Regular Paper

IDEON™: An Extensible Ontology for Designing, Integrating, and Managing Collaborative Distributed Enterprises

Azad M. Madni,* Weiwen Lin, and Carla C. Madni

Intelligent Systems Technology, Inc., 2800 28th Street, Suite 306, Santa Monica, CA 90405

Received July 22, 2000; Accepted September 28, 2000

ABSTRACT

An organization's ability to achieve and sustain competitive advantage in the face of continual change depends, to a large extent, on the adaptability, interoperability, and maintainability of its enterprise management approach and supporting software implementation. In this regard, the major challenges facing organizations are: (a) achieving seamless integration of enterprise design, management and control processes and supporting applications; (b) ensuring interoperability between new and legacy business applications; and (c) adapting business strategies and ongoing operations to changes in the external and internal environments. The latter requires integrated planning and execution of enterprise processes. This paper presents IDEON™, a unified, extensible enterprise ontology that has been designed in response to these needs. Two specific applications of IDEON™ are presented along with the specific extensions for each application. © 2001 John Wiley & Sons, Inc. Syst Eng 4, 35–48, 2001

1. INTRODUCTION

Today, there are several information technology (IT) trends and market dynamics that are driving enterprise management and control requirements. Among these is the fact that IT is expanding from a back-office resource (e.g., manufacturing resource planning/enterprise resource planning) to a front-office (e.g., sales force automation, customer relationship management) enabler of competitive advantage. Change, once viewed as a short period of relative instability, is now a continuous process. At the same time, monolithic enterprises that totally own all of the products, services, and channels required to serve a customer are rapidly being replaced by strategic partnerships, virtual enterprises, and integrated value chains. The need to operate in a rapidly changing

*Author to whom all correspondence should be addressed.

Systems Engineering, Vol. 4, No. 1, 2001

business and technical environment is driving the need for technology infrastructures and application architectures that are increasingly more flexible, interoperable, extensible, and maintainable. In the light of these dynamics, an organization's ability to achieve and sustain a competitive advantage depends, to a large extent, on the flexibility, interoperability, and maintainability of its enterprise management and control capability. It is in this area that enterprise ontologies have the highest payoff.

Interest in ontologies has surged as researchers and system developers have become increasingly more interested in reusing and sharing knowledge across systems (i.e., software applications). However, today there is a major impediment to knowledge sharing given that the different systems use different concepts and terms for describing domains [Schlenoff et al., 2000]. As a result, it is difficult to take knowledge out of one system (e.g., a planner or process modeler) and use it in another system (e.g., a workflow system).

The foregoing problem can be alleviated by developing ontologies that could be used as a foundation for multiple systems (e.g., planning system, workflow system). Such ontologies would ensure that the different applications shared a common terminology, which is the essence of knowledge sharing and reuse. It is not surprising, therefore, that the two major thrusts in ontology research are the development of (a) reusable ontologies that span multiple systems and (b) tools that enable the merging of ontologies and/or translating one to another. The former assures correct interpretation of the knowledge and facilitates the information creation and retrieval process. The latter assures knowledge sharing in a heterogeneous systems environment.

But there is more to ontologies than successful knowledge sharing and reuse. Ontologies fundamentally change the way in which systems are constructed. Today knowledge bases are created from scratch without focusing on sharing or reuse. As a result, it generally takes much too long to create and verify the completeness and traceability of the knowledge contents. However, with an ontology focus this can all change. One can envision a tomorrow in which ontologies in persistent, reusable form are used to effectively and compactly organize the knowledge content of databases. The benefits of this strategy are dramatic reduction in knowledge base development time as well as the creation of robust and reliable knowledge bases from pre-existing, verified components.

It is this thinking that drove the creation of IDEON, a unified enterprise ontology specifically designed to support integrated planning and execution of enterprise processes. While other enterprise ontologies have focused on enterprise analysis and re-engineering, IDEON is focused on integrating and managing planning and execution within collaborative distributed enterprises—a key requirement for supply chain management, command and control, collaborative systems engineering, emergency management, and crisis action planning and execution. IDEON has been successfully employed on two key applications which are presented in this paper. The first application is crisis action planning and execution. The key challenge on this application was to create an ontology that supported plan execution with replanning capabilities in the face of change events. The second application is Integrated Product-Process Development (IPPD). The key challenge on this application was to create an ontology that would support (a) the design and tailoring of systems engineering processes from an IPPD and standards perspective and (b) the execution and management of collaborative systems engineering processes in a distributed environment. What is common to these applications is the integration of model building activities and model-based execution capabilities. IDEON was created to support this class of problems with appropriate domain-specific extensions when needed.

The outline of this paper is as follows. Section 2 defines the term "ontology" and briefly presents the goals and current themes in ontology research. Section 3 reviews the major organizational ontologies in terms of their goals, scope, organizational components, and status. Section 4 presents the origins, objectives, and payoffs of IDEON. Section 5 presents the key design concepts underlying the development of IDEON. Section 6 presents and discusses the four complementary IDEON perspectives—enterprise context view, the organizational view, the process view, and the resource/product view. Section 7 presents two specific applications of IDEON: crisis action planning and execution; and IPPD-enabled complex systems engineering. For each application, specific IDEON extensions are discussed. Section 8 summarizes the IDEON value proposition for designing, integrating, and managing collaborative distributed enterprises.

2. ONTOLOGY RESEARCH

The Webster dictionary [Woolf, 1981] defines an *ontology* as "a particular theory about the nature of being or the kinds of existents." Intelligent systems from the field of computer science are responsible for formally representing these existents, whereas conceptualization provides the basis for formally representing a body of knowledge. *Conceptualization* consists of a set of concepts (e.g., objects), their inter-relationships, as well as other relevant entities about which knowledge is being expressed. Every knowledge model employs some form of conceptualization, implicit or explicit. An explicit specification (or representation) of this conceptualization is called an ontology [Gruber, 1993]. Formally, an ontology consists of a set of terms, their definitions, and axioms that inter-relate them [Gruninger and Fox, 1995]. The set of terms is normally organized as a taxonomy.

From a domain perspective, an ontology is a formal description of the entities within a given domain: the properties they possess, the relationships they participate in, the constraints they are subject to, and the patterns of behavior they exhibit. It provides a common terminology that helps to capture key distinctions among concepts in different domains, which aid in the translation process. An ontology consists of a core and extensions to express information involving concepts that are not part of the core. The main idea behind a core and extensions is not to clutter the core with every conceivable concept that might be useful in specific applications. Rather, the idea is to provide modular extensions to the core thereby allowing a user to tailor the ontology to suit his/her application needs.

