InterSymp – 2008

20th International Conference on Systems Research, Informatics and Cybernetics

(July 24 - 30, 2008, Baden-Baden, Germany)

Pre-Conference Proceedings of the Focus Symposium on

Intelligent Software Tools and Services

Friday, July 25, 2008

Focus Symposium Chair:

Jens Pohl
Executive Director, Collaborative Agent Design Research Center
and Professor of Architecture
College of Architecture and Environmental Design
California Polytechnic State University
San Luis Obispo, California, USA

Sponsored by:

The International Institute for Advanced Studies
in Systems Research and Cybernetics
and
Society for Applied Systems Research

Professor George E. Lasker
Chairman

ISBN: 978-1-897233-65-8

Preface

Increasingly we are looking upon computers as enjoyable and valuable partners in our recreational and professional endeavors. However, the word *partner* suggests a level of interaction that requires some degree of *understanding* of the context of the interaction on both sides of the partnership. In other words, the notion of a human-computer partnership presupposes that computers can be programmed with software to display intelligent behavior. This proposition is naturally met with a great deal of skepticism and even opposition. Surely intelligence is a human attribute that cannot be replicated in a non-living device such an electronic machine? Isn't intelligent behavior closely associated with the human body and mind?

This is certainly true if we consider intelligence and human intelligence to be synonymous. However, perhaps it is also reasonable to argue that there are several forms of intelligence and that a computer can be programmed to perform tasks that are intelligent in as much as they incorporate reasoning and learning capabilities.

Webster's Dictionary (Random 1999) defines intelligence as the "… capacity for learning, reasoning, and understanding;". This definition suggests that there are component capabilities that contribute to the concept of intelligence. Further, these component capabilities are not necessarily equally powerful. In other words, it may be argued that there are levels of intelligence and that at the lowest level such capabilities must include at least the ability to remember. Higher levels of intelligence include reasoning, learning, discovering, and creating. Certainly at least some of these intelligent capabilities can be embedded in computer software. For example, computers excel at storing and recalling data in virtually unlimited quantities and over very long periods of time. Computers can reason about data quite effectively, if adequate context is made available with the data. Also, computers have been shown to have learning-like capabilities, and computers can discover information through associations and pattern matching.

It is not intended to suggest that computer intelligence is equal or even similar to human intelligence, but rather that computer intelligence and human intelligence may be applied in parallel to complement each other. Furthermore, a strong case can be made in support of the view that there is an urgent need for intelligent computer capabilities due to the mounting expectations of accuracy, quality and timeliness in a globally connected environment of rapidly increasing complexity.

There are essentially two compelling reasons why computer software must increasingly incorporate more and more *intelligent* capabilities. The first reason relates to the current data-processing bottleneck. Advances in computer hardware technology over the past several decades have made it possible to store vast amounts of data in electronic form. Based on past manual information handling practices and implicit acceptance of the principle that the interpretation of data into information and knowledge is the responsibility of the human operators of the computer-based data storage devices, emphasis was placed on storage efficiency rather than processing effectiveness. Typically, data file and database management methodologies focused on the storage, retrieval and manipulation of data transactions, rather than the *context* within which the collected data would later become useful in planning, monitoring, assessment, and decision-making tasks.

The reasons for this data-processing bottleneck are twofold. First, large organizations are forced to focus their attention and efforts on the almost overwhelming tasks involved in converting unordered data into purposefully ordered data. This involves, in particular, the establishment of gateways to a large number of heterogeneous data sources, the validation and integration of these sources, the standardization of nomenclatures, and the collection of data elements into logical data models. Second, with the almost exclusive emphasis on the slicing and dicing of data, rather than the capture and preservation of relationships, the interpretation of the massive and continuously increasing volume of data is left to the users of the data. The experience and knowledge stored in the human cognitive system serves as the necessary context for the interpretation and utilization of the ordered data in monitoring, planning and decision-making processes. However, the burden imposed on the human user of having to interpret large amounts of data at the lowest levels of context has resulted in a wasteful and often ineffective application of valuable and scarce human resources. In particular, it often leads to late or non-recognition of patterns, overlooked consequences, missed opportunities, incomplete and inaccurate assessments, inability to respond in a timely manner, marginal decisions, and unnecessary human burn-out. These are symptoms of an incomplete information management environment. An environment that relies entirely on the capture of data and the ability of its human users to add the relationships to convert the data into information and thereby provide the context that is required for all effective planning and decision-making endeavors.

The second reason is somewhat different in nature. It relates to the complexity of networked computer and communication systems, and the increased reliance of organizations on the reliability of such information technology environments as the key enabler of their effectiveness, profitability and continued existence. The economic impact on an organization that is required to manually coordinate and maintain hundreds of interfaces between data-processing systems and applications that have no *understanding* of the data that they are required to exchange is enormous. Ensuing costs are not only related to the requirement for human resources and technical maintenance, but also to the indirect consequences of an information systems environment that has hundreds of potential failure points.

Recent industry studies have found that more than 40% of computer system disruptions and failures are due to human error. However, the root cause of these human errors was not found to be lack of training, but system complexity. When we consider that computer downtime due to security breaches and recovery actions can cost as much as (US)$2 million per hour for banks and brokerage firms, the need for computer-based systems that are capable of controlling themselves (i.e., have autonomic capabilities) assumes critical importance. A core requirement of autonomic computing is the ability of a computer-based information system to recover from conditions that already have caused or will likely cause some part(s) of the system to fail. This kind of self-healing capability requires a system to continuously monitor itself so that it can identify, analyze and take mitigating actions, preferably before the disruption takes place. In addition, the system should be able to learn from its own experience by maintaining a knowledge base of past conditions that have caused malfunctions and the corrective measures that were taken.

In summary, the continued expansion of networks (e.g., the Internet and its successors) will provide seamless connectivity among countless nodes on a global scale. While the collection of data has already increased enormously over the past decade, the availability of such a global

network is likely to increase the volume of data by several orders of magnitude. Such a volume of raw data is likely to choke the global network regardless of any advances in communication and computer hardware technology. To overcome this very real problem there is a need to collect data in context so that only the data that are relevant and useful are collected and transmitted within the networked environment. Most (if not all) of the necessary filtering must be achieved automatically for at least three reasons. First, organizations cannot afford to utilize human resources for repetitive tasks that are tedious and require few human intellectual skills. Second, even if an organization could afford to waste its human resources in this manner it would soon exhaust its resources under an ever-increasing data load. Third, it does not make sense for an organization to burn-out its skilled human resources on low-level tasks and then not have them available for the higher-level exploitation of the information and knowledge generated by the lower level tasks.

Finally, the increased reliance on computer-based information systems mandates a level of reliability and security that cannot be achieved through manual means alone. The alternative, an autonomic computing capability, requires the software that controls the operation of the system to have some understanding of system components and their interaction. In other words, autonomic computing software demands a similar internal information-centric representation of context that is required in support of the knowledge management activities in an organization. In both cases the availability of data in context is a prerequisite for the reasoning capabilities of software agents (i.e., the automatic interpretation of data by the computer).

Jens Pohl, May 2008

(jpohl@calpoly.edu) (www.cadrc.calpoly.edu)

InterSymp-2008
International Conference on Systems Research, Informatics and Cybernetics

Focus Symposium
on
Intelligent Software Tools and Services

Friday, July 25, 2008

TABLE OF CONTENTS

"Challenging Computer Software Frontiers and the Human Resistance to Change" **9**
***Jens G. Pohl**, PhD, Collaborative Agent Design Research Center, Cal Poly, San Luis Obispo, California, US.*

.

"Intelligent Tool to Support Evolution of Information Systems" **29**
***Bogdan Czejdo**, Fayetteville State University, Fayetteville, North Carolina, US.*
***Thompson Cummings**, Saint George's University, Grenada.*
***Mikolaj Baszun**, Warsaw University of Technology, Warsaw, Poland.*
***Janusz Czejdo**, University of Edinboro, Edinboro, Pennsylvania, US.*

"Automated Metadata Tagging, Taxonomy Management and Auto-Classification of Information in an Enterprise Environment" **43**
***Stephen M. Wolfe,** Col (PhD), Air Force Institute for Operational Health, Brooks City-Base, Texas, US.*
***David S. Sanchez**, Maj, Air Force Institute for Operational Health and Chartwell Education Group, Washington, DC, US.*
***Simon A. Chapple,** Squadron Leader, MBChB, DAvMed, Royal Air Force, 711th Human Performance Wing, Brooks City-Base, TX.*

"Towards Joining Systems Science and Engineering of Semantics-Oriented Natural Language Processing Systems" **55**
***Vladimir Fomichov**, PhD, Department of Innovations and Business in the Sphere of Informational Technologies, Faculty of Business Informatics, State University –Higher School of Economics, Kirpichnaya str. 33, Moscow, Russia.*

"Similarity Assessment Techniques" **67**
Michael Zang, *CDM Technologies Inc., San Luis Obispo, California, US.*
Adam Gray*, CDM Technologies Inc., San Luis Obispo, California, US.*
Michael Hobbs*, CDM Technologies Inc., San Luis Obispo, California, US.*

"Perspective Models: A Mechanism for Achieving Interoperability Among Expressive, Personalized Domain Views" **81**
Kym J. Pohl, *CDM Technologies Inc., San Luis Obispo, California, US.*

"Semantic-Based Design Agents in Virtual Environments" **91**
Rabee M. Reffat*, PhD, Architecture Department, King Fahd University of Petroleum and Minerals, Saudi Arabia.*
Yaman Khaeruzzaman*, Information and Computer Science Department, KFUPM, Saudi Arabia.*
I Putu Raharja*, Information and Computer Science Department, KFUPM, Saudi Arabia.*

"Virtual Trading System: A Direction of Marketing in the New Age" **101**
Gahangir Hossain*, Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Bangladesh.*

"ICODES: A Multi-Agent System in Practice" **119**
Jens G. Pohl*, PhD****,*** *Collaborative Agent Design Research Center, Cal Poly, San Luis Obispo, California, US.*

Challenging Computer Software Frontiers and the Human Resistance to Change

Jens Pohl, Ph.D.

Executive Director, Collaborative Agent Design Research Center (CADRC)
California Polytechnic State University (Cal Poly)
San Luis Obispo, California, USA

Abstract

This paper examines the driving and opposing forces that are governing the current paradigm shift from a data-processing information technology environment without software intelligence to an information-centric environment in which data changes are automatically interpreted within the context of the application domain. The driving forces are related to the large quantity of data and the complexity of networked systems that both call for software intelligence. The opposing forces are non-technical and due to the natural human resistance to change.

Based on this background the paper describes current information-centric technology, proposes a vision of intelligent software system capabilities, and identifies four areas of necessary research. Most urgent among these are the ability to dynamically extend and merge ontologies and semantic search capabilities that can be initiated either by human users or software agents. Longer term research interests that pose a more severe challenge are related to the translation of emerging theoretical *hierarchical temporal memory (HTM)* concepts into usable software capabilities and the automated interpretation of graphical images such as those recorded by surveillance video cameras.

Keywords: agents, cognition, context, data-centric, extensible ontologies, human nature, HTM, image interpretation, information-centric, paradigm shift, representation, resistance to change, semantic search, situatedness, software, SOA, TEGRID.

Periods of accelerated change

Over the past hundred years there have been many fundamental changes in our human values and the way we perceive our environment (Figure 1). The Industrial Age placed great value on physical products and devised ingenious ways to maximize the manual contributions of its human work force in a subservient role to a highly automated mass production process. In the Information Age the focus has moved from the physical capabilities of the human work force to the intellectual capabilities and potential of its individual members. The attendant symptoms of this profound shift are the replacement of mass production with computer controlled mass customization, virtual products as opposed to physical products, and the creation and exploitation of knowledge. However, the rate of change is by no means constant.

Throughout history there have been periods of rapid and profound change. More often than not, and certainly in recent times, the precipitating factors have been technological and/or political in

nature. Sometimes these factors have gained momentum over time in a cumulative manner such as the French Revolution in the 18th Century, and at other times they have descended on society more abruptly. The terrorist attacks on the United States (US) that occurred on September 11, 2001 (9/11) are an example of the latter. In either case such periods of change have typically been accompanied by a great deal of human tension.

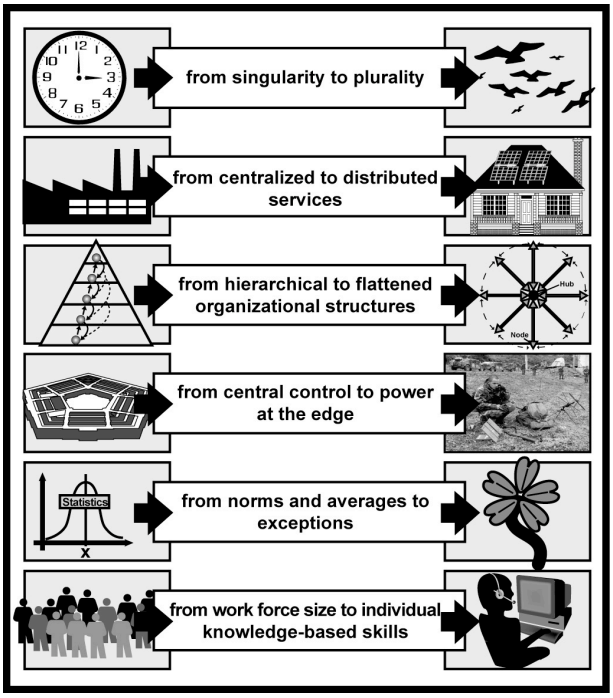


Figure 1: Many fundamental changes

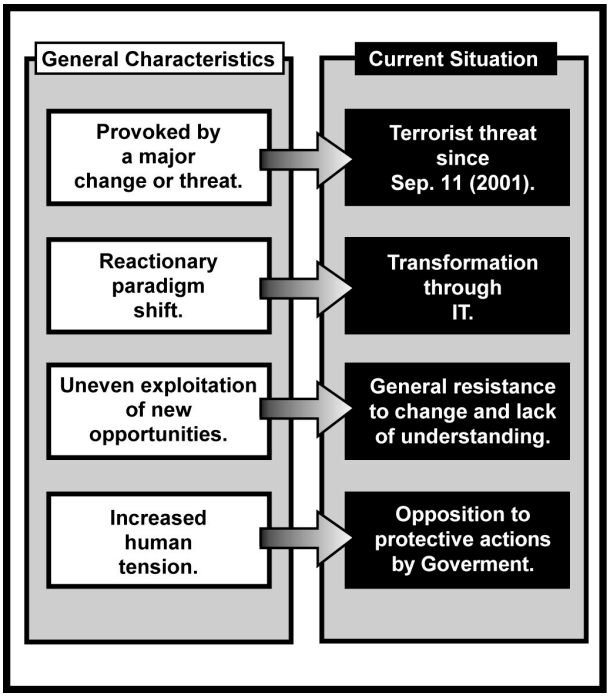


Figure 2: Periods of accelerated change

It is the dual purpose of this paper to explore some of the underlying reasons for the tensions that accompany periods of rapid change and to discuss the technological advances in computer software that are emerging as a natural byproduct. These advances tend to fall into two categories, namely: the implementation of theories and methodologies that have been under development for some time but were not exploited because there did not appear to be a compelling need for their immediate application; and, requirements for additional advances that become apparent as this existing knowledge transitions from focused research projects to broader and larger scale utilization. Typically, the first category manifests itself as a paradigm shift that is accompanied by an order of magnitude increase in capabilities and inevitably demands fundamental changes in the performance and management of existing tasks. The second category becomes apparent as human expectations for higher levels of exploitation of the new capabilities identify the need for additional capabilities.

The origin of a paradigm shift is normally associated with compelling needs that are often of a threatening nature (Figure 2). To counter such threats society is forced to be critical of existing methodologies and processes, to be innovative, and to seek new capabilities that will improve its chances of survival. Therefore, the paradigm shift itself is borne out of fear as the primary source of tension. In the post-9/11 world the US Government found it necessary to initiate a degree of mobilization and reorganization that was unprecedented since World War II. In particular, the

urgent requirement to protect the public from terrorist threats focused attention on information systems for identification, surveillance, and intelligence gathering purposes. It was soon realized that due to the enormous quantity of data involved the computer-based information systems would need to be able to assist the human users in the interpretation of the data that they are processing. This requirement has initiated a paradigm shift from computer-based data-processing to intelligent information management.

A secondary source of tension soon arises as further technical challenges and opportunities for the increased exploitation of the new capabilities emerge. This source of tension is not as severe as the primary forces that precipitated the paradigm shift because it is more narrowly focused on the research community and its funding organizations. The additional capabilities that become available tend to be incremental in nature and are therefore perceived to be less disruptive. Even though these complementing innovations may be even more profound in their enabling capabilities, since society is already engaged in a paradigm shift they become part of the mainstream of change and are therefore more readily accepted. In the post-9/11 world these emerging research challenges are related to the development of software methodologies that will improve the versatility and reliability of the automated transformation of data into actionable information and the intelligent management of this information.

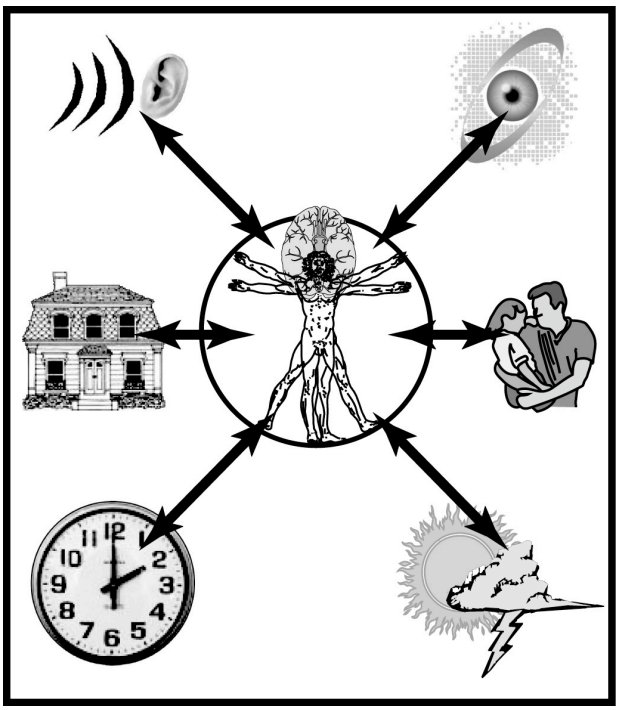


Figure 3: Situated in our environment

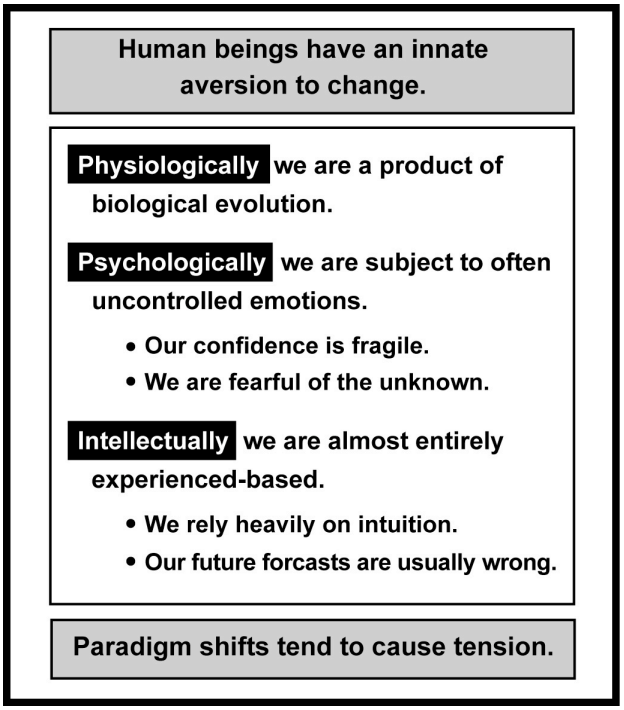


Figure 4: Human resistance to change

Humans are *situated* in their environment

To explore the source of the resistance to change and attendant tensions that inevitably accompany a paradigm shift it is necessary to understand that we human beings are very much influenced by our surroundings. As shown in Figure 3, we are *situated* in our environment not

only in terms of our physical existence but also in terms of our psychological needs and understanding of ourselves (Brooks 1990). We depend on our surroundings for both our mental and physical wellbeing and stability. Consequently, we view with a great deal of anxiety and discomfort anything that threatens to separate us from our environment, or comes between us and our familiar surroundings.

This extreme form of *situatedness* is a direct outcome of the evolutionary core of our existence. The notion of evolution presupposes an incremental development process within an environment that represents both the stimulation for evolution and the context within which that evolution takes place. It follows, first, that the stimulation must always precede the incremental evolution that invariably follows. In this respect we human beings are naturally reactive, rather than proactive. Second, while we voluntarily and involuntarily continuously adapt to our environment, through this evolutionary adaptation process we also influence and therefore change our environment. Third, our evolution is a rather slow process. We would certainly expect this to be the case in a biological sense. The agents of evolution such as mutation, imitation, exploration, and credit assignment, must work through countless steps of trial and error and depend on a multitude of events to achieve even the smallest biological change (Waldrop 1992, Kauffman 1992, Holland 1995, Pohl 1999).

In comparison to biological evolution our brain and cognitive system appears to be capable of adapting to change at a somewhat faster rate. Whereas biological evolution proceeds over time periods measured in millenniums, the evolution of our perception and understanding of the environment in which we exist tends to extend over generational time periods. However, while our cognitive evolution is of orders faster than our biological evolution it is still quite slow in comparison with the actual rate of change that can occur in our environment.

Human resistance to change

Clearly, at least in the short term, the experience-based nature of our cognitive system creates a general resistance to change (Figure 4). The latter is exacerbated by a very strong survival instinct that manifests itself in a desire for certainty as a source of absolute security (Figure 5). Driven by the desire to survive at all costs we hang onto our past experience as insurance. In this respect much of the confidence that we have in being able to meet the challenges of the future rests on our performance in having met the challenges of the past (i.e., our success in solving past problems). We therefore tend to cling to the false belief that the methods we have used successfully in the past will be successful in the future, even though the conditions may have changed. As a corollary, from an emotional viewpoint we are inclined to perceive (at least subconsciously) any venture into new and unknown territory as a potential devaluation of our existing (i.e., past) experience.

This absolute faith in and adherence to our experience manifests itself in several human behavioral characteristics that could be termed limitations. First among these limitations is a strong aversion to change. Typically, we change only subject to evidence that failure to change will threaten our current existence in a significant way. The current paradigm shift from data-centric to information-centric computer software serves as an example. Although the digital computer was originally conceived as a very fast computational machine capable of reducing the

time required for the solution of large numbers of mathematical equations from days to seconds, it soon emerged as a data storage and processing facility. This was mainly due to the need for record keeping accelerated by the growth of commerce and industry driven by major improvements in the ability to travel and communicate over long distances. As a result new opportunities for interaction, leading to cooperation, and eventually collaboration, presented themselves. As the intensity of these activities and the tempo of daily life increased so also did the competition among the human players. However, it did not occur to these players for at least two decades that the functions of the computer could extend beyond the rote storage and processing of data to the representation of information as a basis for automatic reasoning capabilities.

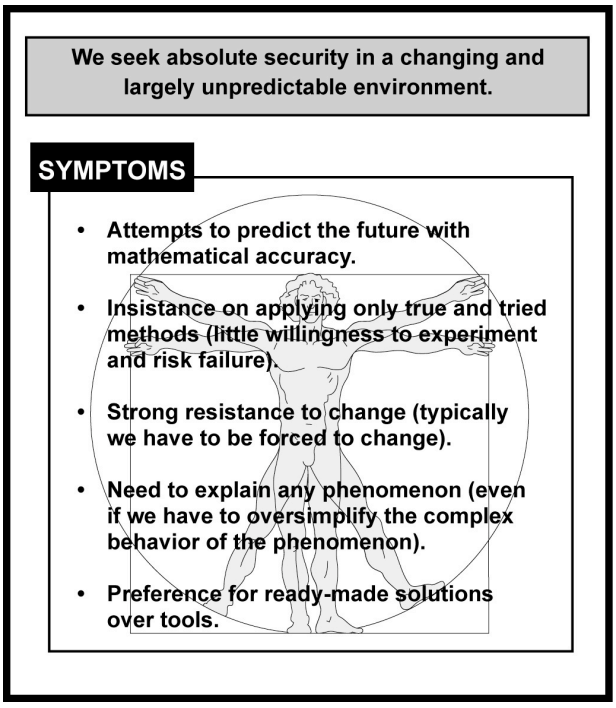


Figure 5: Insecurity as a source of tension

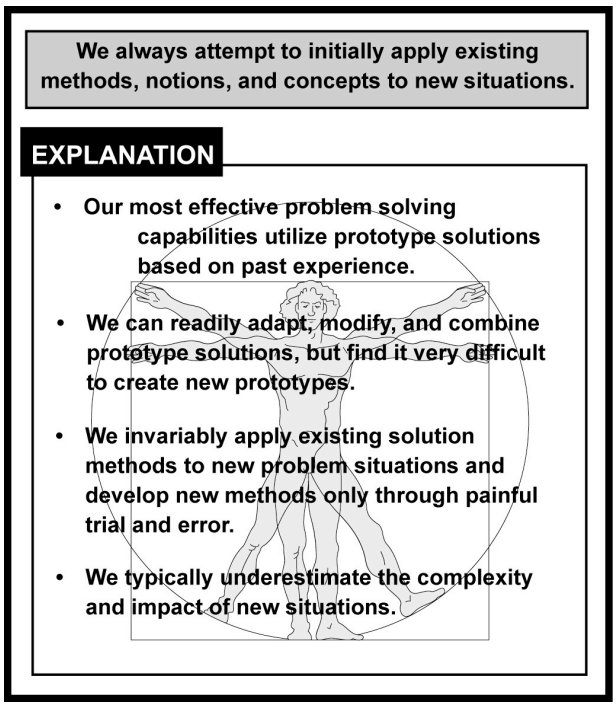


Figure 6: Dealing with new situations

Prior to the events of 9/11 the gradual realization that human-computer interaction could be raised to the level of meaningful collaboration came not as a result of creative discovery, but because the requirement of interpreting the vast amount of computer-stored data simply outstripped the availability of human resources. In other words, it was not the opportunity for using computers in this far more useful role, but the necessity of dealing with an overwhelming volume of data that was gradually persuading computer users to elevate data-processing to information representation in support of automatic reasoning capabilities. Subsequent to 9/11 the absolute necessity of automating at least the lower levels of intelligence gathering and analysis has begun to accelerate the transition from persuasion to conviction. Driven by the realization that the US can no longer afford to depend on the mostly manual processing of intelligence data, key government officials responsible for implementing a vastly improved infostructure have begun to seriously pursue an information-centric software architecture (Cooper 2002).

A second limitation is our apparent inability to resist the temptation of applying old and tried methods to new situations, even though the characteristics of the new situation are actually quite unlike the situations in which the existing methods were found to be useful (Figure 6). This typically casts us into an involuntary experimental role, in which we learn from our initial failures. Examples abound, ranging from the development of new materials (e.g., the flawed introduction of plastics as a structural building material in the 1950s) to the reluctance of the military to change their intelligence gathering and war fighting strategies long after the conclusion of the Cold War era in the 1990s (Wood 2001).

A third limitation is our tendency to view new incremental solutions as final comprehensive solutions. A well known example of such a problem situation was the insistence of astronomers from the 2nd to the 15th Century, despite mounting evidence to the contrary, that the heavenly bodies revolve in perfect circular paths around the Earth (Taylor 1949, 108-129). This forced astronomers to progressively modify an increasingly complex geometric model of concentric circles revolving at different speeds and on different axes to reproduce the apparently erratic movement of the planets when viewed from Earth. Neither the current scientific paradigm nor the religious dogma of the church allowed the increasingly flawed conceptual solution of Ptolemaic epicycles to be discarded. Despite the obviously extreme nature of this historical example, it is worthy of mention because it clearly demonstrates how vulnerable the rational side of the human cognitive system is to emotional influences (Pohl et al.1997, 10-11).

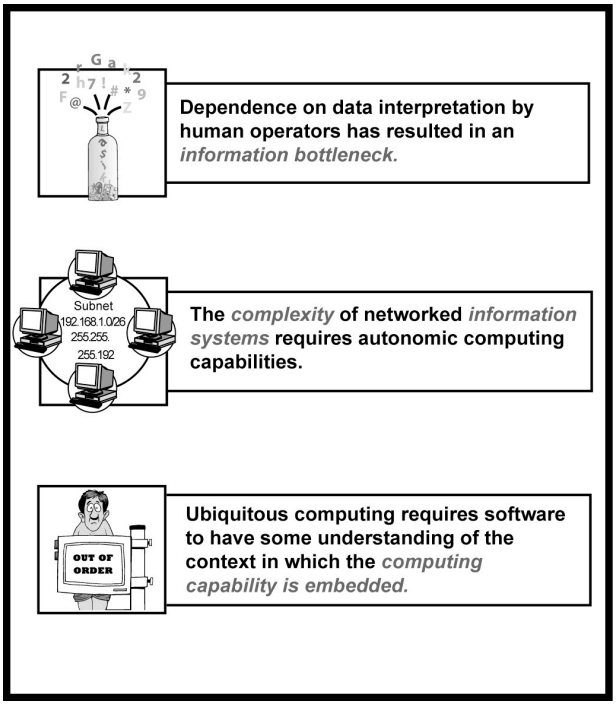


Figure 7: Why do we need context?

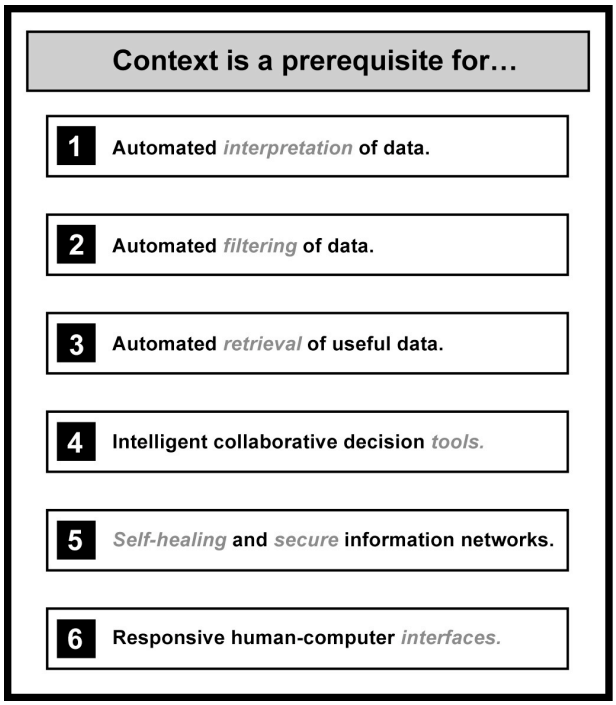


Figure 8: Where should we apply context?

The current paradigm shift

There are essentially two compelling reasons why computer software must increasingly incorporate more and more *intelligent* capabilities. The first reason relates to the current data-processing bottleneck. Advances in computer hardware technology over the past several decades have made it possible to store vast amounts of data in electronic form. Based on past manual information handling practices and implicit acceptance of the principle that the interpretation of data into information and knowledge is the responsibility of the human operators of the computer-based data storage devices, emphasis was placed on storage efficiency rather than processing effectiveness. Typically, data file and database management methodologies focused on the storage, retrieval and manipulation of data transactions, rather than the *context* within which the collected data would later become useful in planning, monitoring, assessment, and decision-making tasks (Figure 7).

The second reason is somewhat different in nature. It relates to the complexity of networked computer and communication systems, and the increased reliance of organizations on the reliability of such information technology environments as the key enabler of their effectiveness, profitability and continued existence.

Increasingly software is being recognized as the vehicle for computers to take over tasks that cannot be completely predefined at the time the software is developed. The impetus for this desire to elevate computers beyond data-processing, visualization and predefined problem-solving capabilities, is the need for organizations and individuals to be able to respond more quickly to changes in their environment. Computer software that has no *understanding* of the data that it is processing must be designed to execute predefined actions in a predetermined manner. Such software performs very well in all cases where it is applied under its specified design conditions and performs increasingly poorly, if at all, when the real world conditions vary from those design specifications. Instead, what is needed is software that incorporates tools that can autonomously adapt to changes in the application environment (Figure 8).

Adaptable software presupposes the ability to perform some degree of automated reasoning. However, the critical prerequisite for reasoning is the situational context within which the reasoning activity is framed. It is therefore not surprising that the evolution of computer software in recent years has been largely preoccupied with the relationship between the computational capabilities and the representation of the data that feed these capabilities. Several decades before the sobering events of 9/11 the theoretical foundations were laid for the transition from data-processing to information-centric computer software. One could argue that the historical path from unconnected atomic data elements, to data structures, relational databases, data objects, object-oriented databases, object models, and ontologies, has been driven by the desire to provide information context in support of automated reasoning capabilities.

Computer software research challenges

An information-centric computer-based environment extends beyond the ability to automatically interpret data into areas that are related to interoperability, flexibility, intelligent analysis and

evaluation capabilities, discovery, and security. Combined with the principles of a service-oriented architecture (SOA) in a distributed implementation, the vision that emerges is profoundly different from the vast majority of existing software systems.

What is suggested is a software environment in which functional capabilities are seamlessly available without the user being aware whether a particular capability is provided by one or more services that are internal to the enabling environment or by an external legacy application that is being accessed through an interoperability bridge. Any data that are being exchanged among internal or external services are shared within the context from which the data derive meaning. The services themselves are not necessarily preconfigured but may be discovered during execution on an as-needed basis. This implies that services are able to automatically configure themselves in conformance with the operational environment and the governing interface protocols.

All of these capabilities are essentially technically feasible today and form part of the notion of a SOA. This notion is by no means new in the software industry, however, it was not until web services came along that SOA principles could be readily implemented (Erl 2005). Initial attempts to provide the required communication infrastructure, such as the Distributed Computing Environment (DCE) and the Common Object Request Broker Architecture (CORBA) did not gain the necessary general acceptance (Mowbray and Zahavi 1995, Rosenberry et al. 1992). Web services and SOA are similar in that they both support the notion of discovery (Gollery 2002). Web services employ the Universal Description Discovery and Integration (UDDI) mechanism for providing access to a directory of web services, while SOA services are published in the form of an Extensible Markup Language (XML) interface.

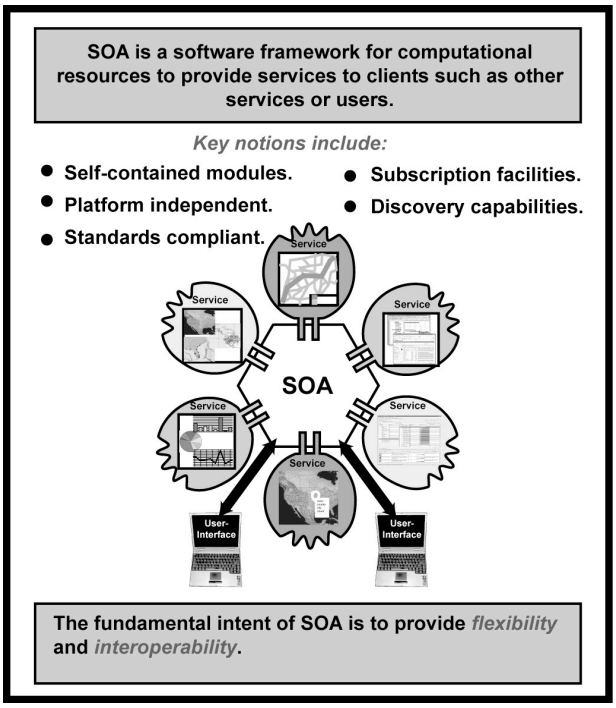


Figure 9: Service-Oriented Architecture (SOA)

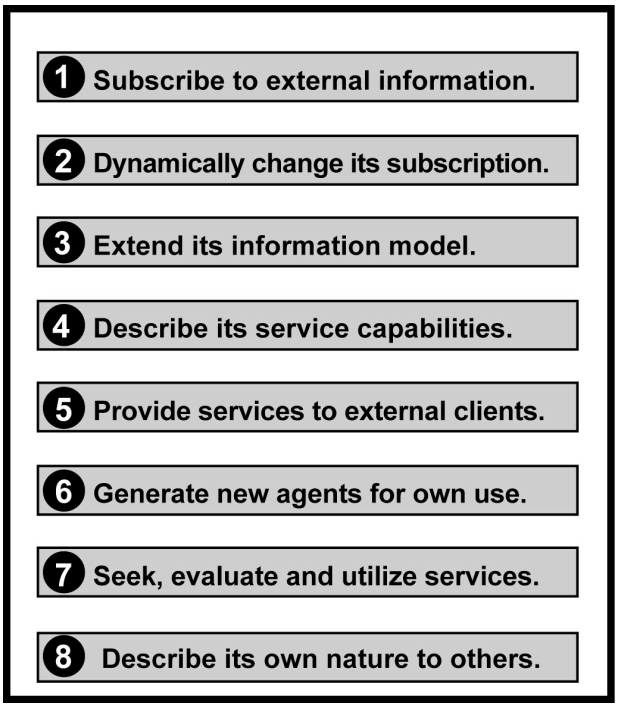


Figure 10: TEGRID capabilities

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 (Figure 9).

In such a software environment any individual service can be designed to meet the following technical specifications:

- Self-sufficiency, interoperability, discovery capabilities, and tools with intelligence.
- Platform independence with self-installing, self-configuring, and self-scaling capabilities.
- For the more domain-centric services the ability to expose functionality through objectified, domain-centric client interfaces and interact asynchronous with clients.
- Adherence to industry-standard patterns (e.g., JavaBeans, Property Change Management, etc.).
- The ability to operate in terms of application-specific notions and concerns.
- Information-centric representation of *context* to support meaningful human-to-agent and agent-to-agent collaboration.

However, as impressive as these interoperability and functional capabilities may be in comparison with existing legacy systems they represent only the beginning of what is implied by an information-centric system environment. The vision is that of a semantic web environment in which autonomous software services with the ability to interpret data imported from other services are able to combine their abilities to accomplish some useful intent. This intent may range from simply finding a particular item of information to the more sophisticated tasks of discovering patterns of data changes, identifying and utilizing previously unknown resources, and providing intelligent decision-assistance in complex and time-critical problem situations.

An example of such an environment is the TEGRID proof-of-concept system, demonstrated by the Collaborative Agent Design Research Center (CADRC) during an Office of Naval Research Conference in 2002 (Gollery and Pohl 2002). TEGRID featured several kinds of web service providers, each implementing a set of operations in support of the exchange of the information that was critical to the functioning of the system. These operations included subscription, information transfer, warning and alert generation, discovery, and assignment. Other operations, less critical to the proper functioning of the system, could have been added for real world implementations.

[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.

TEGRID utilized a number of standard Internet protocols and elements. These elements were combined into executing software entities capable of seeking and discovering existing web services, extending their own information models through the information model of any discovered web service, and automatically reasoning about the state of their internal information models. Each of these software entities consisted of three principal components: a web server; a semantic web service; and, an information-centric application. The web server utilized standard Hypertext Transfer Protocol (HTTP), serving as the gateway for gaining access to other existing web services[2]. The semantic web service (i.e., a web service with an internal information model) was accessed through the web server utilizing standard protocols (e.g., UDDI, SOAP, WSDL, SML). Its purpose was to provide programmed functionality[3]. The addition of an internal information model in a semantic web service allows the storage of semantic level descriptions (i.e., information) and the performance of limited operations, such as reasoning, on these semantic descriptions. The information-centric applications were designed to take advantage of the resources provided by a number of semantic web services, enabling them to reason about the usefulness of each service and support more sophisticated discovery strategies. In particular, the application component was able to construct relationships among the information models of different services, with the ability to integrate services without requiring agreement on a common information model.

Incorporating the three components described above, these TEGRID software entities were minimally equipped to operate in an Internet environment as autonomous software entities, capable of: discovering needed services; accepting services from external offerers; providing services to external requesters; gaining context through an internal information model; automatically reasoning about available information; extending their information model during execution; extending their service capabilities during execution; and, learning from their collaborations (Figure 10). Specifically, they were able to operate as *autonomous* entities and discover the capabilities of other entities. Each entity had a sense of *intent* to accomplish one or more objectives, ranging from the desire to achieve a goal (e.g., maintain situation awareness, coordinate the response to a time critical situation, or undertake a predetermined course of action following the occurrence of a particular event) to the willingness to provide one or more services to other entities.

Near term and longer term research challenges

While TEGRID did demonstrate the potential feasibility of a fully functional information-centric software environment it also identified capability gaps that call for further research. Attempts to work around these technical shortcomings led to some rather primitive solutions that flawed the

[2] Web servers primarily provide access to Hypertext Markup Language (HTML) data sources and perform only simple operations that enable access to externally programmed functionality. However, these simple operations currently form the building blocks of the World Wide Web.

[3] Clients to a standard web service are usually restricted to those services that implement specific predefined interfaces. However, the implementation of web services in the Internet environment allows organizations to provide access to applications that accept and return complex objects. Web service standards also include a limited form of registration and discovery, which provide the ability to *advertise* a set of services in such a way that prospective client programs can find services that meet their needs.

overall achievement of the TEGRID demonstration. These included the ability to share portions of the internal knowledge model of a discovered service with the discovering service and the ability of a service to undertake semantic searches.

Extensible Ontologies: Currently the ontologies of information-centric systems are essentially static in nature. In other words, changes and extensions to the information representation structure cannot be implemented dynamically during the execution of an application. Yet, for several reasons it is highly desirable for ontologies to progressively evolve during the operation of information systems. First, this would allow an information system to automatically extend the granularity of a high level core ontology, representing general concepts and notions, into a biased and much more detailed application-specific domain (Figure 11). Second, the ability to dynamically extend an ontology would allow an information system to capture the representation of new objects and relationships and automatically build them into the existing representation structure, thereby dynamically extending the *context* of the decision-making environment within the computer. Third, the dynamic generation of components of an existing information representation structure appears to be a prerequisite for the automatic extraction of information from unstructured data (e.g., free-format text). Fourth, a promising approach for achieving interoperability among multiple applications, at the information level, is based on the concept of a core overarching ontology that is linked to multiple application-specific ontologies, often referred to as facades (Pohl 2001). The latter are viewed as perspective filters of the core ontology, biased to reflect the native characteristics of a specific application domain. Finally, the ability of a semantic web service to merge part of the ontology of a discovered service with its own internal ontology would be paramount to a low level learning capability (Figure 12).

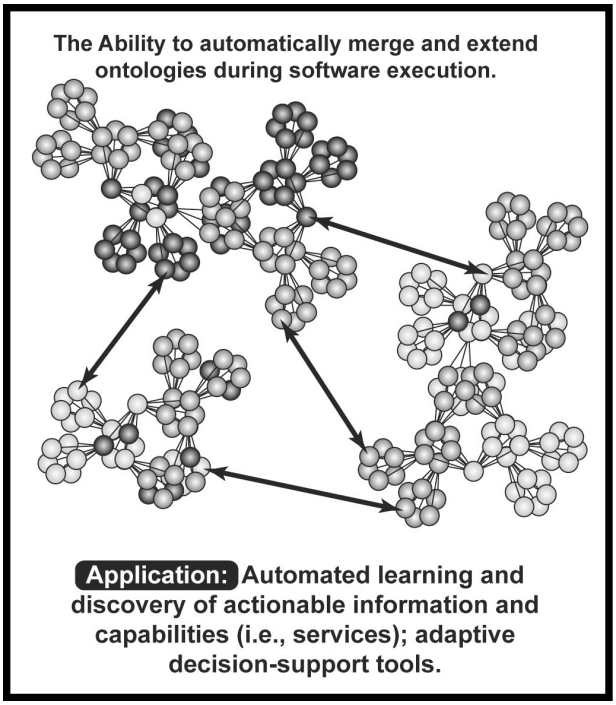


Figure 11: Extensible ontologies

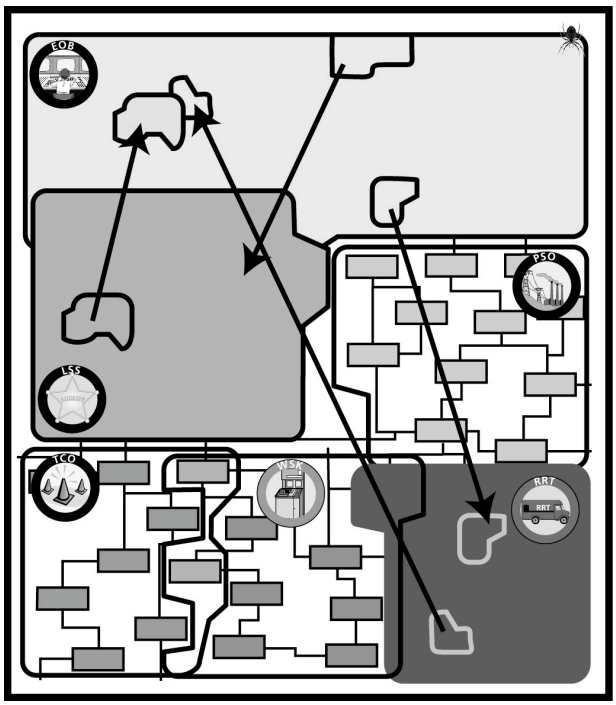


Figure 12: Merging information in TEGRID

Closely associated to the need for dynamically generated ontologies are two related research problems. The first problem deals with the inflexibility of predefined software agents. Typically, the capabilities of software agents are defined at the development stage of an information system. Changes to these capabilities cannot be easily implemented by the user, but normally require the intervention of the software developer. It would be highly desirable for the user or a semantic service to be able to define the capabilities of an agent and have the system automatically create and implement this new agent during normal execution. While some technical capabilities for the dynamic creation of software agents currently exists, these methods are largely limited to predefined functional specifications.

