An Intelligent Supply Chain Planning and Execution Environment

Jens Pohl, PhD
Executive Director, Collaborative Agent Design Research Center (CADRC)
Director, KML Center (Scott AFB, Illinois)
California Polytechnic State University (Cal Poly), San Luis Obispo, California

Abstract

Logistic planning and execution processes in a supply-chain are subject to a high level of complexity because of the number of parties and issues involved, the number of relationships that exist among them, and the dynamic nature of the execution environment. The large volume of data flowing through a sizable computer-based logistic planning and execution management environment that is based on rote data-processing principles tends to overwhelm the human users. As a result many opportunities for improving the efficiency of supply-chain processes and thereby reducing costs are overlooked by the human users, who are forced into a reactive mode.

Similar data deluge symptoms are being experienced in other domains such as Internet searches where the number of website hits returned for a single query can easily exceed several million. The data deluge problem could be overcome if the context of the query could be defined by the user and executed by the search engine in a context-based manner. This would require the representation of a virtual model of real world context in the search software. The same need for the representation of context in software exists also in the cyber security domain where data encryption must be supplemented by the profiling of users and the continuous monitoring and automated interpretation of network behavior.

This paper discusses the design concepts and implementation principles, and describes the end-state capabilities of a computer-based intelligent logistic planning and execution environment that includes a virtual model of real world supply-chain context and multiple agent groups that are able to interact with each other and the human users. Implemented in a service-oriented architecture (SOA) based infrastructure, the virtual context model provided by a multi-layer ontology and the collaborative agents are able to continuously monitor the state of the supply-chain by interpreting the flow of data in the appropriate context. This allows the agents to rapidly re-plan in case of supply-chain interruptions, discover and act on opportunities for improvements, and identify patterns and trends based on the continuous analysis of historical data. As a result the human users are relieved from lower level data interpretation tasks and provided with actionable information for reactive and proactive planning and execution management functions. The author suggests that order of magnitude improvements in efficiency and reduction in cost are achievable with context-based information-centric software systems.

1. Supply-Chain and Logistics

Organizations exist for some purpose and in virtually all cases this purpose involves the creation and delivery of products, in the form of goods and/or services. To achieve its purpose the organization uses a variety of resources such as people, information, materials and/or components, and money, to perform operations that result in the delivery of products to its customers. The required operations may include any number of activities such as manufacture, transportation, training, serving, and selling, and typically involve many activities and relationships that need to be coordinated within a network of interacting entities. The Chartered Institute of Logistics and Transport (1998) defines supply-chain and logistics as follows:

> *"The supply-chain is a sequence of events intended to satisfy a customer. It can include procurement, manufacture, distribution and waste disposal, together with associated transport, storage and information technology."*
>
> *"Logistics is the time related positioning of resources or the strategic management of the total supply-chain."*

The principal objectives of supply-chain management are normally focused on optimizing the sequence of operations in combination with the resources that are required to perform the operations so that the expectations of the customer are satisfied at least cost to the organization. There are many factors that can make it difficult to achieve an optimum supply-chain management outcome (Waters 2007). The logistical functions involved comprise a series of related activities, including acquisition, receiving, warehousing, inventory management, order processing, transportation, distribution, and so on. The workflow processes involved are often quite complex and typically involve several parties with different skill sets and objectives. In a global supply-chain the need to move goods and services across national borders increases the potential complexity by an order of magnitude. At the same time the desire to minimize inventory increases the risk factor and makes it incumbent on the organization to proactively anticipate disruptive events and effectively react to disruptions when they inevitable occur.

A large scale global supply-chain is a very complex undertaking that involves a high level of risk (Handfield 2008, Handfield et al. 2008, Manuj et al. 2007). Much of the risk is associated with factors that cannot be directly controlled by the organization. These include unavailability of essential resources or components, inclement weather conditions, traffic congestion, custom delays at national borders, breakdown of essential equipment, terrorism and criminal activities, and unforeseen surges in customer demand that can all lead to unexpected disruptions of the end-to-end supply-chain. In recent years with the increase in customer expectations, competition, and political volatility the anticipation and ability to react under time critical conditions to such disruptions has placed an emphasis on effective supply-chain *event* management.

Clearly, such a complex, dynamically changing and time critical undertaking requires sophisticated information management support and can benefit greatly from automated monitoring, planning, tracking, and intelligent decision-assistance services. This paper proposes an enterprise-wide intelligent information management environment based on currently available computer hardware and software technology that is capable of providing the required level of support. It is generally understood that current operational trends and advances in information technology are inevitably leading to the eventual realization of the proposed information management capabilities. However, the opportunity exists to accelerate this progress and reap the significant business benefits that will accrue to the organization that captures the leading share of the supply-chain management software market that has been projected at $5.5 billion in 2011 (AMR 2007).

2. The Inherent Complexity of Logistical Planning and Execution

Logistical planning and execution within a supply-chain can have all of the characteristics that are commonly associated with the family of *complex problems*. These characteristics include: many entities and issues that are related to each other; large volume of data that needs to be categorized and analyzed to extract useful information; the reliability of some of the data may be questionable; incomplete data in some areas requiring time critical decisions to be made with partial information; and, a dynamically changing and largely unpredictable execution environment (Pohl 2008, 49-59).

Swaminathan et al. (1998) have identified two categories of supply-chain elements, namely structural elements and control elements. Structural elements such as vendors, manufacturers, suppliers, distribution centers, and conveyances are concerned with the acquisition, transportation and delivery of goods and services. Control elements such as demand and supply, inventory, routing, and the availability of information govern the flow of processes within the supply-chain. The interrelationships among these two groups of elements are responsible for the complex nature of the supply-chain. The degree to which these complex interactions can be effectively managed is greatly dependent on the accuracy of demand forecasting, the continuous flow of timely and reliable information, the availability of resources such as supplies and conveyances, and a host of external factors such as weather conditions, route closures, accidents, and criminal actions. These external factors are largely unpredictable and have the potential of severely disrupting the supply-chain, despite the most careful attention to planning and execution monitoring.

3. Desirable Capabilities of an Intelligent Supply-Chain Environment

Some importance is attached to the term *environment* in preference to the more conventional nomenclature that would refer to a related set of software components that are intended to interoperate as a *system*. The use of the term environment is intended to convey a level of integration of capabilities that is seamless and transparent to the user. In other words, persons engaged in the logistic planning, monitoring and decision-making processes should not be conscious of the underlying software and inter-process communication infrastructure that is necessary to support the operation of the environment. The objective is for the human users to be immersed in their management activities to the extent that both the automated capabilities operating mostly in background and the capabilities explicitly requested by the user at any particular time operating in foreground are an integral part of the process. Ideally, the human user should perceive the logistic management activities and the environment within which these activities are being performed as being synonymous.

From a general point of view there are at least two overriding requirements for an intelligent computer-based decision-making environment. The first requirement relates to the representation of information within the environment. The software must have some level of *understanding* of the information context that underlies the interactions of the human user with the environment. This is fundamental to any meaningful human-computer interaction that is akin to a partnership. The level to which this *understanding* can be elevated will largely determine the assistance capabilities and essentially the value of the software environment to the human user.

The second requirement is related to the need for collaboration. In a broad sense this includes not only the ability to interact with human stakeholders who play a role in the supply-chain, such as planning and management personnel, vendors, remote distribution centers, shippers, and customs officials, but also non-human sources of information and capabilities. All of these interactions between human participants in the logistic processes, data sources, and software-based problem solving capabilities, must be able to be performed seamlessly without the user having to be concerned about access protocols, data formats, or system interoperability issues.

While these overall requirements would at first sight appear to be utopian compared with the state of computer-based environments that exist today (2010), the technology needed for the creation of such environments has been rapidly emerging during the past decade and is now largely available. However, before addressing the technical software design aspects it is perhaps appropriate to delve more deeply into the functional requirements of an intelligent logistic planning and execution environment.

3.1 Emphasis on partnership

A desirable logistic information management environment is one that assists and extends the capabilities of the human user rather than replaces the human element. Human beings and computers are complementary in many respects. The strengths of human decision makers in the areas of conceptualization, intuition, and creativity are the weaknesses of the computer. Conversely, the strengths of the computer in computation speed, parallelism, accuracy, and the persistent storage of almost unlimited detailed data are human weaknesses. It therefore makes a great deal of sense to view a computer-based supply-chain environment as a partnership between human and computer-based resources and capabilities.

This is not intended to suggest that the ability to automate functional sequences in the computer-based environment should be strictly confined to operations that are performed in response to user actions and requests. Apart from the monitoring of problem solving activities, the detection of conflicts, and the execution of evaluation, search and planning sequences, the computer-based environment should be able to undertake proactive tasks. The latter should include not only anticipation of the likely near-term need for external data sources that need to be acquired by the environment, but also the exploration of alternative solution strategies that the environment considers promising even though the user may be currently pursuing another path.

In this partnership a high level of interaction between the human user and the computer-based environment is a necessary feature. It provides opportunities for the planning and management personnel to guide the environment in those areas of the decision-making process, such as conceptualization and intuition, where the skills of the user are likely to be far superior to those of the computer. Particularly prominent among these areas are conflict resolution and risk assessment. While it would be of considerable assistance to the human users to be alerted to conflicts and for the nature of the conflicts to be clearly identified, there are advantages for the resolution of such conflicts to be undertaken in collaboration with the users.

It follows that the capabilities of the computer-based environment should be designed with the objective of assisting and complementing the user in a teaming role. Such tools are interactive by nature, capable of engaging in collaboration with the user to acquire additional information to help better understand the situation being analyzed. These tools are also able to provide insight into the reasoning processes that they are applying, thereby allowing the human planners and decision-makers to gain confidence in their inferencing capabilities as well as make subtle adjustments in the logic being applied. The author's past experience with multi-agent decision-support applications has shown that tools that are engineered for collaboration with each other and the human user provide opportunities for augmenting their capabilities through user interaction during execution (Pohl et al. 1997). It is therefore suggested that these kinds of tools better assist the human users in dealing with the complexities of the logistic processes involved in the supply-chain. In other words, a collaborative approach affords the necessary visibility and agility to deal with the large number of considerations across a far reaching set of domains that characterizes the supply-chain.

3.2 Collaborative and distributed

Supply-chains, or complex problem environments in general, normally involve many parties that collaborate from widely distributed geographical locations and utilize information resources that are equally dispersed. A computer-based logistic planning and execution environment can take advantage of the distributed participation by itself assuming a distributed architecture. Such an architecture typically consists of several components that can execute on more than one computer. Both the information flow among these components and the computing power required to support the system as a whole can be decentralized. This greatly reduces the potential for communication bottlenecks and increases the computation speed through parallelism.

Another advantage of the distributed approach is the ability to modify some components of the system while the system as a whole continues to operate with the remaining components. Similarly, the malfunction or complete failure of one component does not necessarily jeopardize the entire system. This is not so much a matter of redundancy, although the distributed architecture lends itself to the provision of a high degree of redundancy, but rather a direct result of the physical independence of the components. While the components may be closely integrated from a logical point of view they can operate in their own autonomous physical environment.

3.3 An open architecture

The high degree of uncertainty that pervades complex problem environments, such as logistic planning and execution, extends beyond the decision-making activity of the collaborating planners and decision-makers to the configuration of the computer-based environment itself. The components of a design environment are likely to change over time, through modification, replacement, deletion, and extension. It should be possible to implement these changes in a seamless fashion through common application programming interfaces and shared resources. Service-Oriented Architecture (SOA) concepts align well with this principle by treating the required planning, monitoring, and decision-assistance functionality as a composition of discrete, self-contained software services with a very low degree of coupling between components (Erl 2008).

3.4 Tools rather than solutions

The computer-based logistics environment should offer a set of tools rather than solutions to a predetermined set of problems. The indeterminate nature of the supply-chain does not allow us to predict, with any degree of certainty, either the specific circumstances of a future problem situation or the precise terms of the solution. Under these circumstances it is far more constructive to provide tools that will extend the capabilities of the human decision-maker in a highly interactive problem solving environment.

In this sense a tool is defined more broadly than a sequence of algorithms, heuristics or procedures that are applied largely on the direction of a user. Tools can be self-activating, be capable of at least semi-autonomous behavior, and cooperate with each other and users in employing and providing services.

3.5 Expressive internal representation

The ability of the computer-based environment to convey a sense of having some level of understanding of the meaning of the data and in particular the concepts being processed is the single most important prerequisite for a collaborative information management environment (Assal et al. 2009). An expressive representation of the real world supply-chain entities and concepts that define the problem space forms the basis of the interactions between the users and the information management environment and, also, the degree of intelligence that can be embedded within its components (Figures 1 and 2).

