehensive tool suite for developing such applications. Based on existing standards and open technologies like Java, SOAP, OWL, XML, and the Cognitive Agent Architecture (Cougaar), ActiveEdge saves significant research time and development effort as well as takes advantage of proven capabilities. The ActiveEdge architecture, as shown in Figure 1, is composed of a series of layers that each provides specific services and capabilities. It provides real-time intelligence and event management by using agents. ActiveEdge manages real-time processing of high volumes of data (from sensor signals, external systems, etc.) and transforms this data into intelligent and actionable business information. The actionable information is then delivered as powerful dashboards to individuals or pushed to systems in or outside the enterprise.

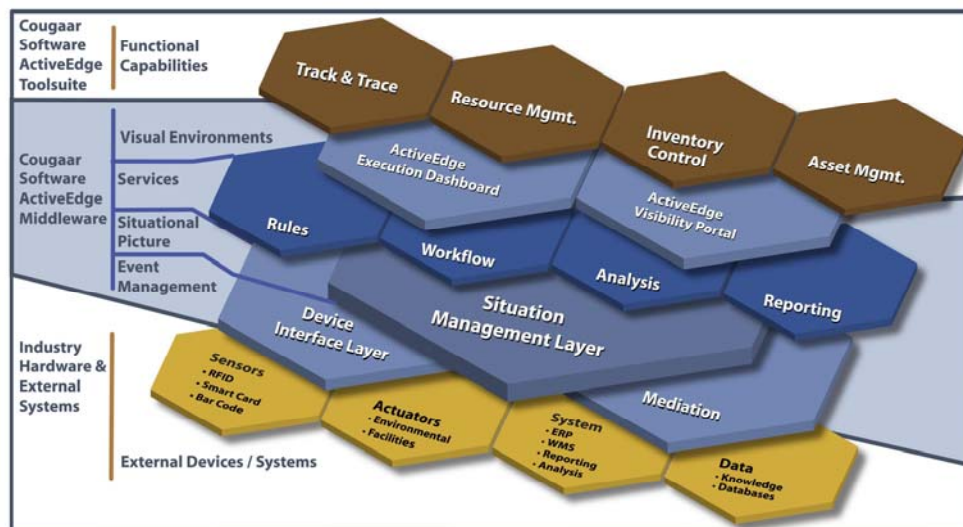


Figure 1 The Logical ActiveEdge Architecture

ActiveEdge provides a user-friendly visualization tool that lets users quickly and easily create task-specific, collaborative applications. In addition, users can modify process flows graphically without changing any code—they can easily ‘drag-and-drop’ components from one application to another.

In the middleware layer, the robust ActiveEdge Distributed Data Environment (DDE) offers advanced information management and data mediation capabilities and can support volumes of data from diverse, distributed sources and models, integrating that data into a single environment. Based on this mediated data, the ActiveEdge Situation Management component provides situation awareness and active visibility into your operations by presenting a virtual world view of your entire enterprise activities. In addition, it provides capabilities for advanced monitoring and alerts, automated rules and workflow, and quick, dynamic adjustment of those rules and workflow as needed.

How Does It Work?

Leveraging the power of intelligent agents, all of the ActiveEdge components work together to transform your raw data into current, actionable knowledge.