Current goals of ontology research are to: (1) make ontologies sharable through common formalisms and tools; (2) develop ontology content (also called ontology design); (3) compare and translate ontologies; and (4) compose new ontologies from existing ones. Recent work in ontology design spans ontologies that represent general world knowledge, domain-specific ontologies, and knowledge representation systems that embody ontological frameworks. The ontology engineering community agrees on the key benefits of integrating ontologies—sharing and reuse of knowledge. Making ontologies interoperable is highly useful but also challenging. The key to achieving a reasonably high degree of interoperability is to: (a) compare and contrast existing ontologies to see how they represent the basic types and aspects of knowledge; and (b) use the results of this analysis to identify and resolve the critical issues in integrating the different ontologies.

3. ORGANIZATIONAL ONTOLOGIES

A number of ontologies for modeling organizations exist in the ontology literature. While the design goals of these various ontologies are quite different, each of them provides valuable insights for identifying and validating key enterprise concepts. The relevant organizational ontologies include: TOVE [Fox, Chionglo, and Fadel, 1992; Gruninger and Fox, 1995], KIF [Genesereth and Fikes, 1992], Enterprise Ontology [Uschold et al., 1998], Open Information Model [Meta Data Coalition, 1999], and CIMOSA [Kosanke, 1995]. In the following paragraphs, each of these ontologies is reviewed with respect to the goals, scope, accomplishments/status, and organizational components addressed.

3.1. TOVE [http://www.eil.utoronto.ca/tove/toveont.html]

The Toronto Virtual Enterprise (TOVE) project is a project at the Enterprise Integration Laboratory, University of Toronto. The goal of TOVE is to construct a data model that is expressive enough to represent all aspects of enterprise knowledge at both the generic level and the application level. Specifically, the data model is intended to: (1) provide a shared terminology for the enterprise that each agent can jointly understand and use; (2) define the meaning of each term as precisely and unambiguously as possible; (3) implement the semantics in a set of axioms that enable TOVE to automatically deduce the answer to many "common sense" questions about the enterprise; and (4) define a symbology for depicting terms and concepts graphically. Table I presents the concepts and their subclass relations (subclasses) included in the TOVE ontology.

The TOVE project was unique in that it employed a formal approach to ontology design and evaluation. First, ontology specialists collaborated closely with administrative and engineering personnel from various types of industrial firms (e.g., IBM, Canada; BHP Steel, Australia; Toyo Engineering, Japan) to identify specific problems that arise in actual enterprises. Because the ontology was expressed in KIF, it allowed automated deduction. So analysis was done in terms of deducing facts from the ontology represented in KIF. Specifically, the problems identified were used as a guide to create a comprehensive set of *competency* questions that an enterprise ontology should be able to answer. The competency questions identified were then used to guide the selection of concepts and relations that were included in TOVE. Next, the competency questions and all concepts and axioms were formalized in first-order logic to create an initial version of the ontology. Finally, the formalized competency questions were used to validate and finalize TOVE.

It should be noted that TOVE is not a single ontology, but a set of several individual ontologies that links the various logical parts of an enterprise model through appropriate relations. These ontologies represent activities, states (including time), products, organization, and activity-based cost management. Within each of these ontologies, a number of hierarchical structures (two or three levels deep) are used to represent clusters of

Table I. Tove's Mini-Theory on Organization

• Organization-Entity	• Organization Goals
• Organization-Group	• Communication-Link
• Organization-Individual	• Empowerment
• Organization-Role	• Authority
• Organization Position	
• Coordination Speech Acts: Propose, Counter-Propose, Accept, Reject, Cancel, Commit, De-commit, Satisfy, Fail	

knowledge. Finally, axioms and relations are used to link knowledge between the various clusters.

The TOVE project employed a more formal approach to ontology design than was used by the other ontologies described in this paper. Therefore, it is more relevant than the others to this paper. There are, however, some important "gaps" in the TOVE ontology, which may well be a result of the ambitious scope of the TOVE project.

3.2. Knowledge Interchange Format [http://logic.stanford.edu/kif/Hypertext/kif-manual.html]

The Knowledge Interchange Format (KIF) is an example of an ontology definition language. KIF provides facilities for defining objects, functions, and relations. KIF, which has declarative semantics, is based on first-order predicate calculus. It provides for the representation of meta-knowledge and nonmonotonic reasoning rules. KIF could be legitimately considered an ontology in that it does contain a certain view of the world. While its representation is far from being at the level of detail of the other ontologies, KIF does axiomatize microtheories of numbers, sets, and lists.

3.3. Enterprise Ontology [http://www.aiai.ed.ac.uk/~entprise/]

The Enterprise Ontology is an outcome of a project at the University of Edinburgh, AIAI Institute. It has a similar motivation as TOVE. Its goal is to obtain an enterprise-wide view of an organization which can then be used to provide a method and computer tools that "help capture aspects of a business and analyze these to identify and compare options for meeting the business requirements." The main motivation behind the Enterprise Ontology is to facilitate enterprise design and analysis by supporting communications among humans as opposed to machines. Table II presents the major concepts used in the Enterprise Ontology.

Because the primary objective of this project was to facilitate communications between humans, there was less of an attempt to precisely formalize the concepts. In contrast to the TOVE, the semantics of many of the concepts in the Enterprise Ontology are limited to the specification of their superclass or the relations that they participate in, with no attempt at axiomatization. On the other hand, Enterprise Ontology developers have made a painstaking effort to specify the intended meaning of each term in a way that would enable non-technical users to easily understand and agree on the meaning of the different terms.

The Enterprise Ontology researchers worked with IBM in the United Kingdom and Lloyd's Register, on projects where re-engineering was the focus. Given the less formal nature of this ontology (in comparison to TOVE), the Enterprise Ontology was successfully used as a knowledge capture tool. Specifically, it was used to mediate the different intuitions of people on a re-engineering team.

3.4. Open Information Model [http://www.mdcinfo.com/OIM/index.html]

The Open Information Model (OIM) is a project sponsored by Meta Data Coalition, a nonprofit consortium of vendors and end-users whose goal is to provide a vendor-neutral and technology-independent specification of enterprise meta-data. The goal of OIM, in particular, is to define a meta-data specification in the application development and data warehousing domains. Its main uses are: modeling the structure, processes, and rules of a business; designing and managing process libraries; animating and simulating business processes; and interchanging business modeling information between tools and enterprise resource planning systems. The major components of the OIM are: Analysis and Design Models, Objects and Components Models, Database and Data Warehousing Models, Knowledge Management Models, and Business Engineering Models. Table III presents the major concepts included in the Business Engineering Models compo-

Table II. Key Concepts in Enterprise Ontology

- Person, Machine, Corporation, Partner, Stakeholder
- Employment Contract
- Share, Shareholder
- Partnership
- Legal Entity, Legal Ownership, Non-Legal Ownership, Ownership, Owner
- Asset
- Organizational Unit
- Manage, Delegate
- Management Link
- Hold Purpose, Intended Purpose, Holder, Strategic Purpose, Objective, Vision, Mission, Goal
- Help Achieve, Strategy, Strategic Planning, Strategic Action, Decision, Assumption, Critical Assumption, Non-critical Assumption, Influence Factor, Critical Influence Factor, Non-Critical Influence Factor, Risk

nent of the OIM, which deals with the organizational aspects of enterprise models. As shown in the table, the concepts are subdivided into four major categories: Business Goals, Organizational Elements, Business Processes, and Business Rules.