The second problem relates to the capture of information by the system. Ideally, all input should be captured by the system at the point of entry, as information (i.e., within the context of an ontology). In practice, however, much of the input from external sources is in the form of data (e.g., voice recognition, data-centric applications, free text messages, signals, and so on). While several available technologies such case-based classification[4], similarity assessment methods[5], and text-based similarity methods have been applied and tested in diverse application domains their combination in a hybrid data interpretation and information fusion system environment requires further research.

Semantic Search Capabilities: The scope of database query facilities desirable for the kind of semantic services envisioned in a TEGRID environment far exceed traditional database management system (DBMS) functions. They presuppose a level of embedded intelligence that has not been available in the past. Some of these desirable features include: conceptual searches instead of factual searches; automatically generated search strategies instead of predetermined search commands; multiple database access instead of single database access; analyzed search results instead of direct (i.e., raw) search results; and, automatic query generation instead of requested searches only (Figure 13).

A traditional DBMS typically supports only factual searches. In other words, users and applications must be able to define precisely and without ambiguity what data they require. In complex problem situations users rarely know exactly what information they require. Often they can define in only conceptual terms the kind of information that they are seeking. Also, they would like to be able to rely on the DBMS to automatically broaden the search with a view to *discovering* information.

---

[4] Classification techniques inherently concern determining the similarity between objects that share, to varying degrees, a common set of features. Case-based classification works as follows: for a new object or a case to be labeled, a case-based classifier retrieves the most closely matching previously labeled cases from a database of cases, called a *case base*, and assigns the label from the retrieved cases as the label for the new object.

[5] Classifying elements in a complex and multifaceted domain tends to require the amalgamation of multiple classification methods that each excel in different aspects of similarity assessment. The relative performance of each individual method is domain-specific and often difficult to predict without real-world usage. By wrapping the classification methods as distinct similarity assessment methods, each calculating its own similarity score, domain-specific selection and relative weighting of those methods can be achieved.

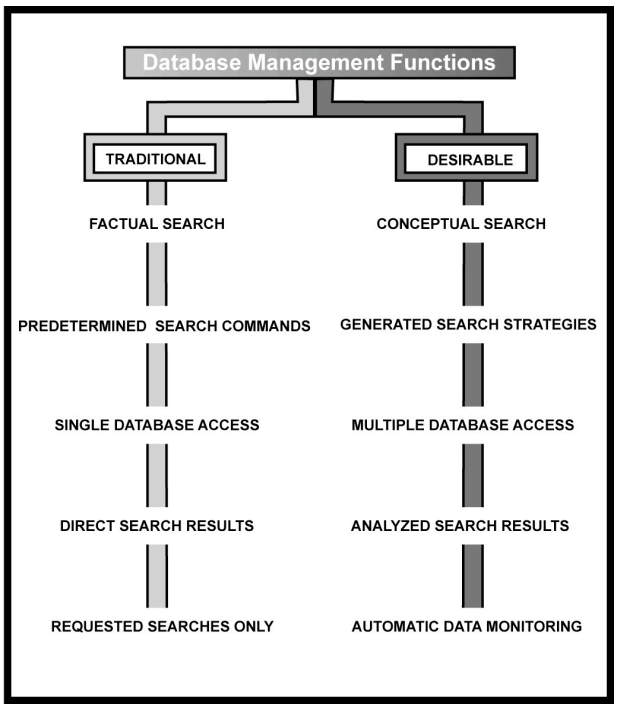


Figure 13: Comparison of directed and semantic search capabilities

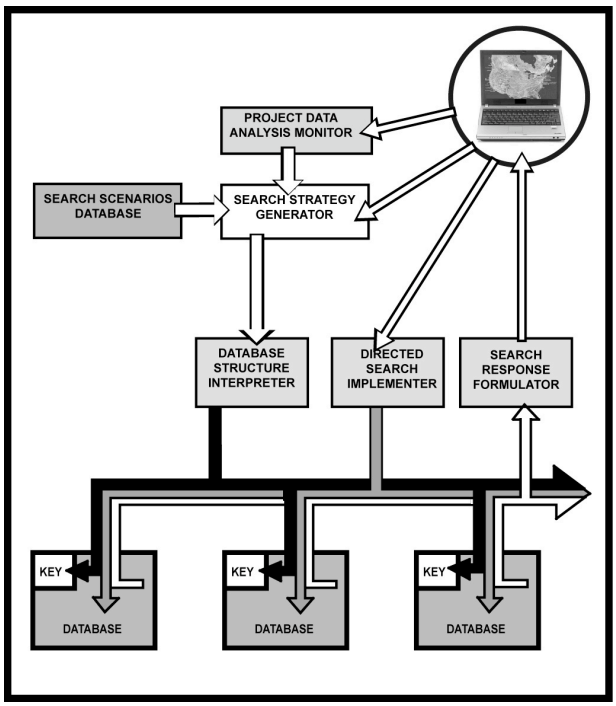


Figure 14: A conceptual semantic search environment

This suggests, in the first instance, that an intelligent DBMS should be able to formulate search strategies based on incomplete definitions. It should be able to infer, from rather vague information requests and its own knowledge of the requester and the problem context, a set of executable query procedures. To facilitate this process the DBMS should maintain a history of past information requests, the directed search protocols that it generated in response to these requests, and at least some measure of the relative success of the previous search operation.

A traditional DBMS normally provides access to only a single database. A knowledge-based decision-support environment is likely to involve many information sources, housed in a heterogeneous mixture of distributed databases. Therefore, through the internal-level database representations discussed earlier, the DBMS must be able to access multiple databases. Using the mapping functions that link these internal representations an intelligent DBMS should be capable of formulating the mechanisms required to retrieve the desired data from each source, even though the internal data structures of the sources may differ widely. Particularly when search results are derived from multiple sources and the query requests themselves are vague and conceptual in nature, there is a need for the retrieved information to be reviewed and evaluated before it is presented to the requester. This type of search response formulation facility has not been necessary in a traditional DBMS, where users are required to adhere to predetermined query protocols that are restricted to a single database.

Finally, all of these capabilities (i.e., conceptual searches, dynamic query generation, multiple database access, and search response formulation) must be able to be initiated

not only by the user but also by any of the computer-based agents that are currently participating in the decision-making environment. These agents may be involved in any number of tasks that require the import of additional information from external databases into their individual knowledge domains.

A conceptual model of an intelligent DBMS interface with the capabilities described above should be able to support the following typical information search scenario that might occur in an integrated and distributed, collaborative, multi-agent, decision-support environment (Figure 14). Queries that are formulated either by the user or generated automatically by a computer-based agent are channeled to a Search Strategy Generator. The latter will query a Search Scenario Database to determine whether an appropriate search strategy already exists from a previous search. If not, a new search strategy is generated, and also stored in the Search Scenarios Database for future use. The search strategy is sent to the Database Structure Interpreter, which automatically formulates access protocols to all databases that will be involved in the proposed search. The required access and protocol information, together with the search strategy, are sent to the Directed Search Implementer, which conducts the required database searches. The results of the search are sent to a Research Response Formulator, where the raw search results are analyzed, evaluated and combined into an intelligent response to be returned to the originator of the query.

The proposition that the DBMS interface should be able to deal with incomplete search requests warrants further discussion. When searching for information, partial matching is often better than no response. In traditional query systems, a database record either matches a query or it does not. A *flexible* query system, such as the human brain, can handle inexact queries and provide best guesses and a degree of confidence for how well the available information matches the query (Pohl et al. 1992 and 1994). For example, let us assume that a military commander is searching for a means of trapping a given enemy force in a particular sector of the battlefield and formulates a *something like* a choke point query. In a flexible query system a *something like* operator would provide the opportunity to match in a partial sense, such as: terrain conditions that slow down the movement of troops; unexpected physical obstacles that require the enemy to abruptly change direction; subterfuge that causes enemy confusion; and so on. These conditions can all, to varying extent, represent *something like* a choke point that would be validated by a degree of match qualification.

Flexible query processing systems are fairly common. For example, most automated library systems have some level of subject searching by partial keyword or words allowing users to browse through a variety of related topics. Even word-processing programs include spelling checkers, which by their very nature search for similar or related spellings. However, even a flexible query system cannot automatically form hypotheses, since the system does not know what to ask for.

The ability to search for *something like* is only a starting point. How can the system be prompted to search for vaguely or conceptually related information? For example, how can the system discover the intuitive connection between a physical choke point, such as a narrow cross-corridor in a mountainous battlefield, and a precision fire maneuver aimed

at concentrating enemy forces in an exposed area. In other words, how can the system show the commander that the precision fire maneuver option can satisfy the same intent as the cross-corridor option? In addition, the system must not overwhelm the commander with an unmanageable number of such intuitive speculations. To discover knowledge it is necessary to: form a hypothesis; generate some queries; view and analyze the results; perhaps modify the hypothesis and generate new queries; and, repeat this cycle until a pattern emerges. This pattern may then provide insight and advice for intuitive searches. The goal is to automate this process with a *discovery* facility that repeatedly queries the prototype knowledge bases and monitors the reactions and information utilized by the decision-maker, until the required knowledge is discovered.

In addition to these two research challenges that are of immediate near term importance as key enabling capabilities during the current transition to an information-centric software environment, there are several other desirable capabilities that are longer term undertakings because they require major research efforts. These include the ability to extract and store the invariant core component of a solution (e.g., plan, design, strategy) in a way that will allow the complete solution to be automatically regenerated in the future (Hawkins and Blakeslee 2004). Any breakthrough in this area, commonly referred to as *hierarchical temporal memory* is likely to have significant impact on the design and capabilities of future decision-support systems. A second area is the automated interpretation of images. With the increased implementation of surveillance technology (e.g., video cameras) there is an urgent need for software systems that are able to continuously monitor and automatically interpret any significant changes in the images that are being recorded.

Hierarchical Temporal Memory (HTM): There is a tendency for us human beings to succumb to the temptation of believing that the goal we have finally reached is the ultimate solution to the problem that we may have been working on for some time. In fact, what appear to be solutions to major problems typically turn out to be mere stepping stones in an endless evolutionary sequence of problem solving and increased understanding.

For example, the computer was initially conceived as a high speed numerical calculator. However, this turned out to be really only the beginning of digital computer technology. It was soon realized that the ability to store and process data (i.e., both numeric and textual) is even more important. This led to new hardware and software solutions in the form of greatly increased storage density devices (e.g., disk drives) and formal data management languages (e.g., relational database management systems and the Standard Query Language (SQL)). As the data storage capacities of the new hardware devices have increased from kilobytes to megabytes to gigabytes it has become increasingly clear that we are essentially storing and analyzing data without context. The context is provided by the users who interpret the results of the data analysis within the context of their experience-based knowledge and understandings. As explained at the beginning of this paper, the complete reliance on the human interpretation of the rapidly increasing quantity of data created a bottleneck. To overcome this human bottleneck, methodologies were devised for constructing context models of real world problem situations in software. These context models are in the form of ontologies that provide an information

structure that is rich in relationships and allows data to be automatically interpreted within the context provided by the ontology.

Again, ontologies are not an ultimate solution but only a stepping stone in the quest for more intelligent computer software tools and services. It could be suggested that the issue is not only related to the representation of context. Software tools, whether intelligent or not, are largely based on the notion of generating solutions based on the interpretation of data in context. Would it not be more productive to find a way of representing and storing solutions (i.e., designs) that can be rapidly retrieved, instead of computing each design from first principles? Such designs could be operational sequences representing entire solutions or, emulating the functions of the human brain's neocortex, only the essential components that can be later quickly assembled into an entire solution (Hawkins and Blakeslee 2004).

The research challenge is twofold, to find a way of extracting the core components of a design and being able to later automatically reassemble the complete design from the core components. Hawkins (2007) and his colleagues at Numenta[6] have developed the Hierarchical Temporal Memory (HTM) theory and a set of tools to emulate some of what they believe to be the functional capabilities of the neocortex of the human brain. In particular, they see the neocortex to be a hierarchical structure like the roots and trunk of a tree. Sensory stimuli enter at the roots level and are hierarchically assembled into progressively more complex and complete configurations (i.e., patterns or designs) at the trunk level. As shown in Figure 15, Hawkins (2007, 23) explains this concept in terms of the hierarchical assembly of an object (i.e., a dog). At the lowest level the key components are spread among many nodes in a fragmented manner. However, at progressively higher levels these components are assembled into the image of a dog.

Specific software research questions that need to be addressed include: What should be the granularity of the partial solution components?; How should the components be assembled?; How can the appropriate components be identified and rapidly retrieved?; How should the solution components be stored?; Will there still be a need for an ontology-like framework to support the rapid identification and retrieval of the components?; and, Should there be a learning component that automatically generates solution components and stores them for future use? A learning capability would certainly be very useful since it would allow the progressive accumulation of a vast knowledge base of partial solution components that can be rapidly adapted and assembled into complete solution.

A reliable HTM capability would have a profound impact on the design of intelligent software tools. Instead of requiring solutions (e.g., a plan) to be developed from the bottom up each time they are required, it would be possible to identify and reassemble an archived past solution. If the solution does not entirely fit the current problem situation it could be modified, much the same way as the human brain modifies prototype solutions and rarely creates a new solution from first principles (Gero et al. 1988, Pohl et al. 1997, 52-55).

[6] Numenta is a California company headquartered in Menlo Park, founded in 2005 by Jeff Hawkins, Donna Dubinsky and Dileep George.

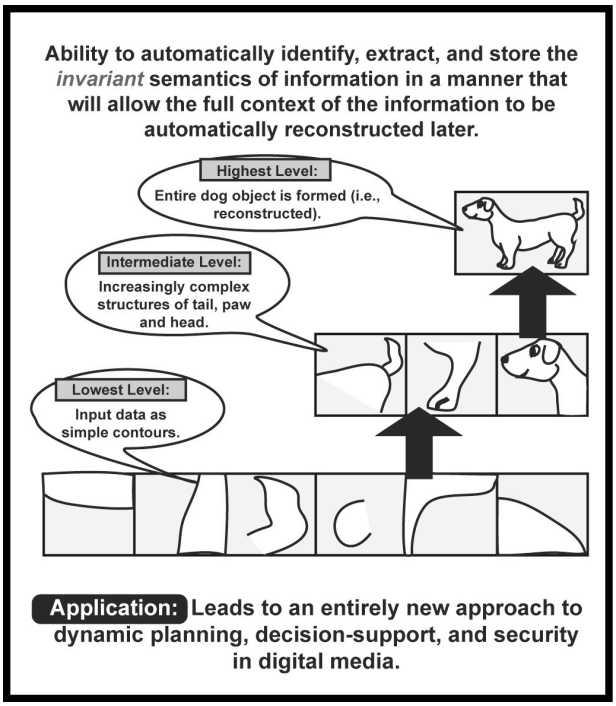


Figure 15: Hierarchical Temporal Memory

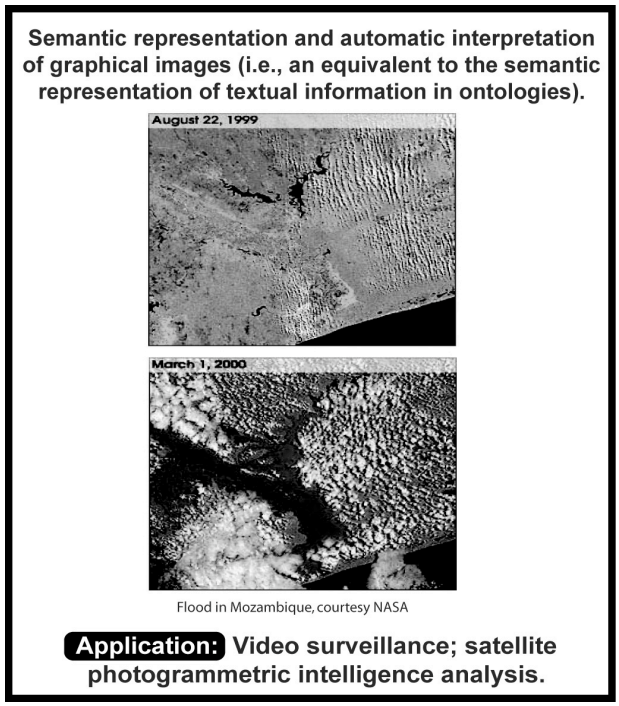


Figure 16: Image interpretation

Automated Image Interpretation: With the increased emphasis on surveillance and personnel identification there is a need for software tools that are capable of automatically identifying the content of video and graphical images. While much headway has been made in recent years in the development of software that is capable of comparing video clips with archived video images and the application of biometric algorithms for personnel identification, this is not sufficient.

The continuous monitoring of video cameras by human observers is cost prohibitive and singularly ineffective. Not only do the capabilities of the human cognitive system degrade over time when required to undertake monotonous tasks, but the reliability of human observers under these conditions is questionable. The research challenge is to develop an ontology-like representation that will support the automatic detection and interpretation of changes in video images. The representation should be of sufficient granularity to detect and interpret changes in a scene, beyond the entry or exit of a person or other object.

The capabilities that have been developed to date are largely focused on video recognition technology in which typically an image is converted into a set of attributes, referred to as an image signature. This provides insufficient context for software agents to reason about smaller changes in a scene that could have significant impact on a particular situation such as a hostage or security surveillance setting. It should be possible to reason about image changes at the same level of granularity as is currently possible with textual data in ontology-based software systems.

Conclusions

We are living in one of the most exciting times in human history for very unfortunate reasons. Information technology is advancing at an accelerated rate and has become the enabler of the individual. Global connectivity combined with inexpensive personal computing devices and powerful software tools are allowing a single person to achieve what was a few decades ago the province of an organization comprising many persons. However, the driving forces of these technological advances are of a sinister nature (Pohl 2004). We are facing unpredictable enemies that are forcing governments to impose security measures that are beginning to seriously impact our everyday activities, particularly in the realm of travel.

Apart from these political forces the technical advances themselves are driving the need for further innovation. For example, global connectivity has greatly increased competition in the commercial arena. Today even the most local market place is within easy reach of the most distant potential competitor. Therefore, simply to survive, there is an increasing need for greater efficiency, continuous vigilance, and tools for planning and re-planning in a dynamically changing environment. These tools must be responsive and adaptive. They must be available to the user when needed, be able to exchange data with external sources, and be capable of seamlessly interoperating with other tools and services. Such capabilities require a level of machine intelligence that cannot be achieved with rote data-processing software.

In this paper the author has attempted to define areas in which research challenges exist and the underlying characteristics of human nature that tend to oppose the necessary motivation for pursuing these challenges. While the tensions created in a paradigm shift that is caused by revolutionary changes in technology can be quite severe and slow down the rate of change, history has shown that it will never succeed in preventing the eventual acceptance and exploitation of the new capabilities.

References

Brooks R. (1990); ‘Elephants Don’t Play Chess’; in Maes P. (ed.) Designing Autonomous Agents, MIT/Elsevier, Cambridge, Massachusetts (pp.3-7).

Cooper S. I. (2002); ‘Homeland Security: A View Through the Eyes of Janus’; Office of Naval Research (ONR) Workshop Series on Collaborative Decision-Support Systems’; Proceedings, September 18-19, Quantico, Virginia (pp. 127-138).

Erl T. (2005); ‘Service-Oriented Architecture (SOA): Concepts, Technology, and Design’; Prentice Hall Service-Oriented Computing Series, Prentice Hall, Englewood Cliffs, New Jersey.

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.

Gollery S. (2002); ‘The Role of Discovery in Context-Building Decision-Support Systems’; Office of Naval Research Workshop on Collaborative Decision-Support Systems, Quantico,

Virginia, 18-19 September (Proceedings available from CADRC Center, Cal Poly, One Grand Avenue (Bdg. 117T), San Luis Obispo, California 93407).

Gollery S. and J. Pohl (2002); 'The TEGRID Semantic Web Application: A Demonstration System with Discovery, Reasoning and Learning Capabilities'; Office of Naval Research (ONR) Workshop Series on Collaborative Decision-Support Systems, hosted by the Collaborative Agent Design Research Center (CADRC) of Cal Poly (San Luis Obispo) in Quantico, VA, September 18-19.

Hawkins J. (2007); 'Why Can't a Computer be More Like a Brain? – Learn Like a Human'; IEEE Spectrum, April (pp. 21-27).

Hawkins J. and D. Blakeslee (2004); 'On Intelligence'; Times Books, Henry Holt and Company, New York, New York.

Holland J. H. (1995); 'Hidden Order: How Adaptation Builds Complexity'; Addison-Wesley, Reading, Massachusetts.

Kauffman S. A. (1992); 'Origins of Order: Self-Organization and Selection in Evolution'; Oxford University Press, Oxford, England.

Mowbray T. and R. Zahavi (1995); 'The Essential CORBA: Systems Integration Using Distributed Objects'; Wiley, New York, New York.

Pohl J. (2004); 'Interoperability and the Need for Intelligent Software'; 6th Office of Naval Research (ONR) Workshop on Collaborative Decision-Support Systems, Quantico, VA, Sep.8-9.

Pohl K. (2001); 'Perspective Filters as a Means for Interoperability Among Information-Centric Decision-Support Systems'; Office of Naval Research (ONR) Workshop hosted by the CAD Research Center in Quantico, VA, June 5-7.

Pohl J. (1999); 'Some Notions of Complex Adaptive Systems and Their Relationship to Our World'; in Pohl J. and T. Fowler (eds.) Advances in Computer-Based and Web-Based Collaborative Systems, focus symposium: International Conference on Systems Research, Informatics and Cybernetics (InterSymp-99), Baden-Baden, Germany, August 2-6, (pp.9-24).

Pohl J., A. Chapman, K. Pohl, J. Primrose and A Wozniak (1997); 'Decision-Support Systems: Notions, Prototypes, and In-Use Applications'; Collaborative Agent Design Research Center, Technical Report CADRU-11-97, Cal Poly, San Luis Obispo, CA 93407 (pp.10-11).

Pohl J., L. Myers and A. Chapman (1994); 'Thoughts on the Evolution of Computer-Assisted Design'; Collaborative Agent Design Research Center, Technical Report CADRU-09-94, Cal Poly, San Luis Obispo, California, September.

Pohl J., J. La Porta, K. Pohl and J. Snyder (1992); 'AEDOT Prototype (1.1): An Implementation of the ICADS Model'; Collaborative Agent Design Research Center, Technical Report CADRU-07-92, Cal Poly, San Luis Obispo, California.

Rosenberry W., D. Kenney and G Fisher (1992); 'Understanding DCE'; O'Reilly and Associates, Sebastopol, California.

Taylor S. (1949); 'Science Past and Present'; Heinemann, London, England (pp.108-129).

Waldrop M. (1992); 'Complexity: The Emerging Science at the Edge of Order and Chaos'; Simon and Schuster, New York.

Wood A. (2001); 'Military Experimentation: Considerations and Applications'; Office of Naval Research Workshop Series on Collaborative Decision-Support Systems'; June 5-7, Quantico, Virginia (pp. 1-8).

INTELLIGENT TOOL TO SUPPORT EVOLUTION OF INFORMATION SYSTEMS

Bogdan Czejdo[1], Thompson Cummings[2], Mikolaj Baszun[3], Janusz Czejdo[4],

[1]Fayetteville State University, Fayetteville, North Carolina
[2]Saint George's University, Grenada
[3]Warsaw University of Technology, Warsaw, Poland
[4]University of Edinboro, Edinboro, Pennsylvania

Abstract

Recently, research and practical efforts intensified in the area of evolution of Information Systems (IS). The need to support IS evolution is caused by a variety of reasons. In this paper we concentrate on evolution of IS data repositories caused by changes of various data hierarchies. We propose an intelligent software tool to accept evolution rules and to use them to support data and application evolution. More specifically, the role of the intelligent tool is to generate database software for evolution of IS data repository and to identify application software modules that would continue to work for new IS data repository.

Key words

Information Systems (IS), system evolution, schema evolution, application evolution

1. INTRODUCTION

Information Systems (IS) currently support and underlie most of our human activities. The importance of maintenance phase of development of IS was stressed by practitioners and by researchers for many years. This effort, however, has not resulted yet in an appropriate tool to support evolving IS. There is still a gap between a relatively informal guidance for building maintainable IS and the support for IS evolution. Therefore, we can see intensification of research and practical efforts in the area of Information Systems (IS) supporting data and application evolution.

The need to support IS evolution is caused by variety of reasons including dynamicity of data sources, changing processing requirements, and using new technologies (Rudensteiner et al. 2000, Eder and Koncilla 2001, Eder et al. 2001, Eder et al. 2002). Therefore, there are many aspects of IS evolution that need to be addressed. In this paper we concentrate on dynamicity of data sources causing evolution of IS data repositories. Our approach is to capture changes of various data hierarchies and use them as rules to implement evolution of IS data repository and evolution of applications. Rather than describing atomic schema changes, our approach is based on changes of larger conceptual schema components referred to as hierarchies. This approach allow us to: (a) specify schema evolution on conceptual level: provide an initial framework to specify transformations, use conceptual structures vs. implementation schemas (UML vs.

relational schemas) allowing e.g. for non-recursive specification in recursive implementation, automate conversion from conceptual to implementation level, provide evolution specification on any level: conceptual or implementation. (b) specify application evolution with references to both conceptual and implementation schema: provide warning when schema evolution causes need for change of application modules, provide recommendation on how to change application modules so that they would work with new schema, support what-if analysis allowing evolution manager to evaluate potential problems with applications if different paths of evolution can be considered, improve quality of applications by making them less dependable on schema changes.

We differentiate between two types of data hierarchies: component hierarchies and classification hierarchies. Each type of data hierarchy can have different representations in both conceptual schema and implementation (relational) schema.

Our approach is applicable to a wide variety of IS that use a global data repository that changes with time. Adjusting to evolving IS data repositories, is of crucial importance for accuracy and timeliness of data analysis.

IS repository most often contains a variety of data hierarchies such as a product structure, a production organization, which are described by diagrams but they might be given in an implicit way. Modeling some of these hierarchies was discussed extensively in the literature (Kim and Seo 1991, Czejdo et al. 1996). In this paper we describe a systematic method for modeling evolution of various types of hierarchies.

This paper is organized as follows. In Section 2 we present the architecture of an Information System with global data repository. In Section 3, we discuss data hierarchy modeling and meta-modeling. Section 4 describes the general framework for intelligent tool for evolution of an information system with data hierarchies. Modeling of evolution of hierarchies is discussed in Section 5. Section 6 illustrates the evolution of hierarchies in data repository on the example of the Spaceship Information System.

2. AN INFORMATION SYSTEM WITH DATA REPOSITORY CONTAINING DATA HIERARCHIES

Our discussion concentrates on evolution of an Information System (IS) with data repository containing various data hierarchies. Architecture of such system is shown in Figure 1. Repository with data hierarchies is processed by applications for data hierarchies. These applications allow, typically, for some data aggregations resulting in the requested data analysis. Such applications can be specific, hierarchy dependent, or independent. Therefore the proper classification of data hierarchies and the proper classification of applications and their relationships with data hierarchies are of crucial importance for our project.

Fig. 1. An Information System with Repository containing Data Hierarchies

As an example of such system let us use Spaceship Information System, designed to maintain and analyze information about different spaceships used by different companies. Let us consider a fragment of data repository for such system as shown in Figure 2.

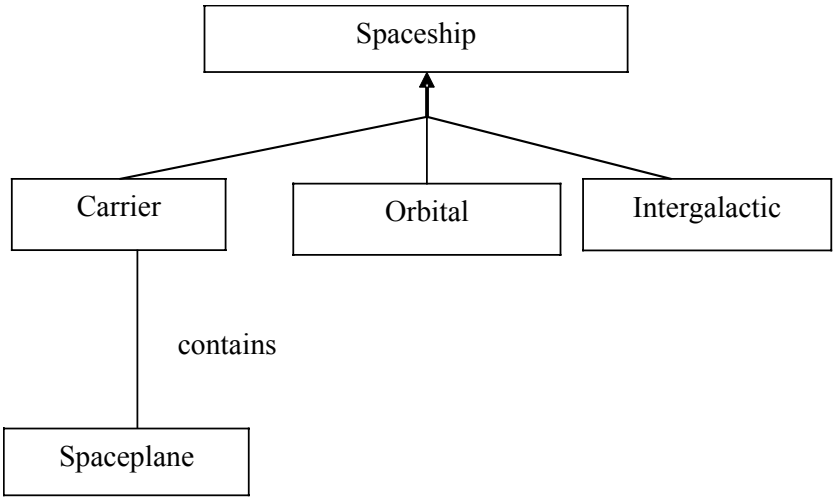


Fig. 2. A schema fragment for Spaceship Information System

There are two important data hierarchies that can be used while modeling data in any information system. The first hierarchy identifies components, subcomponents, etc. of objects.

We typically refer to such hierarchy as a component hierarchy (Figure 3).

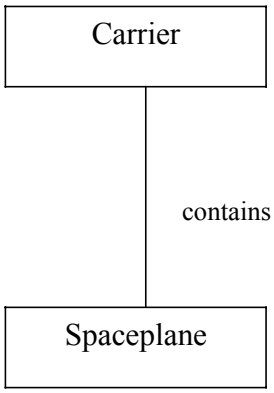


Fig. 3. Component Hierarchy in Spaceship Information System

Another important hierarchy specifies categories, subcategories, etc. of objects. We typically refer to such hierarchy as classification hierarchy (Figure 4).

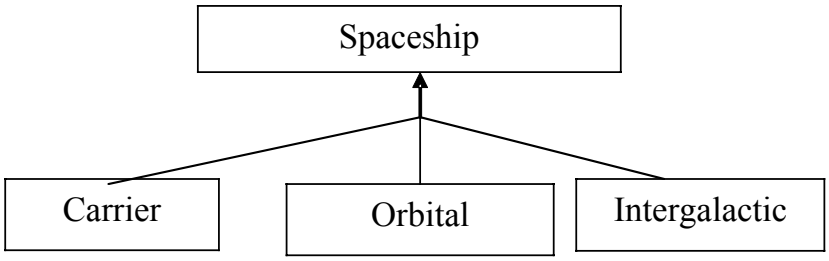


Fig. 4. Classification Hierarchy in Spaceship Information System

3. DATA HIERARCHY MODELING AND META-MODELING

The IS repository can contain one or more component and classification data hierarchies. These hierarchies describe various relationships between data such as a production structure, organizational units (divisions, departments, branches etc.), a structure of products, classification of products, etc.

The component hierarchy graph as shown in Figure 3 is typically based on part-of/consists-of relationship between entity sets. Within component hierarchy we distinguish two subtypes of hierarchies. The first subtype includes component hierarchies that explicitly identify and name all levels. We call this subtype simple component hierarchy. In general, this hierarchy describes typical organizational and/or production structure as shown in Figure 3. Each level corresponds to homogeneous enterprise objects (objects with identical properties) whereas different levels can contain heterogeneous enterprise objects (objects with different properties).

The second subtype of component hierarchy is recursive component hierarchy since it is modeled by a consists_of or is_part_of recursive relationship. This subtype is used to describe the hierarchy where the level names are of lesser importance, the objects are homogenous (they have the same attributes) and the levels can change often. The recursive component hierarchy is typically used to model parts that are built from other parts.

The classification hierarchy is a different hierarchy. It is based on is_a or subclass relationships and is used to describe classifications of entities into types and subtypes. The simple classification hierarchy describes groups with different properties, and all levels are explicitly given in the schema as shown in Figure 4. The recursive classification hierarchy is when the hierarchy is represented by a recursive relationship specifying implicitly hierarchy levels. The recursive classification hierarchy describes groups with identical properties. The meta-model showing classification of hierarchies is shown in Figure 5.

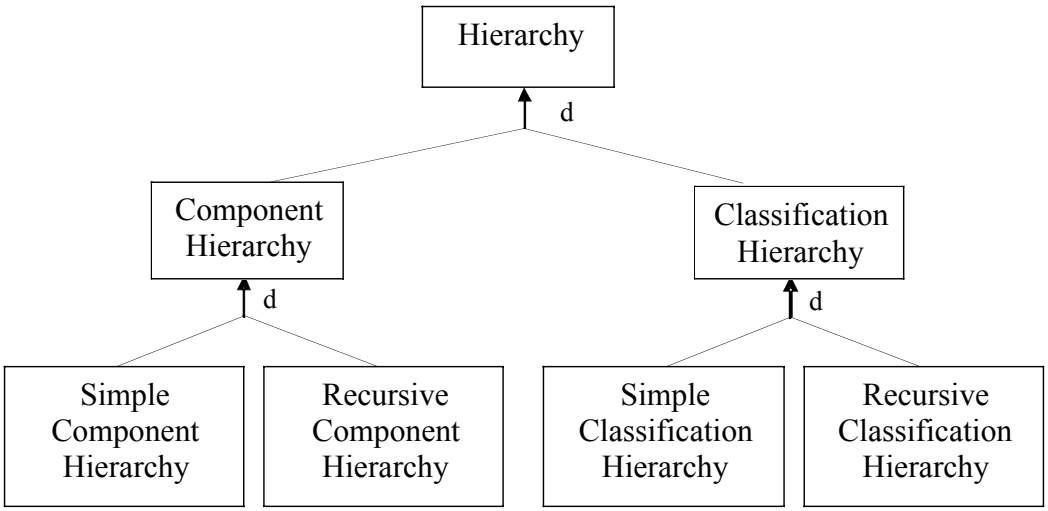


Fig. 5. Meta-model for Data Hierarchies (Classification Hierarchy)

The recursive hierarchies require more discussion. Let us discuss first recursive component hierarchies. Generally, the recursive relationship can be used for modeling of components of homogeneous objects. More precisely, this type of modeling can be appropriate when both atomic (not consisting of any other components) and complex components are homogenous (they have the same attributes) as show in Figure 6a. We refer to such hierarchy as homogenous recursive component hierarchy.

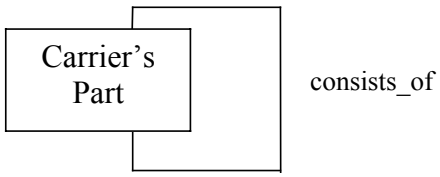


Fig. 6a. Schema fragment for homogenous recursive component hierarchy

The recursive modeling can be also appropriate when atomic and complex components are not homogenous but they are homogenous among themselves as show in Figure 6b. We refer to such hierarchy as partially-homogenous recursive component hierarchy .

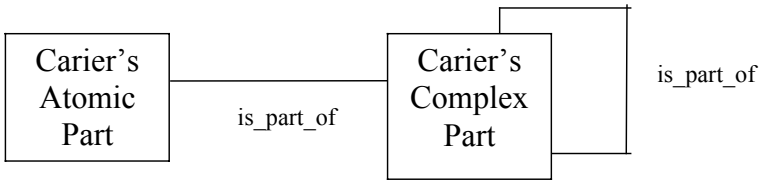


Fig. 6b. Schema fragment for partially-homogenous recursive component hierarchy.

Another way to look at partially-homogenous recursive component hierarchy is to treat it as a homogenous component hierarchy consisting of complex components (homogenous recursive component hierarchy for complex parts) and atomic components.

In the recursive models above, we assumed that the hierarchy graph describing the structure of the Carrier can be very unbalanced with some branches very long and others much shorter. In addition, the number of levels can change when the Carrier is redesigned. In such situation, it was no need to identify and assign any meaning to different levels. However, in some situations we might want to assign a level to a graph represented by a recursive model. We will refer to such a hierarchy as a recursive component hierarchy with levels as shown in Figure 7 for simple recursive hierarchy. Obviously, there is also a possibility to have a partially-homogenous recursive component hierarchy with levels.

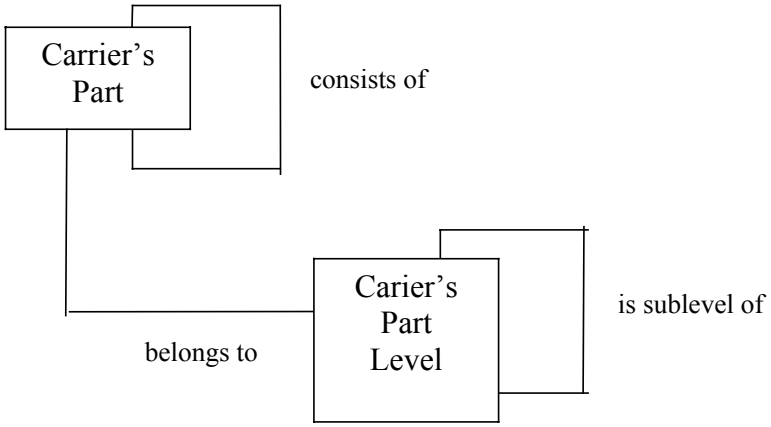


Fig. 7. Homogenous recursive component hierarchy with levels.

It is interesting to notice that the relationship *is_sublevel_of* can be derived from the recursive relationship *consists-of*. Typically, however, it would be convenient to specify level hierarchy explicitly and use it as a constraint for the relationship *consists_of* to verify its correctness.

The discussion for recursive component hierarchy is also generally valid for recursive classification hierarchy. There are only few application and interpretation differences. First, all instances are usually stored in one container separate from the type hierarchy resulting in a partially-homogenous recursive classification hierarchy as shown in Figure 8 for Spaceship Information System. Second, levels are usually not used for classification hierarchies and as a result recursive classification hierarchy with levels will be used very seldom. Third, for classification hierarchy we have a very rare case when subtypes are not named. To avoid unnecessary complications in our discussion, we assume that in such a case the subtypes would be assigned system generated names.

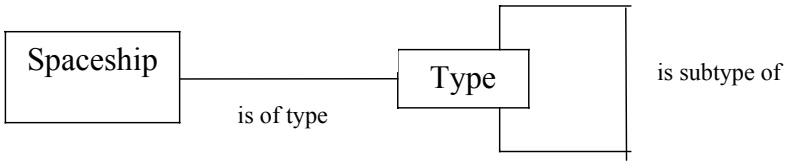


Fig. 8. Hierarchy schema for spaceship recursive classification

4. INTELLIGENT TOOL FOR EVOLUTION OF AN INFORMATION SYSTEM WITH DATA HIERARCHIES

Most IS assume, that database schemas are static and that only the data content changes. However, this assumption doesn't hold in the real world applications. Changes occur frequently. Very often those changes concern data component or classification hierarchy (e.g., assigning an object to another subclass, merging two subclasses, etc.). After such change, queries involving data affected by the change begin to yield incorrect results. Contemporary, most IS are unable to handle such changes, which hinders their functionality.

There is a need for an intelligent tool to better support such changes. In this paper we discuss the tool to allow IS conversion by permitting the user to access all new data items and to maximize access to old data items. Our intelligent tool is based on Meta-Repository and can use many forms of evolution description. The system architecture involving such tool is shown in Figure 9. We will concentrate in this paper on one of the most important tasks, namely, how to consolidate old data repository and the new data repository. But we will also discuss the method of assembling New Application modules.

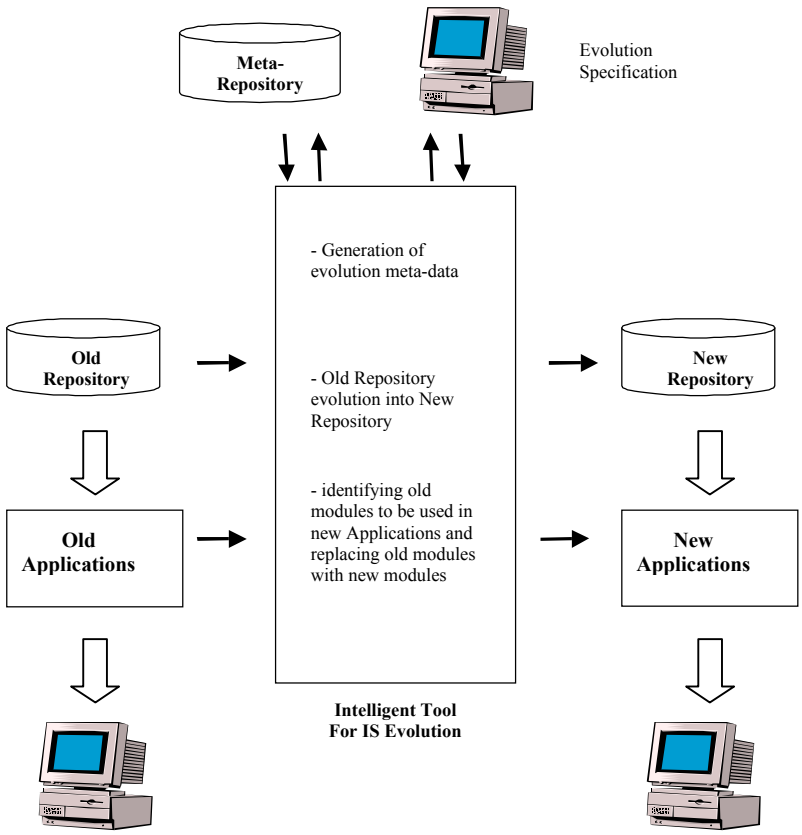


Fig. 9. Intelligent Tool for IS Evolution

Meta-Repository identified in Figure 9 can be used to store all conceptual hierarchies and implementation relational hierarchies as shown in Figure 10. Conceptual hierarchy is related with an implementation hierarchy by the relationship *is_implemented_by* between these two classes. Relational implementations of design schema ~~and~~ are generated usually automatically by a typical conversion of UML diagram into relational model.

Maintaining information about identified hierarchies in two forms in Meta-Repository is crucial for our IS evolution system for several reasons. First, evolution specification can be expressed on any layer conceptual or relational. The evolution can then propagate to other layer automatically. More specifically, this creates a good basis for an intelligent evolution system since it allows a user to specify a conceptual model using the simple non-recursive method rather than more difficult recursive relationships that might be required flexibility reason. Second, evolution specification can be related through hierarchy schemas to application modules' descriptors. Intelligent tool for IS evolution can use this relationships to determine what application modules will work with the new data repository, what applications modules need to be modified (with method of modification suggested), and what new application modules need to be written with

the option of using available templates.

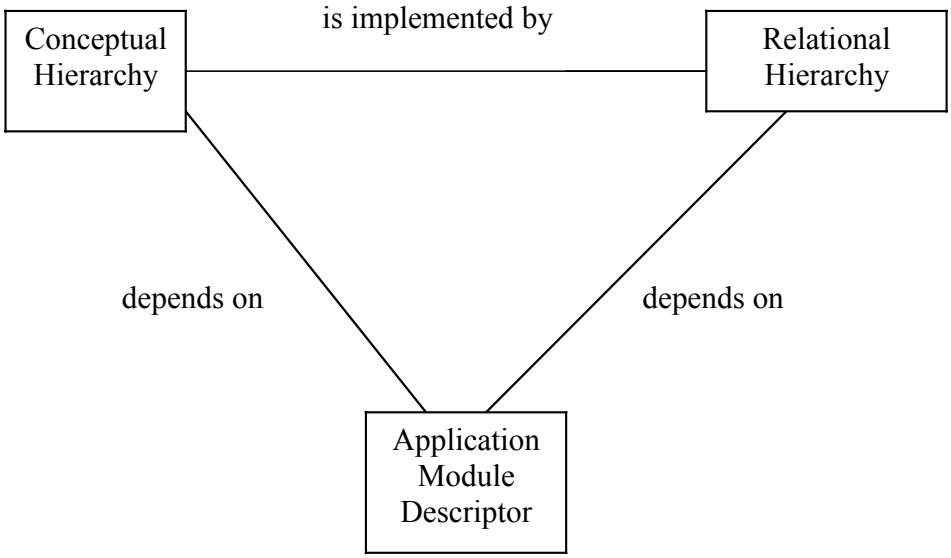


Fig. 10. Model for Meta-Repository

5. EVOLUTION OF HIERARCHIES

The general objective of schema evolution is to provide a new schema that will integrate old and new data and allow a user to view data using a uniform environment. The issue of schema evolution is difficult due to the semantic heterogeneity between old and new schema that appears in the form of schematic and data conflicts among component databases. In this paper we concentrate on conflicts related to incompatibility of data hierarchies discussed in previous sections.