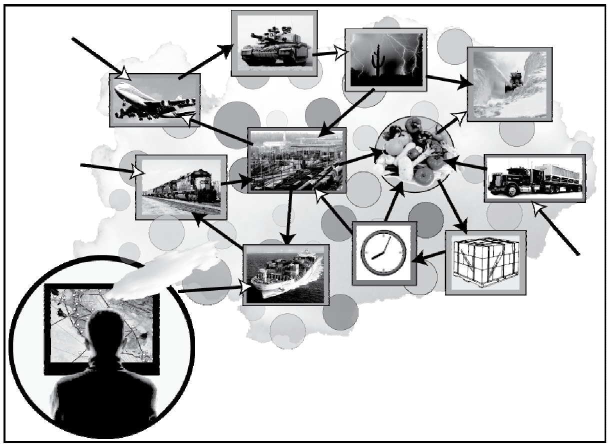


Figure 1: Virtual model of the supply-chain entities and their interrelationships

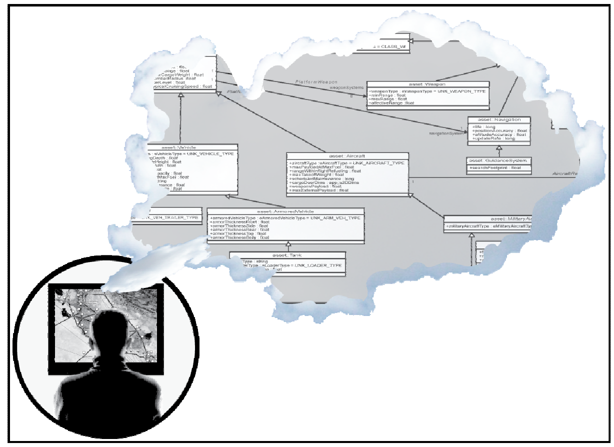


Figure 2: Ontology representation of the supply-chain that is machine processable

To the logistic planning and management personnel the supply-chain consists of real world entities such as requisitions, contracts, goods, services, conveyances, routes, points of embarkation and debarkation, distribution centers, schedules, delivery windows, and costs, as well as related concepts such as efficiency, security, performance, risk, and trust. Each of these notions has properties and relationships that determine their behavior under certain conditions. These semantic descriptors form the basis of collaboration among human problem solvers and are therefore likewise the fundamental subject matter of concern in an enterprise-wide collaborative logistic planning and execution environment.

3.6 Embedded knowledge

The computer-based logistic planning and execution environment should be a knowledge-based environment. In this context knowledge can be described as experience derived from observation and interpretation of past events or phenomena, and the application of methods to past situations. Knowledge-bases capture this experience in the form of rules, case studies, standard practices, and typical descriptions of objects and object systems that can serve as prototypes. Problem solvers typically manipulate these prototypes or patterns through adaptation, refinement, mutation, analogy, and combination, as they apply them to the solution of current problems (Gero et al. 1988, Pohl 2008).

3.7 Decentralized decision-making

While a global supply-chain can be centrally coordinated, the planning and management processes that are required for its efficient operation cannot be centrally controlled. Many of these planning and execution activities will be localized and performed in parallel involving the collaboration of different members of the supply-chain team. In this regard, due to its continuously changing nature, logistic execution is neither a rigidly controlled nor a strongly disciplined activity but rather a process of information seeking, analysis, collaboration, re-planning, and decision-making. For example, intelligent and dynamically interactive software modules that are responsible for pursuing the interests of instances of real world supply-chain objects, such as a particular requisition, a specific conveyance, or a single container, can achieve many of their objectives through employing services and engaging in negotiations that involve only a few nodes of the information management environment. This greatly reduces the propensity for the formation of communication bottlenecks and at the same time increases the amount of parallel activity in the computer-based environment.

The ability to combine in a computer-based information management environment many types of loosely coupled semi-autonomous and autonomous components (i.e., agents), representing a wide range of interests and incorporating different kinds of knowledge and capabilities, provides the environment with a great deal of versatility and potential for problem solving to occur simultaneously at several levels of granularity. This is similar to human problem solving teams in which individual team members work concurrently on different aspects of the problem and communicate in pairs and small groups as they gather information and explore sub-problems.

3.8 Emphasis on conflict identification

The capabilities of a computer-based logistic planning and execution environment should not be bound by the ultimate goal of automatic conflict resolution. Rather, the capabilities of the computing environment should support the identification of the conflict, presenting the human user with as much of the related context as possible. This notion gains in importance as the level of complexity of the logistic planning and management problem increases. The resolution of even mundane conflicts can provide subtle opportunities for advancing towards planning and/or execution objectives. These opportunities are more likely to be recognized by a human user than a computer-based agent. The identification of conflicts is by no means a trivial undertaking. It includes not only the ability to recognize that a conflict actually exists, but also the determination of the kind of conflict and the relationships and related context that describe the conflict and what considerations appear relevant to its resolution. The automatic tracing of these relationships may produce more progress toward a solution than the automatic resolution of the conflict itself.

3.9 Adaptability and agility

Traditionally, software tools categorized as intelligent were engineered for specific scenarios. Consequently, the successful application of these tools depended largely on the degree to which the characteristics of a particular problem component aligned with situations that the tool had been design for. This rigidity has tended to prove quite problematic when these tools were applied to even slight variations of the scenarios that they had been developed or trained for.

In contrast, what the experience of the author has shown is that intelligent tools not only need to support variation, but that these tools should be engineered with such adaptation as a core criterion. Much of this ability to effectively deal with variation is due to the ability of these tools to decompose complex problems into much more manageable components without losing the relationships that tie the components together. To accomplish this, the reasoning capabilities of the tools can be organized as discrete fragments of logic capable of addressing smaller components of the larger problem. If these components are described within an expressive, relationship-rich representation then the connections between the decomposed components are maintained automatically. The effects of addressing each individual component are automatically propagated across the entire expanse of the problem space due to the extensive set of relationships represented within the model that retains their connections and context. The result is a problem solving tool that is agile in its ability to effectively adjust to the variable nature of the dynamically changing supply-chain.

3.10 The human-computer interface

The importance of a high degree of interaction between the human members of the supply-chain team and the various intelligent components of the computer-based information management environment is integral to most of the principles and requirements described above. This interaction is fundamentally facilitated by the information-centric representation core of the environment through which the interacting software components are able to maintain some level of understanding of the current context of the logistic planning and execution activities. However, there are other aspects of the user-interface that must be provided in support of the human-computer interactions. These include two-dimensional and three-dimensional graphical representation capabilities, explanation facilities, and a context-sensitive help system with semantic search support.

At a minimum the graphical capabilities must be powerful enough to include the accurate representation of the current geographical location and state of any transaction moving through the supply-chain, provide near real-time visual access to local conditions, support the animation of alternative movement plans, and allow past movements to be replayed. Technology permitting, the ultimate aim of an intelligent supply-chain environment is to provide a virtual reality user-interface that allows the human users to become fully immersed in the physical and emotional aspects of their logistic planning and execution activities.

Explanation facilities: The author's experience with decision-support systems over the past two decades has lent credence to the supposition that the need for an information management environment to be able to explain how it arrived at certain conclusions increases with the sophistication of the inferencing capabilities embedded in the software environment. At the very least, the intelligent components of the environment should be able to explain their results and methods of analysis. In this regard retrospective reasoning that is capable of providing answers to *what*, *how*, and *why* questions is the most common type of explanation facility available in multi-agent systems (Figure 3).

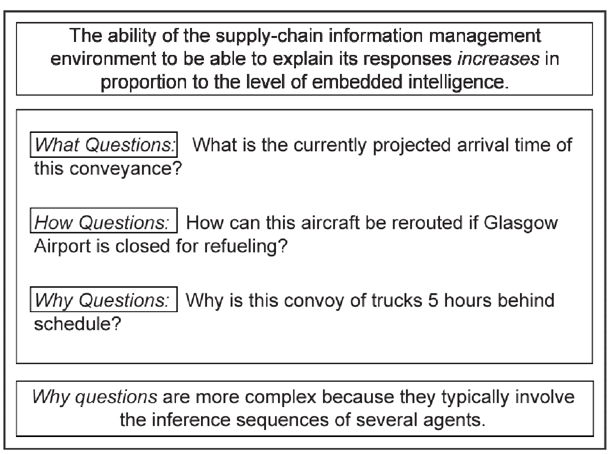


Figure 3: Explanation facilities

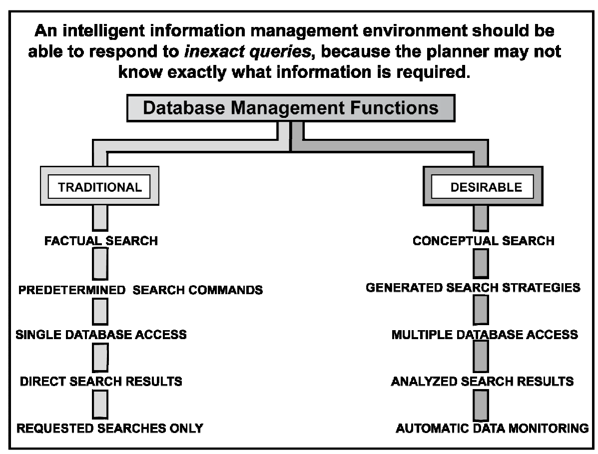


Figure 4: Semantic search facilities

A *what* question requires the explanation or definition of a fact. For example, the user may ask: *What is the currently projected arrival time of this aircraft and what is the certainty factor associated with this projection?* In the past, expert system methodologies based on *format templates* would have allowed the appropriate answer to be collected simply through template values when a match is made with the facts (i.e., aircraft, departure time, wind conditions, etc.) contained in the question (Myers et al. 1993). Today, with the application of ontology-based reasoning capabilities more powerful and direct methods based on the ability of an ontology to represent concepts are available. A *how* question requires an analysis of the sequence of inferences or reasoning that produced the fact. Continuing with the above example, the user may ask: *How can this aircraft be rerouted if Glasgow Airport is closed for refueling?* The answer would require a sequence of inferences by the Fuel, Scheduling and Routing Agents.

Why questions are more complicated. They require reference to the sequence of goals that have driven the progression of inferences (Ellis 1989). For example: *Why is this convoy of trucks 5 hours behind schedule?* In large collaborative systems many agents may have contributed to the inference sequence and will need to participate in the formulation of the answer. This third level of explanation, which requires a summary of justification components, has received considerable attention over the past 30 years. For example: text summary systems such as Frump (Dejong 1982) and Scisor (Jacobs and Rau 1988); fast categorization techniques such as Construe (Hayes and Weinstein 1991); grammatical inference (Fu and Booth 1975) that allows inductive operators to be applied over the sequences of statements produced from successive justifications (Michalski 1983); explanation-based learning (Mitchell et al. 1991); and, case-based reasoning (Shank 1990 and 1991).

Semantic search facilities: While existing computer-based information management systems typically support only factual searches, an intelligent logistical planning and execution environment will provide semantic search capabilities that can deal with inexact queries (Figure 4). Due to the complexity of the problem space the human decision-makers will not always know exactly what information they require. Often they can define only in conceptual terms the kind of information that they are seeking. Also, they would like their query to be automatically broadened with a view to discovering additional information that may be relevant to their current problem solving focus.

The desirability of an information management environment to be able to deal with inexact search requests warrants further discussion. A flexible query capability, such as the human brain, can generate best guesses and a degree of confidence for how well the available information matches the query. For example, let us assume that the user is searching for a *pressure gauge* supply item. Before proceeding with the search the semantic query facility may ask the user to specify further search parameters such as measurement range, required accuracy, or type of fluid to be measured, and allow the user to enter a weighting factor to define the relative importance of each of those parameters that the user has been willing or able to specify. The result of the search would be a list of perhaps 10 *pressure gauge* type supply items ranked in order of probability of satisfying the user's query.

4. The Technical Approach

The desired capabilities of an intelligent logistical planning and execution environment outlined in the previous section call for a distributed system architecture that can be accessed from any physical location, is highly flexile, and totally transparent to the human user. In particular, the user must be shielded from the many protocols and data and content exchange transformations that are required to access capabilities and maintain seamless interoperability among those capabilities. Any member of the supply-chain team, once authenticated during the single sign-on point of entry, should be able to access those capabilities (e.g., intelligent decision-assistance tools and data sources) that are included in the authentication certificate. The focus of the human user should not be on systems, as it still is mostly today, but on the capabilities or *services* that the computer-based environment can provide.

The notion of *services* is well established. Everywhere we see countless examples of tasks being performed by a combination of services, which are able to interoperate in a manner that results in the achievement of a desired objective. Typically, each of these services is not only *recomposable* but also sufficiently *decoupled* from the final objective to be useful for the performance of several somewhat similar tasks that may lead to quite different results. For example, a common knife can be used in the kitchen for preparing vegetables, or for peeling an orange, or for physical combat, or as a makeshift screwdriver. In each case the service provided by the knife is only one of the services that are required to complete the task. Clearly, the ability to design and implement a complex process through the application of many specialized services in a particular sequence has been responsible for most of mankind's achievements in the physical world.

4.1 Service-oriented architecture (SOA)

In the software domain these same concepts have gradually led to the adoption of Service-Oriented Architecture (SOA) principles. While SOA is by no means a new concept in the software industry it was not until Web services became available that the principles of this concept could be readily implemented (Erl 2008, Brown 2008). In the broadest sense SOA is a software framework for computational resources to provide services to customers, such as other services or users. The Organization for the Advancement of Structured Information (OASIS)[1] defines SOA as a *"… paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains"* and *"…provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects with measurable preconditions and expectations"*. This definition underscores the fundamental intent that is embodied in the SOA paradigm, namely *flexibility*. To be as flexible as possible a SOA environment is highly modular, platform independent, compliant with standards, and incorporates mechanisms for identifying, categorizing, provisioning, delivering, and monitoring services.

The principal components of a conceptual SOA implementation scheme (Figure 5) include a Enterprise Service Bus (ESB), one or more portals to external clients with single sign-on facilities, and the enterprise services that facilitate the ability of the user community to perform its operational tasks.