The Open Information Model provides a somewhat odd mixture of organizational-level and system-level constructs. The design rationale is unclear and, despite the attempt to precisely define the meaning of each concept, the concepts are not formally defined. Despite these apparent shortcomings, Microsoft is backing this project. Consequently, the ontology used in the OIM needs to be considered in a review of the literature.

3.5. CIMOSA (Computer Integrated Manufacturing Open System Architecture) [http://cimosa.cnt.pl/Docs/Primer/primer0.htm]

CIMOSA is a project sponsored by a number of European ESPRIT projects. Its primary focus is to "provide a framework for analyzing the evolving requirements of an enterprise and translating these into a system which enables and integrates the functions which match the requirements." In particular, it is intended to provide unambiguous terminology that could serve as a common technical base for Computer Integrated Manufacturing (CIM) users, CIM system developers, and CIM component suppliers.

The CIMOSA reference model is represented as a cube, with the three dimensions being: generation of views, instantiation of building blocks, and derivation of models. CIMOSA also defines four different views for building models: Function, Information, Resource, and Organization. These views are designed to allow the user to work with a specific subset of model elements, i.e., to reduce the complexity of the model by focusing on one aspect of the model at a time. CIMOSA has been used in the ESPRIT projects by independent organizations, and by AMICE member organizations. A few CIMOSA related modeling tools have now become available. The CIMOSA constructs involved in

Table III. OIM Model of Organization (Business Engineering)

Business Goals:
- Business Goal, Vision, Mission, Goal
- Objective and Measure types

Organizational Elements:
- Resource, PhysicalResource, InformationResource, BusinessUnit, Policy

Business Processes:
- Business Process, Classifier, Tasks, Manual Tasks, Automated Tasks, Business Process Model, BusinessProcessGraph, ProcessFlow, Transition, Data Flow, Task State, Process Partition

Business Rules:
- Business Rule: TermRule, FactRule, InferenceRule, DerivationRule, DefinitionRule, ActionRule
- Constraint
- ResourceRuleRole

the organizational view are organizational unit, organizational cell, authority, and responsibility.

CIMOSA has limited expressiveness in support of the organizational view. In addition, its constructs are not formalized to offer precise characterization. Despite these deficiencies, the ontology used in CIMOSA has high visibility due to its use on a very large project sponsored by the European community.

3.6. Process Specification Language [http://www.mel.nist.gov/psl/]

Process Specification Language (PSL) is an ongoing project at the National Institute of Standards and Technology. This project is concerned with the development of a neutral, standard language specification for processes. PSL is intended to serve as an Interlingua for integrating multiple process-related applications throughout the manufacturing life cycle. As an interchange language, PSL is unique due to the formal semantics that underlie the language. The use of explicit, unambiguous definitions in PSL enable information exchange without having to rely on "hidden assumptions or subjective mappings" [Schlenoff et al., 2000]. The motivation for the PSL project was the recognition that manufacturing engineering and business software applications both use process information but in uniquely different ways. For example, process information is used by manufacturing process planning, production scheduling, product realization process modeling, workflow control, and project management applications. However, the underlying process representation varies from one application to the next. Compounding the problem is the fact that the same terms used in different applications often mean different things. For example, in a conventional workflow system, a resource is primarily thought of as information needed to make the necessary decisions whereas in a process planning system, a resource is primarily thought of as a person or a machine. Integrating these two systems can produce nothing but confusion, if one were to ignore the differences in semantics of these applications when translating to a neutral standard. The PSL project directly attacks this problem.

3.7. Other Relevant Ontologies

Two additional ontologies developed for very specific applications are worth mentioning because they contain certain relevant concepts. These are CYC [Lenat, 1995], which was developed for creating a common sense knowledge base, and PLINIUS [van der Vet and Mars, 1993], which was developed for use in ceramics research.

4. IDEON: ORIGINS, OBJECTIVES AND PAYOFFS

IDEON is a unified enterprise ontology that provides a common foundation for designing, reinventing, managing, and controlling collaborative, distributed enterprises [Madni and Lin, 1997–1998; Madni, Madni, and Lin, 1998]. It consists of: (a) a set of "business" objects that represent common entities within an enterprise context; (b) relations that link these objects in specific ways to establish business configurations; and (c) business rules that govern the behavior of various business configurations. The design of IDEON was inspired by the recognition that a conceptually unified ontology can make: (a) enterprise applications development more straightforward; (b) enterprise integration more systematic; (c) enterprise process management more effective; and (d) enterprise software maintenance much simpler. IDEON is readily extensible to vertical application domains. IDEON is compliant with the NIST Process Specification Language (PSL) [Schlenoff et al., 2000] requirements and can be directly mapped to popular implementation models such as CORBA IDLs, KIF, and DCOM. IDEON-based systems can potentially provide several benefits including:

a. Promoting common understanding between system developers and users of enterprise management applications.
b. Providing users with requested information about the enterprise and its environment.
c. Answering complex queries by navigating appropriate links.
d. Supporting different types of analyses such as syntactic correctness checking, Activity-Based Costing (ABC), Critical Path Analysis (CPA), inter-process mismatch analysis, enterprise-process mismatch analysis, and requirement tracking.
e. Supporting process streamlining as well as organizational and process reengineering.
f. Guiding enterprise application integration.
g. Providing decision support in virtual/traditional enterprise integration, resource planning, requirement/performance review, and process control.
h. Supporting collaborative work, adaptive workflow execution, progress tracking, and automated or mixed-initiative process management.

5. KEY DESIGN CONCEPTS

IDEON is based on the following four high-level design concepts that collectively satisfy the requirements of a flexible, interoperable, and maintainable enterprise management and control solution:

1. *Neutrality*. Since IDEON is an ontology (i.e., it describes concepts and their inter-relationships), it is notation-independent and implementation-neutral. In this paper, IDEON is represented as an object-oriented model using the Unified Modeling Language (UML) notation. IDEON can also be represented as CORBA IDLs (an interface definition language from the Object Management Group), or in Knowledge Interchange Format (KIF), or any other modeling language notation.
2. *Extensibility*. The IDEON core is readily extensible to various application domains such as product design, manufacturing, health care, financial services, and military command and control. The IDEON core also provides the basis for establishing compatibility with other relevant enterprise ontologies that address one or more specific aspects of an enterprise.
3. *Complementarity*. IDEON is based on the recognition that multiple complementary enterprise perspectives are needed to model the various aspects of an enterprise and to understand, design, manage, and control enterprise operations.
4. *Interoperability*. IDEON is designed to interoperate with other special-purpose enterprise process or plan ontologies and modeling notations.