The evolution of hierarchies can be described by transformation rules. The transformation rules can be applied to schema or data instances. The transformation rules for data instances can be set based or individual instance based.

Let us assume that the Initial Data Repository contains the information about spaceships as shown in Figure 2. This conceptual UML diagram contains two hierarchies: classification hierarchy and component hierarchy. Each hierarchy can evolve in various ways.

The evolution can be captured by rules. Most of the rules can be expressed graphically. Each rule consists of two parts: source and target diagrams. Let us first discuss evolution rules for classification hierarchy. Let us consider a simple source schema diagram indicating that all three categories: Carrier, Orbital Spaceship and Intergalactic Spaceship are subcategories of Spaceship as show in Figure 4. The simple evolution of classification hierarchy can require creation of a new category, deleting existing category or category rearrangement within the same level. As an

example let us consider Rule 1 to combine categories Orbital and Intergalactic into one category Main Spaceship. The target schema diagram of Rule 1 is shown in Figure 11 whereas the source schema diagram for Rule 1 is shown in Figure 4. The instance diagrams are not needed in Rule 1 since the instance rearrangements can be generated automatically from schema diagram changes. In general, rules for rearranging of existing categories such as Rule 1 and rules for deleting existing categories usually do not require instance diagrams. However rules for inserting new categories and new objects require instance rule diagrams. The more complex evolution of classification hierarchy can require category rearrangement within different levels.

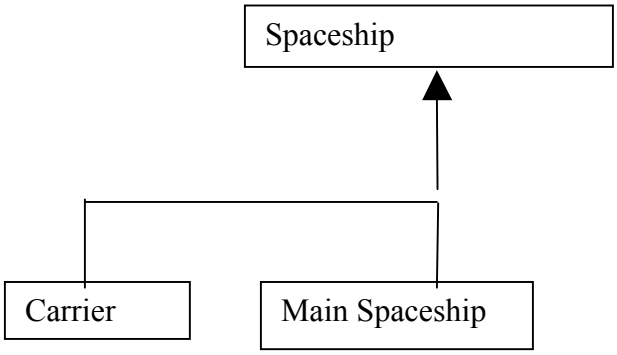


Fig. 11 Target Schema Diagram for Rule 1

Let us next discuss evolution rules for component hierarchy. Let us consider a simple source schema diagram indicating that Carrier objects can have related Spaceplane as show in Figure 3. The simple evolution of component hierarchy could require creation of a new component level. As an example let us consider Rule 2 for creation of a new object set called Intermediate Carrier. The target schema diagram of Rule 2 is shown in Figure 12 whereas the source schema diagram is shown in Figure 3. The instance rules are needed since the new class needs to be populated and relationship instances established. The other rules can involve removing intermediate component level from the middle of hierarchy.

In general the evolution rules can be more complex because of variety of reasons:

1. transformations involving many hierarchies. In most cases the evolution of superposition of hierarchies is equivalent to superposition of evolution of individual hierarchies. However, there are some exceptions.
2. the hierarchies are usually connected to the "non-hierarchical" schema part. The "hierarchical" schema evolution can affect "non-hierarchical" schema and vice-versa.
3. the evolution of hierarchies can coexist with evolution of attributes.

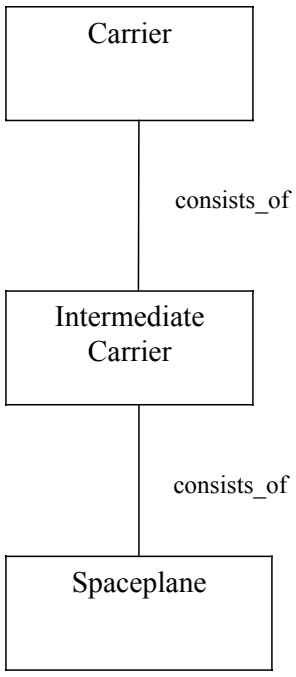


Fig. 12. Target Schema Diagram for Rule 2.

6. EVOLUTION IN SPACESHIP INFORMATION SYSTEM - CASE STUDY

As an example of complex evolution rule let us consider again Spaceship Information System modeled as shown in Figure 2. This model contains a superposition of hierarchies i.e. two combined hierarchies: simple classification hierarchy and simple component hierarchy. Classification hierarchy is based on three subcategories: Carrier, Orbital Spaceship and Intergalactic Spaceship. Component hierarchy is based on consist-of relationship between Carrier and Spaceplane.

Let us discuss a complex Rule 3 of evolution that requires rearranging Spaceplane to become subtype of Spaceship. This rearrangement captures a new company innovation allowing any Spaceplane to be connected to any Carrier. Let us assume that the source schema is similar to Figure 2, except that new entity set Company is added on the top, as shown Figure 13. The target schema diagram is given in Figure 13 and shows the rearranged Spaceplane to become subtype of Spaceship.

Let us see if schema diagrams are sufficient for rule specification. The rule processor will perform the following steps. First, it recognizes that the entity type Spaceplane is removed as a component of Carrier. Second, it recognizes that Spaceplane entity type is created as a new category of Spaceship. Third, in the absence of instance rules, it proceeds with default instance processing which is as follows: generate code to identify all Spaceplane instances, and insert identified instances into Spaceplane new subset. Obviously, marking that this instances are also

of Spaceship type follows.

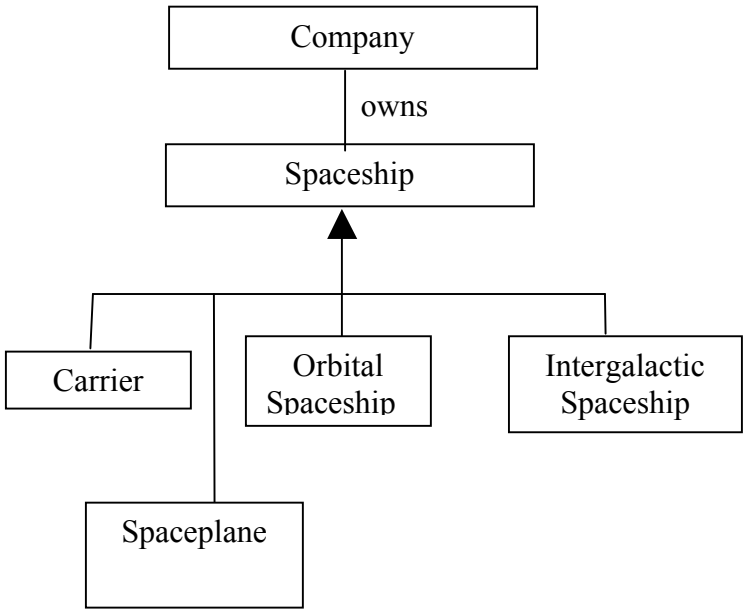


Fig. 13 Target Schema Diagram for Rule 3.

If we stop here, we would be loosing in our analysis information about ownership of spaceplanes Obviously, this is not our intention. Therefore rule processor needs to identify any external relationships i.e. owns between Company and Spaceship instances of Spaceplane type. In this case, these relationship instances can be constructed by joining two initial relationships *owns* and *has_component*. In general, this combined relationship can be generated from one of multiple possible paths. This case, however, will be very rare since hierarchies typically impose unambiguous interpretation.

7. CONCLUSIONS

In this paper, we discussed the problem of the data hierarchy evolution resulting from the changes in the data repositories. We showed how an old data repository for IS can be transformed to new repository by applying appropriate transformation rules. The transformation rules are based on expanding, collapsing of rearranging hierarchies.

Based on transformation rules we designed an intelligent tool to better support evolution of Information Systems. In this paper we discussed the tool to allow IS evolution allowing the user to access all new data items and to maximize access to old data items. Our intelligent tool is based on Meta-Repository and can use multi-form evolution description.

BIBLIOGRAPHY

Czejdo, B., T. Morzy and M. Matysiak (1996); "Hierarchy and Version Modeling", Proc. Of Symposium on Expert Systems and AI, ESDA '96, Montpellier, France.

Rudensteiner E. A., Koeller A., and Zhang X. (2000): *Maintaining Data Warehouses over Changing Information Sources*, Communications of the ACM, vol. 43, No. 6.

Eder J. and Koncilla C., (2001); *Changes of Dimensions Data in Temporal Data Warehouses*, Proceedings of the DaWak Conf. 2001, pp.85-94.

Eder J., Koncilla C., and Morzy T. (2001); *A Model for a Temporal Data Warehouse*, Proceedings of the Intl. OESSEO 2001 Conf. Rome, Italy, pp.86-97.

Eder J., Koncilla C., and Morzy T. (2002): *The COMET Metamodel for Temporal Data Warehouse*, Proceedings of the CAISE 2002, pp.83-99.

W.Kim and J.Seo, (1991). Classifying Schematic and Data Heterogenity in Multidatabase System. IEEE Computer 24(12), 12-18.

Automated Metadata Tagging, Taxonomy Management and Auto-Classification of Information in an Enterprise Environment

Stephen M. Wolfe, Colonel, USAF, MSC, DBA [1]
David S. Sanchez, Major, USAFR, MSC, M.S.H.S.[2]
Simon A. Chapple, Squadron Leader, MBChB, DAvMed, Royal Air Force[1]

[1]711th Human Performance Wing, Brooks City-Base, TX
[2]Air Force Reserve Command Robins AFB, GA

Abstract

The paper addresses the outcomes associated with automated metadata tagging, taxonomy management, and auto-classification of information and how those outcomes positively affect the implementation of an enterprise content management program.

The ability of an enterprise to organize its information has a direct impact on performance. Business rules, such as file plans, provide a structure facilitating retrieval of information by the end-user. In large organizations the ineffective implementation of these business rules can have a detrimental effect on organizational outcomes, in addition to having an adverse impact on efficiency goals.

In this paper the authors discuss using Service Oriented Architecture (SOA) compliant services manifested thru web parts to develop enterprise-wide, highly-relevant meta tags associated with Human Systems Integration in aviation and automatically tag and classify content producing three unique outcomes: (1) an increase in the value of information; (2) the elimination of manual meta-tagging of information and; (3) a dramatic increase in information retrieval precision using faceted searching within Enterprise Search tools.

Introduction and Challenges

In short, transforming information into knowledge to enhance decision-making works only when those with a need for knowledge can find requisite information in a timely manner. According to the Gartner Group, "80% of business is conducted on unstructured information that doubles every three months (i)(White, 2005)." Organizations that do not proactively manage these information assets risk a direct negative impact on their financial bottom line. If the right people cannot find the right information at the right time, decision making has less probability of positioning an organization for success; attendant risks include escalating costs, lost opportunities and decreased productivity.

A 2001 study conducted by Interactive Data Corporation entitled "Quantifying Enterprise Search" yielded the following information regarding the time employees spend on information retrieval (ii)(IDC, 2002):

- 25% of their time (9.5 hours per week) they are searching for information;

- 15% of their time they are duplicating information they cannot find;
- Up to 50% of the time searchers cannot find the information they are seeking and;
- 40% cannot find the information they need to do their jobs.

For an organization of 1,000 knowledge workers, this conservatively translates into a cost of $6M per year spent recreating information that already exists, or searching for information that is either non-existent or is available but simply cannot be found (iii)(IDC, 2002). Enterprise search however is only one piece of the puzzle.

Information and Records Management policy is no longer just a financial compliance issue. It impacts vastly different industries that need to document compliance for a wide range of regulatory bodies, demonstrate multi-national legal compliance, and illustrate a comprehensive audit framework. Developing the policy, processes, deployment, and management of a records management solution involves strong commitment by organizational leadership. In most organizations, electronic content is unmanaged at the enterprise level and for many, managing electronic content is no longer an option, it is an operational requirement.

In light of these new operational requirements, enterprise content management directors are asking themselves:

- "How can we force governance at the desktop?";
- "How do we get our staff to upload appropriate information to the property fields of every document that they create to enable any end user to retrieve that information at a later time?" and;
- "As we troll through terabytes of data, how can our staff retrieve information in a faceted way and deliver value to the organization while enhancing the end-user experience?"

The Root of the Problem

Transforming raw information into actionable knowledge requires information awareness. Information retrieval can occur via a search engine or browsing a virtual file folder for its content; both processes involve the use of metadata, a.k.a. data about data.

For the purpose of this discussion we will focus on two types of metadata: syntactic and semantic. Syntactic metadata describes what the data "looks" like and how it is organized (iv)(NOAA, 2006). Semantic metadata is contextually relevant or domain-specific information about content based on an industry-specific or enterprise-specific custom metadata model or ontology (v)(Sheth, 2003).

Information and Records Management programs and search engines rely on metadata to store and retrieve information. When an individual creates a document they have the option to add subjective semantic metadata to the properties of the document they created. These meta-tags determine not only where a piece of information is filed but also the "retrievability" of that information at a later date. At this point the individual is faced with a "behavioral" issue – "do I or do I not populate 'meta-tags' that will reside within the properties of the document?" If an

individual elects to create meta-tags they are, more often than not, created from a subjective point of view and are incomplete most of the time. If the manual meta-tagging process does not occur this "behavioral" decision significantly reduces the chance that this piece of information will be retrievable at a later date.

Information becomes more actionable when more semantic metadata is present within the properties of a document or record. Even if an organization implements a program focused on 100% compliance in terms of manually creating meta-tags they are still faced with meta-tagging every piece of information that was created prior to the implementation of their new program. While this type of compliance program is a worthy initiative it can be cost prohibitive.

One alternative to the costly and ineffective process of manual meta-tagging is the integration of automated metadata generation, taxonomy management by subject matter experts, and subsequent auto-classification of unstructured information to multiple virtual folders as part of an organizational information and records management program.

Automated Metadata Generation and Taxonomy Management

Automated generation of metadata involves being able to extract both keywords and compound terms from a document or corpus of documents that are highly correlated to a particular concept. If we were to attempt to manually generate highly relevant metadata around the concept of weather as it relates to aviation, we would need to ensure that the source of our metadata was relevant to our selected area of interest.

In figure 1 we have a taxonomy that was developed by ontologists and validated by subject matter experts in the areas of aviation occurrences, human factors in aviation, and phases of flight. When we select the category of "weather" we see that there are 3 keywords and 1 compound term that if present within a document would result in the automatic meta-tagging of the document with the concept of "weather" and the automatic classification of that document to the weather folder.

On the surface, the 4 clues of "icing", "thunderstorm", "turbulence encounter", and "windshear" appear to be highly relevant to the category of weather but without the benefit of a team of meteorologists from the Federal Aviation Administration one would be at a loss to create additional semantics that could facilitate the automatic metadata tagging process and serve as metadata in its own right. To solve the problem of metadata generation a team from the Air Force Research Laboratories' Human Systems Integration (HSI) Directorate indexed documents from the Human Factors Directorates of the Federal Aviation Administration and the Naval Postgraduate School from which highly relevant metadata could be generated.

Selecting the link "Suggest clues for class" (see figure 1) resulted in a set of over 20 additional compound terms, keywords, and acronyms that were related to the concept of weather (see figure 2). "Low level wind", "level wind shear", "microbursts", "windshear training", "pilot weather knowledge", and "hazardous weather" were then added to the original 4 terms for the weather

category, providing us now with 10 highly relevant terms and concepts that, when present within a document, result in automatic metadata tagging of that document and automatic classification to single or multiple virtual folders (see figure 3).

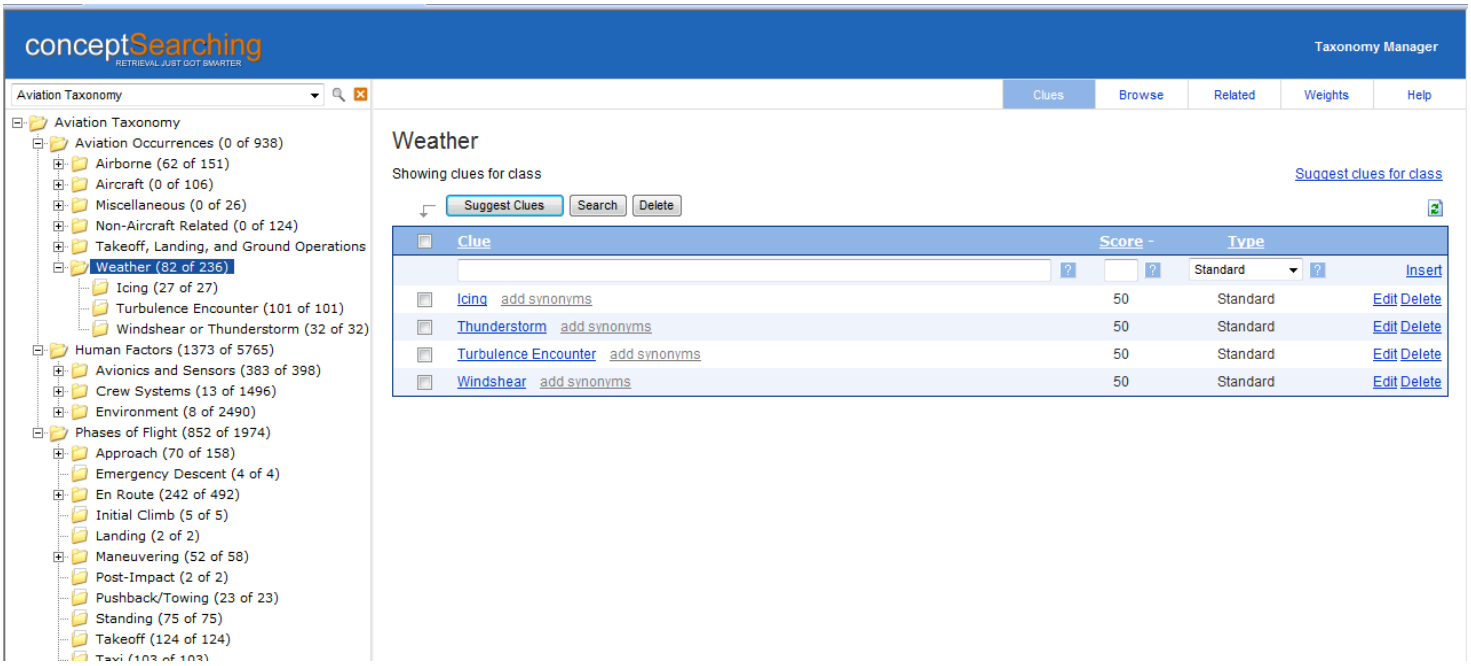


Figure 1: Metadata Associated with Weather as it Relates to Aviation

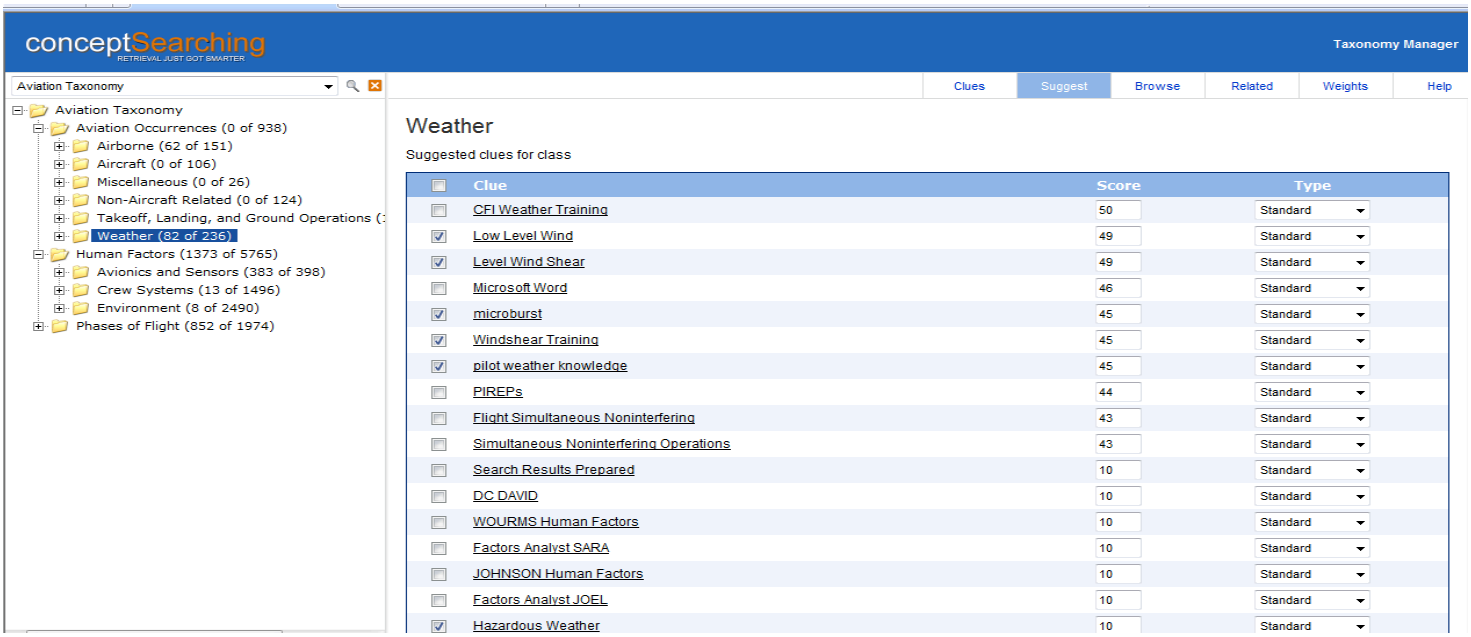


Figure 2: Suggested Metadata for Weather from the Federal Aviation Administration

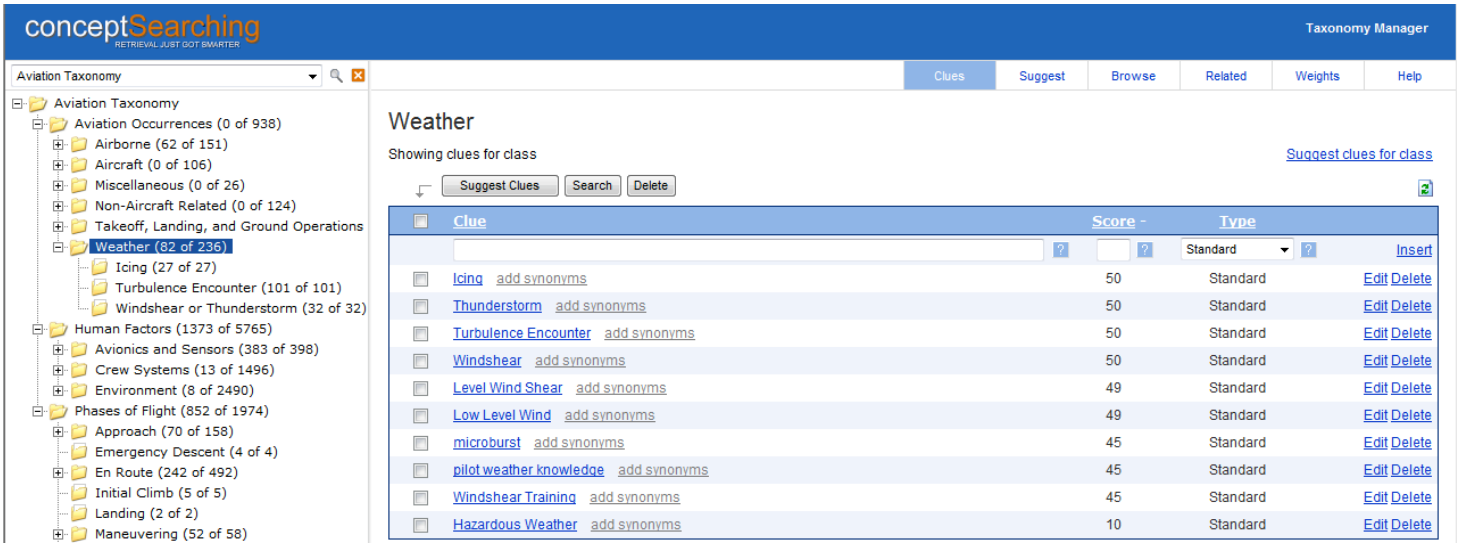


Figure 3: Suggested Metadata added to Original Metadata for Weather

Once highly relevant meta-tags have been created, information residing in document libraries can be tagged automatically with data that is relevant to specific functions, products, and services. In figure 4 we see a set of pdf files contained within a document library on a Microsoft Office SharePoint Server (MOSS). Based upon its semantic content, Newsletter 0102 was automatically meta-tagged with 5 terms or concepts associated with a combined Aviation/Human Factors taxonomy.

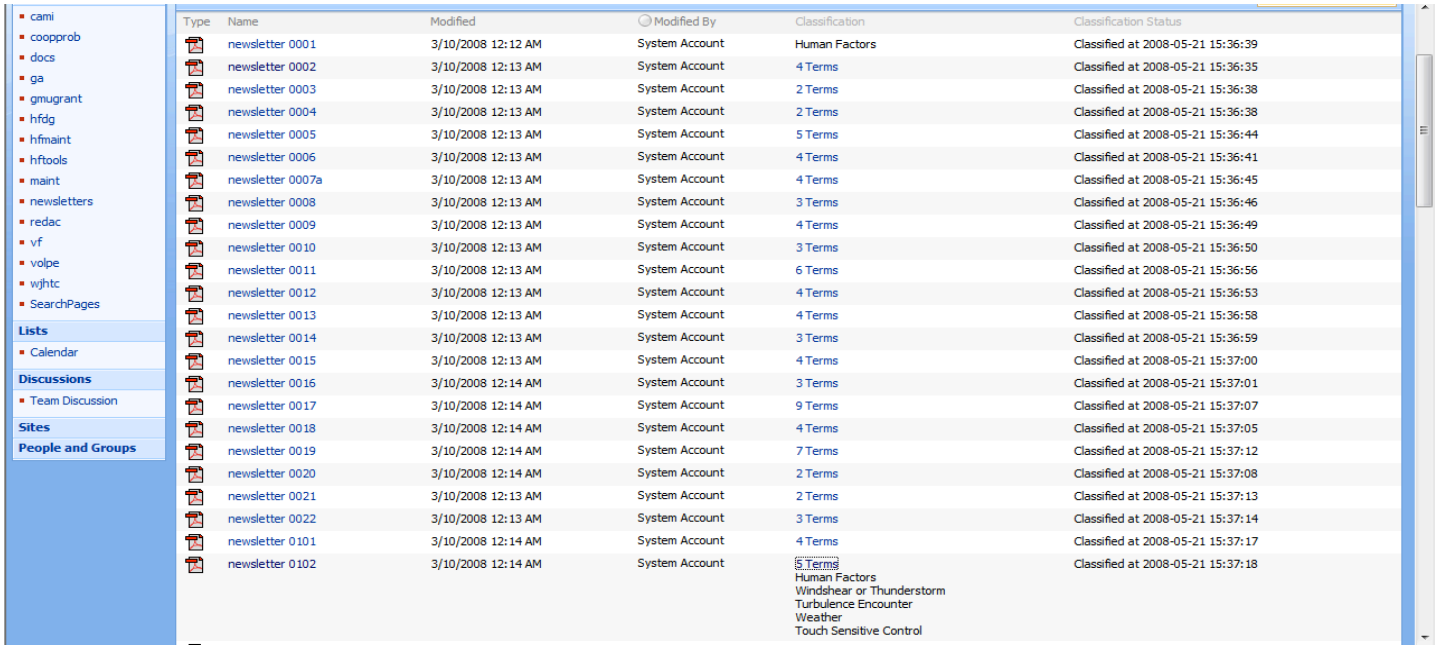


Figure 4: Document Library Content in MOSS

When you open the "properties" of Newsletter 0102 in SharePoint you see the 5 meta-tags that were automatically added to this document (see figure 5). These tags include "human factors", "windshear or thunderstorms", "turbulence encounter", "weather", and "touch sensitive control." These tags were added without having an individual read the document and subjectively create meta-tags based upon their perspective of what the document was about.

Taking this a step further, let us take a look at the actual document and ask ourselves why Newsletter 0102 was tagged with a meta-tag entitled "turbulence encounter?" A keyword search of the term "turbulence" yields no result (see figure 6). When we open taxonomy manager we see that there are other key words and concepts that if present would result in an automated meta-tagging event (see figure 7). When we take the term "windshear" and search for it in the document (see figure 8) we see that its presence triggered the automated meta-tagging event resulting in Newsletter 0102 being tagged with the compound term "turbulence encounter."

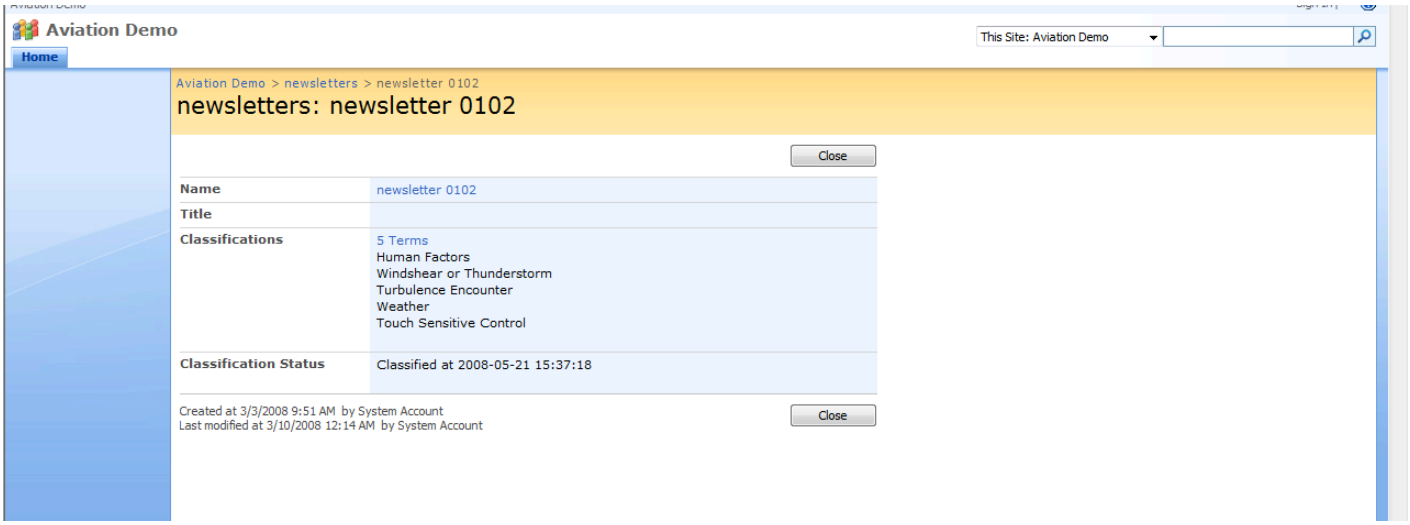


Figure 5: Properties of Newsletter 0102 in MOSS

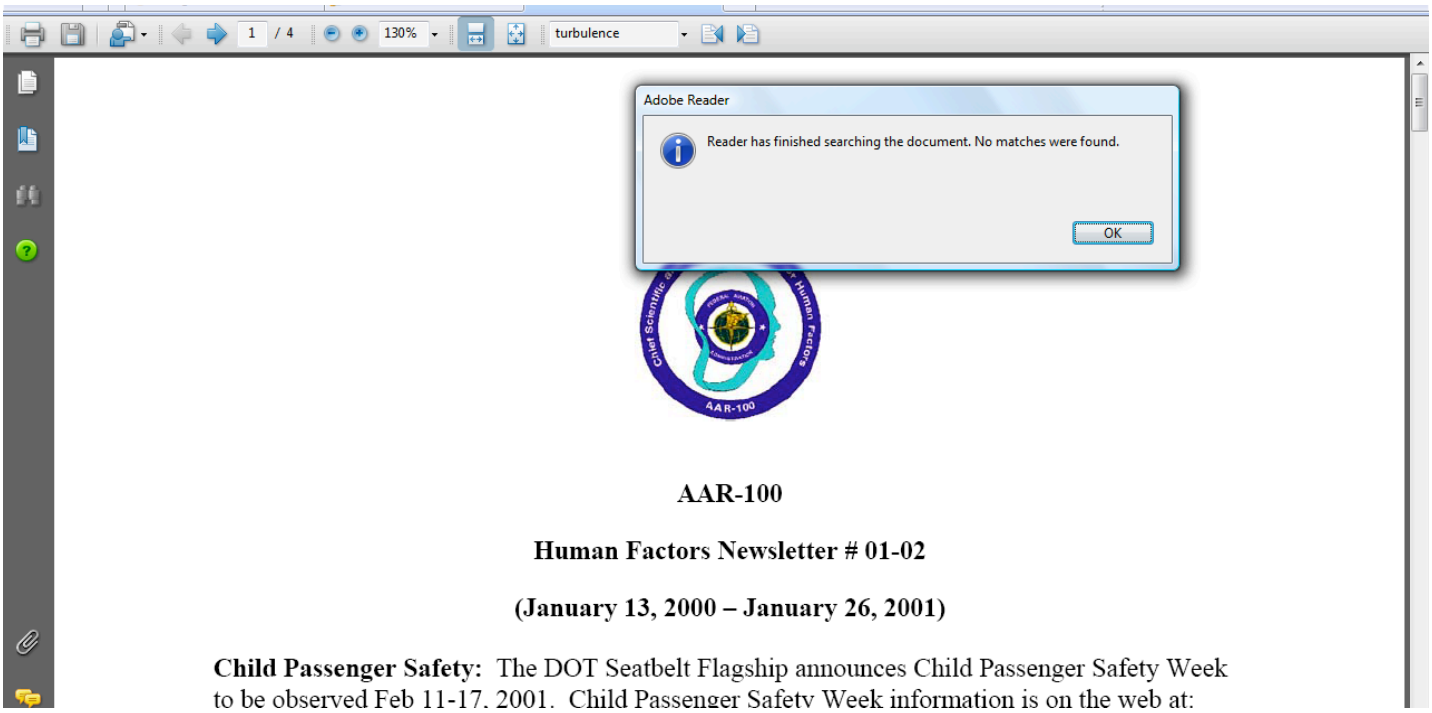


Figure 6: Search for keyword "Turbulence" in Newsletter 0102

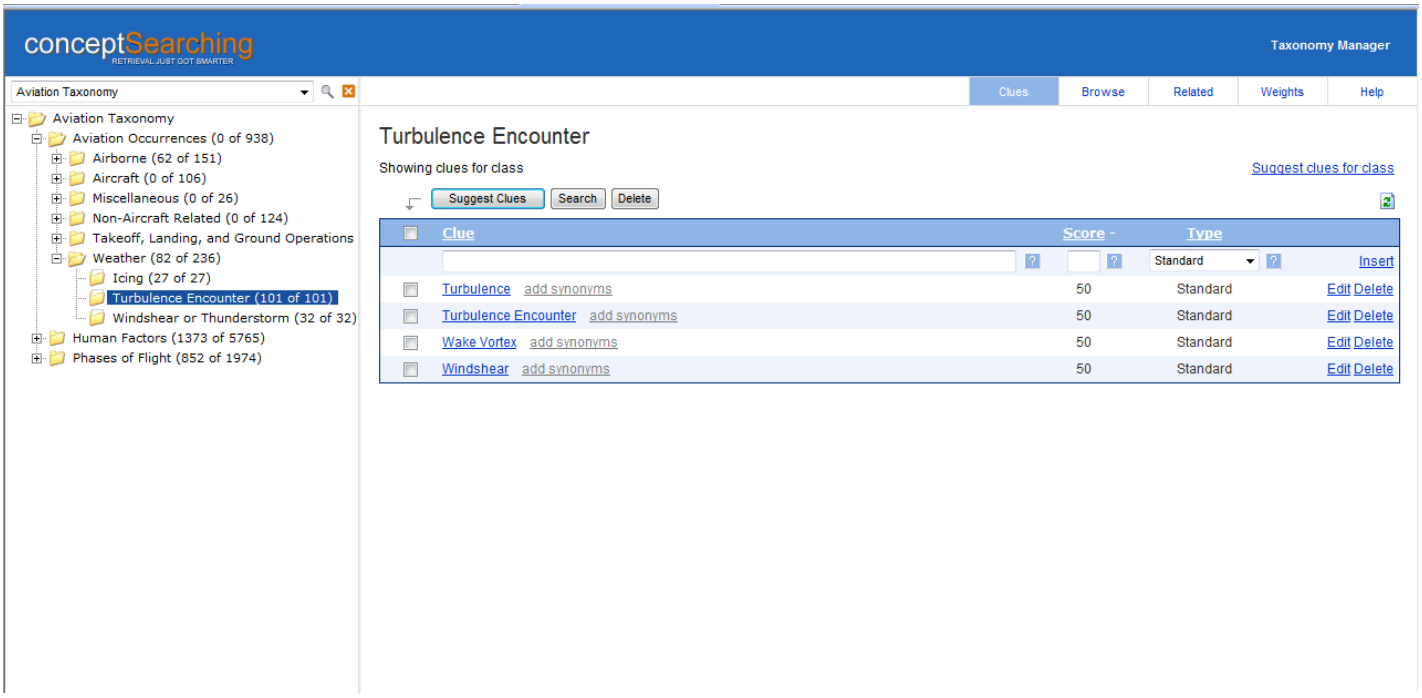


Figure 7: Metadata in Taxonomy Manager relating to the concept of a Turbulence Encounter

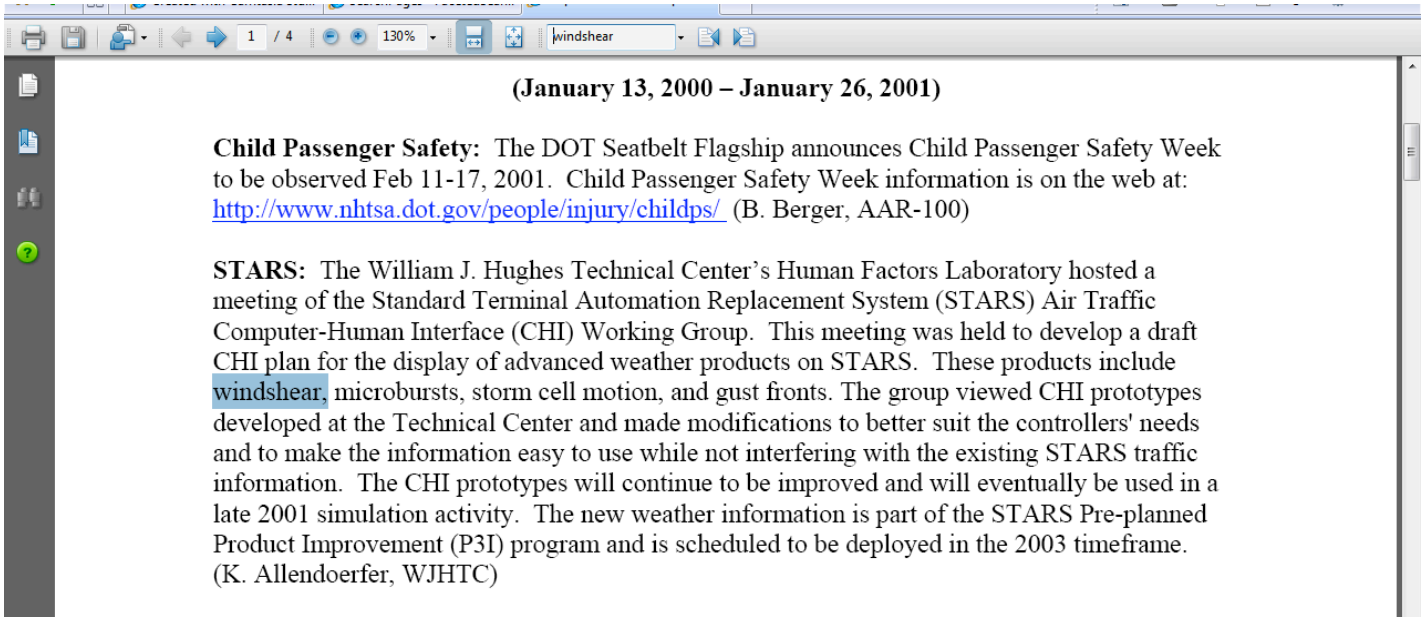


Figure 8: Search for keyword "Windshear" in Newsletter 0102

Auto-Classification and Faceted Searching

A corporate taxonomy imparts a structure from which one can initiate an automated metadata generation process and to which information can be auto-classified and retrieved by searching within a virtual folder structure based upon a corporate taxonomy (see figure 9). In addition, it provides organizations with the ability to cluster enterprise search results by function, product, and geographic region.

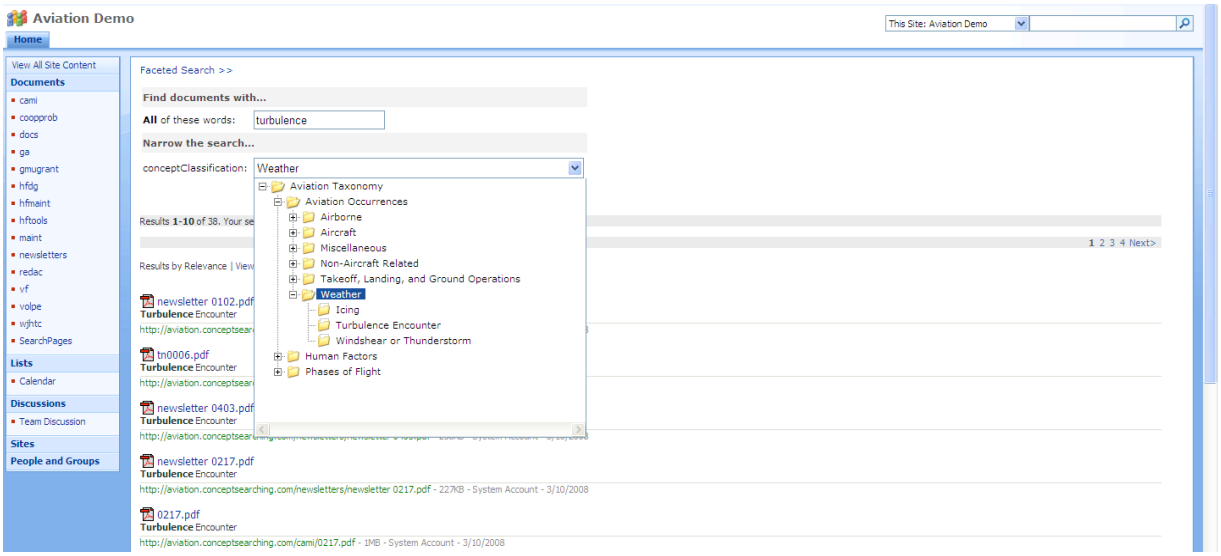


Figure 9: Result set of documents classified to the Weather folder relating to Turbulence

As a contributor to "Code Plex", Microsoft's open source project hosting web site, Leonid Lyublinski leads a team of faceted-search developers who have deployed a set of web parts that provide an intuitive way to cluster and refine search results by category or facet. Categories and facets are implemented using application programming interfaces (APIs) and are stored within the native SharePoint metadata store.

Earlier we identified Newsletter 0102 as a document that was automatically tagged with 5 meta-tags: "human factors", "windshear or thunderstorms", "turbulence encounter", "weather", and "touch sensitive control." In theory we should be able to select two pieces of metadata for this document and Newsletter 0102 should reside at the intersection of those two pieces of metadata.

In figure 10 we conduct a search for the term "weather" using enterprise search within SharePoint. Using metadata contained within Microsoft's propriety index about 101 documents are returned. To the right we see a clustering of search results by facet in addition to a display of a total number of hits within those facets. These facets are dynamically generated based upon the end-user query and come from the corporate taxonomy, in this case aviation classifications.