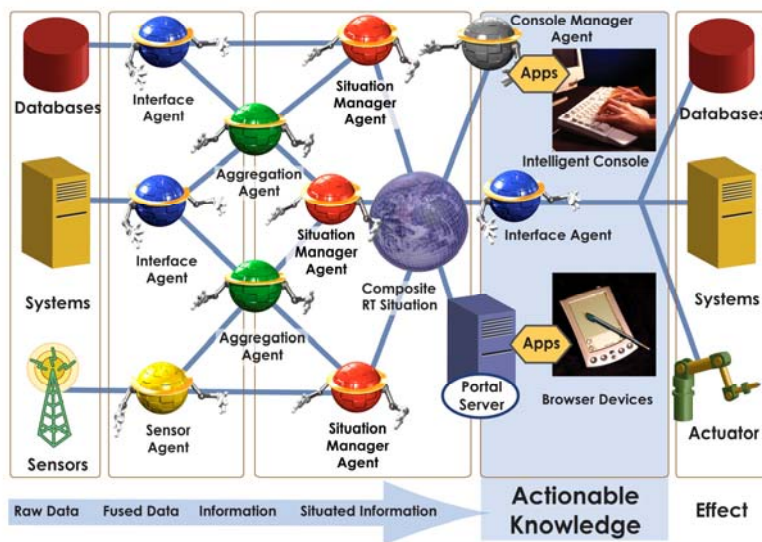


Figure 2 Transforming Data into Knowledge

1. In this diagram, raw data from sensors, external systems, and databases enter the system through Device Interface and Sensor Agents.
2. There, the raw data is filtered and fused into data events.
3. Aggregation Agents assemble the fused information as needed to provide just the right level of detail.
4. The Situation Manager agents then further fuse and reason about the events in relation to current processes, activities, and the environment. This creates and maintains a situational view of current information, and this view serves as the basis for your business applications, giving your users the information they need to understand the current business situation and to act upon it appropriately.
5. In addition, interface agents at the situation level give your legacy systems and databases current information updates in the proper form.

How Does this Apply in the Real World?

In this section, we provide a couple of examples for situations in which situational software is advantageous.

Scenario 1

Consider the following military scenario. A military unit uses a system that acts as a distributed virtual machine, interlinking devices in the environment to maintain a current situational picture of a soldier, the global/overall mission, and the current environment. Figure 1 below depicts a tactical scenario in which soldiers and vehicles operate together in a shared network, executing operations in a distributed, collaborative fashion. The panel on the left shows how soldiers collaborate on activities and use situational information during the mission in changing environments. The panel on the right depicts the underlying layer of intelligent agents that operate behind the tactical information displays to gather, fuse, and present the data and manage the visual information processes relative to the situation. As the situation evolves, the agents continually work to keep the tactical displays relevant and appropriate with special consideration for the users' current activities and mental states.

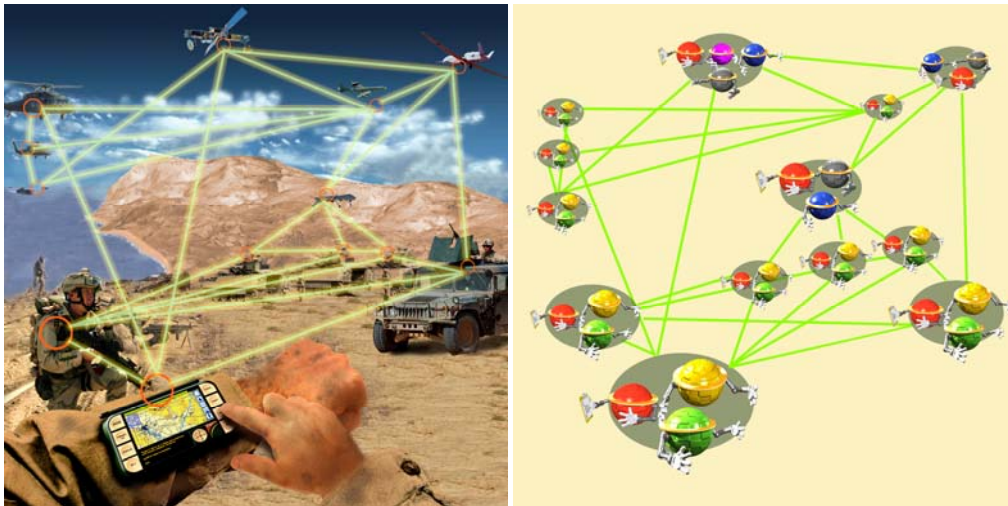


Figure 3 Example Tactical Situation

Scenario 2

Now, consider this healthcare scenario. A physician uses a mobile device with software that is aware of the care situation of current patients. The mobile device can access things (such as patient monitoring devices) and information resources (such as patient records), provides capabilities (such as ordering tests or sending chart data), knows other persons involved in the care of this patient (such as other doctors). The device software ultimately provides communication and collaboration that allow the physician to work optimally with the patient and other care providers in this situation. When the physician's situation changes, for example, the physician moves to the next treatment room with the next patient, the mobile device recognizes the change in situation and provides appropriate support for the next situation. As the situation of the second patient is different, the mobile device will likely access different information resources, provide different capabilities, and connect the physician to different people.

Situation Management in ActiveEdge

In the middleware layer of ActiveEdge, our powerful Distributed Data Environment (DDE) supports secure, selective, and seamless data integration and collaboration among sites in the same organization as well as with employees, partners, and customers at different sites. The ActiveEdge DDE is a state-of-the-art data mediation system that supports advanced mediation and mapping, selective data replication, and intelligent querying using standard and object query constructs as well as higher level, advanced semantic-based queries.