6. IDEON PERSPECTIVES

IDEON employs four complementary perspectives (or views) to capture the key concepts and relationships that characterize an enterprise. Each perspective is described below.

6.1. Enterprise Context View

The enterprise context view (Fig. 1) represents the interaction between an enterprise and its external environment. In this view, an "Environment" is composed of multiple related "Enterprises." Enterprise has three special relationships with other organizations: PartnerOrganization, Customer, and CompetingOrganization. Allowing these three objects to inherit attributes and relationships from the Enterprise class enables the development of a detailed model of the external organizations. This feature makes it easier to study the impact of changes in other organizations and makes it easier to form virtual enterprises dynamically. This representation allows us to study the behavior of a virtual enterprise (i.e., an enterprise which is composed of several cooperating organizations that come together over a finite time window to accomplish an objective). However, this representation does not mandate a detailed model of each subject. The importance of each subject and the available knowledge should be used to decide the level of modeling detail. This representation also makes it possible to combine multiple models into an aggregate model that can be analyzed and controlled.

Each Enterprise (organization) employs "sensors" to observe the Environment. The observed information

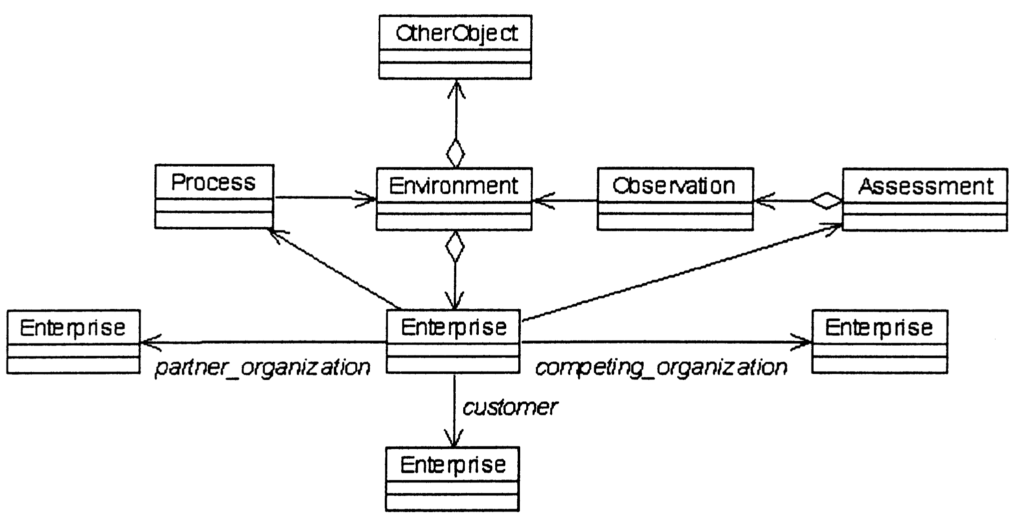


Figure 1. IDEON ontology—enterprise context view.

(e.g., customer survey, customer analysis) is used by the enterprise to assess the state of the Environment.

Based on the Assessment of the Environment (context), the Enterprise can perform certain processes/operations (e.g., develop new product) with the goal of achieving a specific effect on the Environment (e.g., increase market share). It should be noted that most such operations can be expected to have an impact on the Enterprise itself as well as on other organizations. In certain special cases, the operation of an Enterprise may only have internal impact without any externally observable effect. This observe–respond cycle continues indefinitely in the life of an enterprise until the enterprise ceases to exist.

6.2. Enterprise Organizational View

The enterprise organizational view (Fig. 2) is a structural view of the enterprise which complements the enterprise context view. An Enterprise may be composed of multiple, smaller sub-Enterprises with potentially the same structure as the parent Enterprise. This recursive definition of Enterprise allows the study of an organization at any level using the same analysis approach. Each Enterprise has specific Goals to accomplish in a particular environment (e.g., manufacturing). Based on the Assessment about a particular environment, an Enterprise could adopt/set up the right Strategies to achieve its Goals. These strategies are implemented through the selection, sequencing, and execution of appropriate Processes that collectively achieve detailed Objectives that are sub-goals of the original Goals. To perform these Processes, an Enterprise must own or have access to appropriate Resources, which may require collaboration with another enterprise.

There are several relationships that are possible between Enterprises. A Single Enterprise may consist of several smaller organizations. This parent–child relationship may occur recursively to form an Enterprise decomposition hierarchy. Two organizations may be sibling Enterprises under the same parent. In this case, they do not operate independently, but cooperate to achieve their goals. This cooperation is modeled through relationships in the Process.

A common way of representing inter-departmental coordination is through a cross-functional process map [Rummler and Brache, 1995] in which the rows are the various individuals, departments, or organizations, and the process representation runs from left to right with explicit representation of coordination relationships among activities across "swim lanes." This is an important representation because most process improvements are the result of improvements in inter-departmental collaboration. A key task for an executive/manager is the management of the "white space" between the departments under his/her purview. Therefore, it is essential to model and analyze the interfaces between and among departments. This map also shows the internal customers of an organization, which is especially important for those organizations that are responsible for "enabling" processes that do not contribute directly to the business goals of the parent Enterprise.

Each Enterprise owns one or more Resources and is managed by exactly one Person (a special case of Resource). This one-manager link is useful for organization charts showing the reporting structure within an organization and for supporting implicit "approval" processes.

An Enterprise has one or more business Objectives to achieve through its operations. Associated with each business Objective is a metric for evaluating achievement or progress. An organization achieves its business

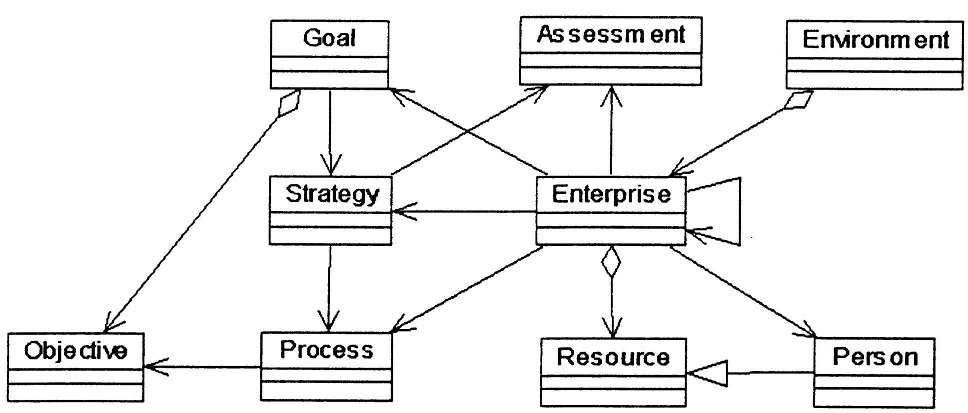


Figure 2. IDEON ontology—enterprise organizational view.