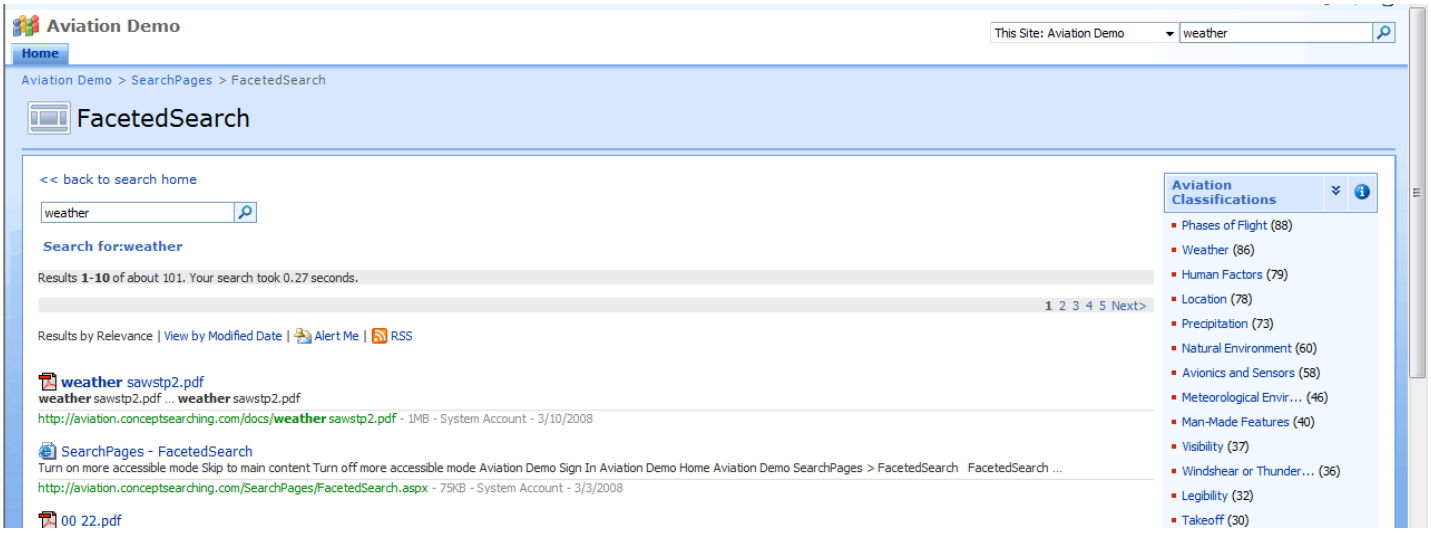


Figure 10: MOSS search results for "Weather"

To refine our search results by facet value we select "windshear or thunderstorms" and in figure 11 our initial result set of 100 documents is culled to 36; Newsletter 0102 appears at the intersection of the two facets, "weather" and "windshear or thunderstorms". In addition to providing a new result set, the facet menu is dynamically updated based on our refined search criteria.

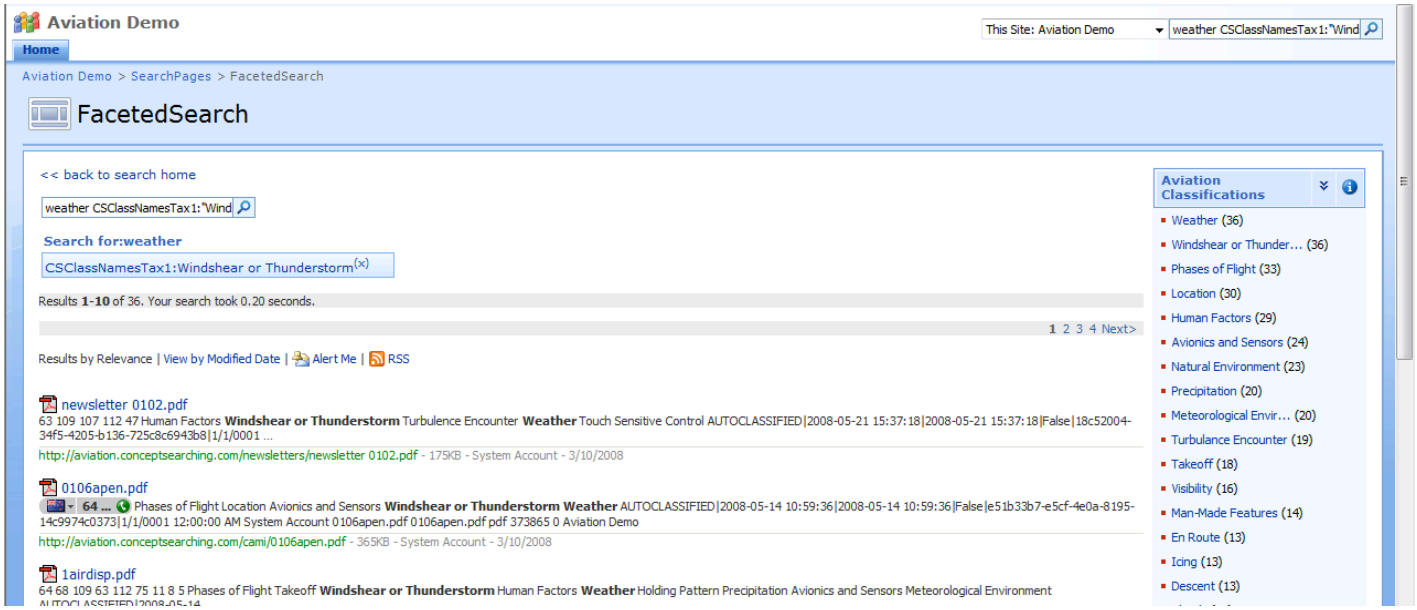


Figure 11: Faceted Search results for "Windshear and Thunderstorms" as a subset of "Weather"

Discussion

Essential to every decision making process is the ability to convert raw information (a.k.a. metadata) into actionable knowledge. An example to illustrate this point follows: at the 711th Human Performance Wing the HSI Directorate is responsible for incorporating a comprehensive strategy into the acquisition process to optimize total system performance, minimize total ownership costs, and ensure that systems are built to accommodate the characteristics of the user population that will operate, maintain, and support the system. All are elements focused on achieving the highest level of integration of human and technology while optimizing human performance.

Despite the existence of a formal Capability Gap Analysis program designed to identify and potentially solve mission capability gaps due to human performance shortcomings, safety events continued to transpire with little warning. In response, the HSI team initiated the first phase of a multiphase project expected to accomplish the following objectives focused on the above issue:

- Identify potential mission capability gaps that may reside within unstructured content located on file servers, mail servers, and document management systems across the Air Force and;
- Provide the Air Force Safety Center with the ability to rapidly organize unstructured information located on flying operations and maintenance file servers as part of its Accident Investigation Board process.

Effective and efficient goal achievement is dependent upon timely access to relevant rich metadata. Building upon foundations started by the Modernization Directorate in the Office of the Air Force Surgeon General, the HSI team expanded its metadata environment to include over 27,000 unique items of metadata (see figures 12 and 13).

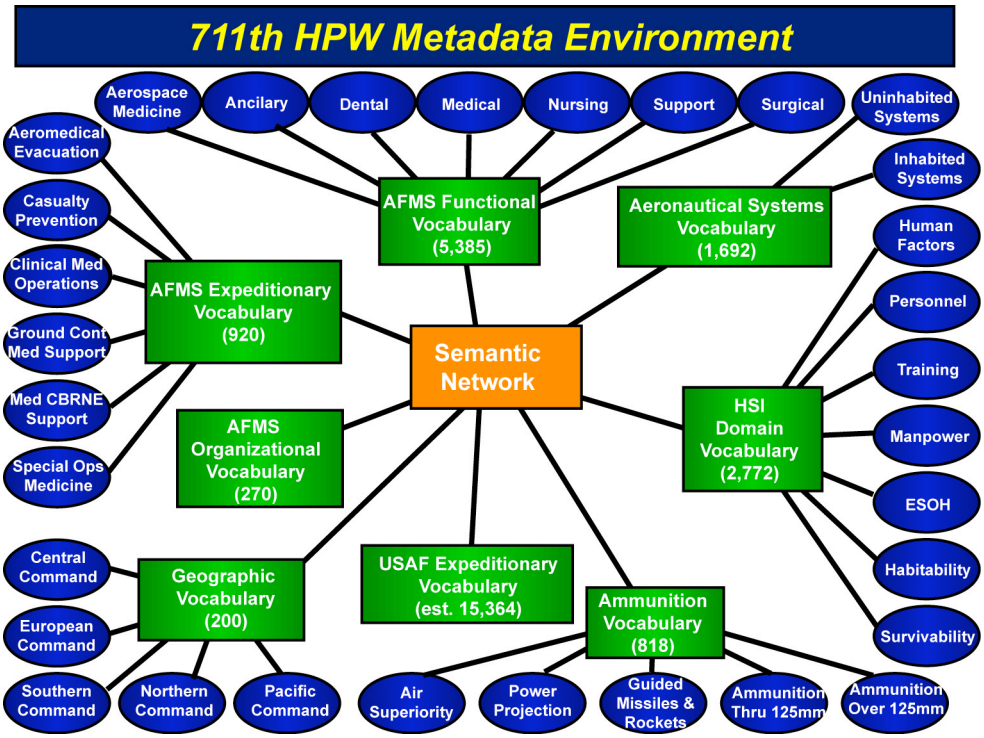


Figure 12: 711th Human Performance Wing Metadata Environment

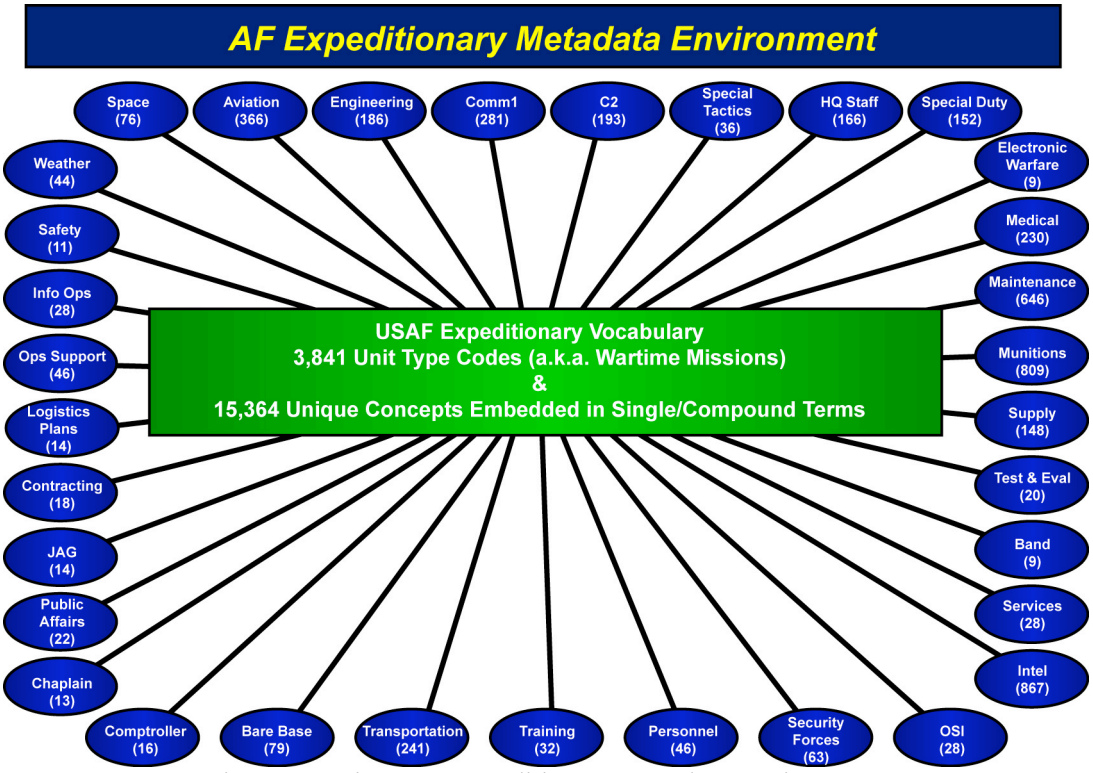


Figure 13: Air Force Expeditionary Metadata Environment

In fiscal year 10 Phase 2 will commence with the deployment of auto-classification capabilities at:

- A flying wing in an effort to identify capability gaps that have not been formally identified or processed and;
- The Air Force Safety Center to reduce the cycle time associated with resolving safety issues that are currently bottlenecked due to the manual classification of information.

Conclusion

The quantifiable benefits of using metadata to harness the power of enterprise information assets cannot be overestimated. Organizational information is an asset that appreciates over time but it easily becomes lost when it becomes unavailable to strategic decision makers. The result is intellectual re-work, sub-standard performance and the inability to access information and knowledge that organizations need to compete and succeed.

While the seeds of innovation are often found within a company's most important asset, its information, it is the inability of an organization to harvest those seeds that leads to a drought of knowledge resulting in decreased productivity, prolonged process timelines, and diminished outcome quality. Automatic metadata generation, taxonomy management by subject matter experts, automatic meta-tagging using corporate taxonomies and auto-classification not only enhance the value of information they also increase its transparency while transforming an information management program from an overwhelming laborious burden into a cost-effective strategic asset.

---

[i] White, C., (2005); Consolidating, Accessing, and Analyzing Unstructured Data, Business Intelligence Network

[ii] IDC, (2002); "Quantifying Enterprise Search"

[iii] IDC

[iv] National Oceanic and Atmospheric Administration, (2006); "Metadata findings for Ocean Observing Systems", Coastal Services Center,

[v] Sheth A., (2003); Semantic Meta Data for Enterprise Information Integration, DM Review, Vol. 13, No. 7, July 2003, pp. 52-54

Towards Joining Systems Science and Engineering of Semantics-Oriented Natural Language Processing Systems

V. A. Fomichov

Department of Innovations and Business
in the Sphere of Informational Technologies
Faculty of Business Informatics
State University – Higher School of Economics
Kirpichnaya str. 33, 105679 Moscow, Russia and IIAS
E-mail: vdrfom@aha.ru and vfomichov@hse.ru
Fax: +7-495-771-3238

Abstract

The paper grounds the necessity of enriching the inventory of formal means, models, and methods destined for designing semantics-oriented natural language processing systems. A special attention is paid to showing the necessity of creating the formal means being convenient for describing structured meanings of arbitrary sentences and discourses pertaining to various fields of professional activity. The possible structure of mathematical models of several new kinds for Systems Science is outlined. The theory of SK-languages (standard knowledge languages) is used for building semantic representations of natural language texts and for representing knowledge about the world.

Keywords

Semantics-oriented natural language processing; semantic representation; theory of K-representations; formal model of a linguistic database; SK-languages

Introduction

Since the pioner works of R. Montague published in the end of the 1960s – the beginning of the 1970s, the studies on developing formal semantics of natural language (NL) have been strongly influenced by the look at structuring the world suggested by mathematical logic, first of all, by first-order predicate logic.

However, it appears that a rich experience of constructing semantics-oriented natural language processing systems (NLPSs) accumulated since the middle of the 1970s provides serious arguments in favor of changing a paradigm of formalizing NL semantics and, with this aim, placing this problem into the focus of interests of Systems Science.

The analysis allows for indicating at least the following arguments in favor of this idea:

1. The first-order logic studies the structure of propositions. However, NL includes also the imperative phrases (commands, etc.) and questions of many kinds.
2. The engineering of semantics-oriented NLPSs needs, first of all, the models of transformers of several kinds. For instance, it needs the models of the subsystems of NLPSs constructing a semantic representation (SR) Semrepr of an analyzed sentence of a NL-discourse as the value of a function of the following arguments: (1) Semcurrent – a surface SR of the currently analyzed sentence E from the discourse D; (2) Semold – a SR of the left segment of D not including the sentence E; (3) Lingbs – a linguistic database, i.e. a collection of the data about the connections of lexical units with the conceptual (informational) units) used by an algorithm of semantic-syntactic analysis of NL-texts; (4) Kb – a knowledge base containing the information about the world. But mathematical logic doesn't consider the models of the kind.
3. NL-texts are formed as a result of the interaction of numerous mechanism of conveying information acting in natural language. Mathematical logic doesn't possess formal means being sufficient for reflecting these mechanisms on conceptual level. That is why mathematical logic doesn't provide sufficiently rich formal tools allowing for representing the results of semantic-syntactic analysis of arbitrary NL-texts. For instance, the first-order logic doesn't allow for building formal semantic analogues of the phrases constructed out of infinitiives with dependent words , of sentences with the word "a notion", and of discourses with the references to the meanings of previous sections and larger parts of discourses.

Due to these reasons, a hypothetical structure of several formal models of the new types for Systems Science is proposed in the next sections. We will not discuss below the question about what kinds of formal means are used in the examples. The purpose of the next sections is only to show the reasonability of undertaking the efforts for constructing the models of the proposed new types.

The Models of Type 1

The models of the first proposed class describe a correspondence between an introduced separate sentence in NL and its semantic representation. The transformation of the inputted sentences into their semantic representations is to be carried out with respect to a linguistic database Lingb and a knowledge base Kb.

Formally, the models of the proposed type 1 describe a class of the systems of the form

(*Linp, Lingbset, Kbset, Lsemrepr, transf, Alg, Proof),*

where *Linp* is an input language consisting of sentences in natural language (NL); *Lingbset* is a set of possible linguistic databases (each of them is a finite set of some interrelated formulas); *Kbset* is a a set of posible knowledge bases (each of them is also a finite set of some interrelated formulas); *Lsemrepr* is a language of semantic representations; *transf* is a mapping from the Carthesian product of the sets *Linp , Lingbset, Kbset* into *Lsemrepr*; *Alg* is an algorithm

implementing the mapping (or transformation) *transf*; *Proof* is a mathematical text being a proof of the correctness of the algorithm *Alg* with respect to the mapping *transf*.
Example.
If Qs1 = ”How many universities in England use the e-learning platform ”Blackboard” for distance education?”, then *transf(*Qs1) can be the string of the form

*Question (x1, ((x1 ≡ Number1(S1)) ∧ Qualitative-composion (S1, university * (Region, England)) ∧ Description1 (arbitrary university * (Element, S1) : _1, Situation(_1, use1 * (Time, #now#) (Agent1, y1)(Process, learning* (Kind1, online))(Object1, certain platform3* (Title, ‘Blackboard’))))))*.

This string includes, in particular, the following fragments: (a) a compund designation of the notion ”a university in England”, (b) a designation of arbitrary element of some set S1 consisting of some universities *arbitrary university * (Element, S1) : _1,* (c) a compund designation of an e-learning platform.

The Contribution of the Theory of K-representations to Constructing the Models of Type 1

The theory of K-representations is an expansion of the theory of K-calculuses and K-languages (the KCL-theory). The basic ideas and results of the KCL-theory are reflected in numerous publications both in Russian and English, in particular, in (Fomichov 1992 – 2005a).

The *first basic constituent* of the theory of K-representations is the theory of SK-languages (standard knowledge languages), stated, in particular, in (Fomichov 1996 – 2005b). The kernel of the theory of SK-languages is a mathematical model describing a system of such 10 partial operations on structured meanings (SMs) of natural language texts (NL-texts) that, using primitive conceptual items as "blocks", we are able to build SMs of arbitrary NL-texts (including articles, textbooks, etc.) and arbitrary pieces of knowledge about the world. The outlines of this model can be found in two papers published by Springer in the series “Lecture Notes in Computer Science” (Fomichov 2002, 2005b). The examples considered in this paper use the class of restricted SK-languages completely defined in (Fomichov 1996).

The analysis of the scientific literature on artificial intelligence theory, mathematical and computational linguistics shows that today the class of SK-languages opens the broadest prospects for building semantic representations (SRs) of NL-texts (i.e., for representing meanings of NL-texts in a formal way).

The expressions of SK-languages will be called below the K-strings. If T is an expression in natural language (NL) and a K-string *E* can be interpreted as a SR of T, then *E* will be called a K-representation (KR) of the expression T.

The *second basic constituent* of the theory of K-representations is a widely applicable mathematical model of a linguistic database (LDB). The model describes the frames expressing the necessary conditions of the existence of semantic relations, in particular, in the word combinations of the following kinds: "Verbal form (verb, participle) + Preposition + Noun", "Verbal form+ Noun", "Noun1 + Preposition + Noun2", "Noun1+ Noun2", "Number designation + Noun", "Attribute+Noun", "Interrogative word + Verb".

The *third basic constituent* of the theory of K-representations is a complicated, strongly structured algorithm carrying out semantic-syntactic analysis of texts from some practically interesting sublanguages of NL. This algorithm, called SemSyn, is based on the elaborated formal model of a linguistic database. The algorithm SemSyn transforms a NL-text in its semantic representation being a K-representation, this algorithm is described in two final chapters of the monograph (Fomichov 2005).

Example. Let T1 = "The antibiotic "Zinnat", produced by the firm "GlaxoWelcome", cures the maladies caused by a coccus flora". Then the algorithm SemSyn constructs the K-representation

*(Situation(e1, producing * (Agent1, certn firm1 " (Name1, "GlaxoWelcome") : x1)*
(Time, current-time)(Product1, certn antibiotic " (Name1, "Zinnat") : x2) ∧
*Situation(e2, curing1 * (Agent1, x2)(Process1, all malady1 * (Cause,*
any flora1 " (Kind1, coccus)))) .

An important feature of this algorithm is that it doesn't construct any syntactic representation of the inputted NL-text but directly finds semantic relations between text units. Since numerous lexical units have several meanings, the algorithm uses the information from a linguistic database and linguistic *context* for choosing one meaning of a lexical unit among several possible meanings.

The other distinguished feature is that a complicated algorithm is completely described with the help of formal tools, that is why it is problem independent and doesn't depend on a programming system. The algorithm is implemented in the Web programming language PHP.

The Models of Type 2

Nowadays there are known computer programs being able to build semantic representations of separate short sentences in NL. However, there are many unsolved questions concerning the semantic-syntactic analysis of the fragments of discourses in the context of the preceding part of a dialogue or preceding part of a discourse. That is why it seems that the engineering of semantics-oriented NLPSs especially needs the models of the next proposed type.

The models of the *second proposed class* are destined for designing the subsystems of NLPSs interpreting a semantic representation of the current fragment of a discourse in the context of semantic representation of the preceding part of a dialogue or preceding part of a discourse. Formally, the models of this class describe the systems of the form

(*Lcontext, Linp, Lingbset, Kbset, Lsem, Lreact, transf, Alg, Proof*) ,

where *Lcontext* is a language for representing the content of the preceding part of a dialogue or a discourse, *Linp* is an input language consisting of underspecified or completely specified semantic representations of NL-expressions (sentences and some fragments of sentences), such NL-expressions can be, in particular, the answers to the clarifying questions of a computer system; *Lingbset* and *Kbset* are (as above) the sets of possible linguistic databases and knowledge bases; the semantic language *Lsem* is destined for representing the deep meaning of the inputs from *Linp* with respect to a semantic representation of the preceding part of a dialogue ot a discourse; *Lreact* is a language for building semantic descriptions of the computer system reactions to the inputted texts; *transf* is a mapping from the Carthesian product of the sets *Lcontext, Linp, Lingbset, Kbset* to the Carthesian product of the sets *Lsem* and *Lreact*; *Alg* is an algorithm implementing the mapping (or transformation) *transf*; *Proof* is a mathematical text being a proof of the correctness of the algorithm *Alg* with respect to the mapping *transf.*

Subclass 1: The models describing the work of a Recommender System.

Since the beginning of the 2000s, a new branch of E-commerce has been quickly developing, this branch is called Recommender Systems (RS). The software applications of this class are destined for consulting the end users of the Internet with the aim of helping to take the decisions about the choice of goods or/and services. The key role in the functioning of many RS plays the interaction with the users by means of Natural language (NL) – English, German, etc.

Consider a particular intrepretation of the components of the models of type 2 destined for the design of NL-interfaces to RS. An input *X1* from *Lcontext* reflects the content (in other terms, the meaning) of the preceding part of a dialogue; an input *X2* from *Linp* is an underspecified (or completely specified in particular cases) semantic representation (SR) of un utterance of the end user; an output *Y1* from *Lsem* is a deep semantic representation of the input *X2* in the context *X1* with respect to a linguistic database *Lingbs* from the set *Lingbset* and to a knowledge base *Kbs* from the set *Kbset*; an output *Y2* from *Lreact* is a semantic description of the computer system reaction to the inputted text with underspecified SR *X2* and deep semantic representation *Y1*. The knowledge base *Kbs* includes a subset of formulas *Userkbs* interpreted as a User Model.

Example. Suppose that an end user of a RS of an Internet-shop applies to the RS with the question Qs1 = ”What models of the cell telephones of the firm Nokia do you have, the price from 300 USD to 450USD?”. Imagine that this question is transfomed in the the semantic representation *X1* of the form

*Question (S1, Qualitative-composition (S1, model1 * (Tech-product, cell-telephone * (Manufacturer, firm1 * (Title, 'Nokia')(Price-diapason, 300/USD, 450/USD))))*.

Having received an answer to this question, the user can submit the next question Qs2 = ”And of the firm ”Siemens”? It is an elliptical question, and the NL-interface to the discussed RS can transform Qs2 into the SR *X2* of the form

*Question (S2, Qualitative-composition (S2,techical-object * (Manufacturer, firm1 * (Title, 'Siemens'))).*

In the English language, the question Qs2 can have one of the following two meanings in the context of the question Qs1:
Meaning 1: The end user wants to get information about all available models of cell telephones produced by the firm "Siemens";
Meaning 2: The end user wants to get information about all available models of cell telephones produced by the firm "Siemens" with the price from 300 USD to 450USD.

That is why the model is to be constructed in such a way that the RS asks the end user to select one of these meanings. A SR of this question, denoted by Clarif-qs, belongs to the language *Lreact.*

Imagine that the end user selects the second meaning. Then, according to the model, the NL-interface to the RS forms Semrepr of the form

*Question (S1, Qualitative-composition (S1, model1 * (Tech-product, cell-telephone * (Manufacturer, firm1 * (Title, 'Siemens')(Price-diapason, 300/USD, 450/USD))))).*

This string expresses the deep meaning of the question Qs2 in the context of the questions Qs1 with SR *X1* and belongs to the language *Lsem.*

Of course, this example represents one of the simplest possible dialogues of a Recommender System with the end user. With respect to the achieved level of studies on NLPSs, many people today are able to elaborate a computer system being able to function in the described way. However, the real dialogues may be much more complex. That is why the practice of designing NLPSs really needs the models of the kind.

The Models of Type 3

The models of the ***third proposed class*** describe the systems of the form

(*Linp, Lpattern, Lingbset, Kbset, Loutput, transf, Alg, Proof*) ,

where *Linp* is an input language consisting of expressions in natural language (NL); *Lpattern* is a language destined for indicating the patterns for extracting information from inputted NL-texts; *Lingbset* is the set of possible linguistic databases (each of them being a finite set of some interrelated formulas); *Kbset* is the set of possible knowledge bases (each of them being also a finite set of some interrelated formulas); *Loutput* is an output language; *transf* is a mapping from the Carthesian product of the sets *Linp , Lpattern, Lingbset, Kbset* to *Loutput*; *Alg* is an algorithm implementing the mapping (or transformation) *transf*; *Proof* is a mathmatical text being a proof of the correctness of the algorithm *Alg* with respect to the mapping *transf.*

Example. The model describes the work of a computer intelligent agent looking in various business texts for the information about any change of the world prices for aluminum (or oil or mais, etc) for N (N = 3, 4,...) or more percents.

Here the language *Linp* consists of NL-texts with commercial information, *Lpattern* contains a semantic representation (SR) of the expression ”any change of the world prices for aluminium (or oil or maize, etc) for N (N = 3, 4,...) or more percents, and *Loutput* consists of SRs of the fragments from inputted NL-texts telling about the changes of the world prices for aluminium (or oil or maize, etc) for N or more percents.

The Models of Type 4

The models of the fourth proposed class are the models of advanced question answering systems, i.e. of intelligent systems being able to find an answer to a request in NL of an end user of a full-text database (of course, it can be Web-based) as a result of semantic-syntactic analysis of NL-texts stored in this database.

Developing the ideas initially stated in (Fomichov 2002), let’s illustrate some desirable properties of formal models reflecting the basic mechanisms of the hypothetical computer intelligent systems of the kind.

Imagine that there is a big city D., and a user of an intelligent full-text database Db1 inputs the question Qs = “Is it true that the ecological situation in the city D. has improved during the year?”, and the date of inputting Qs is Date1.

Suppose that Qs is transformed into the following initial semantic representation *Semrqs1*:

*Question(u1, (u1 _ Truth-value(Better(Ecology(certain city * (Name1, ‘D.’) : y1, Year(Date1)), Ecology(y1, Last-year(Date1)))))) .*

In the expression *Semrqs1*, the element *Ecology* is to be interpreted as the name of the function assigning to the space object *z1* and the year *z2* a statement about the ecological situation in *z1* corresponding to *z2*.

Let’s assume that Db1 has the knowledge base Kb1 including a part Objects-list, and this part contains the K-string *certain city * (Name1, ‘D.’) : v315*. The means that the city D. is associated with the variable *v315*, playing the role of the unique system name of this city. Suppose also that Date1 corresponds to the year 2002. Then *Semrqs1* is transformed into the secondary SR *Semrqs2* of the form

*Question(u1, (u1 _ Truth-value(Better(Ecology(certain city * (Name1, 'D.') : v315, 2002), Ecology(v315, 2001))))) .*

Suppose that there is the newspaper "D. News", and one of its issues published in the same month as Date1 contains the following fragment Fr1: *"The quantity of species of birds who made their nests in the city has reached the number 7 in comparison with the number 5 a year ago. It was announced at the press-conference by Monsieur Paul Loran, Chair of the D. Association of Colleges Presidents".*

Let's consider a possible way of extracting from this fragment the information for formulation of an answer to Qs. The first sentence Sent1 of Fr1 may have the following SR *Semr1a*:

*((Quantity(certn species * (Compos1, bird)(Descr, <S1, P1>)) _ 7) ∧ (P1 _ ∃y1(bird)(Element(y1, S1) ∧ ∃e1 (sit) Is (e1, nesting * (Agent1, y1)(Loc, x1)(Time, 2002)))) ∧ ((Quantity(certn species * (Compos1, bird)(Descr, <S2, P2>)) _ 5) ∧ (P2 _ ∃y2 (bird)(Element(y2, S2) ∧ ∃e2 (sit) Is (e2, nesting * (Agent1, y2)(Loc, x1)(Time,2001))))) : P3 .*

The symbol *certain* is the informational unit corresponding to the word "certain"; *Compos1* is the designation of the binary relation "Qualitative composition of a set"; *P1, P2, P3* are such variables that their type is the distinguished sort "sense of a statement".

Suppose that the second sentence Sent 2 of Fr1 has the following KR *Semr2a:*

*∃e3 (sit) (Is (e3, announcing * (Agent1, x2)(Content1, P3)(Event, certain press-conf : x3)) ∧ (x2 _ certain man *(First-name, 'Paul')(Surname, 'Loran')) ∧ (x2 _ Chair (certn association1 * (Compos1, scholar * (Be, President (any college * (Loc, certn city * (Name1, 'D.') : x4))))))) .*

Here the element *association1* denotes the concept "association consisting of people" (there are also the associations of universities, cities, etc.).

The analysis of the first sentence Sent1 shows that it is impossible to find directly in Sent 1 the information determining the referent of the word "the city". In this case, let's take into account the knowledge about the source containing the fragment Fr1 and about the use of this knowledge for clarifying the referential structure of published discourses.

Imagine that the knowledge base Kb1 of the considered hypothetical intelligent system contains the string

*Follows ((z1 _ arbitr edition * (Title, z2)(Content1, Cont1)) ∧ Associated (z2, arbitr space-object : z3) ∧ Element(w, pos, Cont1) ∧ Sem-class(w, pos, space-object) ∧ No-show-referent (w, pos, Cont1) , Referent (w, pos, Cont1, z3)) .*

Let's interpret this formula as follows. Suppose that: (1) an arbitrary edition *z1* has the title z2 and the content *Cont1*, its title *z2* is associated in any way with the space-object *z3*; (2) the K-string *Cont1* contains the element *w* in the position *pos*, its semantic class is *space-object* (a city, a province, a country, etc.), (3) the text contains no explicit information about the referent of the element *w* in the position *pos* of the formula *Cont1*. Then the argument of the function is the entity denoted by *z3*.

In order to use this knowledge item for the analysis of the fragment Fr1, let's remember that the list of the objects Objects-list (being a part of the knowledge base Kb1) includes the K-string *certain city * (Name1, 'D.') : v315*. Then the system transforms the KR *Semr1a* of the first sentence Sent1 into the formula *Semr1b = (Semr1a ∧ (x1 _ v315))* . The means that at this stage of the analysis the information extracted from Sent1 is associated with the city D.

Assume that the knowledge base Kb1 contains the knowledge items

$$\forall z1(person)\ \forall c1(concept)\ Follows(Head(z1, arbitrary\ association1\ *\ (Compos1, c1)), Is(z1, c1))\ ,$$

$$\forall z1(person)\ Follows\ ((\ z1\ \_\ President((arbitrary\ univ : z2\ \vee\ arbitrary\ college : z3))),\ Qualification(z1, Ph.D.))\ ,$$

and these items are interpreted as follows: (1) if a person *z1* is the head of an association of the type 1 (associations consisting of people), the concept *c1* qualifies each element of this association then *z1* is associated with *c1* too; (2) if a person *z1* is the president of a university or a college then *z1* has at least a Ph.D. degree.

Proceeding from the indicated knowledge items and from Semr2a, the system builds *Semr2b = (Semr2a ∧ (x1 _ v315))* and then infers the formula *Qualification(x2, Ph.D.),* where the variable *x2* denotes Monsieur Paul Loran, Chair of the D. Association of Colleges Presidents.

Let Kb1 contain also the expression

$$Follows\ (\exists e1(sit)\ Is(e1, announcing\ *\ (Agent1, arbitrary\ scholar\ *\ (Qualif, Ph.D.))(Kind\text{-}of\text{-}event, \neg\ personal\text{-}communication)(Content1, Q1)(Time, t1)), Truth\text{-}estimation(Q1, t1, < 0.9, 1>))\ ,$$

Interpreted as follows: if a scholar having a Ph.D. degree announces something, and it is not a personal communication then the estimation of the truth of the announced information has a value in the interval {0.9, 1.0]. Here the substring *∃ e1(sit) Is(e1, announcing ** is to be read as "There is an event *e1* of the type "announcing" such that".

So let's imagine that, proceeding from the semantic representations Semr1b and Semr2b (the secondary KRs of the first and second sentences of the fragment Fr1) and the mentioned knowledge items from Kb1, the system infers the expression

((*Quantity(certain species * (Compos1, bird)(Descr, <S1, P1>)) _ 7) ∧ (P1 _ ∃ y1(bird)(Element(y1, S1) ∧ ∃ e1 (sit) Is (e1, nesting * (Agent1, y1)(Loc, v315)(Time, 2002))) ∧* (*Quantity(certain species * (Compos1, bird)(Descr, <S2, P2>)) _ 5) ∧ (P2 _ ∃ y2(bird)(Element(y2, S2) ∧ ∃ e2 (sit) Is (e2, nesting * (Agent1, y2)(Loc, v315)(Time,2001))))) .*

Suppose that Kb1 contains the following knowledge items:

∀ z1(space-object) ∀ t1(year) Follows (Better(Ecolog-sit(z1, bird, t1), Ecolog-sit(z1, bird, t2)), Better(Ecology (z1, t1), Ecology (z1, t2))) ,

*∀ t1(year) ∀ t2(year) Follows(((Set1 _ certain species * (Compos1, bird)(Descript, Q1)) ∧ (Q1 _ ∃ y1(bird)(Element(y1, Set1) ∧ ∃ e1 (sit) Is (e1, nesting * (Agent1, y1)(Loc,x1)(Time, t1)))) ∧ (Set2 _ certain species * (Compos1, bird)(Descript, Q2)) ∧ Analogue (Q2, Q1, < Previous-year (t1), t1>) ∧ Greater (Quantity(Set1), Quantity(Set2))), Better(Ecolog-sit(z1, bird, t1), Ecolog-sit(z1, bird, Previous-year(t1)))) .*

Here the substring *Analogue (Q2, Q1, < Previous-year (t1), t1>)* means the K-string marked by the variable *Q2* can be obtained from the K-string *Q1* by means of replacing the variable *t1* in *Q1* with the string *Previous-year (t1).*

That is why the system finally infers the formulas

Better(Ecolog-sit(v315, bird, 2002), Ecolog-sit(v315, bird, 2001)) , Better(Ecology(v315, 2002), Ecology(v315,2001)) .

Hence the system formulates the answer "YES" and adds the expression of the form *Ground: Fr1 ('D. News', Date1).*

The Models of Type 5

The models of the ***fifth proposed class*** are destined for designing computer intelligent systems extracting knowledge from natural language sentences and complicated discourses for forming and updating a knowledge base (orontology) of an applied intelligent system. Such models describe the systems of the form

(*Lcontext, Linp, Lingbset, Kbset, Lsem1, Lsem2, transf, Alg, Proof) ,*

where *Lcontext* is a language for building a semantic representation of the already processed part of a NL-text, *Linp* is an input language consisting of underspecified or completely specified semantic representations of NL-expressions (sentences and some fragments of sentences); Lingbset and *Kbset* are (as above) the sets of possible linguistic databases and a knowledge bases; the semantic language *Lsem1* is destined for representing the deep meaning of the inputs

from Linp with respect to a semantic representation of the preceeding part of a NL-text; *Lsem2* is a language for representing the knowledge of the required kinds extracted from the expressions of the language Lsem1; *transf* is a mapping from the Carthesian product of the sets *Linp , Lcontext, Lingbset, Kbset* to the Carthesian product of the sets *Lsem1* and *Lsem2*; *Alg* is an algorithm implementing the mapping (or transformation) *transf*; *Proof* is a mathematical text being a proof of the correctness of the algorithm *Alg* with respect to the mapping *transf.*

The Significance of the Models for the Design of Linguistic Processors

The analysis shows the significance of the studies aimed at constructing the formal models of the considered kinds for the engineering of natural language processing systems (NLPSs). In particular, the following factors are distinguished:

1. The algorithms being components of such formal models can be directly used as the algorithms of the principal modules of NLPSs.
2. The descriptions of the mappings *transf*, characterizing the correspondence between the inputs and outputs of the systems, can become the principal parts of the documentation of such programming modules. As a result, the quality of the documentation will considerably increase.
3. The designers of NLPSs will get the compehensible formal means for describing the semantics of lexical units and for building semantic representations of complicated natural language sentences and discourses in arbitrary application domains. This will contribute very much to the transportability of the elaborated software of NLPSs as concerns new tasks and new application domains.

It should be underlined that even the elaboration of the *partial models* of the kind promises to be of high sugnificance for the engineering of NLPSs. The pincipal difference between the complete models and partial models of the considered types consists in the lack of a proof of the correctness of the algorithm Alg with respect to the defined transformation transf. Besides, a partial model may include not mathematicaly complete definition of a transformation *transf* but only a description of some prinipal features of this transformation.

Even in case of partial models the designers of semantics-oriented NLPSs will receive an excellent basis for the preparation of such documentation of a computer system that distinguishes the most precious or original features of the algorithms and/or data structures and creates the good preconditions of transporting the data structures and algorithms to new problems and application domains.

References

Fomichov, V. (1992); Mathematical Models of Natural-Language-Processing Systems as Cybernetic Models of a New Kind; Cybernetica (Belgium), Vol. XXXV, No. 1 (pp. 63-91)

Fomichov, V.A. (1996); A Mathematical Model for Describing Structured Items of Conceptual Level; Informatica. An International Journal of Computing and Informatics (Slovenia); Vol. 20, No. 1 (pp. 5-32)

Fomichov, V.A. (2002); Theory of K-calculuses as a Powerful and Flexible Mathematical Framework for Building Ontologies and Designing Natural Language Processing Systems (ed. Troels Andreasen, Amihai Motro, Henning Christiansen, Henrik Legind Larsen), Flexible Query Answering Systems. 5th International Conference, FQAS 2002, Copenhagen, Denmark, October 27 - 29, 2002. Proceedings; LNAI 2522 (Lecture Notes in Artificial Intelligence, Vol. 2522), Springer Verlag (pp. 183-196)

Fomichov, V.A. (2004); Theory of Standard K-languages as a Model of a Universal Semantic Networking Language; Preconference Proceedings "Intelligent Software Systems for the New Infostructure" (Focus Symposium in conjunction with the 16th Intern. Conf. on Systems Research, Informatics and Cybernetics – InterSymp-2004, July 29 – Aug. 5, 2004, Germany). Focus Symposia Chair: Jens Pohl - CAD Research Center, Cal Poly, San Luis Obispo, CA, USA (pp. 51-61)

Fomichov V.A. (2005a); The Formalization of Designing Natural Language Processing Systems. Moscow, MAX Press, 368 P. (in Russian)

Fomichov V.A. (2005b); Standard K-Languages as a Powerful and Flexible Tool for Building Contracts and Representing Contents of Arbitrary E-Negotiations; K. Bauknecht, B. Proell, H. Werthner (Eds.), The 6th Intern. Conf. on Electronic Commerce and Web Technologies "EC-Web 2005", Copenhagen, Denmark, Aug. 23 - 26, 2005, Proceedings. Lecture Notes in Computer Science. Vol. 3590. Springer Verlag (pp. 138-147)

Similarity Assessment Techniques

Michael Zang, Adam Gray, and Michael Hobbs

CDM Technologies, Inc
2975 McMillan Avenue, Suite 272
San Luís Obispo, CA 93401

Abstract

This paper serves as a layman's introduction to similarity assessment techniques, their distinction from contemporary computing paradigms, and their real-world applications. The following content derives from the experience gained by the authors in applying similarity assessment techniques to resolve the real-world problems of CDM customers, in particular those faced by the United States Transportation Command (USTRANSCOM).

Keywords

Agents, Case-Based Reasoning, Data Cleansing, Data Mapping, Search, Similarity Assessment

Introduction

Study of case-based reasoning (CBR) served as our initial introduction into similarity assessment strategies. CBR is a two-part reasoning process whereby (1) similarity assessment techniques are employed to compare a new, current case against a case base of previously stored cases to find the most similar case in the case base, which is (2) subsequently used to classify and resolve the current case. Case-based reasoning projects such as the Collaborative Agent-Based Control and Help system (COACH)[1] and the Navy Conversational Decision Aid Environment (NaCoDAE) (Breslow and Aha 1998) refactoring project[2] led us to appreciate the applicability of the first step of case-based reasoning (i.e., similarity assessment) to a variety of problems. Having identified the CBR-derived similarity assessment as a broadly applicable problem-solution paradigm, we applied it, with success, to a number of difficult real-world problems, thus initiating a compelling new growth platform for the company.

This paper first describes the paradigm by which most contemporary software-based problem solutions are derived—namely, Boolean logic, shown to be distinct from similarity assessment as a problem-solution paradigm. Similarity assessment is discussed within the context of its heritage, case-based reasoning. Examples of distinct similarity assessment techniques—word, trigram, numeric, vicinal, and mixed-initiative—and similarity assessment applications—search, mapping, and data cleansing—are provided to further illustrate the concept. The paper ends with a summarizing conclusion.

[1] COACH, a research project sponsored by the Office of Naval research from 1999 to 2001, employs case-based reasoning technology to provide analysis, evaluation, and the formulation of guidance for major equipment item repairs aboard US Navy ships.

[2] NaCoDAE refactoring was performed under a collaborative research agreement (CRADA) with the Navy Center for Applied Research in Artificial Intelligence.

Boolean Logic

Boolean logic is a complete system for operations in symbolic, mathematical logic named after its inventor, George Boole, the 19th-century mathematician, who uncovered the algebraic structure of deductive logic, the basis of computer hardware and software. In Boolean logic, elements each contain only two possible values, following various conventions, such as "true" and "false", "yes" and "no", "on" and "off", or "1" and "0". A Boolean expression, such as "X < Y And Y< Z" evaluates to true or false. Relational databases use Boolean logic to perform queries. For example in standard relational query languages a statement such as: "SELECT * FROM EMPLOYEES WHERE (Not LAST_NAME = 'Zang') And (FIRST_NAME = 'Mike' Or 'Michael')", is used to return records from a database. Search engines (e.g., Google) also employ Boolean logic. For example "Search term 1" "Search term 2" is equivalent to "Search term 1 Or Search term 2".