Enterprise Service Bus (ESB): The concept of an Enterprise Service Bus (ESB) greatly facilitates a SOA implementation by providing specifications for the coherent management of services. The ESB provides the communication bridge that facilitates the exchange of messages among services, although the services do not necessarily know anything about each other. According to Erl (2008), ESB specifications typically define the following kinds of message management capabilities:

- *Routing:* The ability to channel a service request to a particular service provider based on some routing criteria (e.g., static or deterministic, content-based, policy-based, rule-based).
- *Protocol Transformation:* The ability to seamlessly transform the sender's message protocol to the receiver's message protocol.

[1] OASIS is an international organization that produces standards. It was formed in 1993 under the name of SGML Open and changed its name to OASIS in 1998 in response to the changing focus from SGML (Standard Generalized Markup Language) to XML (Extensible Markup Language) related standards.

- *Message Transformation:* The ability to convert the structure and format of a message to match the requirements of the receiver.
- *Message Enhancement:* The ability to modify or add to a sender's message to match the content expectations of the receiver.
- *Service Mapping:* The ability to translate a logical business service request into the corresponding physical implementation by providing the location and binding information of the service provider.
- *Message Processing:* The ability to accept a service request and ensure delivery of either the message of a service provider or an error message back to the sender. Requires a queuing capability to prevent the loss of messages.
- *Process Choreography and Orchestration:* The ability to manage multiple services to coordinate a single business service request (i.e., choreograph), including the implementation (i.e., orchestrate). An ESB may utilize a Business Process Execution Language (BPEL) to facilitate the choreographing.
- *Transaction Management:* The ability to manage a service request that involves multiple service providers, so that each service provider can process its portion of the request without regard to the other parts of the request.
- *Access Control and Security:* The ability to provide some level of access control to protect enterprise services from unauthorized messages.

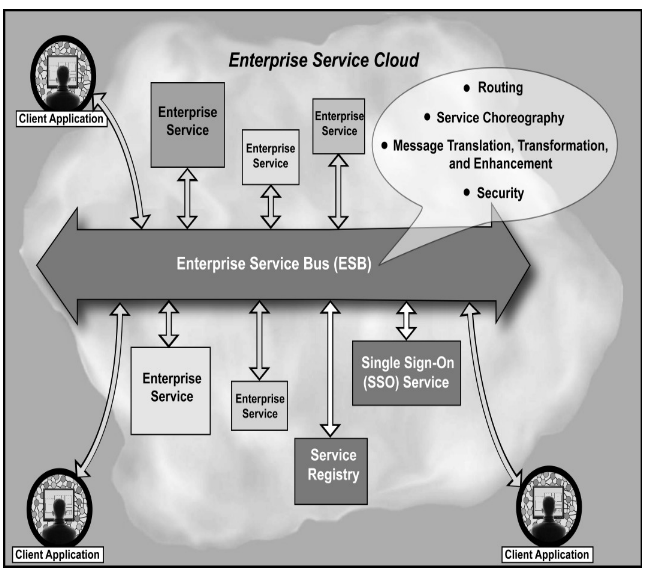


Figure 5: Principal SOA components

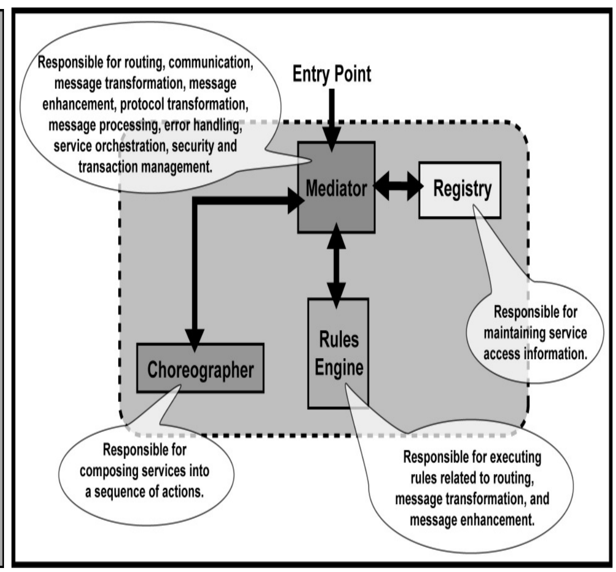


Figure 6: Principal ESB components

There are quite a number of commercial off-the-shelf ESB implementations that satisfy these specifications to varying degrees. A full ESB implementation would include four distinct components (Figure 6): Mediator; Service Registry; Choreographer; and, Rules Engine. The Mediator serves as the entry point for all messages and has by far the largest number of message management responsibilities. It is responsible for routing, communication, message transformation, message enhancement, protocol transformation, message processing, error handling, service orchestration, transaction management, and access control (security).

The Service Registry provides the service mapping information (i.e., the location and binding of each service) to the Mediator. The Choreographer is responsible for the coordination of complex business processes that require the participation of multiple service providers. In some ESB implementations the Choreographer may also serve as an entry point to the ESB. In that case it assumes the additional responsibilities of message processing, transaction management, and access control (security). The Rules Engine provides the logic that is required for the routing, transformation and enhancement of messages. Clearly, the presence of such an engine in combination with an inferencing capability provides a great deal of scope for adding higher levels of intelligence to an ESB implementation.

4.2 Information-centric representation

The methods and procedures that we human beings utilize to make decisions and solve problems rely heavily on our ability to identify, understand and manipulate entities, relationships, and related concepts. Such elements can be readily expressed in software as objects. In this respect, objects are complex symbols that convey meaning by virtue of the explicit and implicit contextual information that they encapsulate within their domain. For example, logistic planners develop shipment plans by reasoning about inventories, conveyances, routes, distribution centers, delivery windows, priority, weather, security, and so on. Each of these objects encapsulates knowledge about its own nature, its relationships with other objects, its behavior within a given environment, and the various constraints and requirements needed to effectively meet its individual performance objectives. This knowledge is contained in the various representational forms of the object as factual characteristics, algorithms, rules, and involvement in past scenarios (whether successful or problematic).

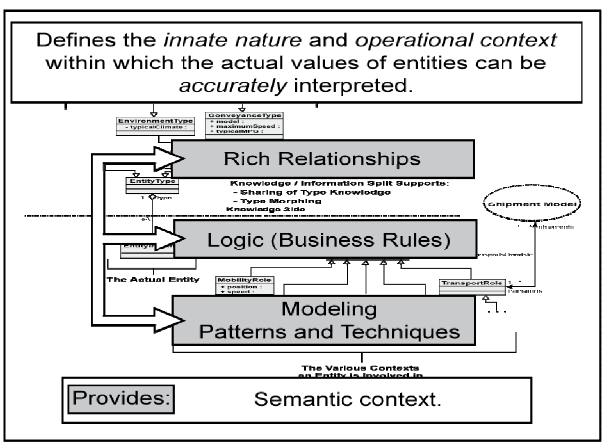


Figure 7: Ontology representation characteristics

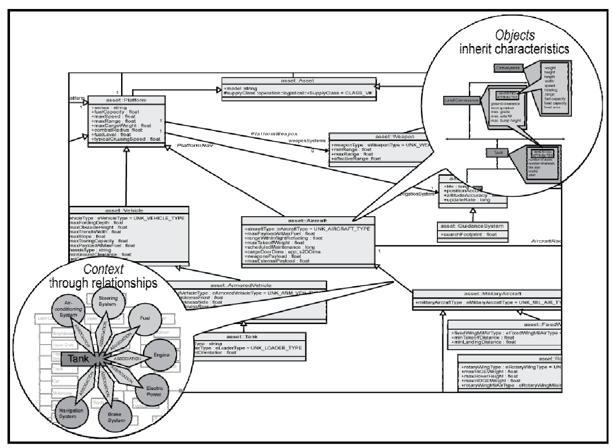


Figure 8: Ontology objects and concepts are machine processable

It is therefore apparent that a critical requirement for effective human-computer interaction in an intelligent supply-chain information management environment is the effective representation of the *context* within which the logistic planning and management activities are taking place. This can be accomplished utilizing an *ontology* (Figure 7). The term ontology is loosely used to describe an information structure that is rich in relationships and provides a virtual representation of some real world environment. As shown in Figure 8, the elements of an ontology include objects and their characteristics, different kinds of relationships among objects, often including the concept of inheritance (Assal et al. 2009). To effectively align ontologies with the dynamics inherent within the real world, it is also important that a set of additional qualities be engineered into such models such as dynamic classification, multiple classification, incremental realization, and the ability to represent something that may not fit into any definition presently available.

Since these elements of an ontology in combination with object-oriented computer languages (e.g., Java, C++) and advanced modeling paradigms (e.g., Web Ontology Language (OWL)) can be automatically interpreted by software, a computer-based information management environment can be endowed with at least a simplistic level of understanding of the real world *context* within which the required planning and execution decisions are being made. This level of understanding is sufficient to provide the necessary context for software agents to automatically interpret data, develop and evaluate plans, detect and explain the causes of conflicts, and generate warnings and alerts.

While an ontology is expressed in object-oriented terms, it is more than an object model. It is designed to describe the entities, concepts, and related semantics of some subject matter domain. Software that incorporates an internal information model, such as an ontology, is often referred to as *information-centric* software. The information model is a virtual representation of the real world domain under consideration and is designed to provide adequate *context* for software agents (typically rule-based) to reason about the current state of the virtual environment.

4.3 Software agents as intelligent tools

On the assumption of an information-centric software architecture that incorporates an ontology-based high level representation of the logistic planning and execution context, the intelligence of the information management environment is largely contributed by the inferencing tools that are available to the human user. Most of these tools will be in the form of invocable services or self-initiating agents. There is a behavioral distinction between services and agents. Services are invoked to perform a discrete activity, returning to their original inactive state after the activity has been completed. Agents on the other hand may be active on a continuous basis, taking the initiative opportunistically whenever they determine that the situation warrants an action. Often these agent actions will invoke services.

There are many types of software agents, ranging from those that emulate symbolic reasoning by processing rules, to highly mathematical pattern matching neural networks, genetic algorithms, and particle swarm optimization techniques. While all of these have capabilities that are applicable to an intelligent supply-chain environment, the symbolic reasoning agents will normally play the most important role and bring the most immediate benefits when a virtual context model (i.e., ontology) has been constructed. Therefore, only symbolic reasoning agents that can interact directly with the ontology-based context model will be discussed in this paper. For these rule-based agents the reasoning process relies heavily on the rich representation of entities and concepts provided by the ontology.

In general terms software agents with symbolic reasoning capabilities may be defined as tools that are situated, autonomous, and flexible (Wooldridge et al. 1999, Wooldridge 1997). They are situated since they receive a continuous flow of operational information generated by the activities within and peripheral to the problem domain environment, and perform acts that may change that environment (e.g., creating alerts, making suggestions, and formulating recommendations). Agent tools are autonomous because they act without the direct intervention of human users, even though they allow the latter to interact with them at any time. In respect to flexibility, agent tools possess the three qualities that define flexibility within the context of the above definition. They are responsive, since they perceive their environment through an internal information model (i.e., ontology) that describes some of the entities and concepts that exist in the real world environment. They are proactive because they can take the initiative in making suggestions or recommendations. They are social, since they can collaborate with other agents or human users, when appropriate, to complete their own problem solving and to help others with their activities.

One important aspect of autonomy in agent applications is the ability of agents to perform tasks whenever such actions may be appropriate. This requires agents to be opportunistic, or continuously looking for an opportunity to execute. In this context opportunity is typically defined by the existence of sufficient information. For example, as the Weather Agent communicates an alert that a particular airport has been closed for the next six hours due to fog, several agents may become involved automatically to undertake analyses (e.g., rerouting alternatives, priority changes, contingency modifications) appropriate to their capability domains.

Service Agents: Agents that are designed to be knowledgeable in a specific domain, and perform planning or assessment tasks in partnership with other agents (i.e., human agents or software agents) are often referred to as Service Agents (Durfee 1988, Durfee and Montgomery 1990, Pohl et al. 1997). The manner in which they participate in the decision-making activities depends on the nature of the situation. Service Agents can be designed to react to changes in the problem state spontaneously through their ability to monitor information changes and respond opportunistically.

In an intelligent supply-chain information management environment Service Agents have knowledge and analysis capabilities in narrow logistic-related domains such as inventory assessment, fuel consumption, scheduling, weather data interpretation, cargo staging, terrain analysis, and maintenance. Typical analysis and inferencing characteristics of Service Agents include:

- Ability to generate alerts based on current state analysis.
- Ability to justify alerts, and analysis results with explanation facilities.
- Ability to broadcast requests for services to other agents.
- Ability to automatically generate queries and access data repositories.
- Ability to temporarily clone themselves to process multiple requests in parallel.
- Ability to undertake proactive explorations opportunistically.

Typical examples of Service Agents for logistical planning and management are described in Appendix A.

Planning Agents: Planning is a reasoning activity that deals with the availability of resources and the actions that need to be taken to complete a given task. Consequently, Planning Agents are designed to reason about the problem state and produce a plan based on the current state of the supply-chain in conjunction with the applicable constraints and objectives. This planning process involves matching the latter with the available resources to produce a course of action that will satisfy the desired objectives. The complexity of the process can be reduced by distributing the basic planning tasks among a set of agents, as follows: identify the constraints and objectives; identify the available resources; note the unavailability of resources; identify the available set of actions or characteristics; and, generate a plan for satisfying the objectives.