Using Semantic Web concepts, which focus on a standard ontology for representing domain knowledge and that allows disparate entities to understand each other, it ensures that various entities can collaborate. Semantic concepts add a new dimension to data mediation. They provide the domain knowledge upon which situational reasoning can occur. From the mediated data, ActiveEdge creates an understandable, real-time picture of the current situation as well as provides advanced event management, execution monitoring, and collaborative decision support based on that picture.

Based on advanced, distributed game theory techniques and concurrent, optimized network representations, situation management is one of the most innovative and unique components of the ActiveEdge architecture. The ActiveEdge Situation Management component is based on our unique situational reasoning framework (SRF), which provides a distributed infrastructure for reasoning about real world scenarios. The situation management component builds a virtual representation of the current situation from various incoming data event streams whether they are random or predictable. This resulting situational picture is a rich, object representation of the state of current operations similar to the virtual world representation in a modern video game. We are able to identify and reason on whether data is redundant or conflicts with other data streams.

The core services of ActiveEdge systems act on the situational picture to support functional processes and applications. And, these functional applications use the situation and core services to perform functional planning, reasoning, display, and analysis for users. With ActiveEdge, situations are monitored, and, depending on defined events, actions are generated according to user-defined business rules and workflows. Software agents can build and/or adjust these business rules and processes based on rule patterns and their ability to learn from prior actions. The situation management component uses situational and domain agents, or reasoners, to translate an event or group of events into a meaningful situation in the current context and situation. The context includes the set of actions the system is performing, the set of events received, and the actions currently operating on the system. The domain controllers correlate events and reason about the changes to the situation that result, considering not only primary impacts, but also secondary and tertiary impacts.

The ActiveEdge Situational Reasoning Framework

The following quote from Col John Boyd (ret), who coined the concept of the “OODA Loop” (Observation, Orientation, Decision, Action), provides the motivation for our ActiveEdge Situational Reasoning Framework (SRF):

“Knowledge of the strategic environment is the first priority. One must also be able to interact with the environment and those within it appropriately. You must be able to observe and orient yourself in such a way that you can indeed survive and prosper by shaping the environment where possible to your own ends, by adapting to it where you must. Doing so requires a complex set of relationships that involve both isolation and interaction. Knowing

when each is appropriate is critical to your success. In OODA Loop fashion, one must continually observe, orient, decide, and act in order to achieve and maintain freedom of action and maximize the chances for survival and prosperity.”

The goal of the ActiveEdge SRF infrastructure is to provide a generic, robust, and distributed framework for situational reasoning. The SRF allows users or software agents (which we will refer to as actors in the remainder of this paper) in the system to define, create, and remove situations dynamically. Actors can interact with situations hosted by the SRF scenario controller in both ‘turn’ and non-‘turn’-based fashions. The SRF incorporates a Graphical Information System (GIS) object model to allow efficient rendering of the situation.

Each actor that participates in a situation receives situational updates filtered by a domain specific perspective that uses the object model and domain knowledge of the relationships between those objects to determine the fidelity of the information sent to an individual actor. The perspectives are a vital component of the infrastructure that enables the system to reason about massive amounts of data.

The ultimate vision for the SRF is a dynamic system where actors, scenarios, and conditions are continually evolving; and the agent-based platform of ActiveEdge provides the secure, robust, and distributed agent environment required to achieve this vision. The SRF can host massive scenarios, and this capability derives from the distributed nature of agent societies. The SRF uses agents to provide an efficient infrastructure for allowing the host of a scenario to send only the necessary pieces of information to other agents participating in the scenario.

Intelligently determining where and how much information is a key element of the infrastructure. It allows it to handle large scenarios composed of massive amounts of scenario artifacts (which are properties related to the scenario) and participants without overwhelming the network infrastructure. The diagram below illustrates the concept model for the SRF. A single scenario controller can host many distinct scenarios at any one time, and the scenario controller responsibilities include:

- Allowing actors to join and leave scenarios dynamically.
- Advertising the types of scenarios the controller can host.
- Advertising the current list of scenario instances that are currently instantiated and running.
- Advertising whether a specific scenario instance currently allows new participants to join it.
- Sending scenario updates to participants based on the rules of the scenario model.
- Receiving actions from scenario participants.
- Resolving and de-conflicting scenario actions.
- Providing domain controls that take resolved scenario actions and affect the scenario artifacts that represent the situational picture.