Special characters, such as truncation and wildcard, provide a higher level of abstraction than the Boolean logic which underlies the operations implementing them. Truncation, '*', allows for search using a shortened (i.e., truncated) form of a word. For example, "adolescen*" will return both "adolescent" and "adolescence". Wild card characters, '?', prove useful in allowing for alternate spellings and other quirks of the English language. For example, "wom?n" will return results for both "women" and "woman." These special characters provide a higher level abstraction than straight Boolean Logic. Note for example, "adolescen*" is not the same as the Boolean expression "adolescent And adolescence" as it may return other perhaps unforeseen values such as "adolescents."

Regular Expressions further the abstraction by providing a language of special characters for identifying strings of text of interest, such as particular characters, words, or patterns of characters. A relatively simple Regular Expression can, for instance, identify the word "car" in isolation or when preceded by the word "blue" or "red", while another can recognize a dollar sign immediately followed by one or more digits, followed, optionally, by a period and exactly two more digits, thus accounting for dollars and cents. While special characters and regular expressions can prove useful during focused searches, processing them across the entire database may consume excessive computer resources (Strickland and Henderson 2005).

Similarity Assessment

Similarity assessment techniques inherently concern objects that share a common set of features to varying degrees. Well-established techniques have been developed in the fields of machine learning—a branch of artificial intelligence—and statistical pattern recognition (Michie et al. 1994). Example approaches include neural networks, support vector machines, decision-tree induction, Bayesian, and case-based classification techniques.

Neural networks consist of interconnected, biologically inspired processors called neurodes that can learn to classify by pre-classified examples. Support vector machines use an optimization technique to find planes that best separate objects into distinguishing classes. Decision-tree induction algorithms typically employ a measure called information gain to select the most promising features, construct a decision tree, and use it to classify new objects. Bayesian classification techniques apply estimations of class-conditional probabilities to predict a label for

a new case or object. Case-based classification reuses the decisions from the closest matching pre-classified objects to label new objects.

The applicability and classification performances of these techniques depend on the representation language of the objects and the amount of available pre-classified data. For example, neural networks and support vector machines work well with many instances of numerical data. In contrast, case-based techniques are well-suited for a mixture of data types with relatively few examples.

Case-based techniques utilize similarity assessment to classify new objects. Classification refers to the task of assigning one or more predefined labels to a previously unlabelled object. Case-based classification works as follows. For a new object or a case to be labeled, similarity assessment techniques are utilized to retrieve the most closely matching previously labeled cases from a database of cases, called a case base, to assign the label from the retrieved cases as the label for the new object (Kibler and Aha 1988).

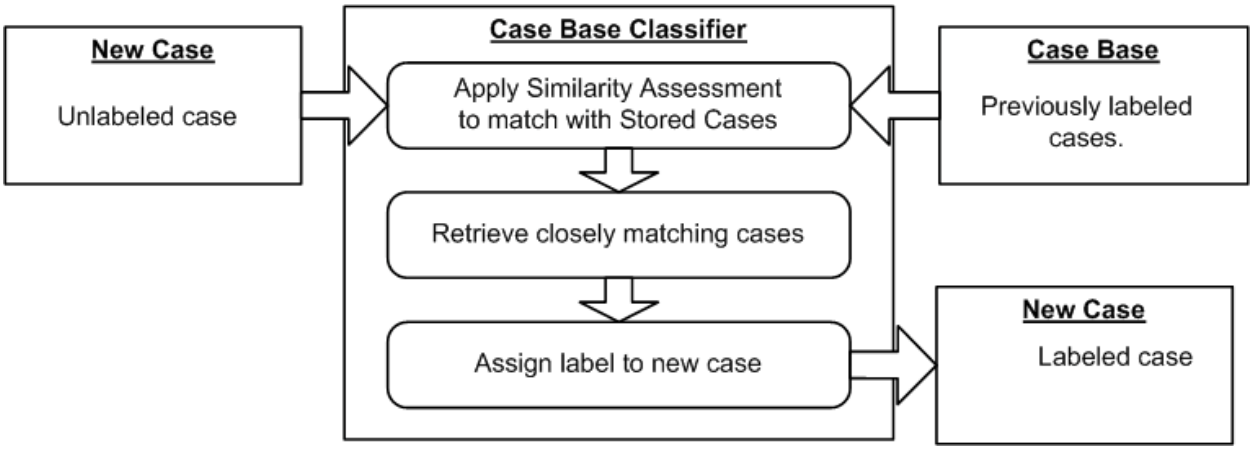


Figure 1: Case-Based Classification

Classification performance depends significantly on two design factors, the case representation and the similarity metric.

> ***Case Representation***: A case is a structured representation of the factors to be used for assessing similarity between two cases. The most common case representation is a list of attributes and values.
>
> ***Similarity Metric***: A similarity metric is an aggregation function that assigns a number between 0 and 1 as a measure of similarity between two cases. A similarity value of 1 implies that the two cases are identical while a similarity value of 0 implies that they are completely distinct.

Examples of similarity or distance metrics include the Euclidean metric, cosine metric, and Hamming distance. Such metrics often specify parameters such as attribute weights to improve classification performance. Nonetheless, when a large number of features are associated with a case base, some prove irrelevant and can reduce classification performance. While counteractive parameters can be set manually, automatic methods can potentially achieve the same result. For

this purpose, several attribute weighting and feature selection methods, such as information gain and rough-set theoretic methods (Gupta et al. 2005), are available.

Word Similarity

The principal problem when comparing the similarity of words is the breadth of naming differences for word-associated concepts. Causes for such differences are described by the taxonomy in Figure 2. Name variation can arise for syntactic or semantic causes. Syntactic distinctions engender commonly used vocabulary (e.g., "code" vs. "id"), conventions (e.g., "Airport_Code" vs. "AirportCode"), and abbreviations (e.g., "Airport" vs. "Arpt"). Semantic distinctions occur as differences in abstraction (e.g., "Ship" vs. "Vessel") and granularity (e.g., "Name" vs. "First Name" and "Last Name").

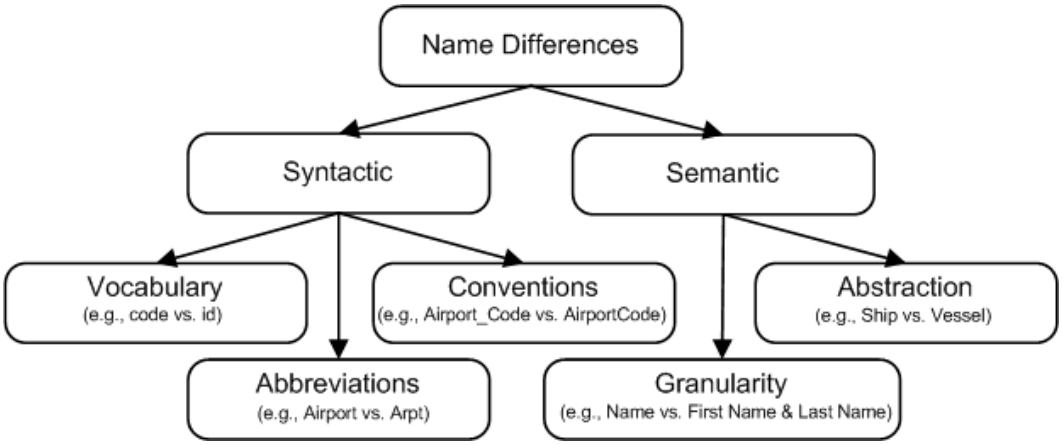


Figure 2: Taxonomy of Naming Differences

Many techniques have been developed to address such terminological variation. CDM has employed the following approaches to aid in detecting the conceptual equivalence of two distinct terms.

- ***Creation of a synonymous terms lexicon,*** by exploiting the mapping that has already been performed. For example, if the field label "Ship Id" has been manually mapped to "Vessel Code" it can be interactively identified that "Ship" is synonymous with "Vessel" while "code" is synonymous with "identifier". Such a resource can be used directly for similarity assessment.
- ***Use of WordNet*** (Fellbaum 1998), a publicly available linguistic ontology, to identify occurrences of terminological variations due to conceptual abstraction. For example, the hyponym (is-a-type-of) relation between concepts (e.g., Ship is-a-type-of Vessel) may be exploited as part of the similarity assessment.
- ***Creation of an abbreviation resource,*** by exploiting the correspondence of data model logical and physical names. The logical names that include non-abbreviated terms (e.g., Airport) and their corresponding physical names that include abbreviated terms (e.g., ARPT) can be used to automatically create a lexicon of abbreviations for use in similarity assessment.

- ***Employment of an order-based abbreviation detection algorithm***, which exploits the characteristic of abbreviations to preserve the relative ordering of the original word. For example, the letter "r" occurs after the "p" in the word "airport." A search utilizing this algorithm will, for instance, return "ARPT," a desired result, while excluding instances of "APRT" (i.e., apartment), an irrelevant, if similar abbreviation. Regularity in abbreviation conventions can be exploited to detect abbreviations dynamically when they are not available in the abbreviations lexicon.

Synonym and abbreviation resources may be employed with a technique commonly known as Bag-of-Words to assess the similarity between textual data. Here the number of primary words in common divided by the number of unique words provides a measure of similarity. Algorithms can exploit a synonym resource to maximize the number of common words. Synonyms may have an associated weighting factor denoting the semantic distance between the word pair. For example, the concept of the word "vessel" is more general than that of "ship," which is more general than that of "oil tanker." To account for variation in semantic abstraction, the synonym pair (vessel, ship) may have a weighting factor of 0.9 and the pair (vessel, oil tanker) a 0.8. The example in Figure 3 shows application of this technique to determine that the expression, "The vessel is arriving" is more similar to "The ship is coming" than "The airplane is coming".

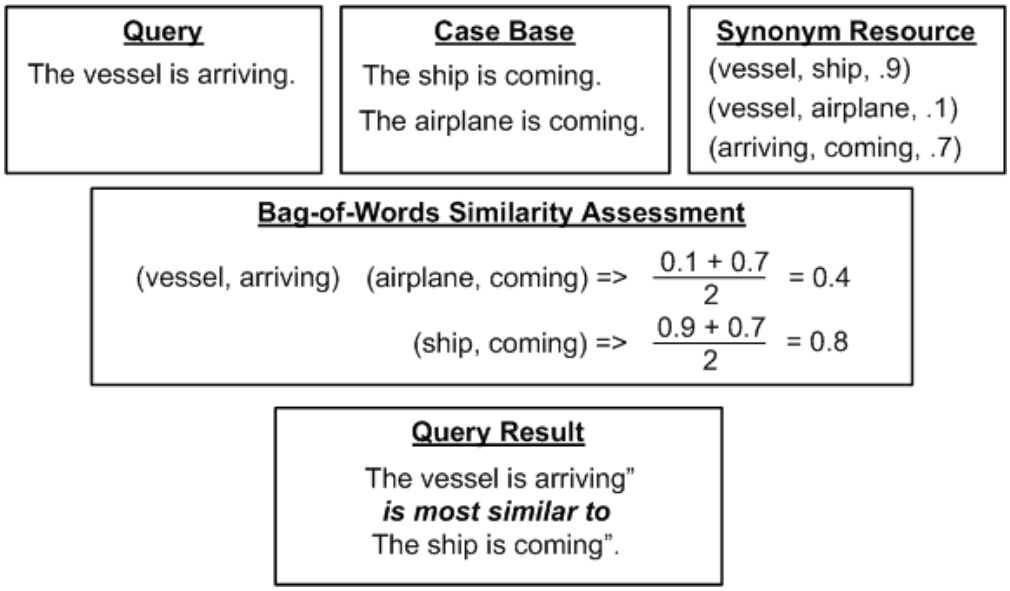


Figure 3: Word Similarity Query Example

Trigram Similarity

Trigrams represent a specific instance of the more general N-gram concept by which text is broken down into all-composing three letter contiguous sequences in order to compare for similarity with other text. N-grams, in turn, represent one of many techniques for producing similarity metrics for what is known in the industry as case-based classification. N-grams are particularly tolerant of spelling variations and misspellings and indeed serve as the underlying technology for contemporary spell-checkers, which present a list of correctly spelled words that

may correspond to a word flagged as incorrectly spelled (i.e., not found in the dictionary of words).

As an example, assume a mechanic wishes to query an intelligent parts database for a pressure gauge. The semi-literate mechanic queries the system with the phrase "presure gage." This query generates the 11 unique trigrams shown in Figure 4. Note that the beginning and ends of words are padded with spaces to emphasize their importance within a word.

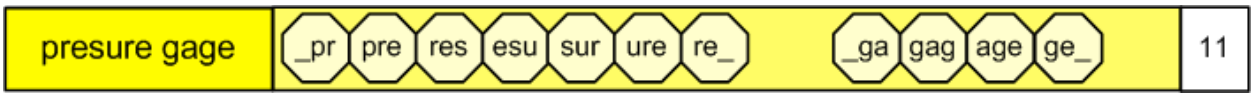


Figure 4: Query Phrase Trigrams

Further assume that the case base (i.e., the parts database employed in a similarity assessment context) contains a pressure gauge, garlic press, and box wrench. The trigrams for these items are shown in Figure 5. The number of occurrences of a particular trigram within the case base—1 or 2 in this example—is shown below each case base item trigram. Note that the trigrams "_pr," "_ga," "pre," and "res" occur in both "pressure gauge" and "garlic press" (i.e., twice as often as do any other trigrams). Statistical analysis of the number of occurrences of each trigram allows them to be weighted for rarity. For example, the twice-occurring trigrams may be weighted at .5 while the single occurring trigrams are weighted at 1. These weights are employed when summing like trigrams.

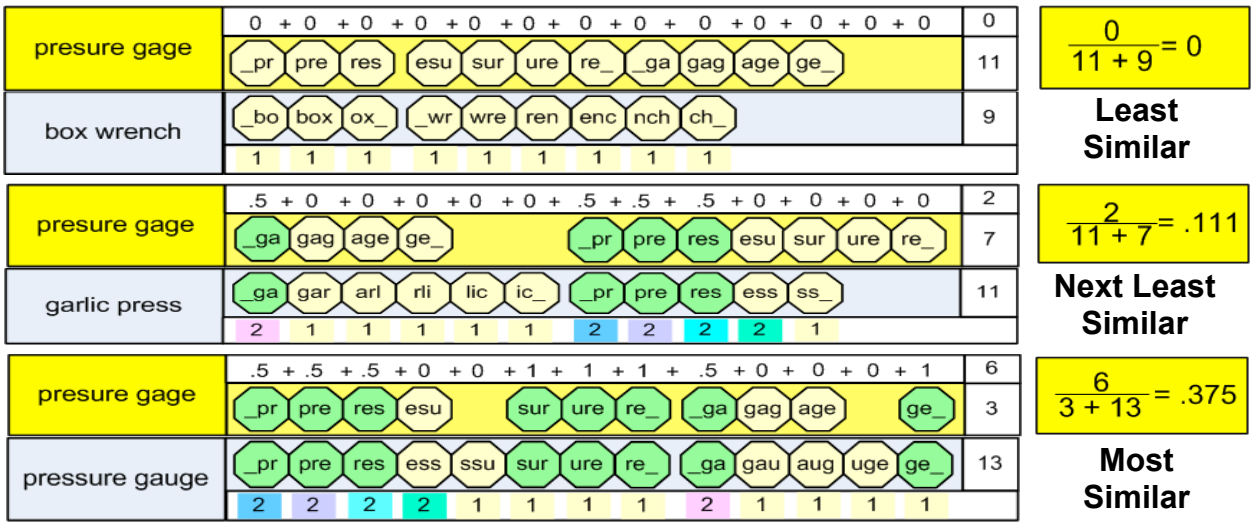


Figure 5: Trigram Similarity Query Example

In the example, the query text "presure gage" shares 0 trigrams with "box wrench. With "garlic press," "presure gage" shares 4 trigrams each weighted at .5 due to their commonality. This produces of score of 4 times 0.5 divided by 18—the total number of unique trigrams contained in both, which equals 0.111. Comparing "presure gage" to pressure gauge produces a score of 0.375. The individual comparison scores are only meaningful in relation to other scores. In this example the case base item "pressure gauge" is clearly the most similar to the query "presure gage" as its score of 0.375 has a 54% difference from the next highest score of .111 for "garlic press."

Numeric Similarity

The similarity assessment of numeric values, including quantitative values (e.g., weights and ages) and displacement values (e.g., heights and lengths), pose inherent difficulties for deriving appropriate similarity scores. Consider the comparison of the weight of a 20 pound dog to that of an 11 pound cat. When attempting to determine the appropriate similarity score, it becomes clear that additional information is needed. If the range of weights of all animals being compared runs from 5 pounds to 20 pounds, then the weights of the dog and cat are relatively dissimilar (i.e., their similarity score should be close to 0). However, if the list of animals includes a 100-ton blue whale, then, relatively speaking, the weights of the dog and cat are quite similar (i.e., their similarity score should approximate 1). Hence, the calculated similarity score should be different in these two cases.

For this reason it is critical that the variance of values across the entire comparison domain be considered when determining the similarity of any two elements. CDM's approach, specified by Equation 1, calculates the similarity between the two numeric values relative to the difference between the maximum and minimum values across the entire domain.

$$Similarity(X,Y) = \frac{((MAX_{val} - MIN_{val}) - |X_{val} - Y_{val}|)^{LOG\_MOD}}{(MAX_{val} - MIN_{val})}$$

Equation 1: Assessing Similarity of Quantitative Values

An additional difficulty arises when comparing values within a domain with a large variance. Taking our weight example, if the 100-ton whale is included in the comparison, a five pound difference in weight leads to only a .00002 difference in similarity score. This makes the calculated difference for a significant number of comparisons negligible (i.e., the difference in weight between the dog and the cat would have almost no effect on similarity score). To counter this issue, CDM uses a logarithmic modifier (LOG_MOD) to accentuate the differences at the lower end of the scale.

Vicinal Similarity

Another valuable technique, vicinal similarity comparison, targets the relative distance between locations (i.e., determines if two locations are in the same vicinity). This type of comparison faces an inherent difficulty in that geographic locations are often represented in multiple ways and include information from multiple fields (e.g., latitudinal and longitudinal values are combined). Hence, geographic information must be translated into a comparable format before similarity calculations can be attempted. CDM's approach is to take the provided format (e.g., latitude/longitude, geospatial, etc.) and convert it into vector [x,y,z] coordinates on the earth. The physical distance between two points is then calculated by determining the angle between the two vectors and scaling the results based on the radius of the earth.

Once a physical distance between two points has been calculated, the problem is essentially one of Numeric similarity, described in the previous section. The appropriate similarity score to assign the results again largely depends on the variance between the locations across the entire domain. For example, when comparing distances between locations in a single zip code, then the locations of two different cities in that zip code might be relatively dissimilar (i.e., their similarity score should be close to 0). However, if you are comparing locations across the entire

world, those same two cities' locations will be, relatively speaking, quite similar (i.e., Their similarity score should approximate 1). The approach taken by CDM, as when handling numeric comparisons, is to determine the similarity between two locations by comparing their physical distance relative to the largest physical distance between two locations across the entire comparison domain. Once again, a logarithmic modifier is again used to magnify the differences at the lower end of the scale when comparing domains with a large variance.

A related vicinal technique compares geographic areas, such as zip codes or countries, rather than specific points on the earth. CDM handles this type of comparison by assigning a specific point on the earth to represent the estimated center of each geographical area. Once this single point has been assigned for all areas within the case representation, similarity assessment can continue as described above.

Mixed-Initiative Assessments

Our experience with similarity assessment techniques, as with artificial intelligence paradigms in general, clearly indicates that specific techniques can perform exceptionally well with one data set and exceptionally poorly with others. All such techniques operate based on pre-established (i.e., built-in) assumptions, thus limiting the utility of any single approach as a solution to every case. The arbitrary or inconsistent performance of single assessment techniques can be counteracted by utilizing the weighted average of a number of distinct techniques to provide a single measure of similarity. Weighting factors allow techniques to be turned off (i.e., weight = 0), turned on (i.e., weight =1), or set to have less influence (i.e., 0 < weight < 1) depending on their domain performance individually and/or in conjunction with other techniques. This approach was shown to greatly improve the performance of the data mapping application (Gupta et. al. 2008) described later in this paper. Note, however, that the mapping performance gained by the mixed-initiative approach comes at the expense of computational performance. While computational performance is not as much of an issue for an application that can run in the background and post results when complete, it can present a problem for one which requires real-time user interaction, such as with an internet search or database query.

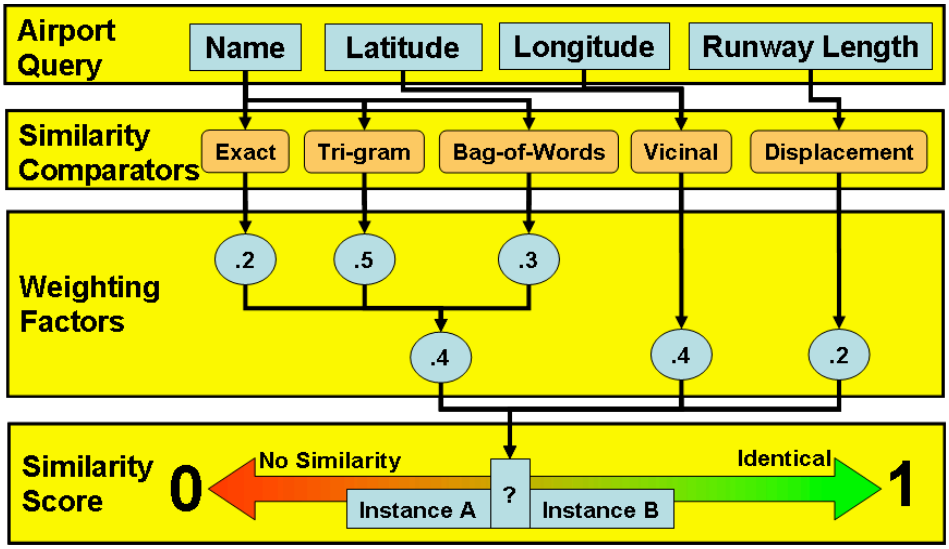


Figure 6: Mixed Initiative Assessment

Figure 6 depicts a mixed-initiative assessment used to correlate airport records contained in two distinct database tables, each standard to a specific domain, Commercial and Military. Both tables contain fields corresponding to name, latitude, longitude, and runway length. The naming conventions, descriptiveness, and data quality, however, vary both across and within the two databases. For example, LAX may be named Los Angeles International, LA Airport, or LAX CIV Runway 1. Geographic locations (i.e., Lat/Long) and runway lengths can be missing, dramatically erroneous, close, or precise. In the example, Names are compared using exact match weighted at .2, Trigram weighted at .5, and Bag-of-Words weighted at .3. The resulting assessment score is combined, with weight .4, with a vicinal comparison of the latitude and longitude, with weight .4, and a numeric comparison of runway length weighted at .2, to produce an overall similarity score.

Search Applications

The data quality and interoperability issues faced by USTRANSCOM are closely associated with reference data (RD) which captures the relatively static information that defines and categorizes enterprise-associated representational entities, thus specifying the universe of content that can be referenced by program-specific data to provide externally meaningful semantic context. As such, a substantial portion of the information exchanged between automated information systems (AISs) is comprised of RD codes which serve as unique identifiers for individual records within a specific reference data set. Contemporary practice results in a significant percentage of exchanged RD code values being rendered invalid or contextually improper, which results in interoperability issues among systems. Such issues introduce operational inefficiencies within the enterprise while degrading the quality of its composite information state, upon which critical business decisions are based. This problem is particularly prevalent in regards to very large and dynamic RD data sets such as National Stock Numbers (NSN), DoD Activity Address Codes (DODAACs), and Geographic Locations (GEOLOCs).

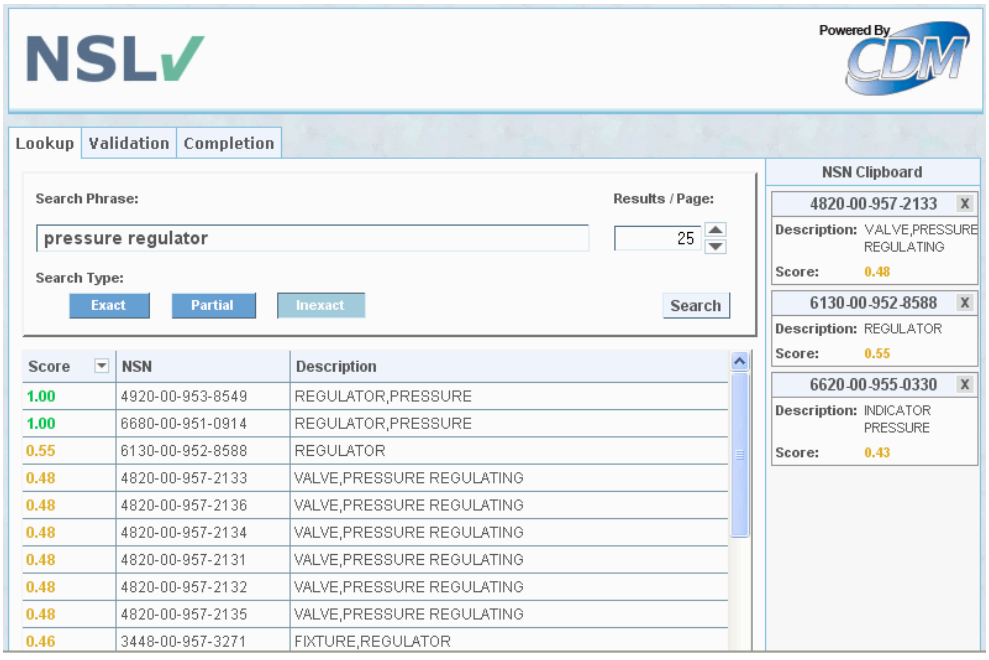


Figure 7: National Stock Number Lookup and Validation Tool

The users and agents of enterprise information systems require intelligent runtime access to RD sets in order to obtain the RD-identifying key codes or associated item characteristic data necessary to perform their system-specific IT tasks. Consider the problem of obtaining the National Stock Number—a 13-character code identifying one of the 8.6 million DoD supply items—to fill in a requisite field on a data form. Knowing what one has or wants does not necessarily provide the code, due to nomenclature variation, imprecision, and the limitations of contemporary database query technology. To confront this problem, CDM developed the National Stock Number Lookup and Validation Tool (NSLV)[3] to assist users in finding National Stock Numbers (NSNs) using free-form textual descriptions of desired items. NSLV employs the trigram similarity assessment technique described in the Figure 5 example. The NSLV screen shot (Figure 7) displays the ranked results of a user query for a pressure regulator.

Mapping Applications

The joint deployment and distribution responsibilities of USTRANSCOM require a substantial level of interoperability across the broad range of technically and functionally diverse automated information systems (AISs) utilized by USTRANSCOM and the individual service branches, as well as the associated commercial suppliers and shippers. This aspect of the USTRANSCOM mission prompted the development of an enterprise-wide data model, known as the Master Model (MM) and a formalized program for the identification, management, and distribution of enterprise reference data, known as the TRANSCOM Reference Data Management (TRDM) program.

The MM enables interoperability at the metadata (i.e., database schema) level while TRDM enables interoperability at the instance-data (i.e., database record) level. However, the human-intensive (thus costly and error prone) development and maintenance of semantic maps is required for these capital investments to provide for increased interoperability levels. At the metadata level, these semantic maps relate elements within AIS-specific data models and interface specifications to elements within the MM. At the instance-data level, they join equivalent or semantically related records (e.g., airport to geographic location records as correlated by latitude and longitude fields).

The Intelligent Mapping Toolkit (IMT)[4] is designed as a set of intelligent collaborative tools to support professional analysts performing labor- and knowledge-intensive semantic mapping tasks within a dynamically evolving information infrastructure. IMT employs a federation of matching agents for case-based similarity assessment and learning. IMT semi-automatically acquires domain-specific lexicons and thesauri to improve its mapping performance. It also provides an explanation capability for mixed-initiative mapping. IMT's primary goal is to suggest mappings to users for final verification and acceptance (Gupta, et. al. 08).

[3] NSLV was developed in the context of an analysis of the USTRANSCOM Reference Data Management program (TRDM) under contract to USTRANSCOM J6, 2006 – 2007.
[4] Sponsored by USTRANSCOM J6, 2005 - 2007

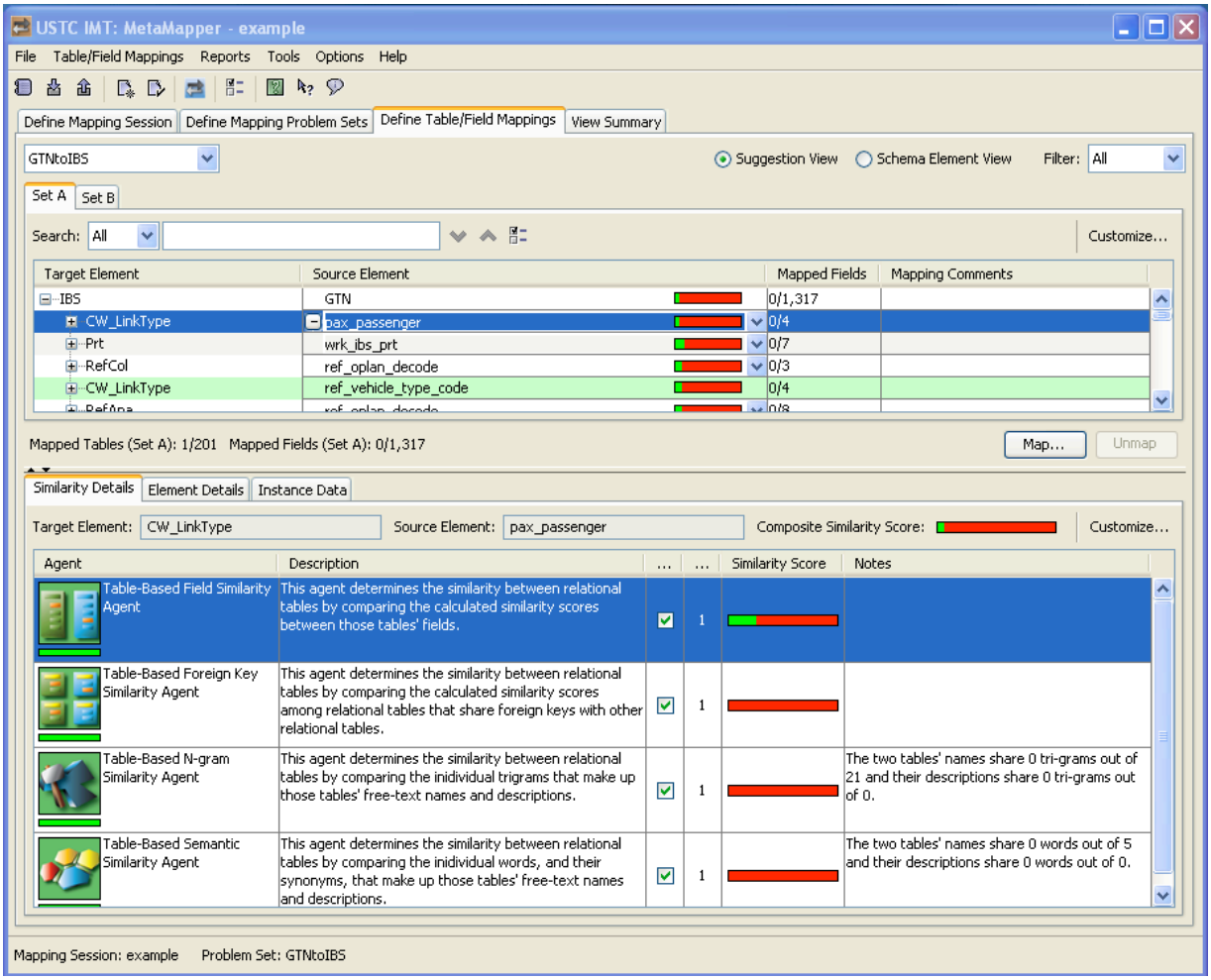


Figure 8: The Intelligent Mapping Toolkit

Data Cleansing Applications

The United States Central Command (USCENTCOM) Top 100 Analysis process provides visibility to the heaviest sustainment items (identified by National Stock Number [NSN]) from two continental United States aerial ports of embarkation—Charleston Air Force Base (AFB), South Carolina, and Dover AFB, Delaware—to the USCENTCOM area of responsibility (AOR) each month. This information is used to support shifting the mode of transportation of the heaviest, most often airlifted commodities to surface transportation, in order to realize significant transportation cost avoidance and greater efficiency for the Department of Defense. The current process is predominantly manual and requires excessive time to analyze the data, especially to overcome data quality issues. The most pressing of these data quality issues fall into four categories: missing item shipping weights, incorrect item shipping weights, missing NSNs, and missing item names.

The Intelligent Data Analysis Application (IDAA)[5] is designed to detect problematic data within imported queries of shipped cargo items and provide support to users in resolving them. Using a

[5] Sponsored by USTRANSCOM J6, 2007 - 2008

database of nearly three million approved cargo types, IDAA matches cargo items by National Stock Number (NSN), item name, weight, and cube. For those items that cannot be definitively matched, suggested cargo types are automatically generated using intelligent similarity-based data comparators.

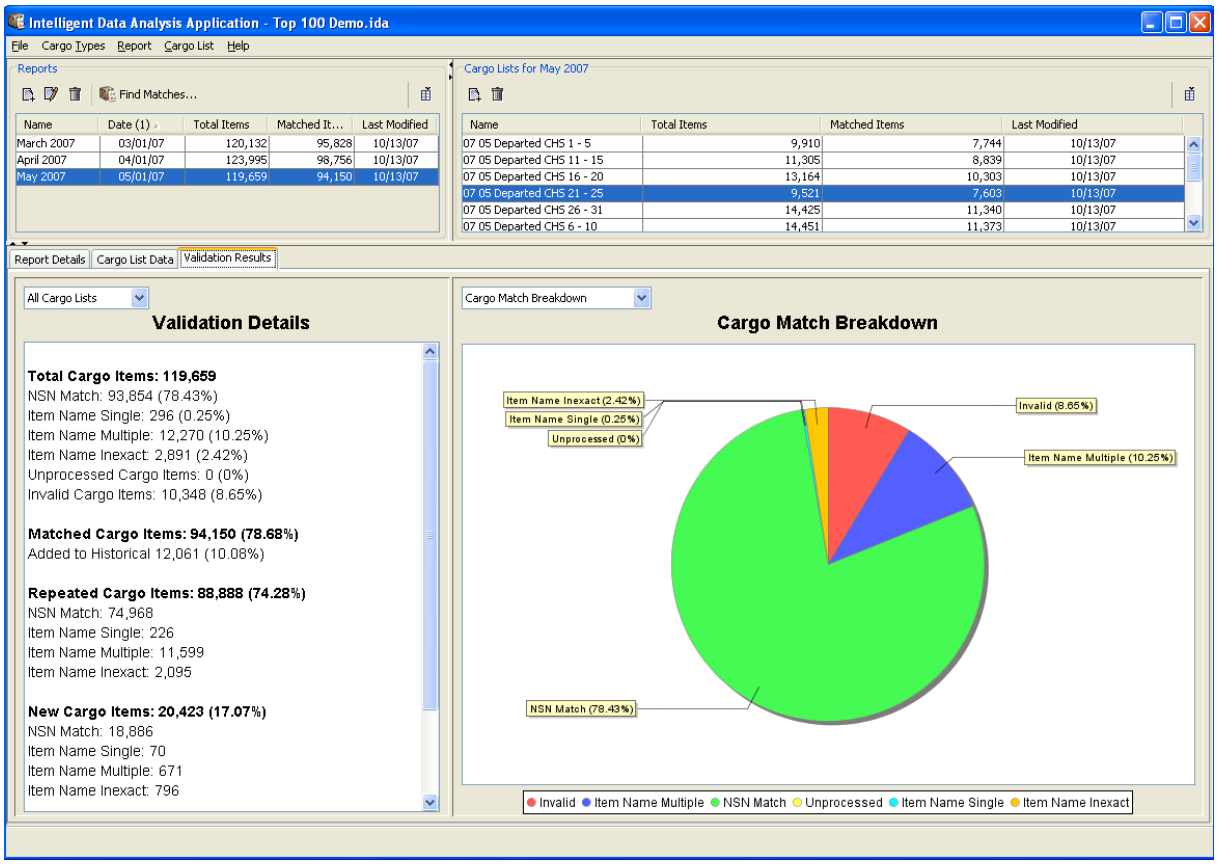


Figure 9: The Intelligent Data Analysis Tool

Conclusion

Similarity assessment techniques provide a unique solution approach applicable to a broad range of problems. They prove to be intuitive to both users and developers, perhaps due to similarities with human problem-solving approaches. Contemporary paradigms are based directly upon Boolean logic in which elements may either be True (1) or False (0). Similarity assessment paradigms provide an analog paradigm which allows elements to have values between 1 and 0 as well. Similarity assessment is derived from the field of case-base reasoning and is particularly effective when employed in conjunction with users to bring pertinent or desired items to their attention in ranked order. Various techniques are available for assessing similarity of textual and numeric data, although performance is strongly dependent on the application domain. This problem can be overcome by combining the results of multiple techniques to produce a single assessment result, given the acceptability in the corresponding loss of computational performance. Similarity assessment techniques have been successfully applied in a variety of search, mapping, and data cleansing applications previously irresolvable employing Boolean logic approaches alone.

References

Breslow, L.A. and Aha, D. W; NaCoDAE: Navy Conversational Decision Aids Environment (1998), Navy Center for Applied Research in Artificial Intelligence, Code 5510, Washington, D.C., December 1998

CDM Technologies, Inc. (2006A); Intelligent Mapping Tool [IMT] Design and Development Report, Task Area C: Intelligent Data Mapping Deliverable; USTRANSCOM J6 contract; 29 September 2006.

Fellbaum, C. (1998); WordNet: An Electronic Lexical Database. MIT Press.

Gupta, K.M.; Moore, P.G.; Aha, D.W.; and Pal, S.K. (2005); Rough-Set Feature Selection Methods for Case-Based Categorization of Text Documents, in S. K. Pal, S. Bandyopadhyay, & S. Biswas [Eds.]; Lecture Notes in Computer Science [LNCS 3776], (pp. 792-798); Heidelberg, Germany: Springer.

Gupta, K.M; Zang, M.; Gray, A.; Aha, D. W.; and Kriege, J. (2008); Enabling the Interoperability of Large-Scale Legacy Systems, Proceedings of the IAAI-08 conference, Emergent Application or Methodologies Papers; Menlo Park, CA. July 2008.

Kibler, D. & Aha, D.W. (1988); Case-Based Classification. E. Rissland & J.A. King [Eds.], Proceedings of AAAI Workshop on Case-Based Reasoning, (pp.62-67); St. Paul, MN.

Michie, D.; Spiegelhalter, D.J.; and Taylor, C.C. [Eds.] (1994); Machine Learning, Neural and Statistical Classification; New York, NY: Ellis Horwood.

Strickland, Jennifer and Henderson, John R. Web page maintained by: Library Webmaster, Ithaca College Library. http://www.ithaca.edu/library/course/expert.html, last modified: 10 October 2005.

USTRANSCOM Corporate Information-Centric Environment [CICE] (2003); USTRANSCOM Architecture and Technical Integration Division Report, prepared by CDM Technologies, Inc. USTRANSCOM TCJ6-A Scott Air Force Base, Illinois; October 2003.

Perspective Models - A Mechanism For Achieving Interoperability Among Expressive, Personalized Domain Views

Kym J. Pohl
CDM Technologies Inc.
2975 McMillan Ave.
San Luis Obispo, CA. 93401
805-541-3750 x233
kpohl@cdmtech.com
http://www.cdmtech.com

Abstract

Accurate and expressive representation of the subject matter over which a context-oriented, decision-support system operates is fundamental to the effectiveness and longevity of the resulting solution. Often taking the form of an ontology, such extensive representational models, by their very nature, are rich in relationships and both coarse and fine-grained objects. It is, however, these qualities enabling rich expression that can significantly increase both the complexity of developing against these models as well as the potential for incurring undesirable performance issues. Further, due to the typically detail-oriented usage inherent in the software-based users (i.e., reasoning agents, etc.) of these models, it is important to recognize that a singular view of the world so to speak is not necessarily appropriate across the entire Ontology user base. In fact, in such highly expressive environments, it is critical to not only recognizing these distinctions in user perspective, but to, in fact, promote and exploit them. It is by acknowledging and consequentially supporting this perspective-based individuality among Ontology users that true representational accuracy and utility is achieved.

Traditionally, software-based users comprising decision-support systems have operated over a singular, common representation. However, in the Perspective Model-enriched environment presented in this paper1, Ontology users are empowered with the ability to effectively perceive the world in accordance with individualized, native views. These views are then seamlessly inter-linked with one another to form a multi-Perspective Model of the target domain capable of supporting rich interoperability. Exclusively operating over personalized Perspective Models, users are not only shielded from the broad-scoped complexities inherent in the more omniscient concerns of the Ontology's entire scope but are also able to both view and interact with it in terms of more native representation.

To be effective, the concept of Perspective Models must be partnered with a supportive model development process. In addition to an explanation of the concept of Perspective Models, this paper also presents a purpose-built development process that supports effective creation of the

potentially numerous sets of models inherent in this type of expressive paradigm. The process offered in this paper effectively parcels the development of individual Perspective Models with the individuals possessing the necessary domain and use-case expertise. In this manner, the development process strives to significantly increase the involvement of the entire set of team members in the modeling activity, both capitalizing on user domain expertise in addition to increasing critical user understanding as well as acceptance of the representation over which their components will operate.

Representing Perspective

Fundamental to context-oriented reasoning is the highly expressive representation over which intricate analysis is performed [8] [12] [13]. Often in the form of an Ontology, such elaborate descriptions form the foundation underpinning the effectiveness of context-oriented, decision-support systems. An Ontology in the scope of this paper1,2 is defined as a highly expressive, typically relationship-rich model of the potentially extensive subject matter over which software components, hereunto referred to as users, reason and otherwise operate.

The Significance of Perspective

Perspective is applied each time we as human beings perceive something. Although certainly at times aligning fairly closely across multiple observers, such perspectives are inherently unique to the individual. Housed within these individualized perspectives is valuable information describing how a particular topic is most suitably represented from a certain point of view. In addition, such perspectives also convey how a particular subject relates to other subject matter seen as relevant by the particular individual. Even when a concept or thing has a common basis among observers, individual perception is typically still biased toward personalized experiences and overall knowledge. Although at times a significant complication for meaningful interaction, such perspective is extremely significant to accurate representation as it is rich in descriptive context. For example, consider the following illustration involving the laptop on which this paper was written. In the case of a software system assisting within the initial manufacturing process, the laptop might be most effectively described in terms of its product-oriented nature. In this sense, the most suitable representation of the laptop would revolve around characteristics relevant to assembly, packaging, and other such manufacturing-oriented concerns. Further, relationships to customer orders and delivery schedules would also be important to represent. In contrast, however, characteristics explicitly describing the laptop's utility in authoring publications or developing software are fairly peripheral, if not completely irrelevant to the target manufacturing domain. However, such perspective may be quite relevant to, for example, the interests of marketing or perhaps even customer-support. Of course both perspectives are quite valid with respect to their individual areas of operation. However, both views would inevitably encompass some of the same subject matter (i.e., laptops) yet describe them in distinctively different manners. The problem arises when users of distinctly different representations of the same subject matter attempt to interact. This situation can produce a significant dilemma. Simply stated, the valuable context that is expressed within individualized perspectives can also

significantly limit the ability for users to interoperate in a meaningful fashion (i.e., in terms of rich context).

However, despite the complications brought on by attempting to capture and exploit distinctive perspective, support for such personalized expression can significantly increase the quality of analysis performed by intelligent software agents operating within a decision-support environment. Perspective-enriched models can successfully capture not only the sometimes subtle distinctions among Ontology users, but by doing so can promote a more expressive description of each user's perception of their world. Unfortunately, due to the complexity inherent in identifying and supporting such subtleties and nuances, representation approaching this level of expression has traditionally been buried as implied assumptions within convoluted business logic or simply omitted entirely. However, when appropriately represented and housed within the context tier of a collaborative environment, such expressiveness can not only be effectively exploited, but is also much more readily accessible to users.