Objectives by providing value to its customer Enterprises (e.g., an external organization in the Environment, or an internal sub-organization in the same enterprise). The value can be in the form of a product sold or services performed.

6.3. Process View

The process view (Fig. 3) represents the (re)planning-execution–control cycle. A Process is a sequence of actions that is taken to achieve a particular Objective. There is one organization (Enterprise) that is responsible for each Process. A Process can be triggered by a particular Event. An intelligent process may be able to modify itself upon the receipt of certain event notifications. The execution of a process may be constrained by one or more Conditions. For example, a "die-casting process cannot start until the temperature indicated by gauge-1 is higher than 200 degrees."

A Condition is represented by a logical expression that is defined in terms of entities in the system. An Event is defined as a change in the state(s) of entities in the system. A Composite Event is defined as a logical combination of other events.

A Process may be related to another Process through causal/temporal dependency relationships. For example, the "shipping" process cannot be started before the "manufacture" process has been completed. A Process can also be related to another Process through the specialization relationship. For example, the "ISTI software development process" is a specialization of the "generic software development process." This is important for indexing processes and for searching a process knowledge base for a particular process. A Process can be further classified into three types: planning process, plan, and activity.

The PlanningProcess is responsible for controlling the entire plan execution and replanning cycle. To this end, it is important for an enterprise to choose the right strategy with a set of processes (plans) for achieving the Goal. This does not happen automatically. An enterprise must implement an agile planning process that knows how to react to any type of event and can control the enterprise such that it stays "on course."

The Plan created by an enterprise can either be a sequence of executable activities (operations), or a high-level plan consisting of several lower level plans (i.e., sub-plans) that are assigned to sub-organizations of that enterprise. In the latter case, the assigned sub-organization can perform further planning to decompose the sub-plan into executable activities.

An executable Activity is a unit of work that cannot be further decomposed in the model. During execution, an activity is performed by one or more human resources with material resources (e.g., aircraft) used as tools, inputs, or references. As a result of the execution of an activity, resources can be created, deleted, or further modified. The execution of an activity could also result in a change in the state of the external environment.

6.4. Resource/Product View

The resource view (Fig. 4) elaborates on the various types of resources that might be needed to execute a process. A Resource can also be the product of a proc-

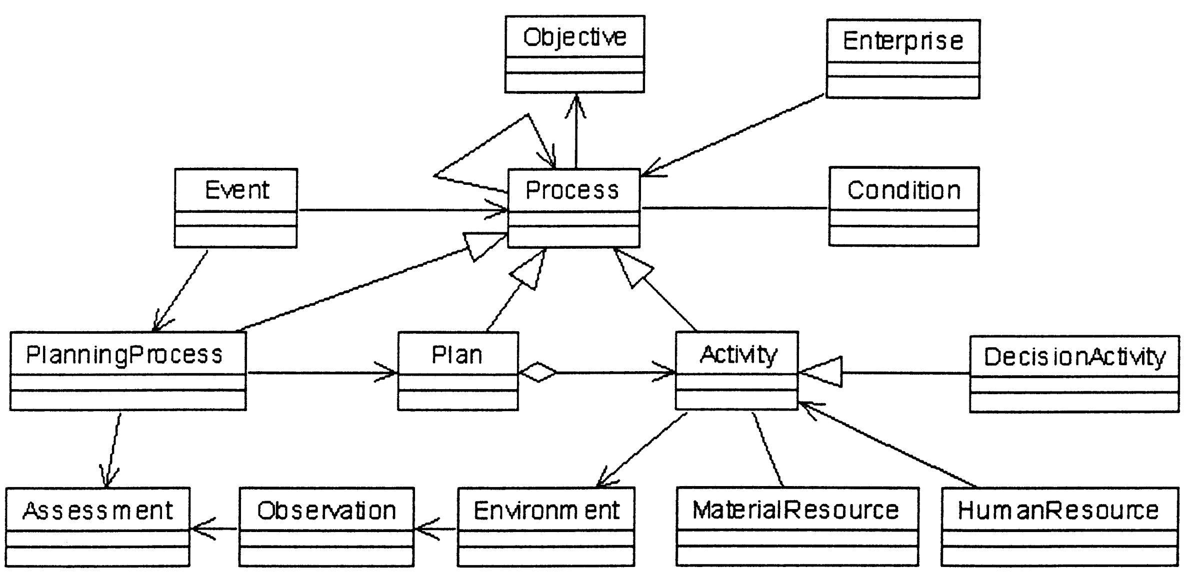


Figure 3. IDEON ontology—process view.

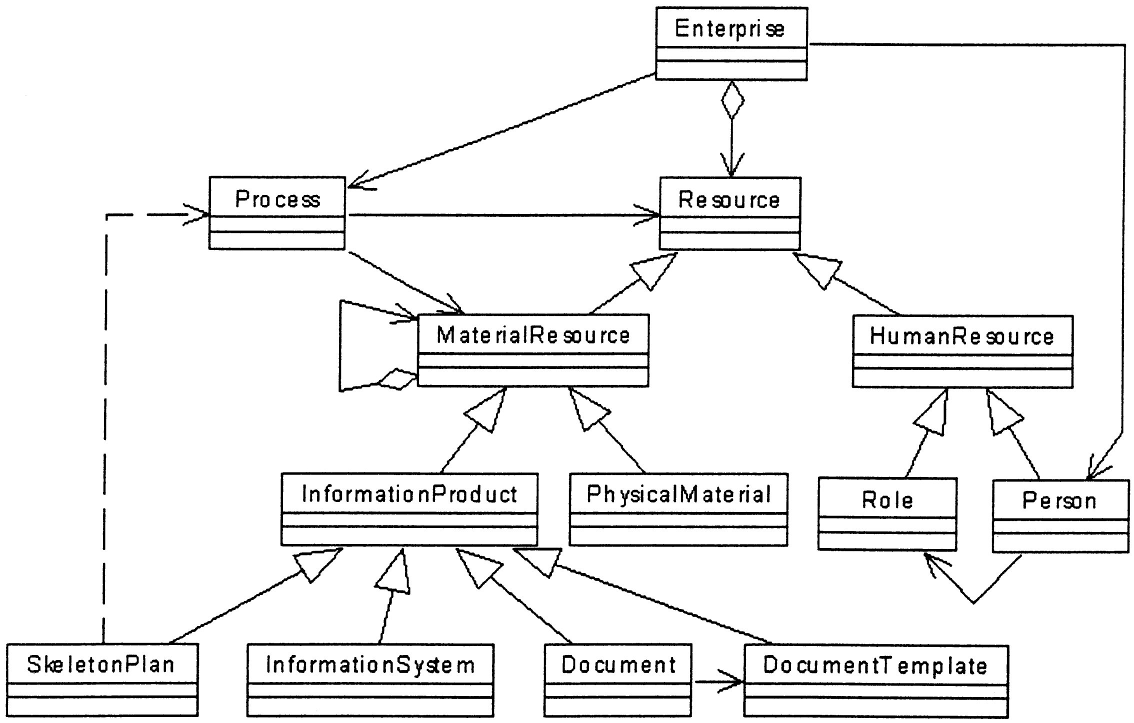


Figure 4. IDEON ontology—resource/product view.

ess. Resources can be classified into Human Resources and Material Resources. A HumanResource can be classified into a Role or a Person. A Role contains information about a particular position, e.g., the qualification, responsibility, and authorization. Role will be used in most process definitions, but one or more Persons have to be assigned to each role before the process can be executed. A Role can be assigned to one or more Persons and a Person can be assigned one or more Roles. A MaterialResource can be further classified into either an InformationProduct or a PhysicalMaterial. This is an important distinction because an InformationProduct can be directly controlled by a process management system while a physical material cannot.