Plan or solution generation is the actual planning activity in the above list of tasks. Many planning systems use specialized search algorithms to generate plans according to given criteria (Blum and Furst 1997). Re-planning, which is also commonly referred to as continual planning and includes dynamic planning, involves the re-evaluation of parts of an existing plan or solution because of a change in the information that has been used in the creation of that plan. Some planning systems take advantage of the feedback obtained from the monitoring and execution of plans to add to their knowledge by employing learning techniques, such as explanation-based learning, partial evaluation, experimentation, automatic abstraction, mixed-initiative planning, and case-based reasoning. There are several approaches to learning in agents, including reinforcement learning, classifier systems, and isolated concurrent learning. Learning techniques also enhance the communication ability of agents (Sen et al. 1994, Veloso et al. 1995).

In a supply-chain environment logistic Planning Agents deal with broader issues that relate to the ability of the shipping plan to meet customer requirements within planning and execution constraints such as the availability of inventory, conveyances, routes, and fuel, as well as delivery windows, cost, and acceptable risk. Typical analysis and inferencing characteristics of Planning Agents include:

- Ability to task Service Agents and request information from Mentor Agents.
- Ability to orchestrate evaluations involving several Service Agents.
- Ability to generate broad current state assessments on request or by alert.
- Ability to act on directions from human users and Coordination Agents.

Typical examples of Planning Agents for logistical supply-chain functions such as route planning, cost estimating, risk assessment, efficiency measurement, and opportunity recognition are described in Appendix B.

Mentor Agents: The purpose of a Mentor Agent is to temporarily provide a passive data element with active capabilities such as communication and limited self-determination (Pohl 1996). Mentor Agents are created either by human users or by Coordination Agents on a temporal basis to track a particular supply-chain object such as a requisition, container, pallet, or conveyance that is of special interest. In this way the instance of an object represented in the context model (i.e., ontology) is empowered to play an active role during its life cycle within the supply-chain (Figure 9).

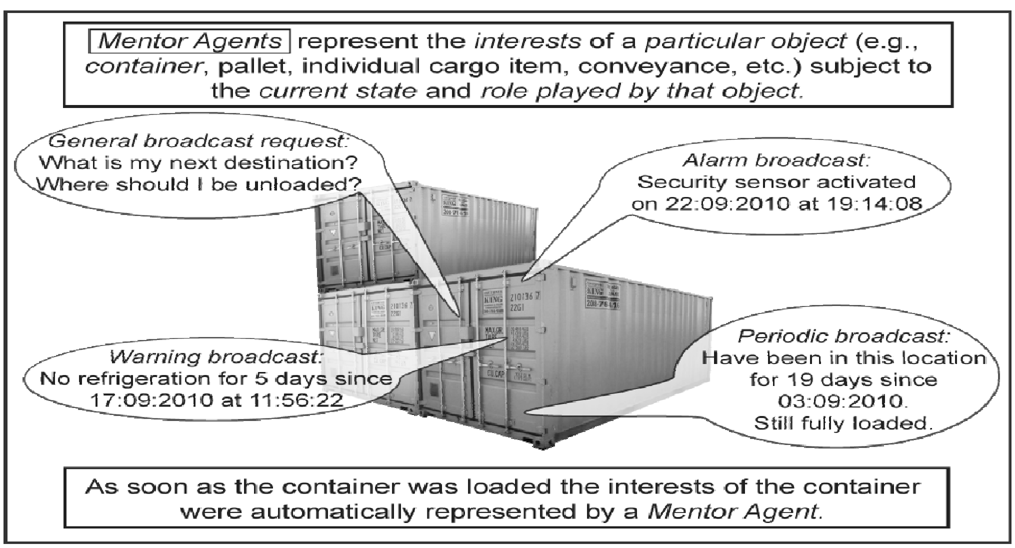


Figure 9: Mentor Agent representing a particular *container* in a shipment

The concept of Mentor Agents brings several potential benefits. First, it increases the granularity of the active participants in the problem solving process. As agents with collaboration capabilities, agentified data elements can pursue their own objectives and perform a significant amount of local problem solving without repeatedly impacting the communication and coordination facilities utilized by the higher level components of the distributed system. Typically, a Mentor Agent is equipped with communication capabilities, process management capabilities, information about its own nature, and objectives. Second, the ability of Mentor Agents to task Service Agents greatly increases the potential for concurrent activities. Multiple Mentor Agents can request the same or different services simultaneously.

Third, groups of Mentor Agents can negotiate among themselves in the case of matters that do not directly affect other higher level components or as a means of developing alternatives for consideration by higher level components. Fourth, by virtue of their communication facilities Mentor Agents are able to maintain their relationships to other aspects of the current state of the supply-chain. In this respect they are the product of *decentralization* rather than *decomposition*. In other words, the concept of Mentor Agents overcomes one of the most serious deficiencies of the rationalistic approach to problem solving; namely, the dilution and loss of relationships that occurs when a complex problem is decomposed into sub-problems. In fact, the relationships are greatly strengthened because they become active communication channels that can be dynamically created and terminated in response to the changing state of the problem situation.

In summary, the capabilities of a Mentor Agent that is created in support of the logistical tasks in an intelligent supply-chain environment would normally include one or more of the following:

- Some understanding of its needs as derived from the context model (i.e., ontology).
- Ability to orient itself geographically and geometrically (i.e., location).
- Ability to communicate and request services from Service Agents.
- Ability to communicate and negotiate with other Mentor Agents.
- Ability to pursue interests proactively leading to alternative recommendations.

Coordination Agents: This group of agents is responsible for facilitating collaboration among human users and software agents. Consequently Coordination Agents require the most intelligence because they need to be able to assess the impact of decisions in individual domains on the particular course of action under consideration (e.g., shipment plan), as well as the overall problem space (e.g., transportation network model).

Particularly in a logistic planning and management environment the most important and demanding role of Coordination Agents is to facilitate collaboration by activating agents and alerting human users of the need for interaction. This requires a relatively high level of understanding of the current state of the supply-chain, which can be only partially fulfilled by currently available artificial intelligence methodologies. Under these circumstances the ability of the human user to assist a Coordination Agent can bridge some of the machine intelligence challenges such as the representation and validation of knowledge that continue to plague the field of machine learning (Forsyth 1989, Thornton 1992, Johnson-Laird 1993). Accordingly Coordination Agents have a greater need than any of the other agent groups to interact with the human agents in the supply-chain information management environment. Through this interaction the human user will be able in several different ways to assist a Coordination Agent by contributing information and knowledge in a collaborative manner. Such human-based assistance may include the setting of priorities, the selection of a particular conflict resolution strategy, the directed invocation of specific agents, or the rejection of certain agent generated recommendations.

Another important function of Coordination Agents is the recognition of conflicts. The emphasis here is on the detection and identification of the causes of a conflict by the agent rather than its resolution. The resolution of a conflict usually involves higher level decisions that have the potential for impacting other areas of the supply-chain. Therefore, apart from very mundane conflicts that could be resolved automatically, the human user should at least be provided with an opportunity to resolve conflicts with wider consequences.

Typical examples of Coordination Agents for logistical supply-chain functions such as collaboration, conflict detection and analysis, threat assessment, and the identification of multi-modal (i.e., air, ship, rail, and truck convoy) transportation alternatives are described in Appendix C.

Governance Agents: While Governance Agents play a particularly important role in military logistic operations, they also have relevance to commercial supply-chains. In both the military and commercial domains these agents are concerned with the measurement of performance, the prevention of security breaches (i.e., theft in the commercial domain), the monitoring of priorities, and the identification of supply-chain trends. Specifically in the military domain, apart from these general functions, Governance Agents are also responsible for ensuring that individual shipment plans are in compliance with Commander's Intent, applicable Rules of Engagement (ROE), and force protection policies.

The role of Governance Agents to identify trends warrants further discussion. The detection of supply-chain trends is almost exclusively considered to be a human role in existing logistical planning and management networks. As a result, due to the large number of transactions that are involved in sizable supply-chains and the dynamically changing nature of the execution phase of operations, many opportunities for proactive planning are overlooked. Particularly under surge conditions in military operations, or when unforeseen events seriously disrupt shipment plans in either the military or commercial domain, the human decision-maker is forced into a reactive role. Unfortunately, it is not uncommon for these disruptions to be either considered one-time incidents that are unlikely to be repeated in the future or for the collection of lessons-learned to be neglected due to human exhaustion. In many cases, the existence of patterns that would, if recognized, lead to operational changes with attendant efficiency improvements and cost savings are not readily discernable without continuous analysis over time.

Governance Agents with access to pattern matching tools such as neural networks can provide powerful trend detection capabilities. Since such tools are able to operate unobtrusively in background on a continuous basis they are able to address the following kinds of questions that are of interest at the executive level of supply-chain management:

- What quantity of any particular commodity or class of supplies (i.e., in the military domain) has been delivered to a specified geographic region or location over a given time period?
- What were the principal choke points where shipments have been delayed during a given time period?
- What has been the average time that certain kinds of shipments have taken over a given time period?

- What have been the relative densities of air, ship, rail, and truck movements over a given time period?
- What have been the principal causes of inter-modal delays or substitutions over a given time period?

Typical examples of Governance Agents for both military and commercial supply-chain functions are described in Appendix D.

4.4 The system environment

Conceptually, as shown in Figure 10, the logistical context provided by the multi-layered ontology allows the various groups of agents to monitor and act on the data that flows on a continues basis through the supply-chain. The primary functions of the Planning Agents are focused on the generation of alternative route plans when needed and the determination of closure when a shipment has been delivered. However, the evaluation of these plans may also involve cost estimating, risk assessment, and the identification of opportunities for improving efficiency and reducing costs. The Coordination Agents are responsible for facilitating collaboration, exploring the availability and suitability of conveyances and arranging multi-modal movement plans. For example, if the Opportunity Agent identifies a partially loaded conveyance then the Collaboration Agent will immediately explore the possibility of backfilling this conveyance with another shipment to the same destination. This exploration may involve one or more Service Agents such as the Scheduling Agent and the Staging Agent to determine whether the existing schedule and staging plan of a candidate shipment can be modified to take advantage of the opportunity.

What is significant is that all of these actions can be undertaken automatically and concurrently for hundreds of shipment plans on a continuous near real-time basis. When events that have the potential for disrupting the supply-chain occur the human users have the necessary tools and actionable information available to take immediate and effective action. At the same time the Governance Agents are systematically analyzing past shipments with a view to identifying patterns and trends within the supply-chain. The purpose of this after-action analysis is to provide a basis for contingency planning and proactive actions that are aimed at reducing risk with attendant increases in efficiency and cost reductions in future transactions.

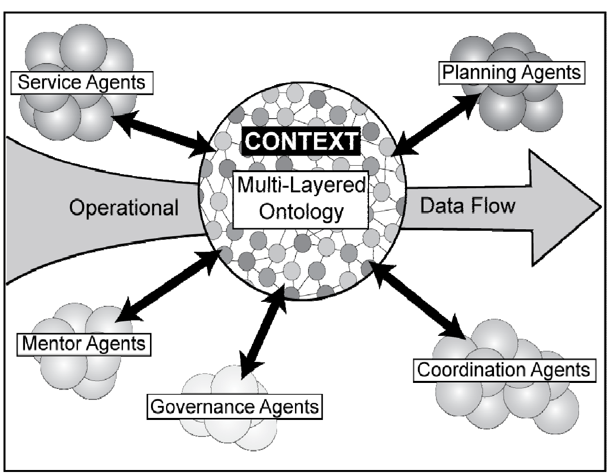


Figure 10: Context-based intelligent tools

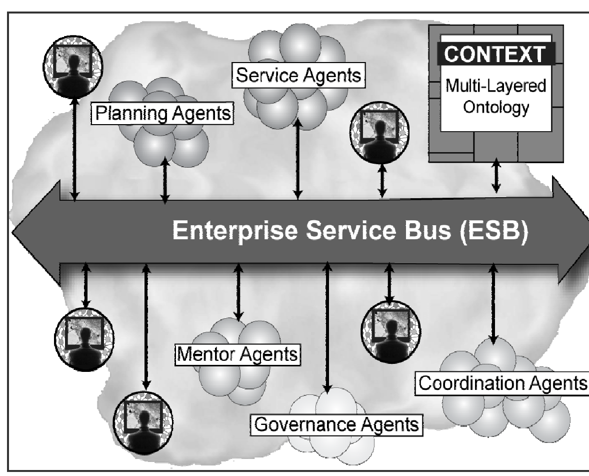


Figure 11: SOA-based system architecture

The system implementation framework is based on SOA principles (Figure 11), with interaction among the various loosely coupled applications and services managed transparently to the human users by an ESB. While many of the agents operate concurrently in an opportunistic mode, the workflow of logistical operations is essentially sequential in character. In a SOA-based system environment the orchestration of such sequences is normally performed by a Business Process Management (BPM) facility.

> ***Business Process Management (BPM):*** BPM is a method for actively defining, executing, monitoring, analyzing, and subsequently refining manual or automated business processes. In other words, a business process is essentially a sequence of related, structured activities (i.e., a workflow) that is intended to achieve an objective or larger task. Such workflows can include interactions between human users, software applications or services, or a combination of both. In a SOA-based information management environment this orchestration is most commonly performed by the Choreographer component of the ESB (Figure 6). Based on SOA principles, a sound BPM design will decompose a complex business process into smaller, more manageable elements that comply with common standards and reuse existing solutions.
>
> The principal components of the BPM capability within the supply-chain information management environment include a Business Process Execution Language (BPEL) engine, a graphical modeling service, business user and system administration interfaces, internal and external system interactions, and persistence. The BPEL is normally XML-based[2] and event driven. The BPEL engine is responsible for systematically issuing the sequence of service and/or user requests that are specified within the specific BPEL script, elegantly handling any related events or issues as they may occur.