Perspective Models

However, even with perspective sufficiently represented within the context tier, the ability for users of such perspective to interact in terms of their individualized views poses a substantially complex interoperability problem. The solution to this interoperability dilemma comprises three elements. The first focuses on the development of a singular, all-encompassing ontology referred to as a Universal Model. As the name implies, Universal Models are an attempt to develop an all-purpose, amalgamation satisfying all possible use-cases and perspectives. In this paradigm, each user would utilize the Universal Model as its primary language for interacting with other users. As a distinct strength of this approach, each user would essentially dialogue with one another in terms of a single representation promoting interoperability in a clear and concise manner void of any context-diminishing translation. Each user would essentially share the same view of the world. However, considering the complexity resulting from collapsing what could possibly be numerous perspective-oriented characteristics into a single description, the resulting model would be severely bloated and would most likely fail to adequately represent any one particular perspective, resulting in a model confusing to utilize.

The second, somewhat related attempt at solving this dilemma addresses the inevitable complexity of the Universal Model approach described above and offers a more delineated organization. In this approach, each particular subject matter is modeled in terms of its fundamental, intrinsic nature. The various perspectives applied to each particular subject are explicitly represented as individual model fragments. These perspective sub models are connected to the subject models they enhance using the role analysis pattern [3]. Such a connection can be conceptualized as something playing a variety of roles with each role representing a particular view on that subject. In this fashion, individual perspectives can be easily managed and clearly discernable from one another. In addition, this approach offers a degree, although limited, of encapsulation and isolation from irrelevant perspectives as users can isolate their interaction with a subject matter to those perspectives that are meaningful to them. Further, additional perspectives can be integrated in a manageable fashion through the incorporation of new roles-based model fragments. As a result, each subject is connected to

model fragments describing the various contexts in which it can be viewed. For example, interaction with the aforementioned laptop subject from a manufacturing-oriented perspective may be in terms of a related ManufacturedProductRole model fragment. However, the problem with this approach is that even though perspectives relating to the same subject matter are somewhat partitioned from one another, they remain integrated into a single model with no explicit management and depending heavily on diligent usage. As such, additional access control may need to be employed to truly isolate users to relevant perspectives. In addition, there is still the dilemma of whether or not a slight difference in two perspectives is worthy to warrant creation of an entirely new Perspective Model fragment. In practice, one would be tempted to collapse subtle differences in perspective into a single, overloaded model fragment, thus compromising accurate expression.

The third, more promising solution to supporting individualized yet interoperating perspectives introduces the notion of a Perspective Model. Based on a semi-stateful façade design pattern [5], Perspective Models allow context-rich subject matter to be viewed by inter-operating users in terms of individualized, native perspective. Perspective models may directly contain their content, derive it from some type of shared source (e.g., an Integration Model), or comprise a combination thereof. While state simply for local consumption is represented and maintained within the Perspective Model itself, derivation is used for material that is shared across users (i.e., the basis for collaboration). In the case of derived content, the function of the Perspective Model may, for example, be to apply more native terminology, structure, or other characteristics that more appropriately represent the manner in which the particular user wishes to see the world. In some cases such mappings, either uni-directional or bi-directional, may be fairly straightforward and easily describable through standard expression grammar. However, in other cases these mappings may be rather complex to the point of requiring customized behavior. In either case, such mappings can be effectively described in terms of a formalized language such as XSLT [1] [9] or CLIPS-based rule sets [6] [11].

Integration Model

As mentioned earlier, derivation is essentially the means for linking together multiple perspectives applied to the same subject matter. While there are a number of approaches to supporting such integration, it is critical that the individuality and bias exhibited by each Perspective Model is preserved in its native form. These models are essentially a user's most familiar and descriptive language with which to interact with the rest of the world (i.e., other users).

The approach presented in this paper to interconnecting disparate perspectives of the same subject matter employs the notion of an Integration Model in conjunction with the façade design pattern [5]. Although not a necessity, employing an Integration Model as a central hub from which interacting models are mapped in and out of avoids the many-to-many mapping paradigm inherent with a more direct perspective-to-perspective connection. With this approach, a central, role-based representation of clearly delineated perspectives, not unlike the second alternative to integrating multiple perspectives described earlier, is developed as a well-structured and delineated combination of individualized perspectives related to the intrinsic subject matter they

enhance. For example, the main subject of our earlier example might take the form of a laptop entity that can play the role of a manufactured product, as well as perhaps the role of a software platform. While the laptop entity would be focused on describing the subject's intrinsic nature, characteristics specific to each of these two perspectives would be housed within each related role.

As a further, diagrammatic description of this connection, Figure 1 describes a logistically-oriented Perspective Model linked to an Integration Model that presents a fairly neutral description of a conveyance. As an aside, note that conceptually such neutrality is not necessarily a prerequisite in that if the Integration Model were more heavily biased toward a particular perspective, it would simply imply that the Perspective Models might need to be more extensive and incorporate additional constraints. However, in the interest of clarity, this example employs a somewhat neutral Integration Model.

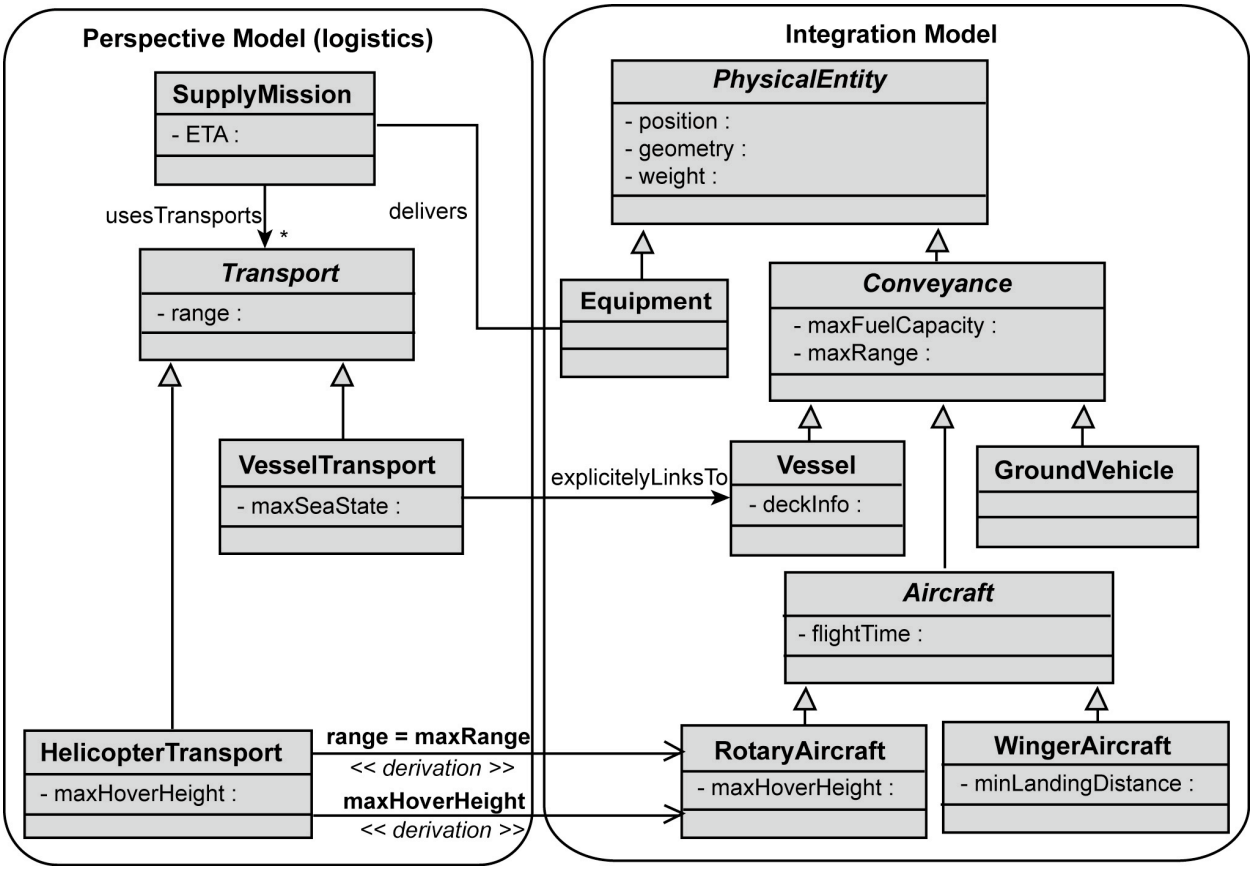


Figure 1 - UML [4] Diagram Illustrating A Logistics Perspective Model Deriving From A Relatively Unbiased Central Integration Model

Central to the logistics perspective presented in Figure 1 is the notion of a transport. Although the logistics perspective may have knowledge of the entire set of conveyance types (i.e., vessels, vehicles, and aircraft) represented in the Integration Model, in respect to the logistics view, only

vessels and rotary aircraft are considered candidate transports. In this situation, it would be valuable to represent this constraint in the Perspective Model employed by the logistics system while still basing such a biased view on the much more neutral representation of the conveyance offered by the Integration Model. As Figure 1 illustrates, representing such refinement can be accomplished by explicitly introducing a constrained notion of a transport in the logistics-oriented Perspective Model. According to the particular perspective, an abstract Transport is defined as taking two specific forms (VesselTransport and HelicopterTransport). At this point, it is immediately apparent that a vehicle is not a candidate to be a transport, from that perspective. In the context of this example, transports can only be VesselTransports or HelicopterTransports. The task now becomes linking this perspective together with the core Integration Model. Relating these two transport types to their conveyance derivation can be achieved in either an explicit or implicit manner. For illustration purposes, the definition of VesselTransport adopts the first method while HelicopterTransport employs the second. The first method defines an explicit, and exposed relationship between the VesselTransport and the core description of a vessel outlined in the conveyance section of the Integration Model.

Utilizing this approach, obtaining the core information relative to the corresponding Vessel from a VesselTransport requires both knowledge of their relationship in addition to a further level of indirection. For reasons of performance and representational precision, both of these requirements may not be desirable.

The second method, illustrated in Figure 1 using HelicopterTransport, overcomes both shortcomings inherent in the first approach. In this case, HelicopterTransport is represented in terms of a façade, or filter of sorts, which transparently connects this biased view to the core RotaryAircraft description housed within the Integration Model. That is, each attribute of RotaryAircraft relevant to the notion of a HelicopterTransport is explicitly declared within the façade. For example, since the maximum range of travel is relevant to the definition of a HelicopterTransport the maxRange attribute of RotaryAircraft (inherited from Conveyance) is subsequently exposed in the HelicopterTransport façade. By virtue of being declared as a derived property, any access to such an attribute would be transparently mapped to the corresponding attribute(s) housed within the Integration Model. In the case of the range attribute of HelicopterTransport, access is transparently directed to the inherited maxRange attribute of RotaryAircraft. Notice also the use of alternative terminology over that used in the Integration Model (i.e., range vs. maxRange). It should also be noted that the derived nature of a façade attribute is not limited to mapping to a single attribute. Rather, the value of a façade attribute may also be derived through specific behavior, perhaps a calculation or algorithm based on the values of multiple attributes residing across several Integration Model objects. In either case, the fact that the value of the façade attribute is derived, and not originating locally, is completely transparent to users of the HelicopterTransport Perspective Model object.

Yet another perspective-oriented enhancement to the core Integration Model illustrated in Figure 1 is the notion of a SupplyMission. Being a fundamental notion of a logistics perspective, a supply mission essentially relates equipment in the form of supply items to the transports by which they will be delivered. Once again, the definition of a logistics-specific notion (i.e., supply item) is derived from a notion defined in the Integration Model (i.e., equipment). In this case, an

explicit relationship is declared linking SupplyMission to zero or more Equipment items. From the perspective of the logistics system equipment scheduled for delivery is perceived as items to be supplied, the term supplyItems is a more appropriate nomenclature. Such enhancement to the

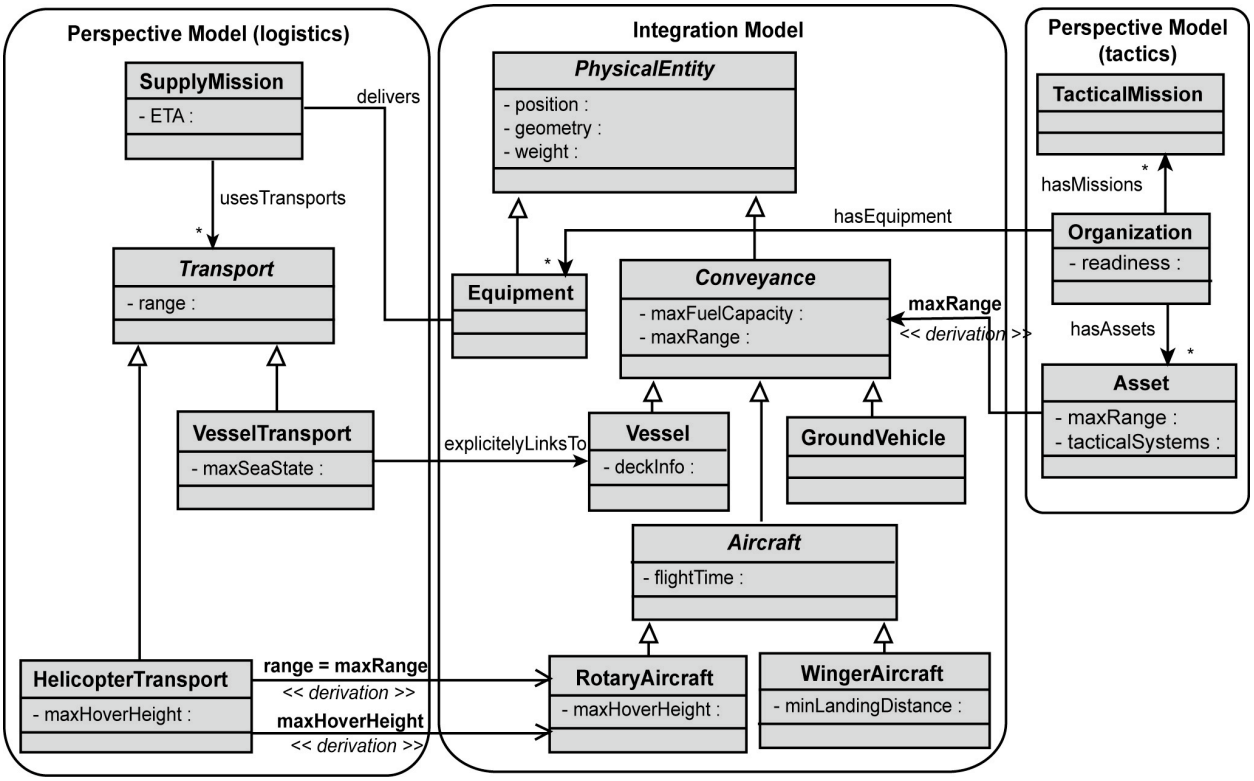


Figure 2 - UML [4] Diagram Illustrating Two Disparate Perspectives Connected Via A Central Integration Model

innate descriptions provided by the Integration Model demonstrates the ability of a Perspective Model to essentially overlay new notions (i.e., supply missions) over existing intrinsically-described subject matter (i.e., equipment and conveyances). To further illustrate how multiple, potentially diverse perspectives can be effectively integrated to support meaningful interoperability, Figure 2 elaborates on the example by introducing an additional perspective on the core subject matter. The additional perspective is concerned with a more tactical view of the domain. Collaboration between these two perspectives is enabled by the common Integration Model from which many of their notions derive. A conveyance is still a conveyance whether viewed in the context of logistics operations or tactical command and control. Although both users may discuss a conveyance from partially disparate perspectives, both can effectively collaborate about a particular conveyance in terms of their own native, biased perspectives.

An Effective Development Process

Perspective models can be a powerful means of capturing and exploiting the expressive nature inherent in individuality. However, to arrive at an effective approach, such a method must be accompanied by a complimentary development process. Traditional approaches to domain model development have typically involved a dedicated knowledge engineer, or group of such individuals, whose task it is to produce a well structured representation of the target domain(s). Following creation of such a model, component developers design and implement functionality in terms of, or at least in a form that is compatible with, this representation. The problem inherent in this approach is essentially twofold. First, while model development is usually driven by a focused study of the domain this study typically does not include the specific use cases of its intended users. After all, the primary purpose of the representation sustaining a context-oriented, decision-support environment is to effectively support the data, information, and knowledge needs of its users. To ensure effective support of these activities, such implicit use-cases should be one-if not the most significant-force that drives model development.

The second pitfall of a conventional modeling approach also deals with the potential disconnect between a subject matter representation and its users. However, in this case the problem manifests itself at a more humanistic level. Critical to the successful application of an often fairly complex representation is the degree to which project team developers embrace, and are able to become familiar with, the various structure and semantics comprising the model. This is especially true in the case of reasoning-based, decision-support systems which tend to operate over complex, highly expressive contexts. To effectively exploit the expressive nature of context-enriched models requires developers to both understand such representation at a semantic level as well as embrace the manner in which it represents their subject matter interests. Many systems have fallen far short of their potential, sometimes to the point of complete failure, due to a lack of team member understanding and buy-in to the manner in which their domain(s) are represented.

The development process offered in this discussion addresses this disconnect by significantly increasing the involvement of model users with the actual model development activity itself. There are a number of benefits to such team member inclusion. First, as component developers research and design their solutions (i.e., software components), they essentially acquire a considerable amount of expertise and knowledge regarding relevant domain(s). Such familiarity goes beyond a fairly deep understanding of the semantics of relevant subject matter and includes valuable insight into the precise means by which particular functionality might most effectively view such content. It is the identification and subsequent capture of such individualized expression that produces a truly accurate representation. Since the focus is on capturing native perspective and bias, there is no need at this stage-in fact it would be potentially polluting-to be concerned with the degree to which these models align with each other. Narrowing the scope of individual Perspective Model development not only promotes the capture of true individuality, but is also a significantly less complex task than developing a singular, all-encompassing model supporting the entire set of interconnected perspectives (i.e., Universal Model). This less complex modeling environment has a direct impact on the amount of expertise and experience required for effectively developing these personalized Perspective Models. While good modeling practices are still quite important in this process, they can be applied within considerably less

complex environments by individuals who may not have the modeling depth of an experienced knowledge engineer. Further, familiarity with model structure and subsequent semantics undoubtedly leads to a significantly stronger bond between component developers and the subject matter representation over which their components operate.

Development of the Integration Model itself is a notably more involved task than that of developing the various Perspective Models. Development of the Integration Model involves the analysis of each Perspective Model with an eye for both identifying and abstracting subject matter existing across the multitude of user perspectives. Further, this subject matter must be modeled in a manner that maintains overall consistency and integrity as well as promotes expandability as additional inclusion of additional content is needed. Considering the complexities involved in this task, in addition to the demand for being both knowledgeable and comfortable with applying various intricate analysis patterns, this activity typically requires a highly experienced, expert modeler. As such, this activity might become the main area of focus for the expert knowledge engineer(s) who have traditionally been responsible for the entire modeling activity.

The final component to building the Integration Model is to describe the derivation logic that effectively ties the various Perspective Models with the central Integration Model. Coupled with some type of code-generation facility capable of managing implementation concerns, such derivation specifications can be designed, communicated, and maintained at the modeling level. Similar to development of the actual Integration Model itself, development of these mappings will likely also require the skills of an experienced knowledge engineer.

Conclusion

To obtain truly accurate, expressive representation, individual perspective must be specifically captured based on the use-cases of its immediate user(s). Interoperability within a diverse, perspective-enriched environment must support meaningful interaction between users that preserves this individualized perspective. Applying Perspective Models interconnected via a unifying Integration Model effectively supports these two objectives. Further, employing a development process where Perspective Model development directly involves the very users themselves leads to a more precise and expressive representation while significantly improving the representation's effectiveness through increased user familiarity and imperative model adoption.

References

[1] Cagle, K., M. Corning, J. Diamond, T. Duynstee, O. Gudmundsson, M. Mason, J. Pinnock, P. Spencer, J. Tang, A. Watt, J. Jirat, P. Tchistopolskii, and J. Tennison, "Professional XSL", Wrox Press Ltd,. Birmingham, UK., 2001

[2] Daconta M., L. Obrst and K. Smith, "The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management", Wiley, Indianapolis, IN., 2003

[3] Fowler, M., "Analysis Patterns: Reusable Object Models", Addison-Wesley, Reading, Massachusetts, 1997.

[4] Fowler, M., "UML Distilled: Applying the Standard Object Modeling Language", Addison-Wesley, Reading, Massachusetts, 1997.

[5] Fowler M., D. Rice, M. Foemmel, E. Hieatt, R. Mee, and R. Stafford, "Patterns of Enterprise Application Architecture", Addison-Wesley, Reading, Massachusetts, 2003

[6] Friedman-Hill, E., "JESS In Action", Manning Publications Co., Greenwich, CT, 2003

[7] Garshol L. and G. Moore (eds.), "The XML Topic Maps (XTM) Syntax", JTC1/SC34:ISO 13250, July 22, 2002, (www.y12.doe.gov/sgml/sc34/document/0328.htm)

[8] Giarratano J. and Riley G., "Expert Systems: Principles and Programming", 2nd Edition, PWS Publishing Company, Boston, MA.

[9] Hunter D., C. Cagle, D. Gibbons, N. Ozu, J. Pinnock, and P. Spencer, "Beginning XML", Wrox Press Ltd., Birmingham, UK., 2000

[10] Karsai G., "Design Tool Integration: An Exercise in Semantic Interoperability", Proceedings of the IEEE Engineering of Computer Based Systems, Edinburgh, UK, March, 2000

[11] NASA, "CLIPS 6.0 Reference Manual", Software Technologies Branch, Lyndon B Space Center, Houston, Texas, 1992

[12] Pohl J., "Information-Centric Decision-Support Systems: A Blueprint for Interoperability", Office of Naval Research (ONR) Workshop hosted by the CAD Research Center in Quantico, VA, June 5-7, 2001

[13] Pohl J, A Chapman, K Pohl, J Primrose and A Wozniak, "Decision-Support Systems: Notions, Prototypes, and In-Use Applications", Technical Report, CADRU-11-97, CAD Research Center, Design Institute, College of Architecture and Environmental Design, Cal Poly, San Luis Obispo, CA, January, 1997

Semantic-based Design Agents in Virtual Environments

Rabee M. Reffat
Associate Professor, Architecture Department,
King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
and
Yaman Khaeruzzaman, **I Putu Raharja**
Information and Computer Science Department, KFUPM
Dhahran, Saudi Arabia

Email: rabee@kfupm.edu.sa

Abstract

This paper presents a developed system of semantic-based design agents in virtual environments to support the architectural design of detached houses. The system consists of agents that have the capacity to recognize design semantics between building objects and their relationships while incrementally learn and update previous knowledge based on users' actions in creating houses in the virtual environment. The System provides designers with the capability to design interactively, test the consequences of actions and to explore different ways of solution refinement which are crucial in designing. The developed System of semantic-based design agents operates within a synchronous multi-users 3D VE namely the Activeworlds platform to support the process of architectural designing.

Keywords: *Virtual Environments, Multi-Agent System, Architectural Design of Houses, and Incremental Learning.*

1. Introduction

Virtual environments have become an alternative 'space' that supports many human activities including design of artifacts. 3D virtual environments incorporated with intelligent agents can be used in architectural design to provide an interactive design support system for collaborative assistance to human designers. In architectural design, building objects and their relationships are fundamental to the act of designing. Designers express ideas and represent elements of design using building objects to abstract concepts and construct situations. Hence, the role of building objects in design is significant. In architectural design, as in many other design disciplines, design composition is an important design activity. The formation and discovery of relationships among parts of a composition are fundamental tasks in designing (Mitchell and McCullough, 1995; Kolarevic, 1997). The relationships among building design objects can be geometrical or non-geometrical in nature. These relationships are called design semantics. Building objects and interrelationships between them are the contents of design semantics. Design semantics are considered as knowledge abstracted from existing entities. Building objects include walls, columns, floors, ceilings, windows, and shading devices while design semantics among these objects may include functional, spatial, esthetical and contextual values. This paper presents a developed system of semantic-based design agents implemented within a virtual environment to

support the architectural design of detached houses. The system consists of agents that have the capacity to recognize design semantics between building objects and their relationships while incrementally learn and update previous knowledge based on users' action. The system provides designers with the capability to design interactively, test the consequences of actions and to explore different ways of solution refinement which are crucial in designing.

2. Utilization of Agents and Virtual Environments in Design

The utilization of agents in virtual environments resembles the real world (or similar) inhabited by autonomous intelligent entities exhibiting a variety of behaviors. These entities can be a virtual representations of life forms (virtual animals and humans), avatars of real-world users entering the system, and others. In fact, the structure and contents of a virtual environment are only restricted by the nature of the target application and the designer's imagination, and the amount of computing power available. Designers are increasingly using virtual environments as platforms of communication and presentation of design intensions. Virtual environments are not only used in academic or professional settings but also for gaming and other consumer activities. Virtual environments are seldom used for creation, development, form-finding and collaboration of architectural design. Virtual environments which enable users active and real-time interactions with design have not yet been used widely for in the process of design. Virtual environments offer new opportunities and solutions to architectural design problems through their involvement in a three-dimensional 3D Virtual Design Environment medium (Schroeder et al, 2001). Recently, intelligent agents incorporated with virtual environments have been utilized to support various design activities.

In collaborative design of structures using intelligent agents, Anumba et al (2003) developed the ADLB system that investigates the use of agents in the collaborative design of light industrial buildings. The system provides peer to peer negotiation between the design agents encapsulated within the ADLIB prototype. It directly addresses the integration of multi-disciplinary perspectives and provides a framework for resolving design conflicts between members of a construction project team. The project examines some of the issues associated with the use of distributed artificial intelligence systems within the construction industry. It describes the potential for the use of agent technology in collaborative design and it presents the key features of an agent-based system for the collaborative design of portal frame structures. The distributed approach proposed in the system will allow individual areas of expertise to be encoded into particular agents, thus modeling the real-world problem of collaborative and concurrent design development in an intuitive, modular and hence expandable manner. In such an agent-based system, as compared with centralized knowledge-based systems, decisions can be taken locally according to local knowledge, allowing greater flexibility. Having agents communicate with each other across the Internet brings great increases in speed of convergence to a satisfactory design, compared with the traditional inter-disciplinary interactions. Some of the other benefits of this approach include: decentralization of traditional and inadequate project command and control structures; effective decomposition of large-scale problems; improved collaborative and concurrent working; and easier and cheaper access to routine specialist information especially as agent-based systems are made available on the World Wide Web.

In design visualization using agent-based virtual environment SYMBAD (Pinho et al, 2006) provides a seamless way of integrating the different architectural teams involved in designing and building and promoting information exchange and awareness of the process as a whole. The agents work with available information about the users' tasks and their current work and provide information on potential problems of the current design. The intent is to cause as little impact as possible on the way designers work, but to promote changes in their way of designing. Ideally, designers would learn about the consequences of their design choices and about the potential problems they may cause in the later stages of the project, and would design in a more informed way. The agents adopt awareness concepts to facilitate the work of the clients and to return only the most appropriate answers. Similar problems must be solved with solutions learned from past experience, so the agents must understand the semantic behind the objects manipulated by the workers to satisfy them accordingly. The semantic comprehension of the problem is built upon a strong ontology used by case-based reasoning component, which helps the agents in their decisions.

In multidisciplinary collaborative design Rosenman et al (2007) developed a collaborative VE for multidisciplinary design based on the need for extending the shared database to take into account the needs of the various views. The 3D virtual world environment provides real-time multi-user collaboration for designers in different locations and allows for the different design disciplines to model their view of a building as different representations. Relationships between the objects in the different models are seen as central to the maintenance of consistency and control while changing the design. It focuses on the extensions of a shared model required to address the following issues: different decomposition schema of the model among the collaborators; relationships environment, and allowing for real-time walkthroughs and collaboration. Agent technology is used to manage the different views, the creation and modification of objects in the 3D virtual world and the necessary relationships with the database(s) belonging to each discipline.

In enhancing the realism of exploring virtual environments using agents, Lam (2008) addressed the mechanism and challenges in the implementation of autonomous agents as actors in virtual environments so as to promote social sense and enliven the environment, hence enhancing realism in the exploration. Autonomous agents are used to populate the environment and induce socio-cultural practice which new users could follow that has resulted in having players preferring the presence of the autonomous agents as they promote the sense of place. Also, by reviewing agents' activities in the environment, players had much better ideas of how to perform (what to do and what can be done) in the environment. Bots in virtual environments are differentiated from non playing characters (NPCs). Bots are agents which have no direct interaction with players, but just exist to populate the environment. NPCs are special characters which players need to interact with. They follow a predefined interaction and conversation script so that they could react appropriately to the players.

The examples outlined above are some of many that reflect a diverse utilization of agents incorporated in virtual environments in the design field. Most of the previous work is useful from the explorative and collaborative perspectives but not as useful from the view of providing a semantic design support on the current design whenever required by the human designers which

is the focus of the work presented in this paper. Such design support is provided by the developed system of semantic-based design agents in the form of design advices responding to designers' request. This is conducted by the agents via observing, recognizing and conceptualizing designers' actions in the virtual environment and advising them according the knowledge available in the System's knowledge-base. Furthermore, agents modify both their behavior and semantic knowledge based on use. Hence, the initial knowledge-base of each agent will be modified and refined overtime based on designers' actions within the virtual environment. Therefore, the virtual environment becomes an intelligent and interactive design medium that relates and responds to what designers do while designing based on progressive and incremental learning of the semantic-based design agents.

3. A System of Semantic-based Design Agents in Virtual Environments

The framework of the implemented system of semantic-based design agents in virtual environments for supporting architectural designing is shown in Figure 1. The system is comprised of Observer Agent, Collaborator Agent, and a set of Design Support Agents that include functional, spatial, esthetical, climate, and context agents. The infrastructure included in the system implementation includes Activeworlds Galaxy Server, Web Server, and MySQL that are installed on a computer server. Designers use the Activeworlds client browser from their local computers to connect to the Activeworlds galaxy Server and the MAS-VDE. The Observer Agent (that operates on the Activeworlds Galaxy Server), detects and records all designers' actions in the MySQL database. Designers' actions include all placed design objects, and any deletion and modification of these objects attributes both graphically and non-graphically. The Observer Agent records all graphical and non graphical information of the design objects placed by each designer tagged with his/her ID in which each designer (user) has a designated ID.

At any stage of the design process, designers can decide to get some architectural semantic design support on their designs so far by clicking on the option of "Getting Design Advice" in the Activeworlds browser which triggers the Collaborator Agent. The Collaborator Agent presents to the designer a list of design support items that include functional, spatial, esthetical, context and climate to select from as shown in Figure 2. Based on users' request to get some design support in terms of advices regarding their designs so far, the Collaborator Agent distributes the inquiry of the required support to corresponding agents accordingly. The Collaborator Agent and Design Support Agents use Active Java Thread to communicate through method-parameter-passing that is read from global variables set by the Collaborator Agent. Global variables are variables defined to be accessible by all agents. Figure 3 illustrates an example the agents' communication logs within the Multi-Agents System Interface using Active-Java-Thread. Each Design Support Agent conducts its tasks based on the goals communicated by the Collaborator Agent. Whenever Design Support Agents complete their required tasks, they acknowledge the Collaborator Agent with their results that are stored in global variables. After tasks completion by all agents, the Collaborator Agent displays the results to corresponding users via PHP-interface at the Activeworlds client browser; an example is shown in Figure 4.

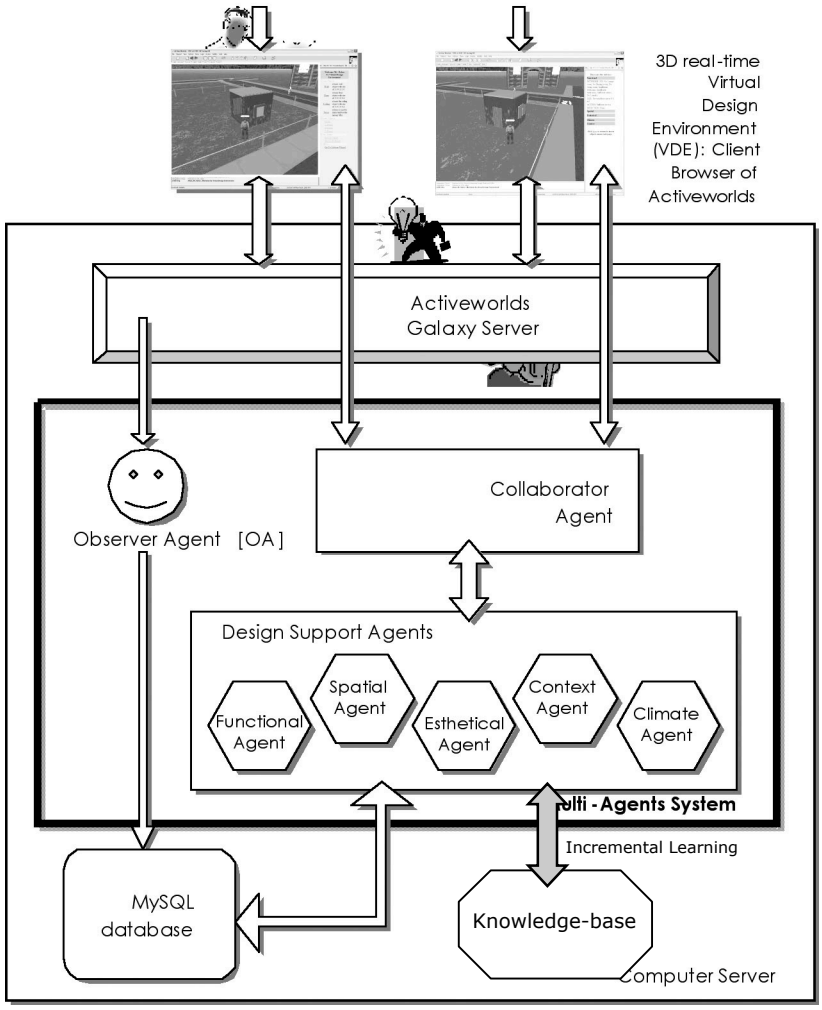


Figure 1. Framework of the implemented system of semantic-based design agents in virtual environments to support architectural designing.

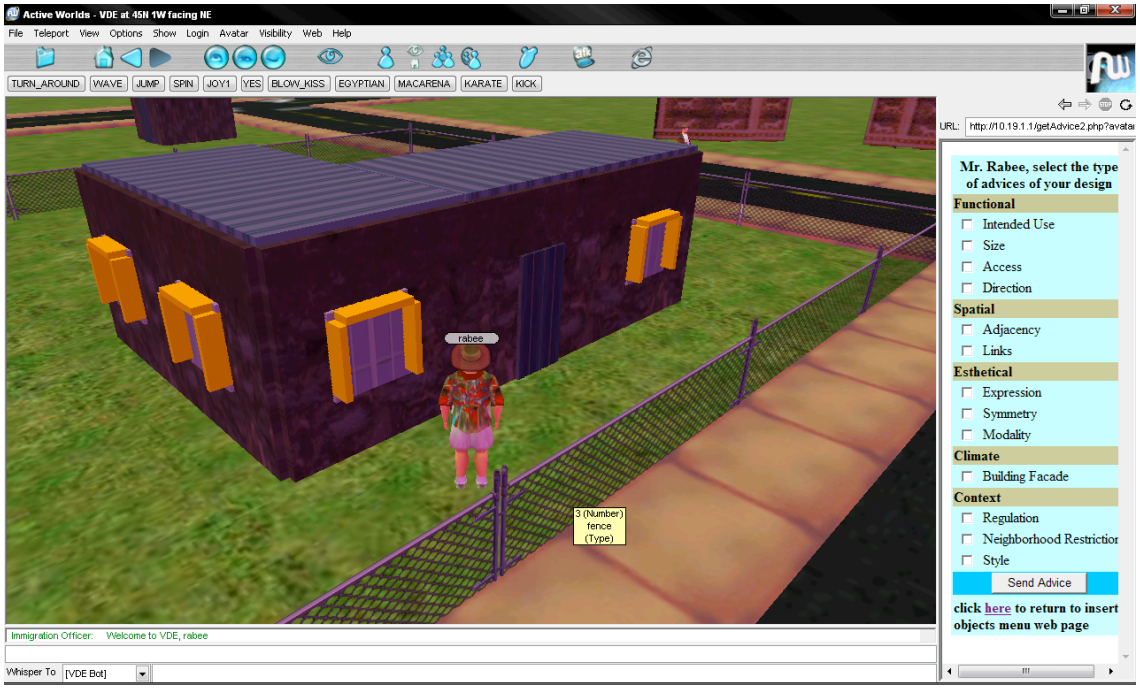


Figure 2. Types of design support (functional, spatial, esthetical, climate and context) that can be requested by the designers and results are presented to the user by the Collaborator Agent.

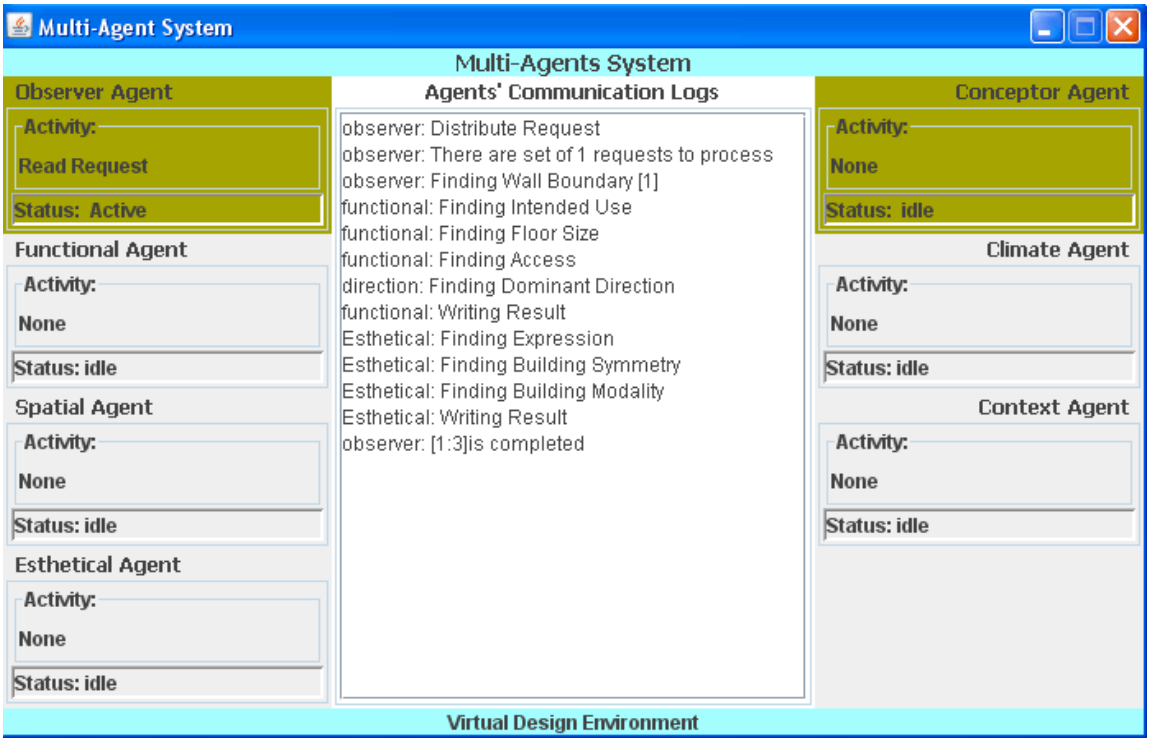


Figure 3. An example the Agents' communication logs of semantic-based design agents using Active-Java-Thread.

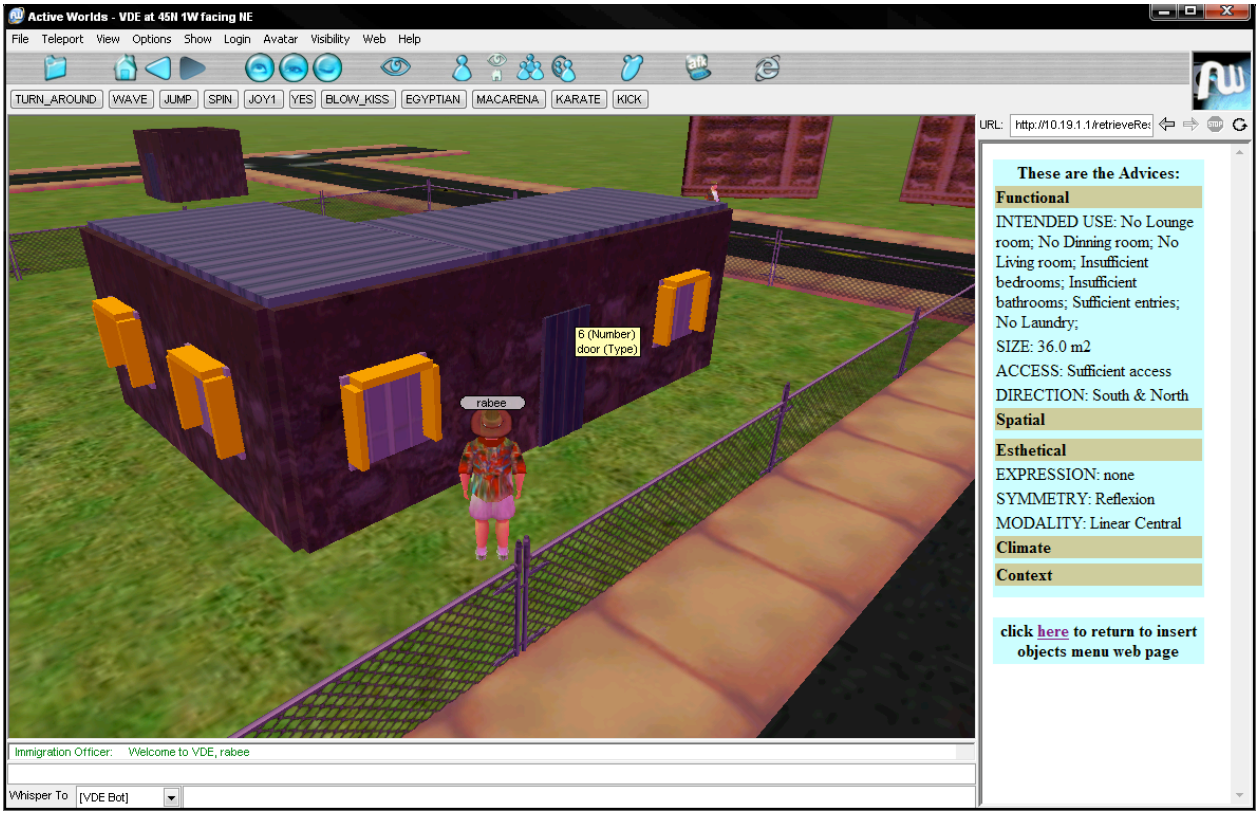
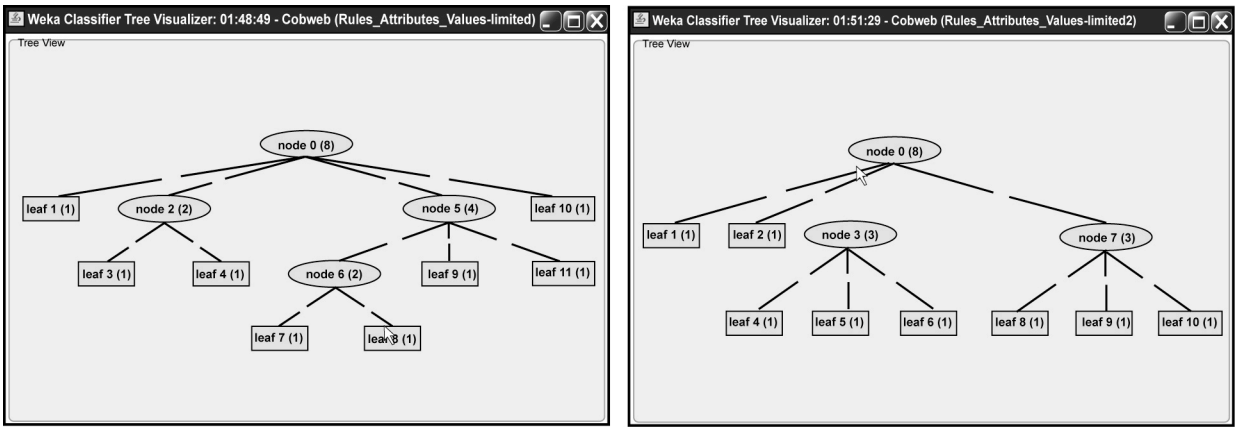


Figure 4. An example of the results (advices) provided and displayed to the designer by the Collaborator Agent based on designer's request on his/her architectural design at one stage of the design process.