An enterprise's products are a subset of all the outputs from the enterprise's processes. A product can be a physical product that can be sold to the customers, a document, a service to the customers, an executable process (or skeletal plan), or a new information system. A Resource object is a resource already owned by an enterprise (to execute processes), or created by a process in that enterprise. A non-human resource (i.e., material resource) created by a process can be a product of that enterprise. Similarly, a trained human (i.e., a human resource) is the product of a training process. A product (material resource) may consist of product components. In such case, the product components are identified in the product breakdown structure (or, Bill Of Material in a manufacturing environment).

7. APPLICATIONS

IDEON has been used to support multiple enterprise applications. Two such successful applications include: Crisis Action Planning and Execution; and Integrated Product-Process Development (IPPD)-enabled systems engineering. Each application is discussed next.

7.1. A Process-Centric Crisis Action Planning and Execution System

Crisis Action Planning and Execution is a complex process that spans situation awareness, execution planning, and execution [USPACOM, 1993]. Crisis Action Planning and Execution is a special class of C^2 processes that encompasses both warfighting operations as well as operations other than war. The typical characteristics of a crisis action planning and execution operation include:

- Multiple "stakeholders" with different "win-conditions" that can be in conflict
- Decisions that must be made in the presence of uncertainty and ambiguity

- Observe–Orient–Decide–Act (OODA) cycle time is a key determinant of success
- A sparse database of past incidents to draw upon

Crisis management teams face several challenges, not the least of which is the need to shrink the cycle time from the recognition of a crisis event to the accomplishment of the mission. In large part, this means being able to compress the execution and replanning cycle. To achieve this overarching objective, crisis management teams need to (a) have the necessary information at their fingertips at the right time; (b) maintain a high degree of situation awareness and concentrate on high-level tasks (i.e., to be offloaded from low-level/routine tasks); (c) be able to try out different scenarios and "quick change" their overall process or plan in response to certain key events; (d) be able to visualize the executing plan from different perspectives and "drill-down" to more detailed levels if necessary; (e) be able to coordinate and collaborate with other stakeholders rapidly and easily whenever the need arises; and (f) seamlessly transition from collaborative (i.e., human-in-the-loop) replanning to automated execution, and vice versa.

For this application, IDEON provided the underlying framework consisting of key "business objects," their relationships and attributes [Madni and Gayton, 1999]. This conceptual framework was used to: (a) create an online catalog of component processes involved in creating a crises response package; (b) compose a crisis response to a specific crisis based on these component processes and their specific tailoring to the crisis at hand; and (c) managing the crisis over the Web using the composed model as a reference or guide during the conduct of a crisis operation. During execution, the system is capable of interactively/automatically adapting the flow, work allocation, and personnel assignments in response to contingency events.

The system concept for an IDEON-based distributed crisis action planning and execution system is shown in Figure 5. Central to this system concept are the capabilities for (a) process design and (b) intelligent process management. These capabilities are based on our core ontology for integrated product-process development [Madni, Madni, and McCoy, 1998]. The *process design* capability allows process engineers/human planners to: (1) design/author processes/plans; (2) verify process/plan correctness with respect to completeness and consistency and store verified processes/plans in a Crisis Action Planning and Execution (CAPE) Assets Library; (3) analyze processes/plans in terms of their resource requirements, cycle times and cost; (4) streamline/simplify processes/plans to eliminate redundancies and extraneous handoffs; and (5) re-engineer processes/plans to exploit promising new C^2-related information technologies.

The *intelligent process manager* enables and oversees plan execution, invokes the various applications

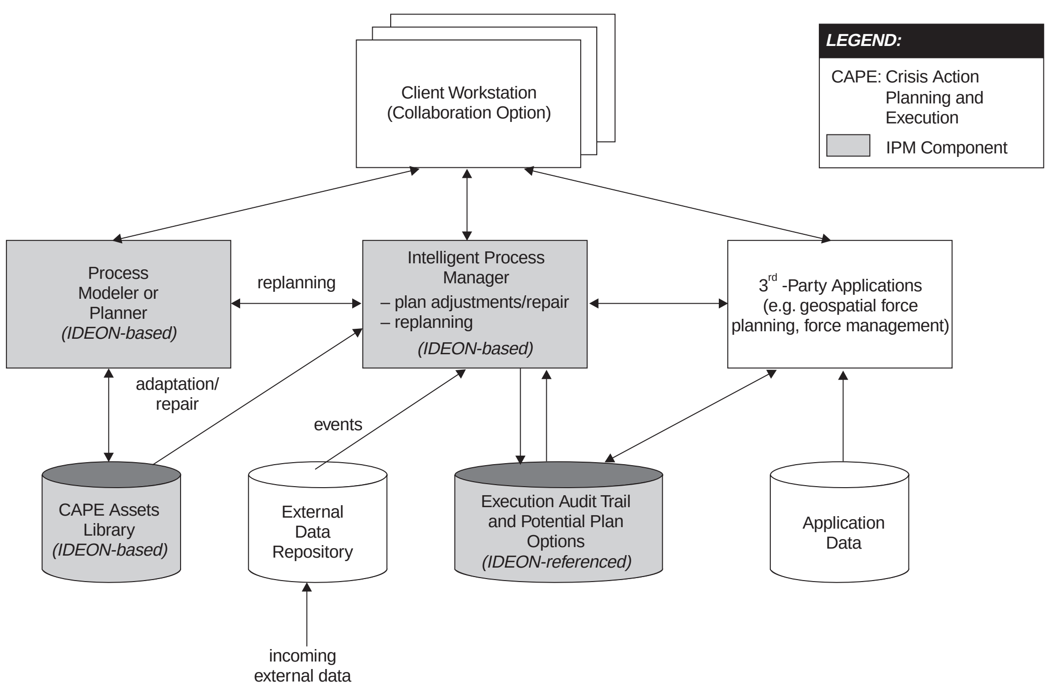


Figure 5. IDEON-based crisis action planning and execution system.

that support the executing plan, and accesses data required for plan execution. The IPM is not only "process-aware" but also "situation-aware" in that it can recognize and respond to key external or internal events. This response can range from automatic plan execution adjustment, to plan repair (e.g., replacing an invalid component with a valid one from the process/plan asset library), to execution support during collaborative replanning after handoff to a virtual collaboration environment.