While BPM and SOA concepts are closely connected, they are certainly not synonymous. Rather, they are complementary. Described more precisely, a SOA-based system environment provides the enabling infrastructure for BPM by separating the functional execution of the business process from its technical implementation. Conversely, BPM offers even the most well architected inventory of SOA functionality (i.e., services) specific objectives. The business process models identified as part of the BPM approach prove to effectively align the software capabilities produced to the actual needs of the users. Too often enterprises suffer from a distinct mismatch between available software functionality and actual user needs.

In addition to those components discussed above, an effective logistics decision-support environment includes a number of other principal components including:

- A web-based application portal that provides the human user with an integrated, highly-interactive canvas (i.e., view) across what may otherwise be a disparate collection of services, information sources (e.g., GIS, databases, etc.), intelligent agents, and external systems. Further, benefiting from the strong presence of BPM principles and functionality complementing the overarching SOA-based enterprise, this rich user interface is purposefully organized around the very business processes that are relevant to the specific type of user (e.g., logistics planner tasked with filling supply orders in an informed and efficient manner, tactical commander (in the military domain) wishing to verify the status of expected supplies, etc.). In other words, orienting the various flavors of the user-interface around relevant business processes provides specific users with a

[2] The Extensible Markup Language (XML) is a general purpose specification that allows the content of a document to be defined separately from the formatting of the document.

graphical, highly-interactive (essentially customized) user-interface that is designed and engineered in terms of the very workflows, terminology, and practices that comprise that user's tasks, objectives, and practices (i.e., business processes). The result is a convenient, highly efficient control panel that fosters an effective partnership between the human users and the software capabilities designed to assist them.

- An ontology service that builds, maintains, and exposes its evolving context to agents and other services that are context-dependent. Such informational services can support synchronization of interested clients with changes occurring within the context they manage via asynchronous service requests that can live for extended periods of time. The result is a means by which clients can subscribe to, and consequentially be notified of, particular events and conditions of interest as they may occur.
- An inference service that may comprise a number of agent communities. An agent community is a collection of related agents in a given domain such as the Planning Agents, Mentor Agents, Service Agents, Coordination Agents, and Governance Agents described in Section 4.3. Each agent utilizes applicable ontology services and other types of services to examine and analyze the current state of a particular transaction sequence or larger supply-chain context.

4.5 The user environment

From the human user's point of view the intelligent logistic planning and execution environment described in this paper is highly interactive and proactive. Not only are the users able to conduct searches for data where the search keys are known (i.e., directed searches) but they are also able to conduct semantic searches when the queries can be only vaguely formulated. In those cases agents with data mapping capabilities will search through one or more databases and return to the user approximately matching query results with computed certainty factors.

At the same time the user is automatically alerted to both opportunities for taking advantage of events that could lead to greater efficiency or lower shipment costs and events that either are already or could potentially disrupt the supply-chain. Since agents are continuously monitoring most aspects of the shipment traffic within the transportation network many of the opportunities for effective intervention that are likely to be overlooked in current data-centric management systems will be brought to the attention of the human user through agent warnings and alerts. In this respect the intelligent logistic planning and execution environment is both reactive and proactive. For example, if any particular shipment is running behind schedule then this will be noted and recorded in a *warning* report by an agent. If a shipment is halted by an obstacle in its path such as traffic congestion, a flooded road or a fogged-in airport then this will be noted by an agent and the user will receive an *alert*. However, agents are also continuously analyzing past shipments to identify patterns and trends, so that these can be related to current or expected near term conditions within the transportation network. This type of analysis may involve multiple Governance, Coordination, Planning, and Service Agents, with the objective of identifying potential supply-chain events and disruptions proactively. For example, the repeated late delivery of shipments in a particular region may suggest the need for considering an alternative inter-modal movement plan.

Data access: Much of the management time in a supply-chain environment is spent on determining the location and status of shipments that have failed to arrive at their destinations within the time windows expected by the requesters. The logistical planning and execution environment must therefore provide in-transit visibility capabilities. These capabilities come as a by-product of the ontology-based context model that treats most of the graphical elements that are displayed in the user-interface on geographical maps as objects with characteristics and relationships. This allows the human user to lodge queries about a particular shipment or group of shipments and pursue such queries to reasonable depth, with the objective of receiving answers to the following kinds of questions: Where is this shipment right now? Where was the shipment last reported to have been seen or identified? What has been the event-by-event or node-to-node history of the shipment from the time it was first requested? What conveyances are available to expedite the movement of this shipment from where it is now to its intended destination?

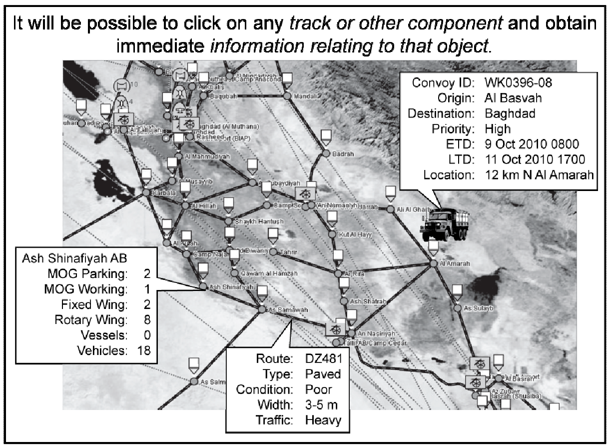


Figure 12: Displayed symbols are objects

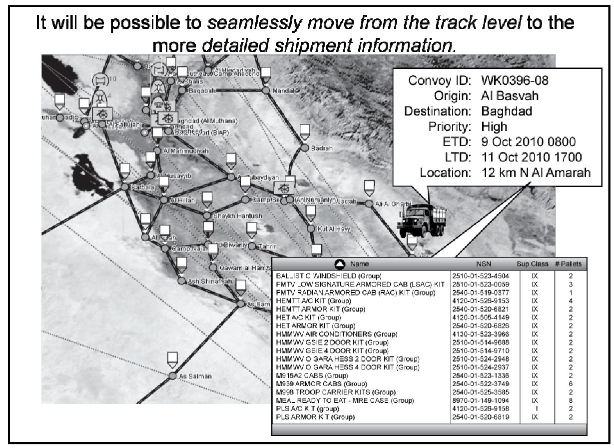


Figure 13: Information on request

As shown in Figure 12, to obtain information about any of the symbols displayed on the map the users simply clicks on the particular symbol (e.g., conveyance, supply center icon, city, or route) with their mouse. A second click allows a user to drill down to more detailed information. For example, in Figure 13 the user is able to seamlessly move from the summary information relating to the current location, destination, priority, and expected delivery window of a truck convoy, to the details of the individual cargo items.

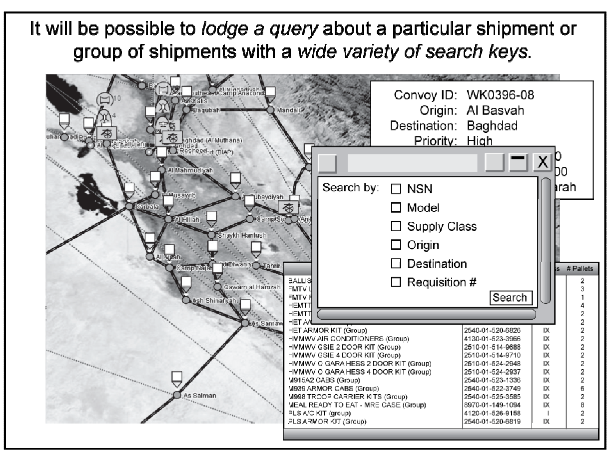


Figure 14: Ability to search on multiple keys

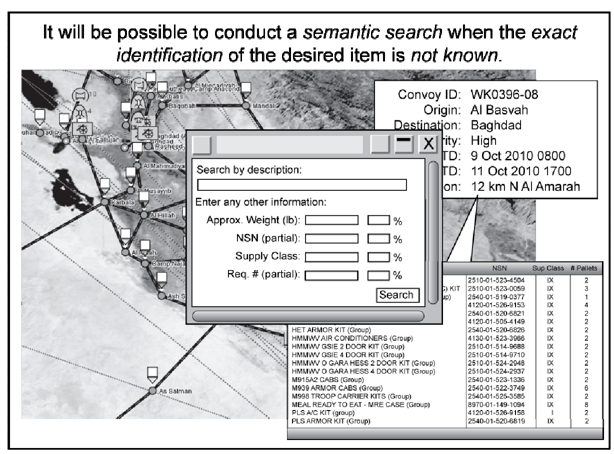


Figure 15: Search with partial information

Not only are the users able to search on multiple keys such as supply item number, supply type, requisition #, and so on (Figure 14), but they can also conduct semantic searches. As shown in Figure 15, the user may describe the kind of supply item in fairly vague terms when the exact identification of the item is not known. For example, the user may know only the kind of supply item and its approximate weight. Based on this partial description the Inventory Agent will search for supply items that are reasonably close to this description and present these to the user with a corresponding certainty factor.

Similarly, either by clicking on a displayed graphic symbol or by employing direct or semantic search capabilities the user is able to obtain a summary of the inventory of all of the supply centers in a particular geographical region (Figure 16) or drill down to the current inventory of a particular supply center (Figure 17). The same data is of course also available to agents based on automatically generated direct queries for use in the generation and evaluation of alternative plans, the assessment of risk, the determination of costs, and any other logistic management task that any particular agent is designed to perform.

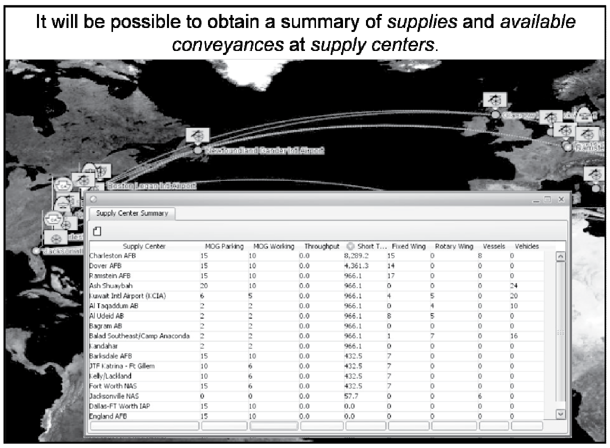


Figure 16: Supply centers inventory summary

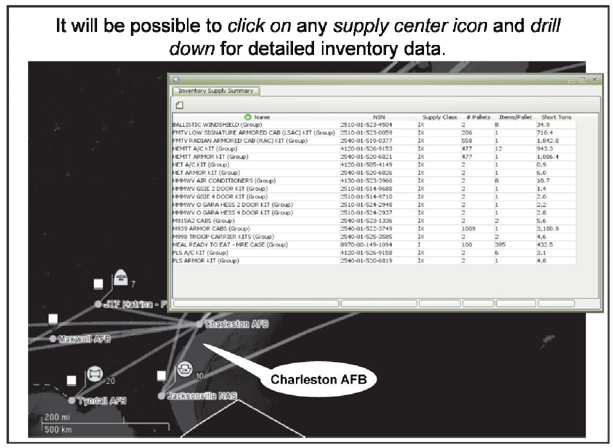


Figure 17: Supply center inventory details

To maintain in-transit visibility the user is able to click on any displayed track and obtain information relating to that track, such as:

- What does the track represent in terms of shipment ID, shipment type, and current transport mode (i.e., conveyance)?
- What is the last reported location of the track and what is the date and time of that location report?
- What is the next destination (i.e., node) of the track and what is/was the planned arrival date and time?

Similarly, the user is able to move seamlessly from the track level data to the more detailed shipment data, to answer questions such as:

- What is the priority of this shipment?
- What is the content of the shipment in terms of quantity and type of supplies?
- What was the origin of the shipment and the start date/time of the movement?
- What is the final destination of the shipment and who requested it? When was it requested? What was the requested delivery date/time? What was the delivery date/time according to the original movement plan? When is it most likely to be actually delivered?
- What is the node-to-node movement plan for this shipment? Where is it now in respect to this plan and what is the remaining unexecuted portion of the plan?

Impact of external factors: Both the formulation and execution of shipment plans is impacted by external factors such as weather conditions, customs requirements at border crossings or points of debarkation in foreign countries, location of criminal or enemy activities, availability of indigenous transportation, terrain, traffic conditions, and so on. In this respect an intelligent toolset is able to accept several on-line data feeds and combine the imported data with sufficient context to allow agents to automatically reason about the implications of the external factors. Candidate data feeds include:

- Weather forecasts on a regional and local level. For example, Figure 18 shows the translation of weather data by the Weather Agent into a weather report that provides actionable information to a human user and is machine processable for inferencing purposes by other software agents.
- Indigenous transportation systems (e.g., major roads, railways, ferries, commercial airline routes) in regions and local areas that may be available for shipments.
- Supplies, conveyances, fuel, and related transportation resources available at transportation hubs and distribution centers (Figure 19).
- Location of criminal and/or enemy activities.
- Infrastructure objects such as power plants, warehouses, railway stations, ferry stations, airports, ocean ports, fuel depots, and so on.