There are five Design Support Agents in the implemented system of semantic-based design agents in virtual environments covering functional, spatial, esthetical, context and climate. The application domain of these Design Support Agents in the implemented system is the architectural design of detached houses within the context of Saudi Arabia. Hence, each Design Support Agent has a predetermined set of design knowledge applicable to the application domain. This knowledge has been extracted from extensive case analysis of various contemporary houses in Saudi Arabia. Such knowledge is stored in the System's knowledge-base.

The knowledge available in the System's knowledge-base is not fixed but rather can be either reinforced or decayed based on the new learned knowledge extracted from the designers' actions manifested in their various house designs constructed in the virtual environment. The Systems adopts the incremental learning algorithm of COBWEB implemented in WEKA software. Consequently, the System's knowledge-base is continuously revised based on designers' use of the System and so are the given advices provided by the Design Support Agents. An example of the System's capability of incremental learning is illustrated in Figure 5 wherein Figure 5 (a) presents the clustering of nine rules in the System's knowledge-base, while Figure 5 (b) presents a re-clustering of the same number of rules in response to the new knowledge extracted from the

results of designers' actions manifested in their created designs (instances) developed in the virtual environment.



(a) (b)

Figure 5. (a) Initial clustering of available knowledge in the System's knowledge-base; and (b) re-clustering of rules available in the System's knowledge-based in response to the new knowledge extracted from the results of designers' actions manifested in their created designs (instances) developed in the virtual environment.

4. Conclusion

A system of semantic-based design agents in virtual environments has been implemented to support the architectural design of detached houses. The developed system of semantic-based design agents operates within a synchronous multi-users 3D VE namely the Activeworlds platform. The system consists of agents that have the capacity to recognize design semantics between building objects and their relationships while incrementally learn and update their previous knowledge based on users' actions. Design support is provided by the developed system of semantic-based design agents in the form of design advices responding to designers' request. Design advices are based on both the current stage of the design and the knowledge available in the System's knowledge-base. Furthermore, agents modify both their behavior and semantic knowledge based on use. Hence, the initial knowledge-base of each agent is modified and refined overtime based on designers' actions within the virtual environment. Consequently, the semantics-based design agents transform virtual environments into intelligent and interactive design medium that relate and respond to what designers do while designing based on progressive and incremental learning capability of the semantic-based design agents.

Acknowledgement

The authors would like to acknowledge and thank the support of King Fahd University of Petroleum and Minerals (KFUPM) for funding this research through the Internal KFUPM Research Project No. INT-311.

References

1. Anumba, C. J. Ugwu, O. O., Newnham, L. and Thorpe, A. (2003); Collaborative design of structures using intelligent agents; Automation in Construction, Vol. 11 (1), (pp. 89-103).
2. Kolarevic, B. (1997); Regulating lines and geometric relations as a framework for exploring shape, dimension and geometric organization in design; in R. Junge (ed.), CAAD Futures 1997, Kluwer, Dordrecht, (pp. 163-170).
3. Lam, S. (2008); Enhancing Realism in Exploring in Virtual Environments; CAADRIA 2008 Proceedings of the 13th International Conference on Computer Aided Architectural Design Research in Asia, Chiang Mai, Thailand, (pp. 662-628).
4. Mitchell, W. J. and McCullough, M. (1995); Digital Design Media; Van Nostrand Reinhold, New York.
5. Pinho, D., Vivacqua, A., Palma, S. and Souza, J. D. (2006); SYMBAD: Similarity based agents for design; Expert Systems with Applications, Vol. 31 (4), (pp 728-733).
6. Rosenman, M. A., Smith, G., Maher, M. L., Ding, L. and Marchant, D. (2007); Multidisciplinary collaborative design in virtual environments; Automation in Construction, 16, (pp. 37 – 44).
7. Schroeder, R., Huxor, A., and Smith, A. (2001); Activeworlds: geography and social interaction in virtual reality; Futures, Vol. 33, (pp. 569–587).

Virtual Trading System: A Direction of Marketing in the New Age

Gahangir Hossain
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
Chittagong-4349 BANGLADESH
E-mail: gahangir@cuet.ac.bd, gahangir@gmail.com

Abstract

The existence of attractive shopping malls or trading centers with physical infrastructure is a matter of question for the next generation informational revolution. As the faster trading is established between consumer and producer by direct negotiation in Business to Consumer (B2C) business environment, it is the next generation marketing policy to design attractive web applications with sufficient web resources that exist in distributed virtual environment and ensures secure transaction of data among the participants in real time basis. The web applications must be more attractive and realistic as like the traditional shopping environments and the client can walk through, can touch the goods and can negotiate with the brokers' or the marketers. It must have to be secure enough so that it surmounts the problem of online transaction. It must use the policy of Private Information Retrieval; the customer should have the freedom to hide herself or her data to the remote database server. In this research some metrics for such virtual trading system is analyzed as a part of its feasibility study.

Keywords

Distributed Virtual Environment (DVE), Knowledge Based System, Expert System, Explanation System, International Trading System and Marketing Research.

Introduction

By definition, international trade is the exchange of capital, goods and service across international boundaries or territories. The major impact on the international trading system includes: industrialization, advance transportation, globalization, multinational corporations and outsourcing. The international trade is the major source of economic revenue for any nation that is considered a world power. Without international trade nations would be limited to the goods and services produced with their own borders. An efficient trading system with perfect security properties and real time transaction system is demanded nowadays. Many organizations find the markets they serve are dynamic and real time with customers, competitors and market conditions continually changing. And marketing efforts that work today cannot be relied upon to be successful in the next generation. Meeting changing conditions requires marketers have sufficient market knowledge in order to make the right adjustments to their frequently changing marketing strategy. For marketers gaining knowledge is accomplished through marketing research with new tools and techniques. As the driver of this generation ICT is reforming all strategies in every aspects of modern civilization. The ICT based marketing research is also

included with the future business world. The market places may be represented by distributed shared virtual environment (VE). A realistic trading took places among the negotiators through online-shared virtual environments (SVEs). Every process will go on in a faster way. Analyzing some VEs, their pitfalls and feasibilities, a basic structure of a SVE for trading system is proposed through the paper.
This paper is organized as following: Section 2 explains some terms and definitions related to the traditional system, marketing research as well as the virtual environments. It also discusses some traditional VEs with their merits and demerits. In section 3, the structure of the proposed VE is explained with some illustrations. A case study on a stock exchange and its representation through VE is briefly shorn in section 4. Finally the conclusion and future works are drawn in section 5.

The State of the Art

In recent years the evolution of marketing research has been dramatic with consumers getting access to a wide variety of tools and techniques to improve their chase for information. Because organizations recognize the power information has in helping create and maintain products that offer value, there is an insatiable appetite to gain even more insight into customers and markets. Marketers in nearly all industries are expected to direct more resources to gathering and analyzing information especially in highly competitive markets. Some of the trends discussed below are directly related to marketers' quest to acquire large amounts of customer in recent age, competitive and market information. Thus the trends of marketing research can be explained as below:

Internet Technologies for present market:

To address the need for more information, marketing companies are developing new methods for collecting data. This has led to the introduction of several new technologies to assist in the information gathering process. Many of these developments are Internet-based technologies that include:

- **Enhanced Tracking through Internet**– Recently the Internet offers the supreme ability to track and monitor customers by its attractive and intelligent websites. Each time a visitor accesses a website they provide customers with extensive information including how they arrived at the website via a search engine and what they did when on the website what products were investigated. In many ways the vast data available through Internet tracking has yet to be used by the majority of customers. However, as tracking software becomes more sophisticated the use of tracking data nowadays used as a research tool.
- **Internet Improves Communication** – Not only is the Internet enabling marketers to monitor customers' website activity, it also offers significant improvement in customer-to-company communication which is vital for marketing research. For instance, the ability to encourage customers to offer feedback on the company's products and service is easy using website popup notices and email reminders. Thus the use of the Internet for conducting online focus group research is expanding.

- **Internet Research Tools** – A large number of Internet services have added options for conducting research. These include the ubiquitous search engines, tools for conducting online surveys, and access to large databases containing previous research studies secondary research.

Other Future Technologies that rises the demand of ICT in marketing

In addition to the Internet, marketing research has benefited from other technological improvements including:

- **Virtual Reality and Simulations** - Consumers can use computer developed virtual worlds to simulate real world marketing activity such as store shopping. While this type of research is mostly performed in a controlled laboratory setting, there are emerging virtual worlds on the Internet where marketers can test concepts and communicate with customers. Our virtual trading is developed as part of the issue.
- **Global Positioning Systems** – GPS enables marketers to track inventory and even track mobile sales and service personnel. Soon GPS is becoming common feature of customers' communication devices, such as cell phones, offering marketers the potential to locate and track customers.
- **Data Analysis Software** – As we will see in the Planning for Market Research Tutorial, research includes gathering information and it also involves analyzing what is collected. A number of software and statistical programs have been refined to give marketers greater insight into what the data really means and probable changes.

This section describes some terms and definitions that are closely related to the deemed virtual trading environment. Some existing virtual environments are also discussed later on.

International Trading vs. Domestic Trading

The main difference between domestic trade and international trade is that international trade is more costly than the domestic trade. The reason is that the border typically imposes *additional costs* such as tariffs, time costs, due to border delays and costs associated with country differences such as language, the legal system or a different culture. The second difference is the *factor of production* such as capital and labors are typically more mobile within a country than across countries. Then trade in good and services can serve as a substitute for trade in factors of production. Instead of importing the factor of production a country can import goods that make intensive use of the factor of production and are thus embodying the respective factor. Research proved that, international trade represents a significant share of GDP in most countries. In a Virtual Trading it is deemed to reduce the costs of International Trades by informatics tools.

BPR vs. TPR

Neither the *additional costs* nor the *factor of production,* it is *the trading process* that has large impact in the information and communication technology (ICT) based next generation

civilization. Business Process Reengineering (BPR) is harder, technology oriented restructuring method that enables radical change but requires major change management skills. A Trading Process Reengineering (TPR) is thus demanded for virtual trading system. Roughly, two types of strategies: business strategy and IT strategy intuitively builds the TPR process. Factors that can be considered for TPR derived directly from a BPR method as below:

1. Is the competition outperforming the company by factors? For TPR process, this problem can be solved by the IT strategy. The competitive learning process of Artificial Neural Network can be used in this purpose.
2. Are there many conflicts among the organizations or the trading partners? The conflicts arise from the dissatisfaction of the bad properties of traditional trading process. The problem can be solved by the automatic negotiation of recently introduced artificial decision support system.
3. Is there an extremely high frequency of meeting? Organization of a meeting in remote place between trading partners located in different places takes more times and efforts for the only output of success or failure. This can be organized via a private video conferencing or private net meeting. Special software driven method can solve this problem.
4. Excessive use of non-structured communication? As for trading it is required for the agreement in paper-based materials, then these needs to send and receive over the mailing process. This takes the major transaction time before processing goods for shipping. Private Information Retrieval Method can be used to negotiate the decisions and to store the data in database server before the retrieval.
5. Is more continuous approach of incremental improvements not possible? This depends on the Kaizen philosophy, that is, every aspect of our life deserves to be continuously improved. That consist five fundamentals elements in its foundation: teamwork, personal discipline, improved morale, quality circles and suggestion for improvement. Kaizen is a daily activity whose purpose goes beyond simple productivity. This philosophy differs from command-and-control philosophy that is most popular in mid-twentieth century for large-scale industry management.

Distributed Virtual Environments (DVEs)

A Distributed Virtual Environment (DVE) is a place for people to meet and communicate naturally staying in scattered or remote places. The environment must contain sufficient real time resource and channels to communication through. Designing such a VE brings a host of complex problem. At first, Shusan et al. (1998) describes a design method for a VE integrating the Unified modeling Language (UML) and a system engineering processes and some pattern language for software development. They developed the case-use narrative analysis and pattern language, the exposed usability issues early in the processes and allowed the user to better define requirements for networking, interaction and content. They designed the virtual playground, the Netgate Mall with the support for their invented technology. A Distributed Virtual Music Environment is introduced by Byungdae et al. (2002). As the name of their institution their philosophy is named as PODIUM (POstech Distributed virtUal music environment). PODIUM allows the user to participate in a shared space and play music with other participants in a

collaborative manner. In addition to plying virtual instruments, users can communicate and interact in various ways to enhance the collaboration and thus the quality of the music played together. They have considered the network delay network delay that causes the considerable problems. They solved the problem by enhancing the feeling of participants by "co-presence" technology, a technology of real time message passing among the participants. Cagatay et al. (2001), experimentally showed the role of touch in shared virtual environment. Instead of multimodal virtual environment they have given attention on the human-human and human machine interaction for considering the VEs. Their goal was to assess the impact of feedback and task performance, to better understand the role of haptic communication between human-human interaction, to study the impact of touch on the subjective sense of collaborating with a human and reported by the participants based on what they could see and feel and to investigate if gender, personality, or emotional experiences of users can affects haptic communications in SVEs. In trading system the experiment is applicable to touch the products in trading process. A Case Study on the hazards in Distributed Virtual Environments System Design is explained by Manuel O. et al. (2002) the system was designed and found the Internet as the main culprit for disruptions in the experience of endusers. They also mentioned that the majority of the problems reside in the existing misconceptions regarding the nature of the Internet leading to inefficient and inadequate network subsystems. Moutzouris, M.(1998), introduces a software application, DataVis, which allows designers to visualize a three-dimensional environment from a two-dimensional CAD drawing. The application allows for the viewing and manipulating of electrical and physical properties of the objects in the environment. Distributed virtual reality software architecture is used to allow networked low-cost personal computers to create and allow the designer to interact with this environment. Tramberend, H.(1999) presented Avocado, their object-oriented framework for the development of distributed, interactive virtual environment applications. Data distribution is achieved by transparent replication of a shared scene graph among the participating processes of a distributed application by a sophisticated group communication system that is used to guarantee state consistency even in the presence of late joining and leaving processes. They also described how the familiar data flow graph found in modern stand-alone 3D-application toolkits extends nicely to the distributed case. *Wray, M. and R. Hawkes (2000)* presented an approach to the problem of implementing and supporting Distributed Virtual Environments (DVEs) on the Internet using an event-based notification system. The three most important characteristics of this approach are generality, scalability, and openness. They described the notification system, how we use it to provide general DVE support, its use in implementing the Living Worlds Virtual Reality Modeling Language (VRML) DVE architecture, and an application in an office environment.

Some existing internet base marketing polices are: the www.*vse.marketwatch* describes the Stock market game and message boards, trade stocks and funds in a collaborative environment. Run your own competition or join others. A VME explained in www.*vme.net* builds community on the World Wide Web by designing and linking your pages to your community and more. *www.pbs.org* Plays a Virtual Market by Rob Meyer While cleaning the cushions of your couch; you find an old gold pocket watch that you don't recognize. The official number of exhibitors in the IFA Virtual Market Place does not correspond with the official number of IFA exhibitors is shown in *www.virtualmarket.ifa-berlin.de*.ITB Virtual Market Place Berlin

www.virtualmarket.itb-berlin.de describes the internet based business policies. A resource guide to agricultural market information *www.aec.msu.edu* selected sites organized by commodity, region or on market analysis. BigBarn *www.bigbarn.com* is the Virtual Farmer's Market. *www1.messe-berlin.de* is IFA Virtual Market Place where you can find an information and communication platform allowing you to access up-to-date information about all exhibitors

Private Information Retrieval

The increasing model of e-commerce infrastructure opens the door for secure transaction of information over the net, keeping some records private as users' choice within a few seconds. A client while frequently retrieves his records seldom wish to hide the identity of the records to the database server. Private Information Retrieval (PIR) protocols allow users to retrieve information from a database while keeping their query private. Existing protocols are Theoretical PIR, Computational PIR, Hardware based PIR (Secure Co-processor), PIR with pre-processing and off-line communication, Almost Optimal PIR, Tagged PIR, and recently introduced PIR with P-cache by Hossain and Haque (2008) with improved performance. The protocol is suggested to use in the virtual commercial environment to ensure the secured information retrieval privately in multi-server distributed system.

The Virtual Trade Market (VTM)

An environment suitable for trading in virtual system may name as Virtual Trade Market. It is being considered in the distributed virtual environment and rigorously termed as Distributed Virtual Trade Market (DVTM).

The Strategy:

Just like a virtual 3D game! The shopping malls or trade centers are created. All information about products or goods are authenticated and directly mapped (indexed or linked) by justifying their availability and visibility from suppliers or from warehouses where these are stored previously. It is possible to negotiate with the marketer via conversation. The conversation is as if as a real conversation, as a virtual person on behalf of the buyer will present in the virtual negotiation room. A virtual negotiation room is well furnished web based room where every trading partner has the access right if he is allowed for negotiation about his business trade. The game is as like as the traditional game but reengineered in different fashion by online real time basis. A complete buying process includes as below steps:

(1) Bayer can log on the virtual center
(2) Bayer can render in the different shops to choose her expected products
(3) Bayer is allowed to negotiate about the product's quality and the price
(4) Bayer feels free and enjoys freedom to walk through the center
(5) Bayer can take refreshments from any fast food shop or wait where he wants on the chairs in the corridors.

(6) Bayer can park her car or associates in any secured zones insight the mall or center where it is allowed.

The operations close to a VTM are summarizes as: transaction management system, components visualization and tracking, fixed and non-fixed trade implementation, shipping management, virtual negotiation algorithms, virtual sales persons, security and authentication of marketer and the user, Indemnification of different user types, product crisis management in web site to warehouse update, Virtual payment system using VISA and Master card and Virtual amusement center for enjoyment of customers in trading. These are explained briefly later on:

Transaction management system

To make business transaction of all types of goods or products a policy is developed as shown below figure-1:

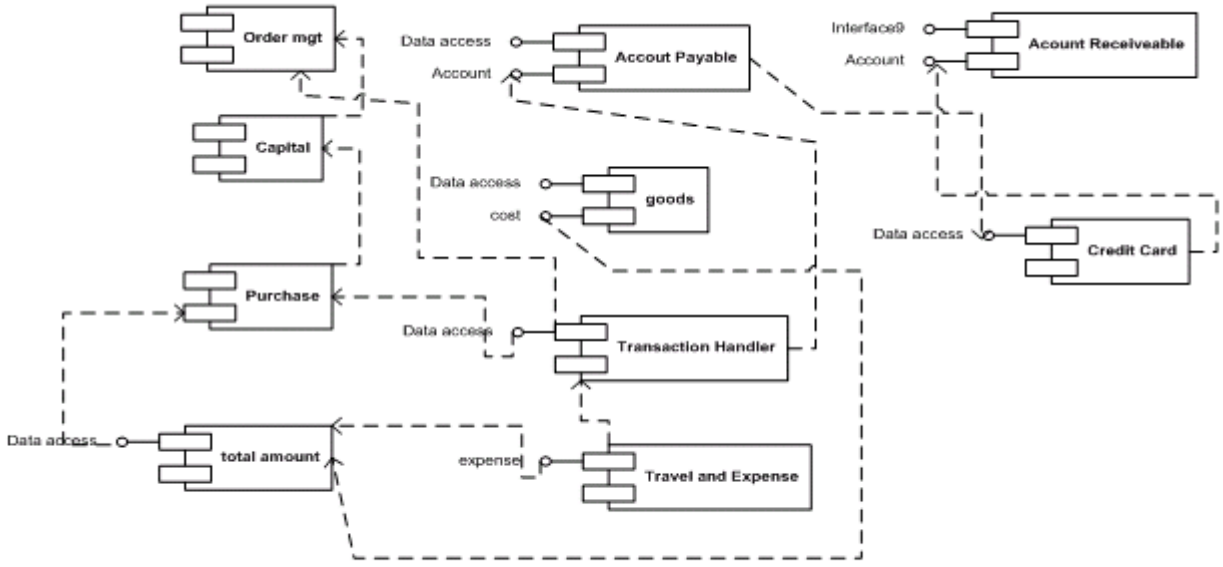


Fig-1: Transaction management System

The transaction management in a VTS is composed form of the transactions between order management, capital management, payable accounts and transaction travel management

.

Components visualization and tracking

The visualization and tracking of components in a virtual environment must be realistic. A component shown in the corresponding 3D website must have to be visual and colorful as it is in the natural. Tracking a product means the interaction of the products availability as far the real time request of a user. Instantaneous videos of the product can solve the requirement. But the major problem is the transformation of the real time videos data of the product from its origin (for example apple collection from an apple garden)

Difference between fixed trade and non-fixed trade in VE can be made by designing separate algorithms with collaboration with the negotiation algorithm. A problem occurs whenever a user sees a product in the virtual shop but it is not actually available. This is termed as product crisis. This is due to the failure of real time data transfer. This can be solved by dedicated communication channel and smart software for management.

Virtual negotiation system between customer and marketer

If a product is being shown by a site but in the mean time it is not possible to supply from its warehouse then it is required to update the site as a real time basis. The management software must be smart enough to update the available information about a product.

Product Shipment policies in VTS

As like the direct shipment the transaction order is placed in the store or warehouse as a real-time basis, instantly. There is no hassle to wait for any office order an automatic transaction by route (virtual transaction route (like the DHL/FedEx/UPS etc. online tracking system).

Security management

A VTM is secured enough from the vindictive persons in the places but in a trap of the hackers. Some mechanism for the hacker protection must include with the whole system. User authentication and payment system for VTM are also a matter of consideration in the security policy.

Virtual Payment system

As for the payment of any products must have to be ensure before transaction, a fastest, authentic and secured fund transfer system is suggested to use along with VISA or Master card policy.

Virtual sells person or user

A part of VTS design, the complete visualization associates the visualization of the partners or the persons with whom we are negotiating. A virtual environment always has the criteria to focus the visualization properties of the environment as if it is natural. Designing a visual sales person is moderated from a normal user updated with the smart negotiation algorithms. It is being observed that multiple users may try to buy a product at a same time then the system must satisfy them concurrently managing in a real time basis.

Others

There are many other reasons over trading why we go to a shopping malls or trading centers. Some are for learning the marketing policy or simply for enjoyment. Virtual amusement park may be designed as an alternative of enjoyments in a VTS. As in traditional system there is a lot of amusement suites included with the systems like shopping mall or trading centers. A policy must be account for consideration to make the virtual persons in enjoyment. The learning can be simulated by learning the use of whole program.

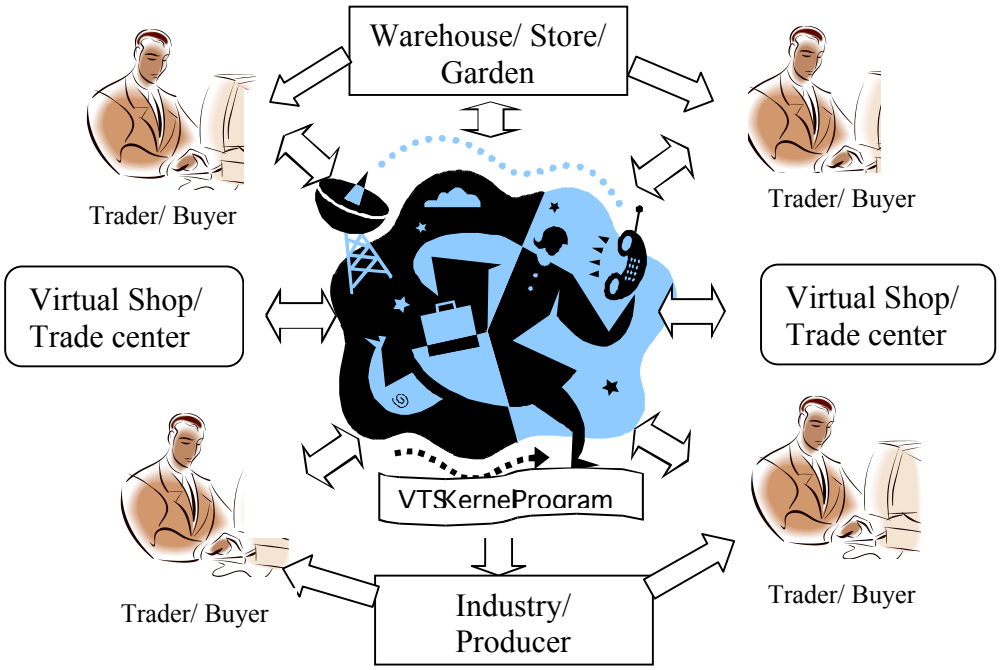


Figure-2: The Trading System Structure

Hardware and Software required for a Virtual Room in a Virtual Trade Center

The hardware required for the virtual trade center is as imagined as like the hardware essential to create live distributed virtual environment. It is assumed that the system is as realistic as it allows the participant to touch the products or each other (hand shaking) through touching the haptic devices with their computer system in distributed shared environment.

The hardware components thus include:

IBM compatible PC (at least Duel Pentium II 300 MHz processor) with high-end 3D graphics accelerator for visualization for the visual objects and a PHANToM (Sensible Technologies Inc.) to simulate haptic sensation and other multimedia devices.

A software module includes:

1. Multithreading techniques for integration of vision and touch
2. Haptic Rendering Techniques
3. User/ Marketer Interface
4. Other modules for transaction, shipping and participants negotiations etc.

These are being considered for the process in order to develop in a shared distributed system.

An example that can be implemented

A Virtual Online Stock Brokerage System

Virtual stock brokerage system is an online stock brokerage system that automates the traditional stock trading using computers and the internet, making the transaction faster and cheaper. This system also gives faster access to stock reports, current market trends and real time stock prices. The system utilizes the three-layer architecture, consisting of a front-end system, middleware and back-end system, as explained in the following figure 3.

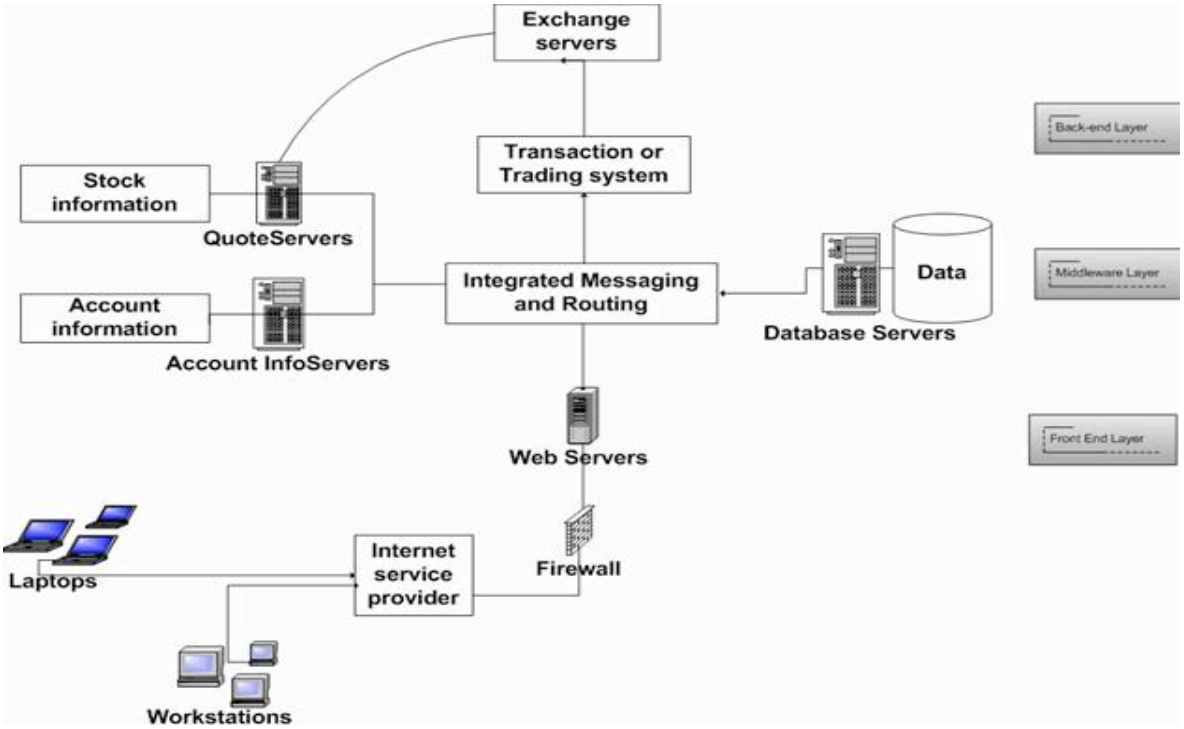


Figure 3: Virtual Stock Brokerage System Architecture

The system design is accomplished using the three-tier architecture explained above and applying the systems engineering principles. Initially use case analysis is done based on the customer/stake holder requirements. From the use case analysis, high-level system requirements are generated and are further synthesized and broke down into low-level requirements. A system structure model consisting of all system components and the interfaces through which components interact with each other is developed. Then the system behavior model is developed. By mapping the chunks of system behavior onto system structure, a simplified model of the logical system is developed. And then the system tradeoff analysis is carried out to come up with various design alternatives from the system specifications, after allocating requirements to the objects and attributes of system structure.

Initial Use Case Modeling

The use cases represent system goals or system functions. A uses case is an abstraction of a system response to external inputs, and accomplishes a task that is important from user's point of view.

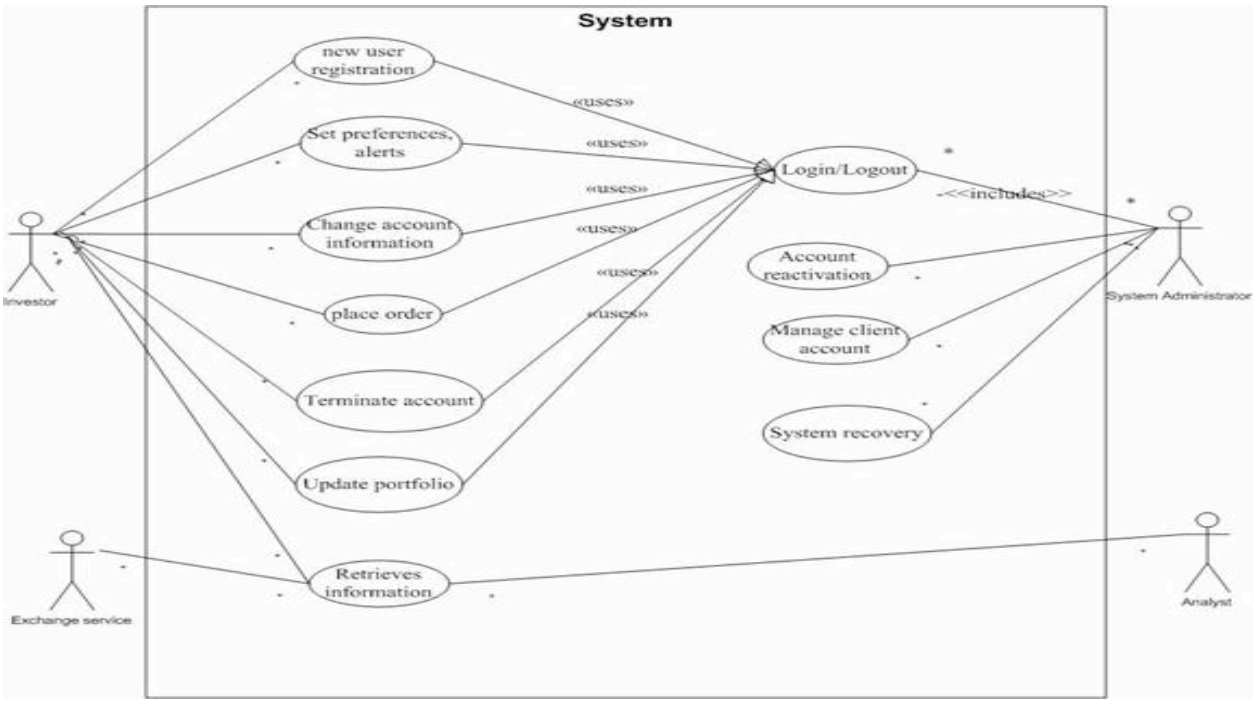


Figure 4: Initial use-case diagram of Virtual Brokerage System.

Activity modeling

Investor gets his/her login id and password, which can be used to use the services of the brokerage system.

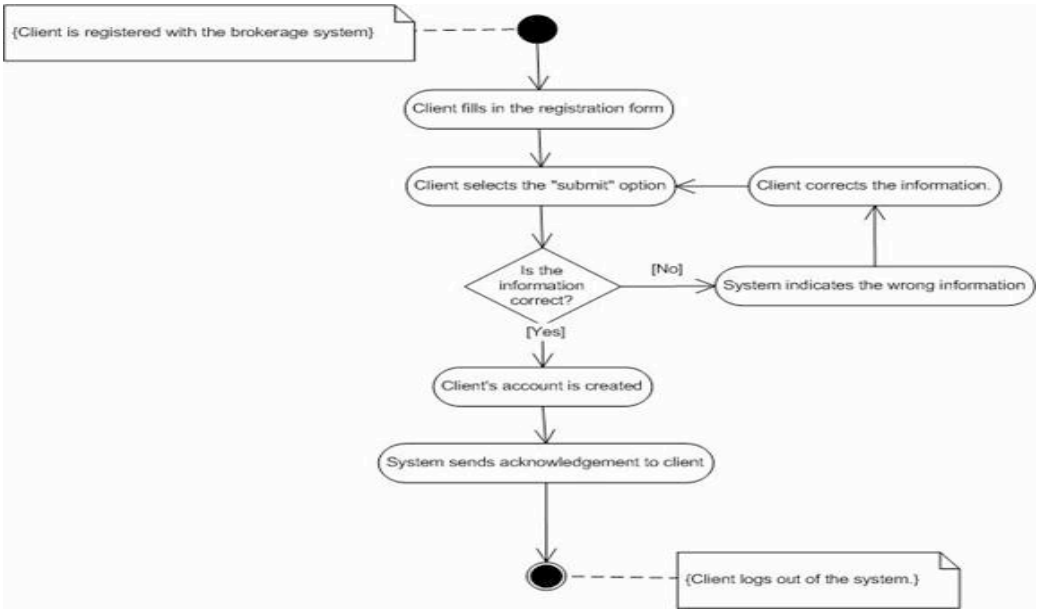


Figure 5: Activity diagram for New User Registration.

If the investor's login and password are not correct, system prompts the investor to reenter the data with a proper message. If the investor forgets his password, "forgot password" option is provided to the investor Post-condition. Investor's account is displayed.

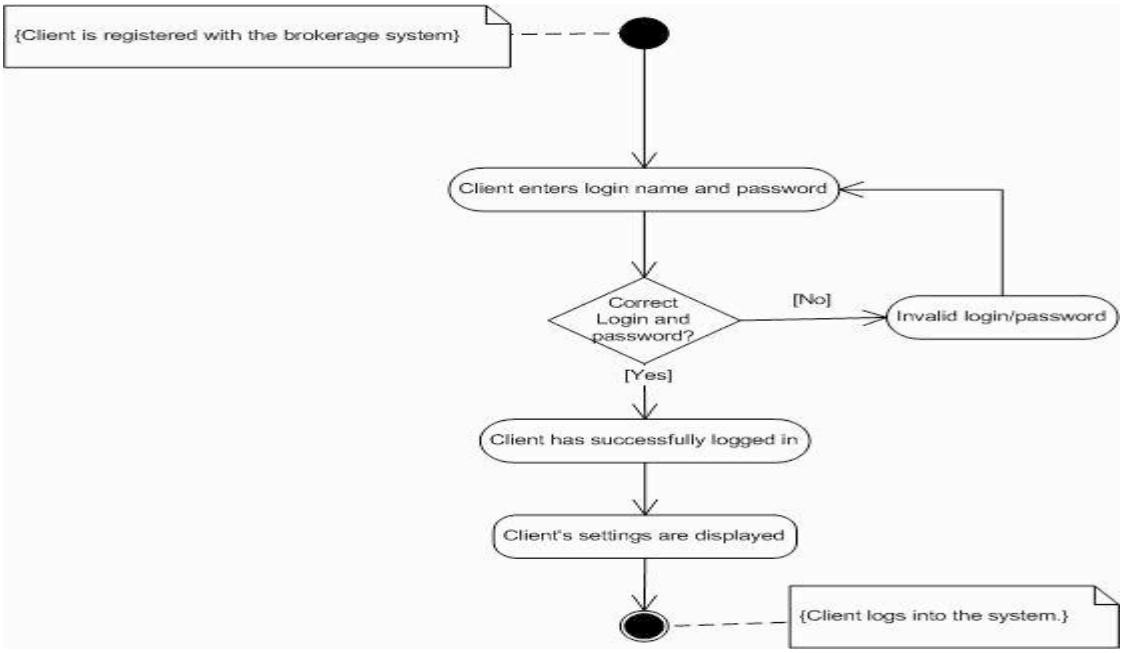


Figure 6: Activity diagram for "Login".

Investor is logged out successfully and has to login again if he wishes to use the services.

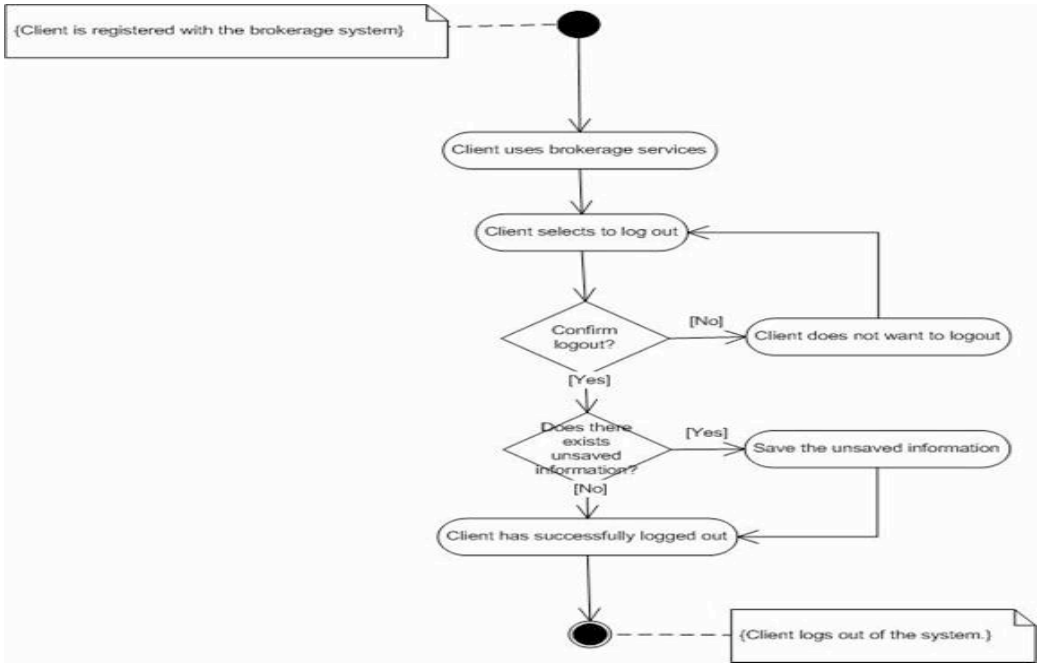


Figure 7: Activity diagram for Logout

Investor has placed an order and that order is recorded in the database.

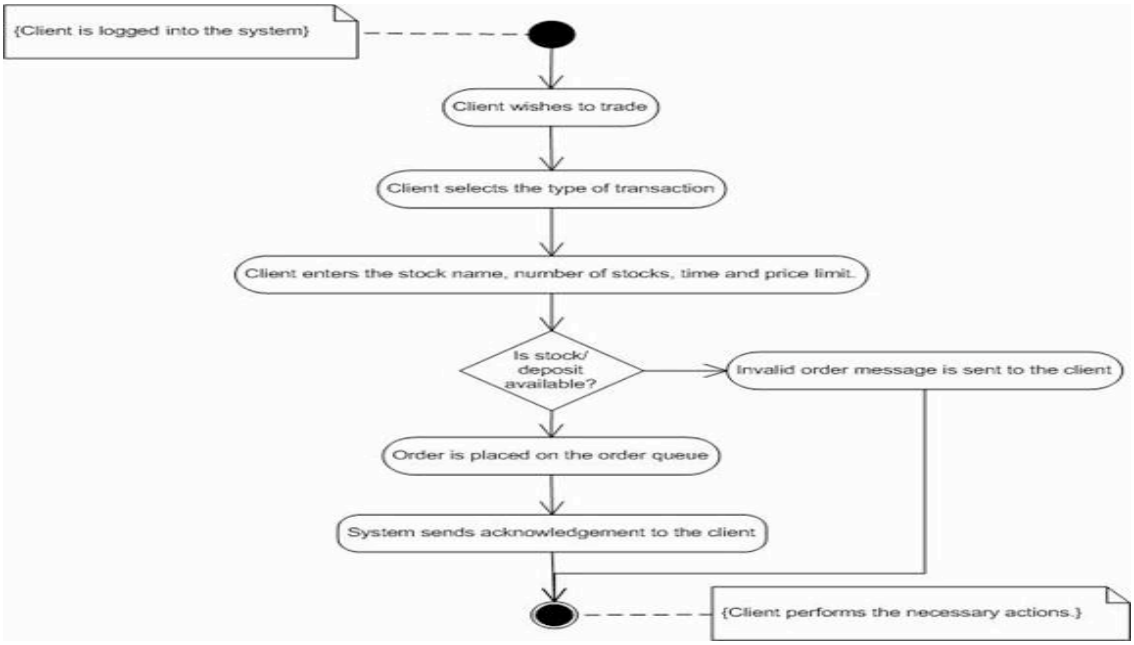


Figure 8: Activity diagram for "Place an order".

Class diagram:

The online brokerage system in virtual environment structurally modeled with sub-system hierarchies as shown in the following class diagram.