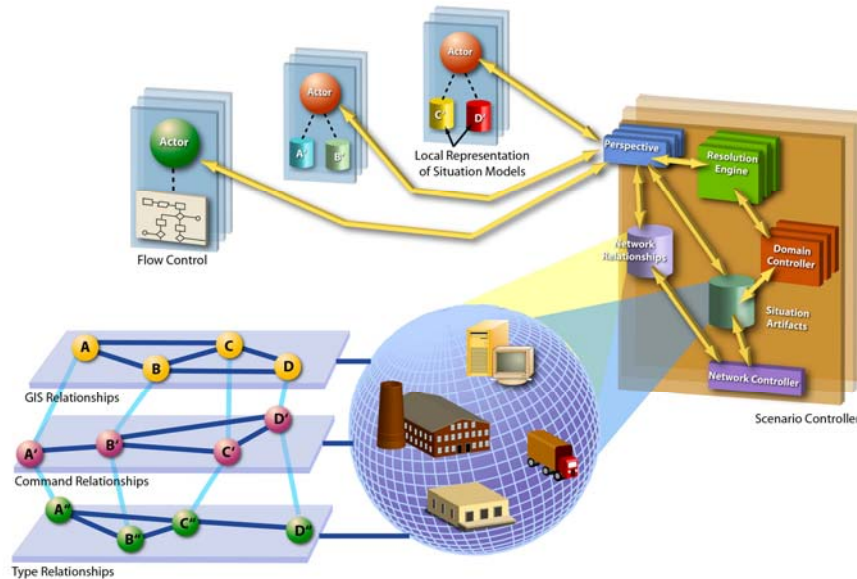


Figure 4 Conceptual View of the Situational Reasoning Framework

One or more actors participate in a scenario hosted by a single scenario controller. Each scenario instance has a Scenario Flow Control Agent that performs these functions for a scenario instance:

- Initializes and de-initializes the scenario on the scenario controller.
- Sends initialization information to each scenario participant (if necessary).
- Controls the turn token in turn-based scenarios by sending the turn token to the appropriate participant and updating the turn token on the scenario controller. This is so that the resolution engines on the scenario controller can validate actions in turn-based scenarios.
- In scenarios where there are winners and losers, the flow controller determines when the game ends and who won or lost the game.

Actors may participate in more than one situation, and there are multiple scenario controllers in the framework. Scenario Controllers can host one or more scenario types and can host multiple instantiations of the same or different scenario types concurrently.

Domain Controllers

The Domain Controllers manipulate objects within a Situational Object Space (SOS). Domain Controllers are triggered by actor actions (i.e., moves), internal events, and other domain controllers. Domain Controllers are domain specific and modify objects in the SOS based on rules defined by the domain experts who create the scenario. Domain Controllers are not directly accessed by actors in the system. Instead, Domain Controllers are invoked by the Resolution Engines after the Resolution Engines have resolved and de-conflicted actions submitted by actors.

The SRF infrastructure provides a non-domain specific way to represent the relationships between as well as the responsibilities of Domain Controllers. The ActiveEdge Workflow Engine provides a straightforward way to define relationships of hierarchical domain controllers. The Domain Controller infrastructure has predefined processes and knowledge for expanding specific activities. Domain Controllers can execute and expand activities into lower level processes until the activity has been decomposed to the point where a low-level activity results in the atomic manipulation of SOS objects.

Domain Controller execution is controlled by Workflow processes. These processes have start predicates that trigger the execution of a process when certain conditions are met. Some example predicates for starting domain controller processes could be:

- Execute every n milliseconds.
- Wait for moves from all players to be fully resolved before executing.
- Process each resolved action individually.

Resolution Engine

The Resolution Engines de-conflict and resolve actions from situation actors. Actions submitted by situation actors are either resolved immediately or in a periodic fashion depending on the type of scenario the action is for. For example, in a turn-based scenario where each actor has a turn and no other actor can take an action until that turn has been resolved, the resolution engine processes the action immediately. In a non-turn based scenario where each actor can submit a move at any time, the resolution engines queue incoming actions and process them in batch mode. This mode of processing requires domain specific resolution engine logic that can de-conflict actions. The resolution engines would notify actors if the submitted move has been accepted or rejected.