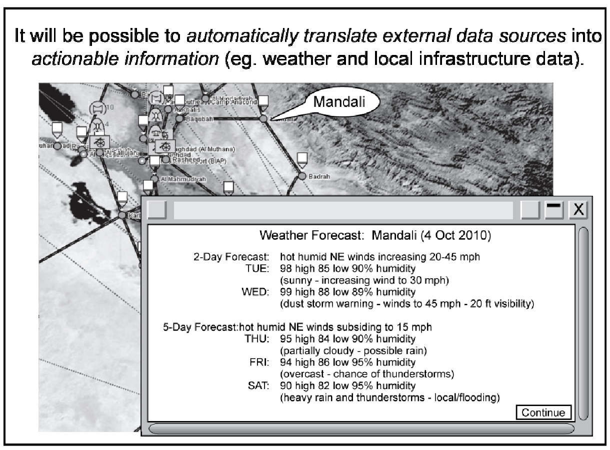


Figure 18: Weather report as actionable information for human and agent consumption

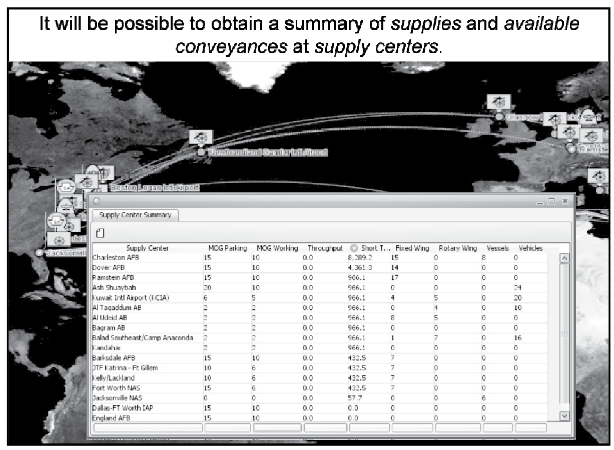


Figure 19: Distribution center inventory and available conveyances

Pattern recognition: As the scale of the adaptive toolset progressively encompasses a more significant portion of the supply-chain enterprise the intelligent agents will have access to an increasingly larger set of historical data. This will allow the implementation of agents with sophisticated analysis and case-based reasoning capabilities. Such agents, operating in a collaborative manner, will be able to analyze past shipments on a continuous basis and be able to respond to the following kinds of questions:

- What quantity of any particular kind of supplies has been delivered over a given time period, what shortages are likely to arise, and when?
- What were the principal choke points where shipments have been delayed during a given time period? Where are choke points likely to occur in the future based on current market forecasts?

- Where have shipments been intercepted by criminal or enemy action over a given time period and what are the risk factors that should be applied to future shipments?
- What has been the average time that certain kinds of shipments have taken over a given time period and how do these times relate to planned future movements?
- What have been the relative densities of air, surface and rail movements over a given time period and how do these densities relate to supply-chain performance?

4.6 Agent collaboration and decision-assistance

Historically, computer-based data-processing systems have been designed to be activated and controlled by human users. In this respect they may be characterized as *passive* decision-assistance environments that with few exceptions respond only when tasked by a human user. For example, the user enters the requirements for certain goods to be shipped between two geographical locations and a movement plan is either interactively formulated or automatically generated if more sophisticated tools are available. In other words, the user directs the system to assist in some predefined manner and the system generates the appropriate response or result to the best of its capabilities. If the users do not request the system to undertake any tasks then the system will be essentially idle.

A context-based (i.e., information-centric) software system with inferencing capabilities provided by agents is in contrast an *active* decision-assistance environment in which data cleansing, monitoring, analysis, planning and re-planning, pattern identification, and exploratory processing will occur on an on-going basis. In fact, under certain circumstances the system environment may be intensely active while the human users are largely inactive. The activities of the system environment are activated at least as much by the data that flows through the system on a continuous basis (Figure 10) as by the interactions of the human users with the system environment. This is largely made possible by the virtual model (i.e., multi-layered ontology) of the real world supply-chain context that allows the agents to autonomously and concurrently interpret and analyze the data flow in the appropriate context.

As an example of a typical sequence of logistical execution management events we will assume the following typical military scenario. A high priority requisition for add-on-armor (AOA) supplies comes to the Defense Logistics Agency (DLA) from Al Udeid in the Iraq theater and enters the Joint Deployment and Distribution Enterprise (JDDE) environment of the United States Transportation Command (USTRANSCOM).

As shown in Figure 20, the Priority Agent sends a *warning* to the Collaboration Agent suggesting that collaboration will be necessary due to the high priority of the request. The Collaboration Agent starts monitoring the requisition and immediately requests the Opportunity Agent to determine whether the requested AOA items are already in theater or in-transit to the theater. The Opportunity Agent invokes the Inventory Agent, which in turn seeks the assistance of the Distribution Center Agent and the Closure Agent to determine whether the requested AOA items are or will be available in the theater by the required date. Concurrently the Inventory Agent with the assistance of the Distribution Center Agent determines whether the required AOA items are in stock at a CONUS[3] supply center.

[3] Continental United States (CONUS) includes the 48 states on the continent of North America that are south of Canada plus the District of Columbia, but excludes the states of Alaska and Hawaii, and all off-shore United States (US) territories and possessions.

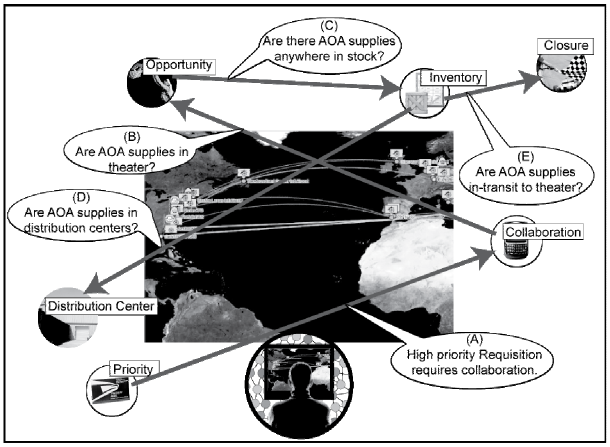


Figure 20: Are the requested AOA supplies available in inventory?

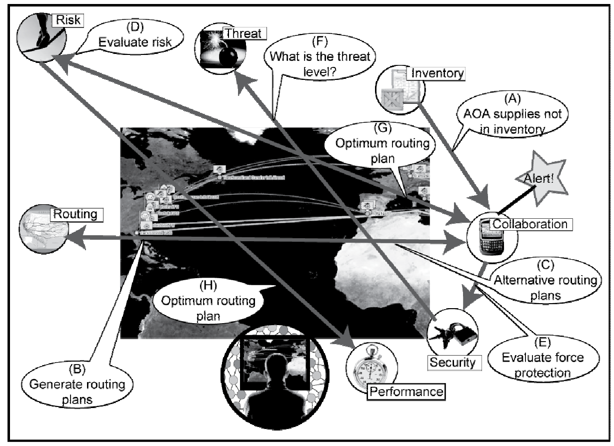


Figure 21: The supplies are not available and must be outsourced.

In Figure 21, the Collaboration Agent determines on the basis of the report received from the Inventory Agent that the requested supplies are not in CONUS inventory and decides to outsource to commercial supplier(s). Concurrently the Routing Agent is invoked by the Collaboration Agent to generate alternative multi-modal route plans from Charleston to Al Udeid and sends the plans to the Security Agent to address force protection issues and the Risk Agent to assess the risk of non-performance. The Security Agent requests the assistance of the Threat Agent in its analysis, while the Risk Agent shares the results of its analysis with both the Collaboration Agent and the Performance Agent.

In the meantime, the Collaboration Agent requests the creation of a Mentor Agent for this requisition (Figure 22). The Mentor Agent keeps track of all matters pertaining to this requisition such as: name of vendor; delivery window of AOA supplies to Charleston for shipping to Al Udeid.

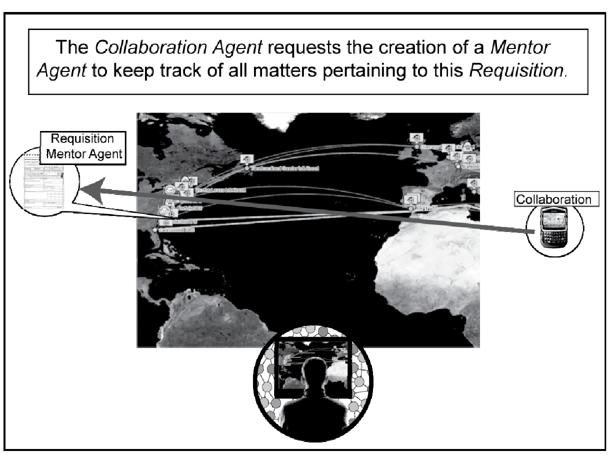


Figure 22: Mentor Agent is assigned to the high priority requisition

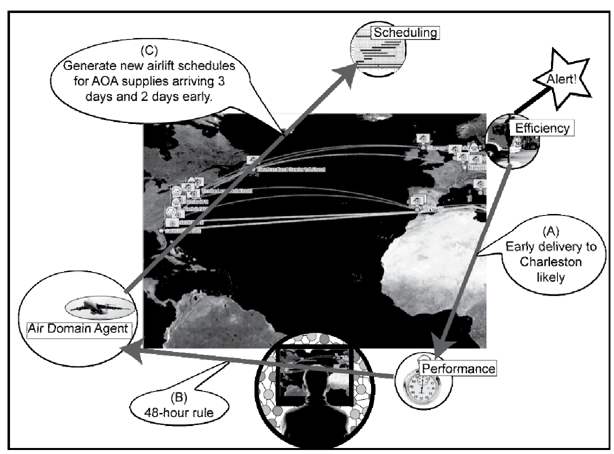


Figure 23: Potential Thanksgiving holiday build-up at Charleston POE[4]

In Figure 23, the Efficiency Agent notices that the delivery window for Charleston is 22-24 November, which is just before the Thanksgiving holiday. It therefore sends an *alert* to the

[4] Point of Embarkation (POE).

Performance Agent indicating that early delivery to Charleston by commercial shippers to accommodate personal holiday plans is likely to cause a build-up of shipments at Charleston. The Performance Agent being aware of the 48-hour rule that does not allow cargo to be staged at Charleston for longer than 48 hours prior to shipping, sends a *warning* to the Air Domain Agent. The latter proactively requests alternative schedules from the Scheduling Agent based on most (i.e., 80%) of the AOA cargo arriving at Charleston 3 days and 2 days before Thanksgiving.

Continuing in Figure 24, the Air Domain Agent determines on the basis of the schedules generated by the Scheduling Agent that the airlift assets available at Charleston will be inadequate and sends an *alert* to the Collaboration Agent. In Figure 25, the Collaboration Agent requests shipping cost estimates based on early and late purchase orders from the Cost Agent and then sends an *alert* to the human user to the likely requirement of commercial airlift with the cost estimates in hand. In the meantime, the Risk Agent assesses the risks involved in early and late purchase decisions. The human user decides on the basis of the high priority of the shipment, and the reports received from the Risk Agent and the Cost Agent that an early decision to order commercial airlift is warranted and approves the necessary purchase orders.

It should be noted that the decision to place an immediate order for commercial airlift, thereby taking advantage of advance notice cost savings, has been made in minutes instead of days (or not until the need for commercial airlift has been noticed at the last moment by human users).

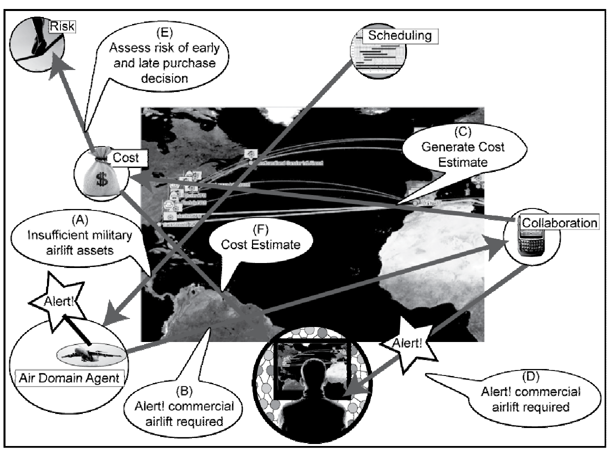


Figure 24: Early decision on commercial airlift required

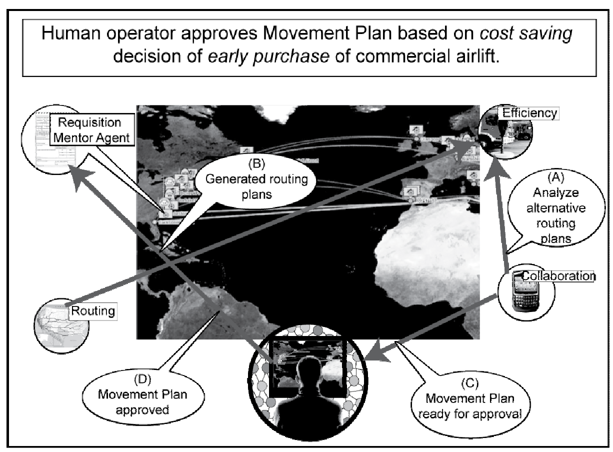


Figure 25: Decision to order commercial airlift made in minutes instead of days

Concurrently, in Figure 25, the Efficiency Agent is invoked by the Collaboration Agent to analyze the alternative plans generated by the Routing Agent, with the objective of determining the optimum movement plan. The human user approves the movement plan based on recommendations received from the Collaboration Agent. Again, recognition of the potential build-up of cargo at Charleston and the need for commercial airlift resources, as well as the decision to place an early purchase order and generate a new shipment plan all occurred in minutes.

By this time the Mentor Agent holds the following information about the requisition:

- Requisition ID, date received, ID of requesting party, and priority.
- Destination and requested delivery window.