IDEON Extensions for Crisis Action Planning and Execution. IDEON customization for this application consisted of: introducing the concept of a "plan," which is defined as (a) the output or product of a planning process and (b) an executable process instance. The specific interpretation is based on usage. In other words, usage context determines whether the plan is viewed as an output or product, or whether it is viewed as an executable process instance. The same type of distinction is made when defining the status of a plan. When a plan is viewed as a product, its status takes on values such as: created, modified, and finalized. When a plan is viewed as a process instance, its status takes on values such as started, ongoing, suspended, aborted, and completed. In addition, specific events were defined that served as criteria for adaptation of the crisis response during actual operations. This capability provides the agility in managing crisis operations.

7.2. Integrated Product-Process Development (IPPD)

The DoD's Integrated Product-Process Development (IPPD) mandate [DoD, 1996] requires system developers to concurrently address product and process issues from a full life-cycle perspective, and to start this activity from the very beginning of the product life-cycle. Integrated Product-Process Development emphasizes the concurrent development of a product along with the processes by which the product is created and supported. A central concept in IPPD is that of Integrated Product Teams, which are responsible for carrying out the IPPD process. A major aspect of the IPPD lifecycle is the modeling, analysis, improvement/redesign, execution, and management of the IPPD process and its integration with product design data/tools.

IDEON was extended to support IPPD-enabled systems engineering. The extended IDEON is referred to as IDEON/IPPD [Madni, Madni, and Lin, 1998].

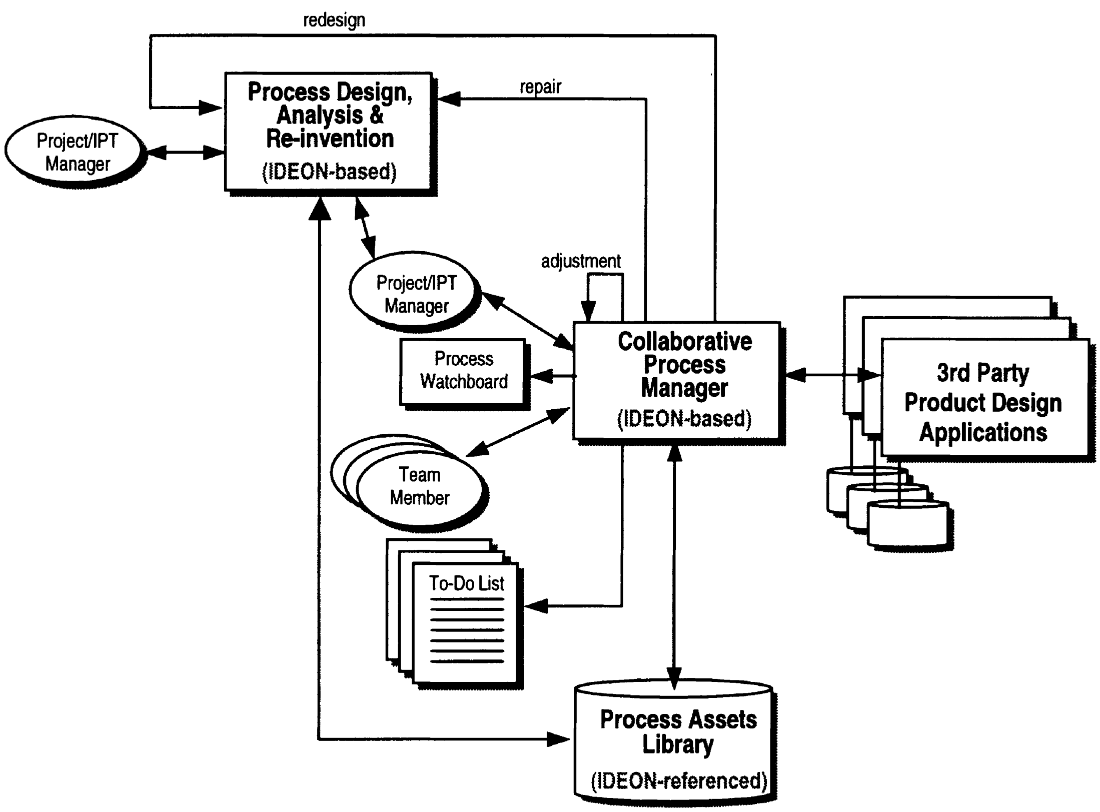


Figure 6. IPPD process management system concept.

IDEON/IPPD provides the conceptual foundation for modeling, analyzing, and redesigning or improving complex systems development processes while taking system constraints into account. A system is considered to be complex [Sage, 1998] when "we cannot understand it through simple cause-and-effect relationships or other standard methods of systems analysis. In a complex system, we cannot reduce the interplay of individual elements to the study of individual elements considered in isolation. Often, several different models of the complete system, each at a different level of abstraction, are needed."

For this application, IDEON provided the underlying conceptual framework for modeling/designing and persistently storing the IPPD process for subsequent operation over the Web [Madni, Madni, and McCoy, 1998]. Figure 6 presents the overall system concept for the resultant system. IDEON extensions for this application included the addition of: (a) new constraints modeled as pre- and post-conditions that define specific "gates" in the IPPD process; (b) the association of states with *each* component of the complex system or highly engineered product; (c) the definition of interfaces between components; (d) the definition of product metadata with linkage to existing product design tools and data; and (e) the incorporation of concepts such as "product specification" with a link to the design that satisfies it.

8. CONCLUDING REMARKS

Competitive advantage in today's environment depends, to a large extent, on whether or not an enterprise can capitalize on IT developments and trends to exploit fast moving business opportunities. Change, once viewed as a short period of relative instability, is now a continuous process. At the same time, the concept of a monolithic enterprise that totally owns all products, services, and channels to serve a customer are rapidly being replaced by strategic partnerships, virtual enterprises, and integrated value chains. The need to operate in a dynamic business and technical environment is driving the need for technology infrastructures and application architectures that are increasingly more flexible, integrable, and maintainable. It is in response to these needs that IDEON was created. Unlike other enterprise ontologies, IDEON enables and/or supports the creation, integration, and management of collaborative, distributed enterprises. Specifically, it supports integrated planning and execution of enterprise processes which are key requirements in command and control, supply chain management, and crisis action planning and execution systems. This paper has presented the underlying principles and key components of IDEON as well as two specific enterprise applications that are based on IDEON. As discussed in this paper, these applications required minimal extensions to IDEON to provide the requisite functionality. An earlier version of IDEON has been implemented in the ProcessEdge™ Enterprise Suite from Intelligent Systems Technology, Inc.