The set of Resolution Engines for a specific scenario contain all of the details and rules that determine the extent to which an action is successful as well as the corresponding penalties or rewards. The Resolution Engines, like the Domain Controllers, are hierarchical in nature. High-level Resolution Engines decompose actions into various sub-actions until a low-level Resolution Engine can determine the extent to which a move is successful. The success or failure of a low-level action is propagated back up to the original high level action where the overall extent of the success or failure of the action can be determined. If the Resolution Engines are functioning in batch mode, the final outcome of the action cannot be determined until all of the actions in that batch have been resolved. For example, an action that originally is determined to be successful may actually be a failure because a higher priority conflicting action causes the Resolution Engine to change the status from successful to unsuccessful.

How Domain Controllers & the Resolution Engine Contribute to the Situational Picture in ActiveEdge

With the approach described in the previous two sections, we can correlate incoming events to update not only the state of the virtual world, the current situational picture, but also to reason about primary and secondary impacts on other aspects of the world. This representation is very extensible, able to capture the uniqueness of a domain through resolution and domain controllers. The resolution controllers de-conflict and prioritize incoming event streams while the domain controllers provide domain specific reasoning about the meaning of an event and the implications.

Domain controllers provide secondary updates to the situation by essentially using recursion on any situation changes. They reason about changes as though the change were a newly posted event. The recursion then stops when no additional changes are detected. This process continues as new and secondary events are being generated. The situation is actually a partitioned, distributed representation that is processed by one or more situation manager agents and is propagated to the other agents in the society. These propagated changes also act like incoming events and are processed through the same resolution and domain controllers, which enables events with wide-ranging effects to be fully realized in the situation.

Actor Perspectives

Each Actor in a situation interacts with the scenario controller through a perspective, and each Actor requires a perspective. The Perspective facilitates all of the Actor's actions and views of the situation. A single perspective can serve multiple Actors or each Actor can have its own perspective, depending on the scenario. For example, in a perfect knowledge situation where each player has exactly the same set of functionality and knowledge of the world, a single perspective may suffice. However, in a less than perfect knowledge situation where each Actor or class of Actor has a different set of constraints on the amount and the fidelity of information about the current situation that the actor receives, then multiple perspectives may be required. The Perspective contains all of the rules that apply to the set of Actors it facilitates. These rules dictate the information the Actors will have access to, and to some extent, constrains the actions they can perform.

A unique feature of the SRF is the mechanism used to determine the set of information that an individual actor in the situation should receive. This mechanism, like many other features of an intelligent agent society, is enabled by the aggregation of many small and relatively simple decisions. Each time an artifact, or property, is added or modified in the scenario controller, the perspective determines if, given the domain-specific rules for which the perspective is designed, the object should be sent. The perspective has no need to perform additional logic to determine the complete set of objects that should be sent to a participant given the fact that a single artifact is being sent to that participant.

The Perspective is not a single entity; instead, it is a set of components (plug-ins, services, and logic providers) that facilitate an Actor's interaction with the scenario controller. For example, in turn-based scenarios, the Perspective components inform an Actor of its turn to move.

Network Management Components

Manipulation of scenario artifacts is handled by the Domain Controllers. When a scenario artifact is modified by a Domain Controller, it triggers the execution of the network management components. The Network Management Components maintain one or more networks of meta-information that describe relationships between artifacts in a scenario. Each network is represented by a graph structure with nodes and edges. The nodes describe some meta-data about an artifact in the scenario, and the edges describe a relationship between two artifacts. The graph may be directed or undirected and connected depending on the domain-specific nature of the relationships the graph represents. The following figure shows a sample network as a fully connected undirected graph.

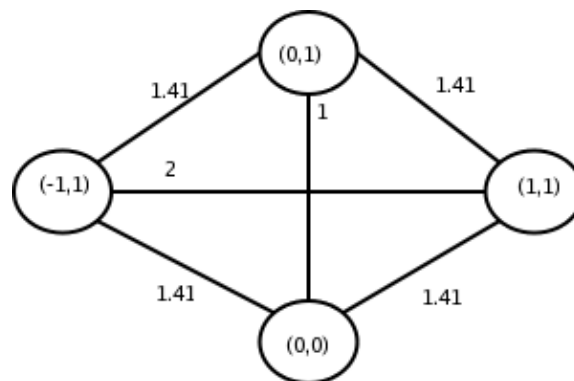


Figure 5 Sample Network Showing a Fully Connected Undirected Graph

In this diagram, each vertices represents a coordinate position of an artifact in the scenario. Each arch represents the magnitude of the distance between each of the vertices. The type of graphs created by the domain specific network management components are not restricted by the SRF architecture; however, implementers should consider the complexity of maintaining the graph when choosing the graph type for their domain.