- Name, NSN[5], number of pallets, number of items per pallet, supply class, and weight of each requested AOA supply item.
- ID of commercial vendor for each outsourced AOA supply item.
- Force protection rating.
- Risk of non-performance rating.
- Estimated costs of supplies.

4.7 Execution scenario examples

During subsequent execution stages the Mentor Agent continues to look after the interests of the high priority requisition and the Collaboration Agent invokes any other agents to assist in the analysis and resolution of unforeseen events until the Closure Agent determines that the transaction has been completed.

The following two execution scenarios are not only typical of the military domain, but could equally well occur in a commercial supply-chain. The shipment plan approved by the human user in Figure 25 includes Glasgow Airport in Scotland as a refueling venue. However, in its continuous monitoring and interpretation of global weather reports the Weather Agent discovers that Glasgow Airport is fog-bound. It immediately sends an *alert* to the Collaboration Agent indicating that Glasgow Airport is fog-bound (Figure 26). The Collaboration Agent requests the Routing Agent to generate an alternative movement plans with the assistance of the Air Domain Agent. Concurrently the Collaboration Agent requests the Efficiency Agent to analyze the alternative plans generated by the Routing Agent to determine an optimum alternative shipment plan. The Efficiency Agent receives input from the Cost Agent and the Security Agent during the analysis. Finally, the human user reviews the recommendations received from the Collaboration Agent and approves the new Movement Plan.

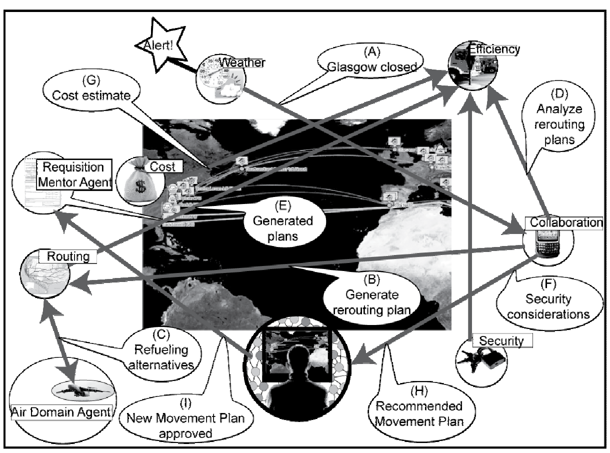


Figure 26: Glasgow Airport is fogged in and flights will need to be rerouted

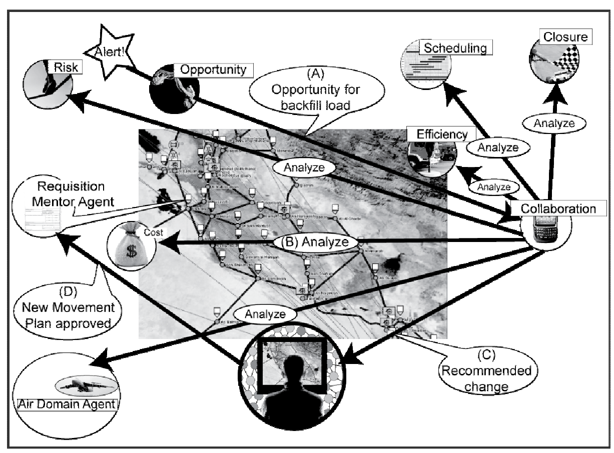


Figure 27: A backfill opportunity is not overlooked by the agents

The second example scenario deals with an opportunity to increase efficiency and reduce costs that would likely be overlooked by human users. Late arrival of another unrelated shipment to the same destination provides an opportunity for part of this shipment to backfill partial aircraft loads from Charleston to Al Udeid. In Figure 27, the Opportunity Agent sends an *alert* to the

[5] National Stock Number (NSN).

Collaboration Agent indicating an opportunity for saving transportation costs and time. It has discovered that due to late arrival at Charleston of some cargo from another requisition there may be a backfill opportunity. The Collaboration Agent immediately undertakes an analysis with the assistance of the Air Domain Agent, the Scheduling Agent, the Cost Agent, the Risk Agent, the Efficiency Agent, and the Closure Agent. The human user reviews the recommendations received from the Collaboration Agent and approves the modified shipment plan. Consequently, the Collaboration Agent informs the Convoy Domain Agent that part of the shipment for this requisition will be airlifted from the POE directly to Al Udeid and will therefore not require road transportation.

5. Conclusions

The inordinately high complexity of logistical planning and management tasks in a global supply-chain is due to the multitude of issues involved (e.g., routing, cost, risk, efficiency, security, priority, weather conditions, priority, inventory, conveyance type, terrain, and so on), the relationships among those issues, the frequency of changes during execution that threaten to disrupt the supply-chain, the time critical nature of shipments, and the diversity of the players involved[6]. Management of this compound complexity requires the assistance of an intelligent software system environment (Figure 28).

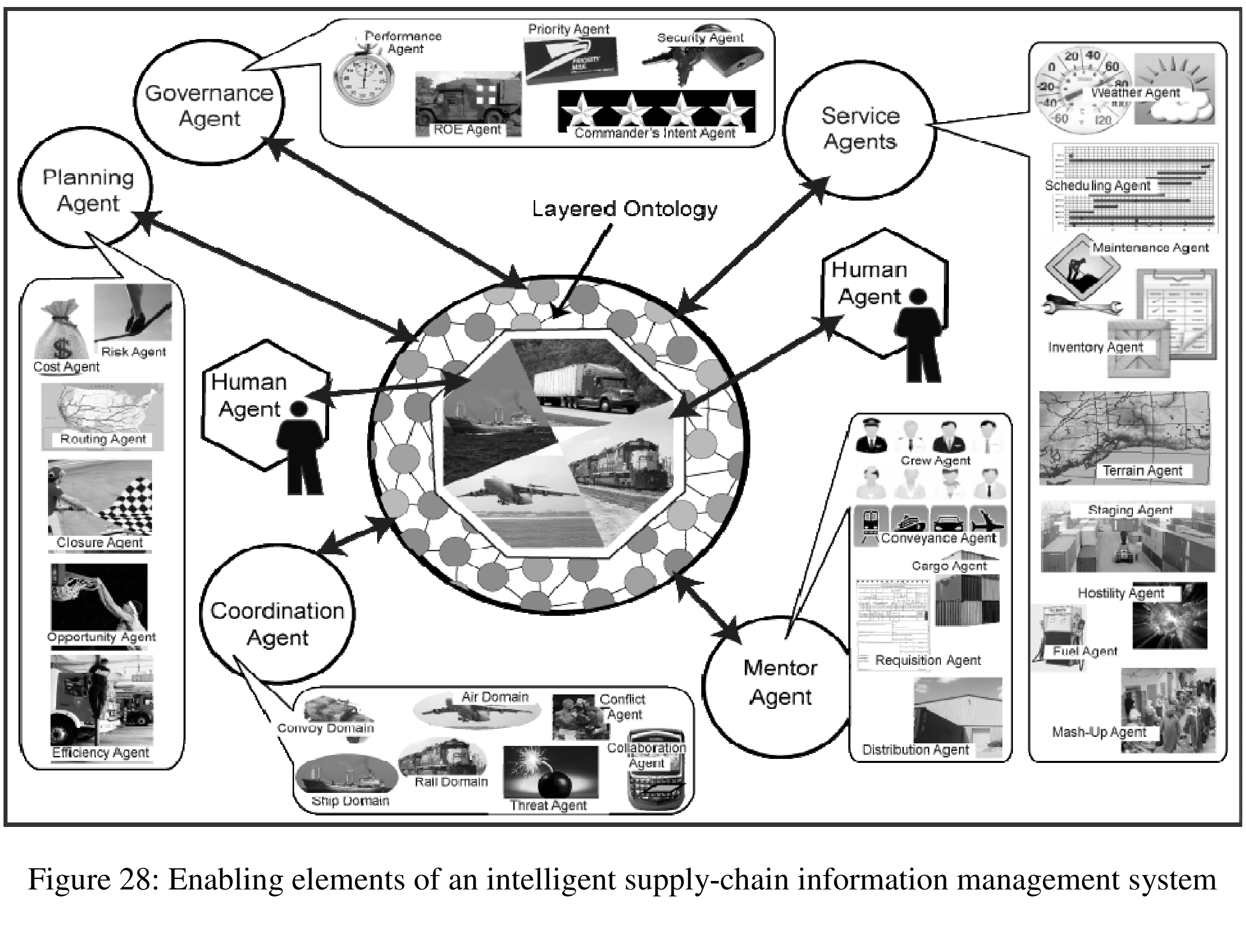


Figure 28: Enabling elements of an intelligent supply-chain information management system

[6] The players or stakeholders in a supply-chain typically have very different objectives. For example, the planner is interested in high efficiency at minimum cost, the shipper is concerned about conveyance reliability and route conditions, while the customers expect to receive their orders on time and in an undamaged state.

As discussed in this paper there are two principal requirements for such an environment. The first requirement is a rich contextual representation of supply-chain information. This can be provided by a virtual model of the real world context within which the logistical management tasks such as the preparation of a multi-modal shipment plan, maintaining in-transit visibility, reacting to unforeseen events, preparing proactively for potential future events, and so on, can be expeditiously performed. The importance of this virtual model of real world context must not be underestimated. As a core requirement it provides the basis of most of the assistance capabilities of the intelligent information management environment described in this paper. Without access to the context provided by the multi-layered ontology the different groups of software agents defined in Section 4.3 and the Appendices could not function as intelligent tools in the manner described in Sections 4.6 and 4.7.

The second requirement is collaboration among the human users, as well as interaction between the human users and the intelligent software tools (e.g., agents) and, as discussed in Sections 4.6 and 4.7, between the intelligent tools themselves. Effective collaboration between any two parties assumes at least some commonality of purpose. Between human parties this commonality is based not only on the understanding that each party has of its own objectives, but also on some level of understanding of the objectives and needs of the other party. In addition, there is a distinctly opportunistic aspect to collaboration. While the general requirement for collaboration and even the protocol that must be adhered to during the process of collaboration may be prescribed, the events that will initiate collaboration are largely unpredictable.

Similar principles of collaboration apply to the interactions between the human users and the software agents, and among the software agents themselves. The human users will expect the agents that they interact with to have some semblance of common understanding of the content of the interaction. This applies equally whether the user is requesting an explanation of an agent-generated result or queries the agent for specific information. Similarly, agents need some understanding of context to determine under what circumstances they should send an alert to human users or other agents. Clearly, the prerequisite for this semblance of understanding is the existence of a virtual model of real world context at the software level.

The current state of technology in software development provides the means for implementing a distributed, collaborative, intelligent, information management environment. Service-oriented architecture (SOA) concepts provide the framework and the guiding principles for developing distributed, service-based systems. The field of ontology representation is sufficiently mature to support the expressive modeling of domain knowledge as the enabling foundation for intelligent software tools or agents. Such agents can continuously monitor the supply-chain, participate in decision-making processes within specific domains, gather and present relevant information to the human user, and opportunistically communicate with human users and other agents.

References

AEDOT (1992); 'AEDOT Prototype 1.1: An Implementation of the ICADS Model'; Technical Report CADRU-07-92, Collaborative Agent Design Research Center, Cal Poly, San Luis Obispo, CA 93407.

AMR Research (2007): http://www.gartner.com/technology/supply-chain/amr-research.jsp

Assal H., K. Pohl and J. Pohl (2009); 'The Representation of Context in Computer Software'; Pre-Conference Proceedings, Focus Symposium on Knowledge Management Systems, InterSymp-2009, Baden-Baden, Germany, 4 August.

Barber K., A. Goel, D. Han, J. Kim, D. Lam, T. Liu, M. MacMahon, C. Martin and R. McKay (2003); 'Infrastructure for Design, Deployment and *Experimentation* of Distributed Agent-based Systems: The Requirements'; The Technologies, and an Example, Autonomous Agents and Multi-Agent Systems. Volume 7, No. 1-2 (pp 49-69).

Blum A. and M. Furst (1997); 'Fast Planning Through Planning Graph Analysis'; Artificial Intelligence, 90 (pp.281-300).

Chartered Institute of Logistics and Transport in the UK (CILT) (1998); 'Members Directory'; CILT, Corby, www.ciltuk.org.uk.

Brown P. (2008); 'Implementing SOA: Total Architecture in Practice'; Addison-Wesley.

Dejong G. (1982); 'An Overview of the Frump System'; Lehnert and Ringle (eds.) Strategies for Natural Language Processing, Lawrence Erlbaum, Hillsdale, New Jersey (pp.149-176).

Diaz C., W. Waiters, J. Pickard, J. Naylor, S. Gollery, P. McGraw, M. Huffman, J. Fanshier, M. Parrott, S. O'Driscoll-Packer, Boone Pendergrast and Evan Sylvester (2006); 'ICODES: Technical and Operational Description'; Technical Report CDM-20-06, CDM Technologies Inc., San Luis Obispo, California, November.

Durfee E. (1988); 'Coordination of Distributed Problem Solvers'; Kluwer Academic, Boston, Massachusetts.

Durfee E. and T. Montgomery (1990); 'A Hierarchical Protocol for Coordination of Multiagent Behavior'; Proc. 8th National Conference on Artificial Intelligence, Boston, Massachusetts (pp.86-93).

Ellis C. (1989); 'Explanation in Intelligent Systems'; in Ellis (ed.) Expert Knowledge and Explanation: The Knowledge-Language Interface, Horwood, England.

Erl T. (2008); 'SOA: Principles of Service Design'; Prentice Hall.