REFERENCES

Department of Defense (DoD), Guide to integrated product and process development (version 1.0), Office of the Under Secretary of Defense (Acquisition and Technology), Washington, DC, February 5, 1996.

M.S. Fox, J.F. Chionglo, and F.G. Fadel, TOVE manual, University of Toronto, Toronto, 1992.

M.R. Genesereth and R.E. Fikes, Knowledge interchange format, version 3.0, reference manual, Knowledge Systems Laboratory, Stanford University, Palo Alto, CA, 1992.

T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing, KSL-93-04, Knowledge Systems Laboratory, Stanford University, Palo Alto, CA, 1993.

M. Gruninger and M.S. Fox, Methodology for the design and evaluation of ontologies, Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Quebec, Canada, August 19–20, 1995.

K. Kosanke, CIMOSA—overview and status, Comput Ind 27(2) (October 1995).

D.B. Lenat, Cyc: A large-scale investment in knowledge infrastructure, Commun ACM 38(11) (November 1995) (see also other articles in this special issue).

A.M. Madni and J.P. Gayton, A process-centric crisis action planning and execution system, Proc 13th Int Conf Syst Eng, Las Vegas, Nevada, August 10–12, 1999, pp. SE205–SE210.

A.M. Madni and W. Lin, ISTI distributed enterprise ontology (IDEON™): An overview, ISTI White Paper ISTI-WP-5/97-1, Intelligent Systems Technology, Santa Monica, CA, 1997–1998.

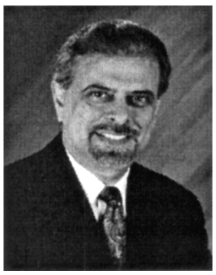
A.M. Madni, C.C. Madni, and W. Lin, IDEON™/IPPD: An ontology for systems engineering process design and management (invited paper), Proc 1998 IEEE Int Conf Syst, Man, Cybernet, San Diego, CA, October 11–14, 1998.

A.M. Madni, C.C. Madni, and W.L. McCoy, Process support for IPPD-enabled systems engineering (invited paper), Proc 1998 IEEE Int Conf Syst, Man, Cybernet, San Diego, CA, October 11–14, 1998, pp. 2585-2590.

Meta Data Coalition, Open Information Model, Version 1.1 (Proposal), November 1999, http://www.mdcinfo.com/OIM/index.html [Open Information Model (last seen: April 17, 2000, last update: unknown)].

G.A. Rummler and A.P. Brache, Improving performance: How to manage the white space on the organization chart, 2nd ed., Jossey-Bass, 1995.

A. Sage, Toward systems ecology, IEEE Comput (February 1998), 107–109.
C. Schlenoff, M. Gruninger, F. Tissot, J. Valois, J. Lubell, and J. Lee, The Process Specification Language (PSL): Overview and version 1.0 specification, NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD, 2000.
M. Uschold, M. King, S. Moralee, and Y. Zorgios, The enterprise ontology. Knowledge Eng Rev 13 (Special Issue on Putting Ontologies to Use) (1998) (also available from AIAI as AIAI-TR-195).
USPACOM, Crisis Action Planning and Execution, Preliminary Functional Economic Analysis (Draft), USPACOM, Nov 30, 1993.
P.E. van der Vet and N.J.I. Mars, Structured system concepts for storing, retrieving, and manipulating chemical information. J Chem Inf Comput Sci 33 (1993) 564–568.
H.B. Woolf (Editor), Webster's new collegiate dictionary, G. & C. Merriam, Springfield, MA, 1981.



Azad M. Madni is the CEO of Intelligent Systems Technology, Inc. He has previously served in a variety of leadership roles in high technology companies. He also provides strategic technology consulting services to corporate executives. He has written over 150 articles, reports, and book chapters in enterprise ontologies, intelligent process management, enterprise modeling and simulation, integrated planning and execution, adaptive decision aiding, collaborative systems engineering, and adaptive instruction in distance learning. Dr. Madni's research has been sponsored by several R&D agencies of the U.S. Government including the Defense Advanced Research Projects Agency, the Air Force Office of Scientific Research, the Air Force Research Laboratory, the Naval Research Laboratory, the Office of Naval Research, the Army Research Institute, the Department of Commerce, the Department of Energy, and the National Aeronautics and Space Administration. He has successfully commercialized his research in enterprise process design, collaborative process management, human–system integration, and concurrent engineering. He has received numerous commendations for technology innovation, transfer, and commercialization from various agencies of the U.S. Government. He has received national acclaim for leading ISTI to several prestigious awards and national rankings, including the 1998, 1999, and 2000 Deloitte & Touche Los Angeles Fast 50, the 1998 and 1999 Technology Fast 500, the SBA's National Tibbetts Award for California, the selection to *Computerworld's* 100 Emerging Companies to Watch in 2000, and the Blue Chip Enterprise Award from the U.S. Chamber of Commerce and Mass Mutual. A past Visiting Industrial Fellow at Caltech's Jet Propulsion Laboratory Center for Space Microelectronics, he is listed in the Marquis' *Who's Who in Science and Engineering* and *Who's Who in Industry and Finance*. Dr. Madni received his B.S., M.S., and Ph.D. in Engineering from the University of California at Los Angeles. He also has an Executive MBA from the AEA/Stanford Executive Institute.

Dr. Weiwen Lin is a Senior Member of the Technical Staff and Technical Director of Intelligent Systems Technology, Inc. At ISTI, he provides technical leadership to project teams engaged in the development and implementation of state-of-the-art distributed objects for collaborative planning and web-based distance learning. He serves as a co-Principal Investigator on DoD and NIST research projects on distributed process management and web-based adaptive instruction. Previously, he was a member of the research department at Anderson Consulting where he designed and implemented the security system for a medical information system. He received his M.S. in Computer Science and Ph.D. in Mechanical Engineering from Stanford University. His research interests include knowledge representation, knowledge management, enterprise modeling, distributed object architectures, and web-based security.

Carla C. Madni is currently the Vice President of Operations of Intelligent Systems Technology, Inc., a position that she has held for the last six years. Previously, she was a Senior Program Manager of Perceptronics where she oversaw and managed multiple R&D programs. Her research interests are knowledge acquisition, distributed problem solving, enterprise modeling, process management, and adaptive instruction. She has been a coinvestigator and program manager on DARPA, Air Force, Navy, and Army R&D programs. She has done extensive consulting on human–system engineering projects for major aerospace companies such as Lockheed Martin, Northrop Grumman, and Hughes. She received her B.S. in Biomedical Engineering from Tulane University and an M.S. in Engineering Systems with specialty in Distributed Problem Solving from UCLA. She is a member of the IEEE, the Human Factors Society, and the National Association of Female Executives.