In the sample graph above, if the artifact at the node labeled (0,0) is modified by the domain controller, the network manager for this graph would determine if the modification changed the position of the artifact. If the position was changed to (0,1), then the network manager would update the node (0,0) and change it to (0,1). As a result of the change, each edge connected to the modified node would be updated to reflect the new distance between the node and every other node in the graph.

The perspective for the scenario is invoked for the modification of the artifact and the modification of the node in the network that references the artifact. The perspective uses the same rules to determine if the artifact should be sent to a specific participant that it will use to determine if the node referencing that artifact should be sent to the participant. If the node is sent, then each edge and the other node connected to that edge will be sent to that participant as well. If the participant that received a node referencing an artifact that it does not have a copy of on its local blackboard, the participant can send a request to the scenario controller for that artifact. It will be up to the perspective to determine if the participant can receive the artifact.

Actors

An Actor is a self-interest driven participant in a scenario. Modeling and observing the strategic interactions between Actors in a scenario is the core capability of the SRF. An Actor is linked to the scenario world through its perspective. Through the Perspective, an Actor can view the situation and submit actions. Actors in a scenario have the capability to send and receive messages from other actors in the scenario. These messages can be point-to-point or multi-cast messages. If a scenario needs to reason about or log communication between actors, the messages must be sent through the scenario controller and then forwarded to the intended recipients.

Situational Object Space

The Situational Object Space (SOS) represents the collection of entities that exist in the playing field scenario. This includes all physical objects and scenario tokens as well as abstract concepts and information relevant to the objects and Actors. Scenario actions, moves, and information queries that are validated by the resolution and domain controllers are eventually manifested as manipulations of objects within this space or result from observation of the state of objects within this space. The SOS contains few, if any, rules. The rules that govern the access and manipulation of the space reside in other components within the framework.

Network Object Space

The Network Object Space (NOS) is a graphical structure layered on top of the SOS. The NOS forms edges that represent relationships between SOS objects. The purpose of the NOS is to provide an efficient data structure that makes it simple to determine the relationships between sets of SOS objects quickly. Components of the system that update and transmit portions of the SOS use the NOS instead of exhaustive searches of the entire SOS.

Domain specific network management plug-ins monitor the SOS for modifications and actively maintain the network structure. When a Domain Controller adds a new object to the SOS, the network management plug-ins create edges between the newly added object and existing objects in the SOS. The network management plug-ins, like domain controllers and resolution engines, are hierarchical and iterative.

In Conclusion

Effective decision-making in dynamic business environments is highly dependent on accurate, timely, and responsive Situation Awareness information—the evolving picture of the state of the environment. In today’s rapidly evolving technological and business environments, situation awareness has become critical to making the right decisions and achieving business goals quickly and most effectively.

With the rise of the Internet and associated Web-based applications and geographically dispersed sites in the modern enterprise, there is a new, more challenging need for infrastructure that supports distributed, instantaneous information gathering and dissemination. This infrastructure must also be able to support the various related systems and devices such as sensors, mobile devices, legacy data systems, and overall business system integration.

The ultimate solution provided by ActiveEdge and the ActiveEdge SRF is a dynamic, distributed, agent-based, situation-aware system in which actors, scenarios, and conditions are continually evolving and where your diverse data becomes relevant and actionable business knowledge. ActiveEdge leverages the dynamic characteristics and cognitive reasoning capabilities of intelligent agent technology to provide advanced, accurate, and comprehensive situation management.

About Cougaar Software

Cougaar Software, Inc. is a leading provider of intelligent solutions for the commercial and military sectors. We were formed in 2001 to transition a powerful suite of advanced, intelligent agent-based technologies from military research to the commercial market. Through this work and subsequent military investments, we have successfully productized this technology in various secure, time-critical operational environments. We have spent the last two years extending and refining our framework to become a total platform, called ActiveEdge, for quickly building and deploying distributed, intelligent agent-based applications.

Because of our unique distributed agent approach and expertise, we can build systems larger and more complex than possible with conventional architectures. Intelligent agent applications enable secure, dynamic collaboration over complex business processes that handle volumes of information to levels of accuracy, timeliness, and quality never before possible.

Please contact us today for more information about our powerful ActiveEdge technology or to schedule a demo of our intelligent architecture platform and products.

Cougaar Software, Inc.
7925 Jones Branch Drive, Suite G300
McLean, VA 22102
Phone: (703) 506-1700
Fax: (703) 506-8010
Email: info@cougaarsoftware.com