Forsyth R. (1989); 'Machine Learning: Principles and Techniques'; Chapman and Hall, Computing Series, London, England.

Fu K. and T. Booth (1975); 'Grammatical Inference: Introduction and Survey'; IEEE Transactions on Systems, Man, and Cybernetics. SMC-5: (pp.95-111, 409-423).

Gero J., M. Maher and W. Zhang (1988); 'Chunking Structural Design Knowledge as Prototypes'; Working Paper, The Architectural Computing Unit, Department of Architectural and Design Science, University of Sydney, Sydney, Australia.

Handfield R. (2008); 'Consumers of Supply Chain Risk Data'; in Handfield R. and K. McCormack (eds.) 'Supply Chain Risk Management: Minimizing Disruptions in Global Sourcing', Auerbach Publications, New York.

Handfield R., J. Blackhurst, D. Elkins and C. Craighead (2008); 'A Framework for Reducing the Impact of Disruptions to the Supply Chain: Observations from Multiple Executives'; in Handfield R. and K. McCormack (eds.) 'Supply Chain Risk Management: Minimizing Disruptions in Global Sourcing', Auerbach Publications, New York (pp. 29-50).

Hayes P. and S. Weinstein (1991); 'Construe-TIS: A System for Content-Based Indexing of a Database of News Stories'; Rappaport and Smith (eds.) Innovative Applications of Artificial Intelligence 2, AAAI Press, Menlo Park, California (pp.47-64).

ICADS (1991); 'ICADS Working Model Version 2 and Future Directions'; Technical Report CADRU-05-91, Collaborative Agent Design Research Center, Cal Poly, San Luis Obispo, CA 93407.

Jacobs P. and L. Rau (1988); 'A Friendly Merger of Conceptual Analysis and Linguistic Processing in a Text Processing System'; Proceedings of the Fourth IEEE AI Applications Conference, IEEE Computer Society Press, Los Alamitos, California (pp.351-356).

Johnson-Laird P. (1993); 'Human and machine Thinking'; Erlbaum, Hillsdale, New Jersey.

Manuj I, J. Dittmann and B. Gaudenzi (2007); 'Risk Management'; in Mentzer J., M Myers and T. Stank (eds.) 'Handbook of Global Supply Management', Sage Publications, London, UK (pp. 319-336).

Michalski R. (1983); 'A Theory and Methodology of Inductive Learning'; Artificial Intelligence, Vol.20 (pp.111-161).

Mitchell T., J. Allen, P. Chalasani, J. Cheng, O. Etzioni, M. Ringuette and J. Schlimmer (1991); 'Theo: A Framework for Self-Improving Systems'; VanLehn (ed.) Architectures for Intelligence, Twenty-Second Carnegie Mellon Symposium on Cognition, Lawrence Erlbaum, Hillsdale, New Jersey (pp.323-355).

Myers L., J. Pohl, J. Cotton, J. Snyder, K. Pohl, S. Chien, S. Aly and T. Rodriguez (1993); 'Object Representation and the ICADS-Kernel Design'; Technical Report CADRU-08-93, Collaborative Agent Design Research Center, Cal Poly, San Luis Obispo, CA 93407, January.

Nibecker J., H. Larsen, X. Pan, C. Warren, R. Chambers, D. Taylor, B. Weber, J. Delos Reyes, C. Maas, and M. Porczak (2007); 'TRANSWAY: Technical and Operational Description'; Technical Report, CDM-21-07, CDM Technologies, Inc., San Luis Obispo, CA 93401, October.

Pan J. and J. Tenenbaum (1991); 'Toward an Intelligent Agent Framework for Enterprise Integration'; Proc. Ninth National Conference on Artificial Intelligence, vol.1, San Diego, California, July 14-19 (pp.206-212).

Pohl J. (2008); 'Cognitive Elements of Human Decision-Making'; in Jain L. and G. Wren (eds.); Intelligent Decision Making: An AI-Based Approach; Springer Verlag, New York (41-76).

Pohl J., A. Chapman, K. Pohl, J. Primrose and A. Wozniak (1997); 'Decision-Support Systems: Notions, Prototypes, and In-Use Applications'; Technical Report, CADRU-11-97, Collaborative Agent Design Research Center, Cal Poly, San Luis Obispo, CA 93407, January.

Pohl J. and L. Myers (1994); 'A Distributed Cooperative Model for Architectural Design'; in Carrara G. and Y. Kalay (eds.) Knowledge-Based Computer-Aided Architectural Design, Elsevier, Amsterdam, The Netherlands.

Pohl K. (2008); 'A Translation Engine in Support of Context-Level Interoperability'; Special Issue on Ontology Driven Interoperability for Agile Applications Using Information Systems: Requirements and Applications for Agent Mediated Decision Support, Intelligent Decision Technologies, 2 (1), January.

Pohl K. (2007); 'Enhancing the Face of Service-Oriented Capabilities'; Pre-Conference Proceedings, Focus Symposium on Representation of Context in Software, InterSymp-2007, Baden-Baden, Germany, 31 July.

Pohl K. (1996); 'KOALA: An Object-Agent Design System'; in Pohl J. (ed.) Proc. Focus Symposium on Advances in Cooperative Environmental Design Systems, InterSymp-96, Baden-Baden, Germany, Aug.14-18 (pp.81-92), Collaborative Agent Design Research Center, Cal Poly, San Luis Obispo, CA 93407.

Swaminathan J., S. Smith and N. Sadeh (1998); 'Modeling Supply Chain Dynamics: A Multiagent Approach'; Decision Sciences, 29(3), Summer (pp. 607-632).

Schank R. and R. Osgood (1990); 'Content Theory of Memory Indexing'; Technical Report 2, The Institute for the Learning Sciences, Northwestern University.

Schank R. (1991); 'Case-Based Teaching: Four Experiences in Educational Sofware Design';

Sen S., M. Sekaran, and J. Hale (1994); 'Learning to Coordinate Without Sharing Information'; in National Conference on Artificial Intelligence (pp.426-431).

Simon H. (1996); 'The Sciences of the Artificial'; 3rd ed., MIT Press, Cambridge, Massachusetts (pp. 111-3).

Taylor D. and H. Assal (2009) 'Using BPM to Support Systems Interoperability'; The International C2 Journal, Vol. 3, No. 1.

Thornton C. (1992); 'Techniques in Computational Learning'; Chapman and Hall, Computing Series, London, England.

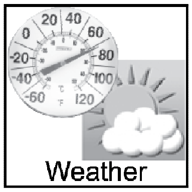
Veloso M., J. Carbonell, A. Perez, D. Borrajo, E. Fink and J. Blythe (1995); 'Integrating Planning and Learning: The PRODIGY Architecture'; Journal of Theoretical and Experimental Artificial Intelligence, 7(1).

Waters D. (ed.) (2007); 'Global Logistics: New Directions in Supply Chain Management'; 5th edition, Kogan Page, Philadelphia.

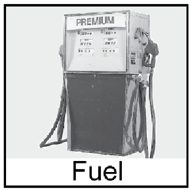
Wooldridge M., N. Jennings and D. Kinny (1999); 'A Methodology for Agent-Oriented Analysis and Design'; Proceedings Third International Conference on Autonomous Agents (Agents-99), Seattle, Washington.

Wooldridge M. (1997); 'Agent-Based Software Engineering'; IEEE Transactions on Software Engineering, 144(1), (pp.26-37), February.

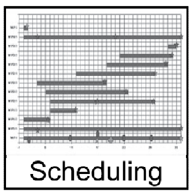
Appendix A: Typical Service Agents



1. The ***Weather Agent*** has the ability to interpret and translate raw weather data into a weather report that has meaning to both the human user and the computer (i.e., is machine processable)



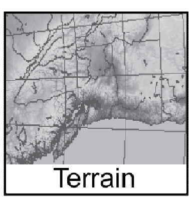
2. The ***Fuel Agent*** has the ability to monitor the fuel consumption of conveyances during movements (through sensor data), project fuel requirements, locate refueling nodes, and assess the fuel capacity at nodes.



3. The ***Scheduling Agent*** is capable of integrating inter-modal movements, taking into consideration the delivery dates of cargo at the POE, the availability of surface and air transportation, and delivery windows.

4. The ***Staging Agent*** is capable of planning the staging of cargo in marshalling yards taking into account projected cargo arrival dates/times, order of loading based on conveyance type and destination, access routes, and space constraints.

5. The ***Inventory Agent*** is responsible for monitoring the inventory of distribution centers and therefore has the ability to access data sources and formulate queries on an on-going basis, as well as in response to requests for inventory information from other agents and human users.



6. The ***Terrain Agent*** has the ability to assess the state of surface routes in terms of traffic congestion, impediments (e.g., flooded areas, land slides, snow, ice), road conditions and grades, and their potential impact on traveling time.

7. The ***Hostility Agent*** is responsible for monitoring potentially hostile activities that could impact shipments moving on surface routes, including theft, narcotics, piracy, terrorism, and enemy actions (in the military domain).

8. The ***Maintenance Agent*** is responsible for monitoring the maintenance requirements of conveyances and therefore has the ability to both access appropriate data sources and to monitor the operational state of conveyances and high value loading facilities through the interpretation of sensor data.

Mash-Up

9. The ***Mash-Up Agent*** is capable of generating a web application that combines data and/or existing Internet functionality (e.g., Google Earth) from multiple sources into an action report, such as an on-the-spot view of a local event (e.g., disaster area survey, cargo loading at an ocean port).

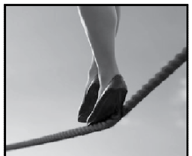
Appendix B: Typical Planning Agents

Routing

1. The ***Routing Agent*** has the ability to plan and re-plan multi-modal routing alternatives under time critical conditions, taking into consideration route conditions, efficiency, cost, and risk.

Cost

2. The ***Cost Agent*** has the ability to rapidly estimate the cost of alternative movement plans during both strategic planning and execution.



Risk

3. The ***Risk Agent*** has the ability to assess the risks associated with alternative movement plans based on past performance, current threat conditions, weather forecasts, and political factors.

Efficiency

4. The ***Efficiency Agent*** is responsible for monitoring the compliance of shipments with planned schedules in a reactive mode, and for identifying potential shipment delays or supply-chain disruptions in a proactive mode.

Opportunity

5. The ***Opportunity Agent*** is capable of identifying potential partial conveyance loading based on the ability to algorithmically assess the number of a particular type of conveyance required for a shipment or based on the analysis of cancelled or modified transactions.

Closure

6. The ***Closure Agent*** is responsible for determining when a shipment has reached its destination and been delivered, thereby signifying that the movement portion of the transaction has been completed.

Load-Planning

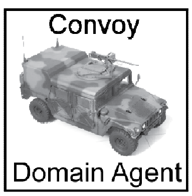
7. The ***Load-Planning Agent*** is capable of generating load-plans for ships, aircraft, railcars, and trucks either automatically or in a user-assistance mode, taking into account cargo size and weight, access path, type of conveyance, stability constraints, hazardous material requirements, and cargo spacing tolerances.

Appendix C: Typical Coordination Agents

1. The ***Conflict Agent*** is capable of detecting conflict conditions that may arise among agents and within the transportation network, and identify the likely causes.

2. The ***Collaboration Agent*** is responsible for facilitating collaboration by activating agents and alerting the human users to the need for interaction.

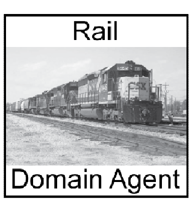
3. The ***Threat Agent*** has the ability to assess threat conditions based on intelligence sources and relate these to individual shipments, as well as the global transportation network by communicating high threat conditions to the *Security Agent*.



4. The ***Convoy Domain Agent*** is capable of matching the need for trucks based on load and shipment schedule with the availability of truck convoy transportation from origin to destination (i.e., between the required POE and POD[7]).

5. The ***Ship Domain Agent*** is capable of matching the need for surface ship transportation, based on cargo list and shipment schedule, with the availability of cargo space on-board vessels moving between the required POE and POD.

6. The ***Air Domain Agent*** is capable of matching the need for airlift, based on cargo list and air transportation schedule, with the availability of aircraft and aircrews at the designated POE.



7. The ***Rail Domain Agent*** is capable of matching the need for railcars, based on cargo list and shipment schedule, with the availability of railcars between the nearest railhead and the designated destination (i.e., between the required POE and POD).

[7] Point of Debarkation (POD).

Appendix D: Typical Governance Agents


Commander's Intent

1. In the military domain the ***Commander's Intent Agent*** has the ability to abstract the principal features of a movement plan to a conceptual level for the generation of Commander's Critical Information Requirements (CCIR). In the commercial domain the equivalent objectives are to identify instances when a movement is in serious danger of not meeting stated company objectives.


Performance

2. The ***Performance Agent*** has the ability to apply metrics and assess not only the quality of an individual movement plan but also its impact on the overall operational efficiency.

Priority

3. The ***Priority Agent*** is responsible for monitoring the assigned priority of shipments and drawing high priority shipments to the attention of the *Collaboration Agent*, as well as alerting other agents and/or the human user if high priority shipments are subject to delay.


Security

4. The ***Security Agent*** receives threat condition assessments from the *Threat Agent* and uses these as a basis for determining the appropriate security or force protection (military domain) precautions that should be applied to shipments.


ROE

5. The ***ROE Agent*** (military domain) in collaboration with the designated human user is responsible for maintaining a repository of supply-chain relevant rules of engagement, monitoring the compliance of shipments to these rules, and alerting the designated human user to any ROE violations.