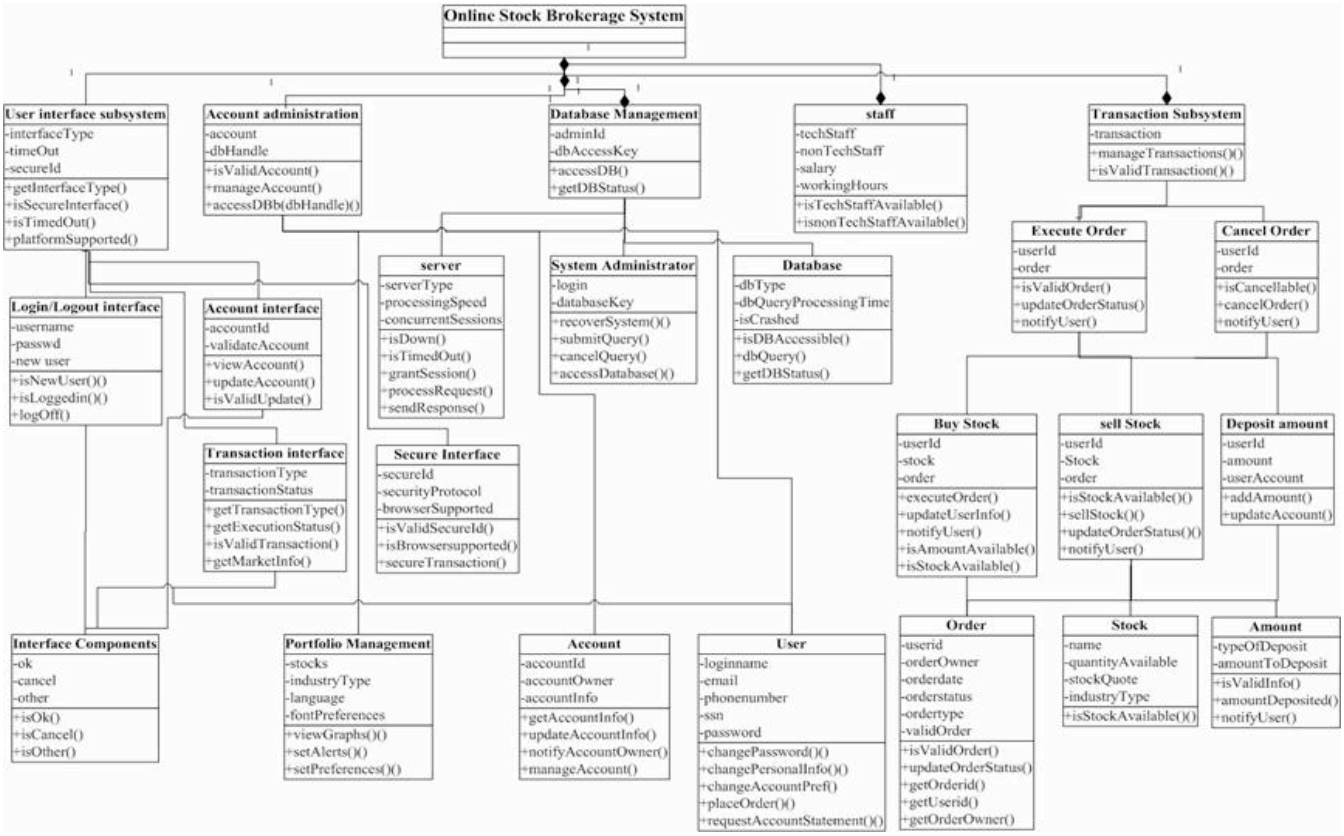


Figure 9. Class diagram for the structural model of the online stock brokerage system

Sequence Diagram

Detailed System Behavior for Client account activities:

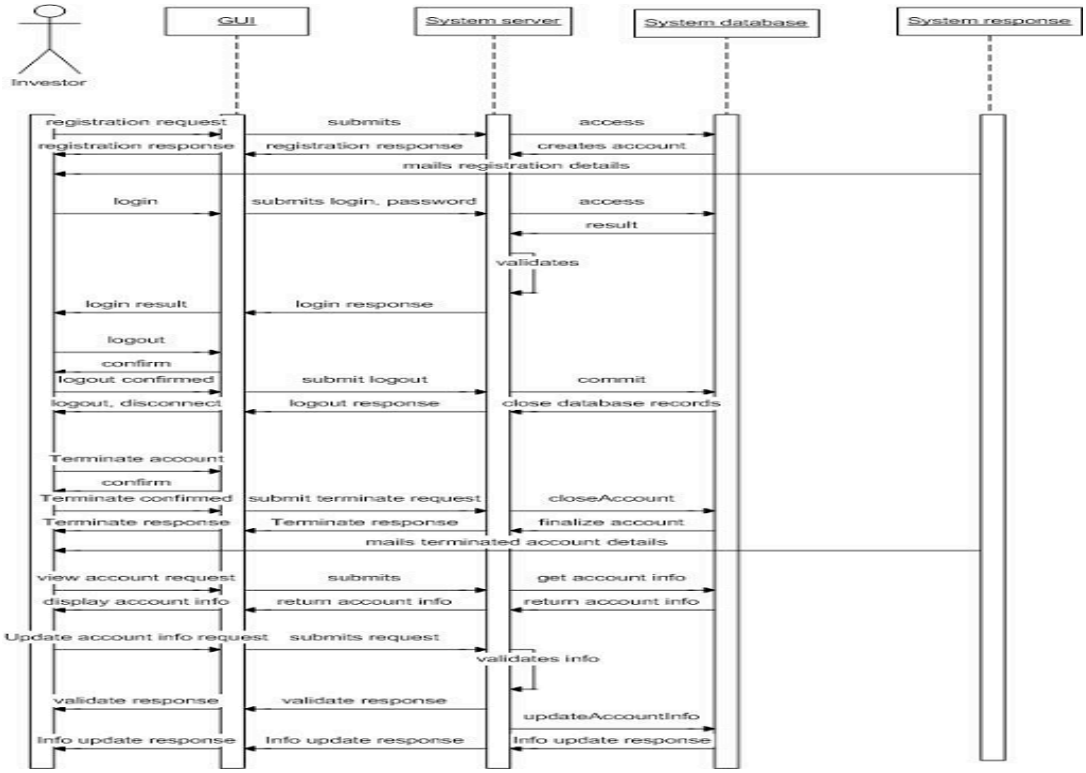


Figure 10. System behavior for Client Activities

By this way all UML modeling of the Virtual Stock Brokerage System can be developed and simulated for test its applicability in real life as a alternative abut faster way of marketing.

Conclusion

This is a proposed virtual system. The experimental design and feasibility analysis of the system is left for the next phase of the subsequent research. We hope that the system will overcome the limitations specially the security factors involves in traditional system. But it lacks the natural environment for enjoyment as we have nowadays walking through a shopping mall or a large trade center. It is obviously suitable for the persons very busy with multiple daily works and expects the trading jobs will be as an automatic manner. Keeping in mind this thought that the next generation peoples will be tremendous busy in their routine works this proposal is thrown for adaptation.

References

Basdogan, C.,Ho,C., Srinivasan, M.A. and M. Slater (2000); An Experimental study on the Role of Touch in Shared Virtual environments; ACM Transaction of Computer-Human Interaction, Vol.7, No.4 (pp. 443-460).

Casaneuva, J. and E. Blake (2000); The Effects of Group Collaboration on Presence in a Collaborative Virtual Environment; Proc. of the Eurographics VE Workshop, 2000.

Hossain G. and A. S. M. L. Haque (2008); "PIR with P-Cache: a New Private Information Retrieval Protocol with Improved Complexities", accepted to publish in the Malaysian Journal of Computer Science (ISSN 0127-9084).

Http://www.vse.marketwatch.com/
Stock market game and message boards. Trade stocks and funds in a collaborative environment. Run your own competition or join others.

Http://www.vme.net/ -VME builds community on the World Wide Web by designing and linking your pages to your community and more.

Http://www.pbs.org/wgbh/nova/stockmarket/virtual.html
Play a Virtual Market by Rob Meyer While cleaning the cushions of your couch, you find an old gold pocket watch that you don't recognize.

Http://www.virtualmarket.ifa-berlin.de/
The official number of exhibitors in the IFA Virtual Market Place does not correspond with the official number of IFA exhibitors

Http://www.virtualmarket.itb-berlin.de/
Welcome to the ITB Virtual Market Place Berlin, March 11 - 15th, 2009. May 11, 2008. Hauptnavigation überspringen.
Http://www.aec.msu.edu/fs2/market/contents.htm
Resource guide to agricultural market information: selected sites organized by commodity, region or on market analysis.

Http://www.bigbarn.co.uk/
BigBarn - The Virtual Farmer's Market. Use BigBarn.co.uk to find your local food producers!

Http://www1.messe-berlin.de/vip8_1/website/MesseBerlin/htdocs/www.ifa-berlin/en/VirtualMarketPlace/index.html - 9k At IFA Virtual Market Place you will find an information and communication platform allowing you to access up-to-date information about all exhibitors.

Jung, B., Hwang, J., Kim, G.J, Lee, E. and H.Kim (2000) Incorporating Co-presence in Distributed Virtual Music Environment; Proceedings of the ACM symposium on Virtual reality software and technology, (pp. 206 – 211).

Kim, G., and J. Hwang (2000); Design and Analysis of Virtual Music Environment, Proc. of Intl. Computer Music Conference, Berlin.

Kim, G. and J. Hwang (1999), Musical Motion: A Medium for Uniting Visualization and Control of Music in the Virtual Environment, VSMM.

Miner, N. and T. Caudel (1998); Computational Requirements and Synchronization Issues for Virtual Acoustic Displays, Presence; Vol 7, No 4.

Moutzouris, M., and B. Trangh (1998); Visualization of Transformer Zones using a Distributed Virtual Reality Architecture; In *Proceedings of the Seventh International WWW Conference (WWW7)*, Brisbane, Australia Vol. 1 (pp.1-9).

Slater, M. et al (1996), Immersion, Presence and Performmace in Virtual Environments: An Experiment with Tri-Dimensional Chess, Proc. of VRST.

Steur, J. (1992), Defining Virtual Reality: Dimensions Determining Telepresence, Journal of Communication, Vol 42, No. 4.

Towell, J. and E. Towell (1977), Presence in Text Based Networked VE or "MUDs", Presence, Vol 6, No 5.

Thomas, B. Sheridan (1992), Musings on telepresence and virtual presence, Presence: Teleoperators and Virtual Environments, Vol.1 No.1, (pp.120-126).

Tramberend, H.(1999);Avocado: a distributed virtual reality framework; Virtual Reality Proceedings, IEEE Vol. 102 (pp.14 – 21).
Wray, M. and R. Hawkes (2000); Distributed Virtual Environments and VRML: an Event-based Architecture; Bristol, BS12 6QZ, UK.

ICODES: A Multi-Agent System in Practice

Jens Pohl, Ph.D.
Executive Director, Collaborative Agent Design Research Center (CADRC)
California Polytechnic State University (Cal Poly)
San Luis Obispo, California, USA

Abstract

This paper describes the Integrated Computerized Deployment System (ICODES) from both an architectural and evolutionary vantage point.

First, ICODES is a logistic software application of ship load-planning tools that utilizes intelligent software agents in a human-computer collaborative mode. As an example of a new generation of intelligent military decision-support systems, ICODES includes expert agents with automatic reasoning and analysis capabilities. This is made possible by an internal virtual representation of the load-planning environment, in terms of conveyance and cargo characteristics and the complex relationships that constitute the context within which load-planning operations are performed. ICODES agents monitor the principal determinants of cargo stowage, including: the placement and segregation requirements for hazardous cargo items; the trim, list, stress, and bending moments of the conveyance structure; the accessibility of stow areas through ramps, cranes, elevators, hatches, and doors; the correct placement of cargo items in respect to fire lanes, no-stow areas, reserved stow areas, and inter-cargo spacing tolerances; and, the accuracy of cargo characteristics (e.g., dimensions, weight, type, and identification codes) relative to standard cargo libraries and associated reference tables.

Second, ICODES is a system that has evolved over the past 10 years and is continuing to evolve from a stand-alone application focused on the load-planning of ships to a distributed environment capable of addressing the assembly and planning for any kind of surface or air conveyance. This transition from single domain to multiple domains and from stand-alone to distributed has been made possible by a scalable service-oriented architecture that emphasizes a multi-layered, multi-tiered design approach.

Background

In 1996, ICODES was selected as the *migration* system for ship load-planning by the United States (US) Department of Defense (DoD). It has been deployed by the US Transportation Command (TRANSCOM) through the Military Traffic Management Command (MTMC)[1] to the US Army since 1999, and the US Marine Corps since 2002. Other users include the US Navy and the British Army. ICODES currently interfaces with several external sources that provide it with cargo data, including the World-Wide Port System (WPS) for the US Army, the Transportation Coordinators' Automated Information for Movement System (TC-AIMS II) for

[1] MTMC was renamed in 2004 as the Surface Deployment and Distribution Command (SDDC).

several military services, the MAGTF Deployment Support System (MDSS II) for the US Marine Corps, and the Integrated Booking System.

In 2007, ICODES was designated by the Distribution Steering Group (DSG), co-chaired by TRANSCOM and the US Joint Forces Command (JFCOM), as the Single Load Planning Capability (SLPC) for all conveyances with a planned release date of 2010. In this new role ICODES Global Services (GS) will be required to integrate and provide seamless access to communities of planners that have previously operated in separate and largely autonomous domains. The fragmented nature of these logistic enclaves has promoted data quality and exchange problems that have manifested themselves in multiple failure points, leading at times to severe supply chain inefficiencies.

The multiple objectives of the SLPC initiative include the following: to improve the flow of data in a multi-modal transportation and distribution environment; to ensure in-transit visibility from origin to destination; to support collaborative planning efforts; to accelerate the staging, loading and unloading of supplies; to reduce labor requirements through automation; and, to increase throughput without sacrificing in-transit visibility during surge periods. These are indeed ambitious objectives considering the number and diversity of personnel involved and the enormous quantity of supplies and equipment involved. For example during 2005, in its assigned role of DoD's Distribution Process Owner, TRANSCOM and its component commands moved over 2.34 million short tons of cargo and more than 1.1 million passengers.

Load-Planning as a Complex Problem

The rapid deployment of military assets from the US to overseas locations is a complex undertaking. It involves the movement of large numbers of tracked and wheeled vehicles, weapon systems, ammunition, power generating and communication facilities, fuel, food supplies, and other equipment and goods, from military bases to the area(s) of operation. Several modes of transportation are typically involved. Depending on the location of the military base the assets are preferably moved by road to the nearest railhead, from where they are loaded onto railcars for transportation to the appropriate air or ocean port of embarkation.

Alternatively, if rail transportation is not an option, all of the cargo must be shepherded through the public road corridor from the base to the port. At the port of embarkation the assets are briefly assembled in staging areas and then loaded onto aircraft or vessels for shipment. Points of debarkation may vary widely from a commercial air or ocean port with fairly good facilities to a secure airfield in the theater or an amphibious landing on a hostile shoreline under fire. Once the cargo has been disembarked in or near the theater it must be transported to its final destination by road, rail, air, or barge. In many cases this becomes an inter-modal affair with the need for frequent re-planning due to changes in priority or as routes in the theater become temporarily unavailable due to inclement weather conditions or enemy activities.

Speed and in-transit visibility are of the essence (Figure 1). The total time required for the loading and unloading of the conveyance is a critical factor and largely determined by the quality of the load-plan. Ship load-planning, for example, has many of the characteristics of a complex problem situation (Figure 2). First, there are continuous information changes. The vessel that arrives at the port may not be the vessel that was expected and that has been planned for. This

means that the existing load-plan is no longer applicable and a new plan has to be developed. Similarly, last minute cargo changes or inoperative lifting equipment may require the existing plan to be modified or completely revised. Second, there are several complex interrelationships. The cargo on any one ship may be destined for several ports of debarkation, requiring careful consideration of loading and unloading sequences. However, these sequences must take into account unloading priorities that may be dictated largely by tactical mission plans. In addition, the placement of individual cargo items on board the ship is subject to hazardous material regulations and practices. These regulations are voluminous, and complex in themselves. At times they are subject to interpretation, based on past experience and detailed knowledge of maritime risks and practices. Finally, the trim and stability characteristics of the ship must be observed throughout the planning process. This includes listing, draft and deck stress limitations.

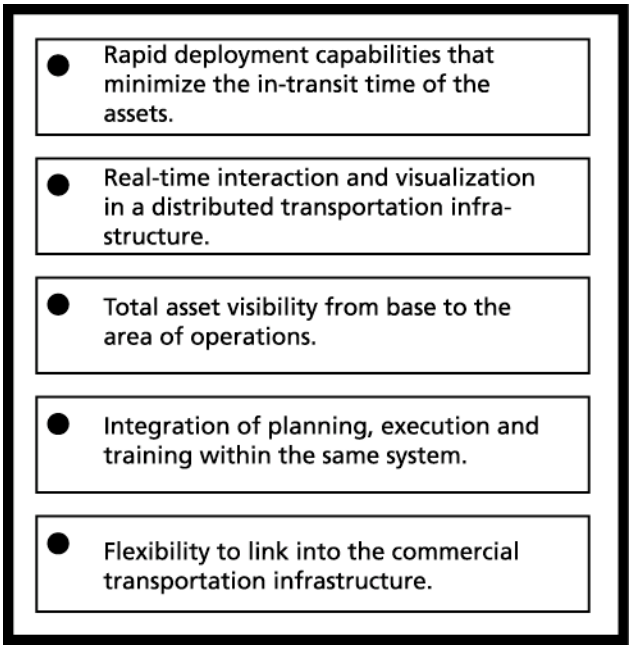


Figure 1: Military deployment objectives

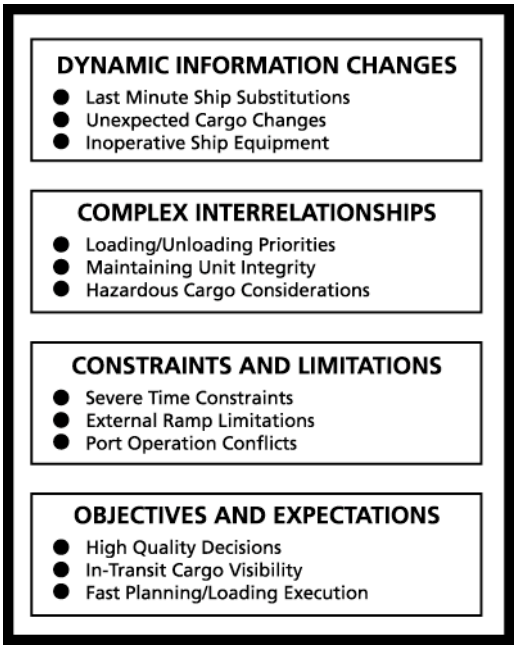


Figure 2: Complexity of ship load-planning

Third, there are many loading and unloading constraints. Some of these constraints are static and others are dynamic in nature. For example, depending on the regional location of an ocean port external ship ramps may not be operable under certain tide conditions, or an airfield may be able to accommodate only a small number of aircraft concurrently on the ground for loading purposes. Local traffic conditions, such as peak hour commuter traffic and rail crossings, may seriously impact the movement of cargo into staging areas or from staging areas to the pier or aircraft loading area. While these constraints are compounded whenever loading operations occur concurrently, the general complexity of the load-planning problem is exacerbated by the number of parties involved. Each of these parties plays an important role in the success of the operation, but may have quite different objectives. Certainly, the objectives of the commercial stevedore crews that may be under contract to carry out the actual loading tasks are likely to differ markedly from the prevailing military objectives that include rapid loading and unloading operations, safety, unit integrity, load density, documentation accuracy, and security.

Initial Functional Requirements

Several general and specific operational and technical objectives were specified by the military sponsor (MTMC) at the beginning of the project in 1994, when ICODES was conceived solely as a stand-alone ship load-planning application. Foremost, it was the vision of the sponsor that ICODES should present itself to the user as a set of collaborative and expert tools, rather than a conglomeration of predefined solution templates. Experience had shown that the problems encountered in the real world of ship load-planning are driven by dynamically changing factors that are often unpredictable. Accordingly, any predetermined solutions based on preconceived requirements were unlikely to adequately address the nuances of the cargo stowage problem encountered under actual operational conditions.

From a general operational viewpoint the ICODES application was required to be magnitudes faster than the existing DOS-based ship load-planning application. It should allow the concurrent planning of four ships, provide the user with continuous assistance in the form of alerts and warnings throughout the load-planning process, incorporate an automatic cargo placement capability, link to several external systems but be capable of operating in a stand-alone mode, and offer a friendly and flexible, graphical user-interface that could be customized by the user to suit individual needs.

Specifically, the ICODES application was required to automatically alert the user of cargo placements within stow areas that are in violation of hazardous material mandates, the trim and stability requirements of the ship, deck strength limitations, or a host of cargo stowage rules such as adjacency tolerances, fire lanes, boom clearances, and movement restrictions (e.g., door and hatch dimensions, crane lifting capacities and reach, ramp and elevator constraints, and stow area heights). For example, in the hazardous material domain these specific objectives required ICODES to be capable of differentiating among the internationally recognized nine classes of hazardous materials, and the sub-groupings or divisions that exist in five of these classes. In addition, ICODES was required to interpret and apply the regulations prescribed in the following four principal reference sources:

The 49 Code of Federal Regulations (49 CFR) that specifies segregation requirements for hazardous cargo shipments in the Continental United States (CONUS).

The International Maritime Dangerous Goods (IMDG) library that applies to all international shipments of hazardous materials.

The Department of Defense Identification Code (DoDIC) library that applies specifically to Class 1 hazardous items (i.e., explosives), namely munitions.

The Dangerous Cargo Manifest National Stock Number (DCMNSN) library that is used primarily by the Marine Corps for identifying and load-planning hazardous cargo items.

Technical Objectives and the Development Environment

The general technical objectives established for ICODES in 1994 included the requirement of an open architecture, the ability to add new and enhance existing user-assistance capabilities over the lifetime of the application, the ability to add future modules to support related functional

areas such as inter-modal transportation (i.e., air, rail, and truck convoys) and the management of staging areas), as well as the ability for the user to create cargo lists and vessels within the application if these were not available within ICODES and could not be imported from existing external sources.

Like most of the planning and decision-support systems developed by the CADRC Center over the past decade ICODES was designed as a suite of Knowledge Management Enterprise Services (KMES®) and implemented within the Integrated Cooperative Decision Making (ICDM) software environment[2]. ICDM is an application development framework for distributed decision-support systems incorporating software agents that collaborate with each other and human users to monitor changes (i.e., events) in the state of problem situations, generate and evaluate alternative plans, and alert human users to immediate and developing resource shortages, failures, threats, and similar adverse conditions. A core component of any ICDM-based application is a virtual representation of the real world problem (i.e., decision-making) domain. This virtual representation takes the form of an internal information model, commonly referred to as an ontology. By providing context (i.e., data plus relationships) the ontology is able to support the automated reasoning capabilities of rule-based software agents.

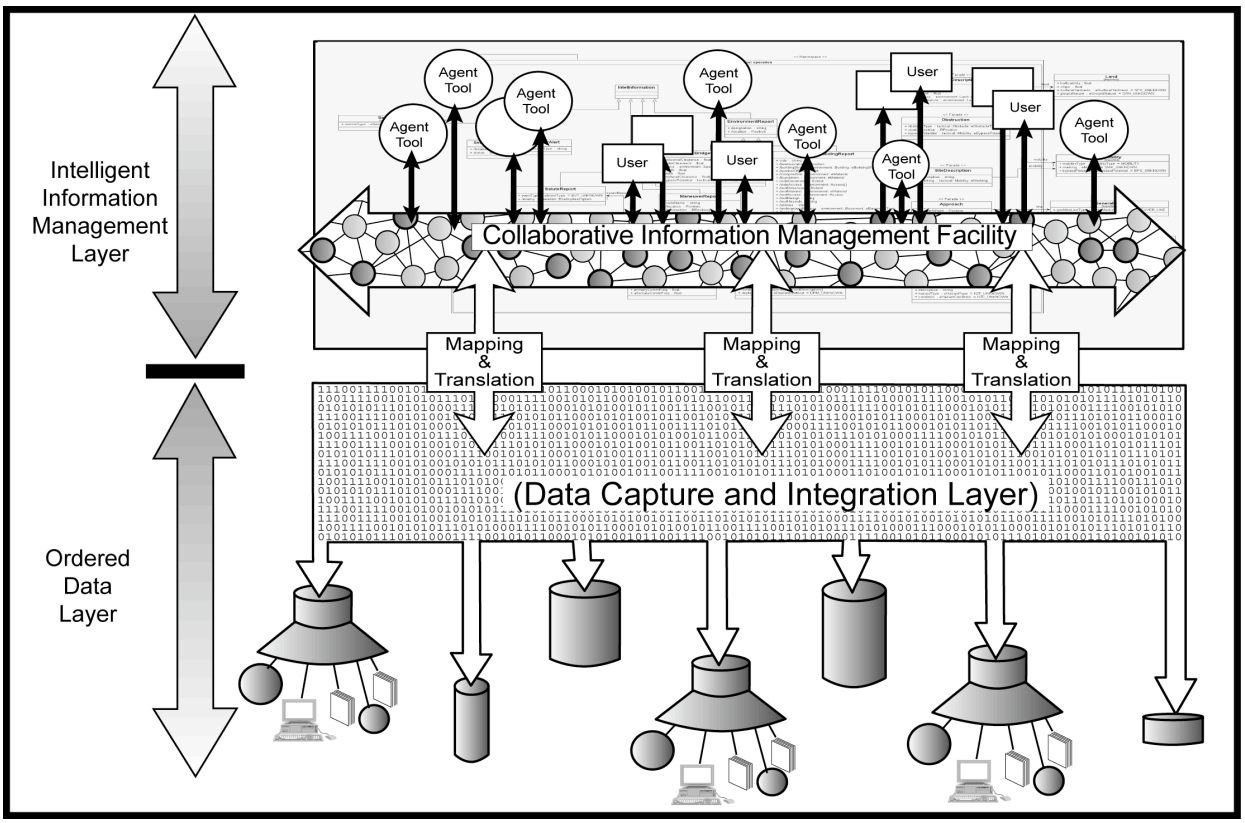


Figure 3: Conceptual KMES-based net-centric architecture

Knowledge Management Enterprise Services (KMES®) are self-contained software modules with clearly defined functional capabilities and interface specifications. They are designed to be

[2] ICDM is a software development toolkit that is proprietary to CDM Technologies, Inc. and available to third parties under licensing agreements (Pohl et al. 2004).

platform independent and to be reusable. Some of these services may have quite narrow capabilities such as the mapping of cargo data imported from an external source to an internal information model or ontology, while others will incorporate larger functional domains such as the optimum placement of cargo of multiple dimensions in a given space, such as a stow area on board a ship or in a marshalling yard.

From a general point of view the KMES® approach to software systems incorporates intelligent agent technology to provide an internal staff of software agents. These agents analyze and categorize incoming signals and data, and then issue warnings and alerts as appropriate. The agents manipulate the incoming data within an internal information-centric representation framework to publish statements of implication, and if so empowered, proceed to develop plans for appropriate action. Legacy data-centric systems can become clients of such an agent-based KMES® software environment through the use of interoperability bridges that map the data model in one system to the information model of the other and allow a two-way exchange of data. Conceptually, a KMES®-based application consists essentially of two component layers (Figure 3): a data-centric Data Capture and Integration Layer that incorporates linkages to existing data sources; and, an Intelligent Information Management Layer that resides on top of the data layer and utilizes software agents with automatic reasoning capabilities, serving as decision-support tools.

A multi-tier architecture is used to logically separate the necessary components of the data layer into levels. The first tier is the data repository, which ensures the persistence of the data level and provides the necessary search capabilities. The second tier is the service level, which provides the interface to the data level and at the same time supports the data access requests that pass through the mapping interface from the Intelligent Information Management Layer to the Data Capture and Integration Layer. It is designed to support request, response, subscribe, and publish functionality. The third tier is the control level, which routes information layer and user requests to the service level for the update, storage and retrieval of data. Finally, a view layer representing the fourth tier serves as a user-interface for the Data Capture and Integration Layer.

The Intelligent Information Management Layer consists of KMES® components in the form of a group of loosely coupled and seamlessly integrated decision-support tools. The core element of each KMES® component is an ontology that provides a relationship-rich model of the particular decision-support domain. Typically, KMES® components are based on a three-tiered architecture incorporating technologies, such as distributed-object servers and inference engines, to provide a framework for collaborative, agent-based decision-support that offers developmental efficiency and architectural extensibility. The three-tiered architecture clearly distinguishes between information, logic, and presentation. Most commonly an information tier consists of a collection of information management servers (i.e., information server, subscription server, etc.), while a logic tier incorporates an agent engine, and a presentation tier is responsible for providing interfaces to human operators and external systems.

The notion of service-oriented is represented as much in the elements of each of these tiers as it is in the functional capabilities of each KMES®. Therefore, even the internal elements of a KMES® communicate through standard interfaces as they provide services to each other. They are, in essence, decoupled software modules that can be replaced with improved modules as the

technology advances. Each of these modules functions in an integrated fashion to form a comprehensive agent-based decision-support execution framework (Pohl 2007).

The ICDM Software Development Framework: For the past two decades the CADRC Center and more recently CDM Technologies have pursued the design and development of agent-based decision-support systems[3] utilizing the ICDM software development toolkit. Not only does ICDM function as an accelerator (i.e., rapid development) and stabilizer (i.e., built-in robustness and fault tolerance) in the development of decision-support systems, but it also provides a concrete vehicle for representing the key concepts and philosophies that the CADRC Center and CDM have found to be useful for the success of KMES®-based systems (Pohl et. al 2004, Pohl 1997). The key design principles on which ICDM is founded are collaboration-intensive, context-based representation, flexibility and adaptability, multi-tiered and multi-layered, within the framework of a service-oriented, distributable architecture. An ICDM-based application is based on an information-centric premise, in the sense that it incorporates an internal information model of objects, their characteristics, and the relationships that associate these objects to each other and the functional capabilities of the application (Myers and Pohl, 1994; Pohl et al. 1992; Pohl K. 2002).

The term information-centric refers to the representation of information, as it is available to software modules, not to the way it is actually stored in a digital machine. This distinction between representation and storage is important, and relevant far beyond the realm of computers. When we write a note with a pencil on a sheet of paper, the content (i.e., meaning) of the note is unrelated to the storage device. A sheet of paper is designed to be a very efficient storage medium that can be easily stacked in sets of hundreds, filed in folders, folded, bound into volumes, and so on. As such, representation can exist at varying levels of abstraction. The lowest level of representation is wrapped data. Wrapped data consists of low-level data, for example a textual e-mail message that is placed inside some sort of an e-mail message object. While it could be argued that the e-mail message is thereby objectified it is clear that the only objectification resides in the shell that contains the data and not the e-mail content. The message is still in a data form offering a limited opportunity for interpretation by software components.

A higher level of representation, commonly referred to as an ontology, endeavors to describe aspects of a domain as collections of inter-related, constrained objects. This allows context to be captured and represented in a manner supportive of software-based reasoning. Apart from their role as services, however, distributed behavioral objects can also be employed as a mechanism for supporting the notion of facades. As one of the fundamental patterns employed in object-oriented design, facades provide a level of derivation attained from the particular representation or ontology on which they are based (Pohl K. 2001). In the case of ICDM and the kinds of ontologies it manages, facades offer a method of supporting and managing an alternative perspective from that modeled in the ontology from which they are derived. In other words, ICDM-based facades allow the perspective inherent in a particular model of a domain to be augmented, or in some way altered to support a more appropriate (i.e., to the façade user) representation of the concepts, notions, and entities over which that user is operating.

[3] CDM Technologies, Inc. is the commercial arm of the Collaborative Agent Design Research Center (CADRC) at California Polytechnic State University (Cal Poly), San Luis Obispo.

Facades can also be utilized to support real-time calculations. In this sense, the façade derivation would involve a calculation or algorithm perhaps based on one or more attributes of the base object(s). For example, consider a stow area on board a ship with its length, width, and height dimensions described in American pound/foot units that is to be accessed by a planning service that understands only Metric kilogram/meter units and also requires space volumes. Utilizing ontology-based facades a model can be easily developed in which, not only the length, width, and height, but also the volume of the space are calculated and presented to the planning service in terms of Metric units. Although there are a number of approaches to supporting calculated attributes in the case where an alternative perspective is to be supported, the façade approach permits an extensible (i.e., one perspective extended from another) and encapsulated (i.e., easily maintainable) solution.

One of ICDM's primary goals is to support a high degree of flexibility in respect to the configuration of its components both at the development and execution levels. ICDM supports the addition, replacement, and reuse of software components in the context of agent-based, decision-support systems, and achieves this goal by reducing inter-component coupling to an absolute minimum. Two key ICDM properties permit this flexibility. First, all collaboration between clients takes place via, and in terms of the informational ontology (i.e., distributed objects). No direct communication exists between collaborators. The result is a collaborative environment in which client identities are essentially irrelevant in respect to this process.

The second property deals with the manner in which clients access and interact with the ontology. ICDM offers a standard interface component known as the Object Management Layer (OML) that both shields accessing clients from the complexity of ontology management as well as providing an abstracted view of the ontology. Clients of OML interact with the ontology via object wrappers based on a set of corresponding ontology-specific templates. Promoting the notion of adaptability, these templates are discovered by OML as a runtime activity. The resulting support for dynamic definition permits elements of the ontology to be extended, eliminated, or even redefined during the course of a runtime session.

From an architectural organizational point of view ICDM strictly adheres to the principle of separation between areas of functionality at both the conceptual (i.e., tier) level and the more concrete (i.e., layer) level. Conceptually, the architecture of an ICDM-based decision-support system is divided into three distinct tiers namely, information, logic, and presentation. To manage its particular domain each tier contains a number of logical layers that work in sequence. As the name suggests the information tier houses both the information and knowledge (i.e., ontology) being operated on, in addition to all of the mechanisms needed to support management, transport, and access. The information is further delineated into layers. The first of these is the OML described above. Below the OML resides the Object Access Layer (OAL) responsible for managing access to the information tier. The OAL exists as a level of abstraction below OML and interfaces directly with the Object Transport Layer (OTL). Based on the CORBA specification (Mowbray and Zahavi 1995) the OTL is responsible for communicating the various requests and subsequent replies for distributed information and behavior issued through the OAL throughout the system. The OTL is the only layer that forms a dependency on an underlying communication protocol. As such, support for alternative communication facilities can be implemented with minimal impact on either the OAL or the OML. This exemplifies the benefits of a layered architecture in supporting component reuse and replacement.

The Logic Tier contains the business rules (i.e., agents) and analysis facilities by which these rules are managed. Although extensible to include other forms of reasoning the current version of ICDM focuses on opportunistic rule-based analysis. Regardless of which form of reasoning is employed this capability is supported by two layers namely, the Business Rule Layer (BRL) and the Business Engine Layer (BEL). The BRL is primarily system-specific and contains the agent-based analysis facilities resident in the system. Execution of agents is in turn managed by the BEL. To integrate the Logic Tier with the Information Tier the BEL interfaces with OML permitting the agents to both access and contribute to the ontology.

The final tier is the Presentation Tier. This tier is responsible for interfacing with the various users of the system. In this sense a user may be a human operator or an external system. In the case of a human operator support is provided through a Graphical User Interface Layer (GUIL) that presents and promotes interaction with the contents of the Information Tier. In the case of an external system, support takes the form of a Translation Layer that manages the mapping of representations between systems. Like the GUIL, access to and from the Information Tier is supported by OML.

User-Interface and Functionality

Implemented in a typical Windows 2000 operating system environment the main screen of ICODES Version 5.2 is shown in Figure 4, as consisting of six components or sections.

(1) The ***Main Menu Bar*** provides access to the nine principal ICODES option groups in the form of pull-down menus.

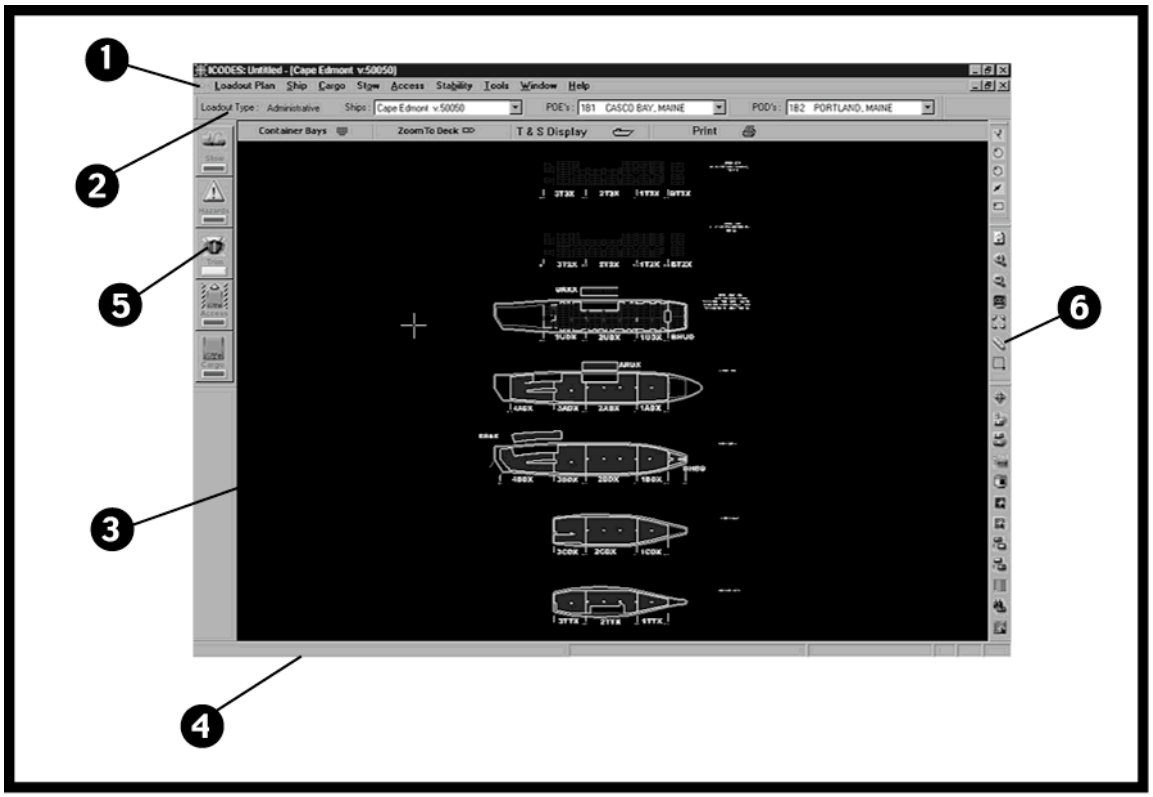


Figure 4: The Main Screen of ICODES Version 5.2 (Released August 2002)

(2) The ***Loadout Banner*** provides information about each of the currently displayed load plans such as plan type(s), ship name(s), ports of embarkation and debarkation, and the measurement units used in each plan.

(3) The ***Graphics Window*** displays the ship drawing(s). It can accommodate multiple ships, with the number of ships that are concurrently displayed limited only by the constraints of the screen size and the memory capacity of the computer.

(4) The ***Message Window***, found at the bottom of the main screen, provides the user with messages relating to the current status of ICODES (e.g., the status of an option selected by the user, or instructions relating to the use of a particular tool).

(5) The ***Agent Status Bar*** on the left side of the main screen provides access to agent reports and explanations of warnings and alerts.

(6) The ***Tool Bars*** on the right side of the main screen contain three groups of tools: stow tools (e.g., rotate, flip, unstow individual cargo items); view manipulation tools (e.g., zoom, pan); and, drawing tools that allow the user to superimpose lines, circles, polygons, and rectangles, on a displayed ship drawing.

ICODES offers a very comprehensive set of editing, saving, restoring, reporting, and special operations options (MTMC 2002). In addition, ICODES recognizes the differences among tactical (emphasizing mission accomplishment), pre-positioning (accommodating the maintenance requirements of pre-loaded regionally positioned ships) and administrative (focusing on the maximum utilization of troop and cargo space) load-plans.

The development of a load-plan can be undertaken in either of two modes. In the *User Stow* mode the user selects a cargo item from a textual cargo lists, ICODES automatically converts the selected item into the appropriate graphic cargo symbol, and once the user has placed the cargo symbol in a stow area the agents assess the impact of the cargo item in that position on both the validity of the load-plan and the condition of the ship. The agents take into account: the path of the cargo item from the dock to its final location on the ship (e.g., availability of ramps, cranes and elevators, and the dimensions of doors, hatches and openings); the segregation and other special requirements related to hazardous materials; and, the trim and stability conditions of the ship.

In the *Assisted Stow* mode the user is able to define specific parameters at the cargo and ship levels and then request ICODES to automatically stow the cargo on one or more ships. Parameters include the establishment of preferences for individual stow areas, the exclusion of stow areas, the specification of spacing distances between cargo items, the orientation of cargo items, and the selection of subs

ets of the cargo list. Once the parameters have been specified (either by default or user selection) ICODES will automatically prepare a load-plan that does not violate any of the rules and regulations known by the agents.

Expert Agent Capabilities

There are many definitions of software agents in the literature (Wooldridge and Jennings 1995; Bradshaw 1997). To the author, a software agent in its simplest form is a software module (i.e., service) that is capable of communicating with other software modules or human agents to facilitate some action. However, at this level of definition an agent is not necessarily intelligent. An intelligent agent would need to communicate using a common language (such as the ontology represented by the Semantic Network in ICODES) to support reasoning capabilities. In addition, an agent may have deep information and expert skills within a narrow domain and would then be referred to as a knowledge-based agent that has the ability to act on its own initiative. Such agents typically collaborate with other software and human agents to accomplish goals, and use local information to manage local resources.

The expert agents in ICODES are designed to assist the load-planner in the knowledge domains of hazardous material, trim and stability of the ship, cargo access paths, cargo attribute verification, and the actual placement of cargo in stow areas. The agents do not communicate directly with each other, but are totally decoupled. In fact, they do not know about each others existence. They collaborate indirectly as clients through a subscription service that allows them to post interests to data changes within the context provided by the ontology.

When the user is developing a load-plan while operating in User Stow mode, the agents will alert the user to any violations by turning the surround of the appropriate agent status window red. The user can then click on the status window to display a window with an explanation of the violation. In fact ICODES provides several different types of agent warnings:

- A ***yellow*** surround of an agent status window provides warning of a situation that could lead to a potential violation.
- An ***orange*** surround of an agent status window indicates that a warning has been acknowledged but still exists.
- A ***red*** surround of an agent status window indicates a violation (i.e., the existence of a serious problem).
- A ***purple*** surround of an agent status window indicates that an alert (i.e., serious violation) has been acknowledged but still exists.

If the user operates in Assisted Stow mode the agents will collaborate to place the cargo in such a manner that there are no violations. Cargo items that could not be placed in any stow area without causing a violation are simply not stowed. Brief summaries of the functional capabilities of each ICODES agent are provided below.

The ***Stow Agent*** supports both manual and automatic load-planning operations. Using default settings in the automatic mode (i.e., Assisted Stow), the Stow Agent attempts to place the heaviest cargo items as low as possible on the ship without causing a violation. This results in a low center of gravity for the ship, which is desirable in most cases. The Assisted-Stow mode provides a comprehensive set of settings. This allows the user to define exclusive and inclusive constraints and preferences in respect to both the cargo that is required to be stowed and the stow areas that have been designated as being available. The Stow Agent checks to see that the placement of a cargo item does not overlap another cargo item, a

fixture of the ship such as a stanchion or fire lane, or if the item is not entirely within a stow area. In Assisted-Stow mode, the user can also set the front/back and side to side spacing requirements of a cargo item (e.g., 18 inches front and back and 6 inches side to side) and the Stow Agent will abide by these settings so as not to stow within that imagery boundary around each cargo item.

Other parameters checked by the Stow Agent include the ports of embarkation and debarkation to ensure that they match the ports indicated in the voyage documents, and the height of each cargo item to ensure that the latter can reach their final stow positions. The Stow Agent automatically adds a safety cushion (specified by the user) to the actual height, which is set by the end-user, to make sure that height plus the cushion does not exceed the maximum allowable height for cargo in that stow area and the access path to the stow area..

While in the Assisted Stow mode ICODES will ensure that the automatically generated load-plan has no violations, in manual mode (i.e., User Stow) ICODES will allow the user to stow cargo items that are in violation. However, the Stow Agent will alert the user of the violations and provide an explanation on request.

The ***Trim and Stability Agent*** checks the placement of cargo items on the ship to see if they violate any desired (i.e., user specified) or mandated maximum draft settings, strengths (i.e., bending of the ship) or deck stress limitations. The Stow Agent in automatic mode will rearrange the placement of cargo during the Assisted Stow process if the placement of cargo causes the upper limits of the strengths properties of the ship to be exceeded. For example, if the predefined stow order requires the middle two stow areas of a deck to be stowed first and second, this would result in a sagging condition of the deck. Under these conditions the Stow Agent will automatically redefine the stow order used by the Assisted-Stow process, so that the placement sequence of the cargo will begin with the forward and aft areas of the deck (thereby preventing the occurrence of a sagging condition).

ICODES calculates the effects of the exact placement of every cargo item stowed on the ship in three different planes. These planes are: forward to aft often referred to as the Longitudinally Center of Gravity; side to side or Transverse Center of Gravity; and, up and down or Vertical Center of Gravity. The Trim and Stability Agent takes into account the combined effects of all of the cargo items, the ballast, and the original condition of the ship to provide the user with fairly accurate estimates of the center of gravity in each of the three planes, as well as an overall assessment of the stability of the ship.

The ***Access Agent*** checks all paths to ensure that a cargo item can be stowed in a particular stow area. This includes openings, doors and hatches, differentiating between cargo that is loaded with cranes through hatches (i.e., LOLO: Lift On Lift Off) and cargo that is driven or pulled into stow areas (i.e., RORO: Roll On Roll Off). Under Assisted Stow conditions, if there is a violation in the stow path of a particular cargo item the Stow Agent will not place this cargo item in that stow area but will attempt to place it in another stow area. In this situation the violation is transmitted indirectly from the Access Agent to the Stow Agent without notification of the user.

In manual mode (i.e., User Stow), on the other hand, if a cargo item is placed in a particular stow area for which all of the possible stow paths register an access violation then the Access

Agents will inform the user that the cargo item has a violation for every path to the stowed location. In addition, the Stow Agent will identify for the user the shortest stow path and the nature of the violation that is associated with that path.

ICODES allows the user to edit the ship characteristics, including the usability properties of the cranes and the dimensions of doors, openings and hatches. Since the Access Agent utilizes the current ship characteristics as the existing constraint conditions, these changes will be reflected in the actions of the Stow Agent in automatic mode and the alerts provided by the Access Agent in manual mode.

The ***Cargo Agent*** checks the characteristics of each cargo item against the expected characteristics for that cargo item recorded in the Marine Equipment Characteristics File (MECF) or Tech Data cargo libraries. Not all cargo characteristics can be verified in this manner. These cargo libraries currently contain more than 20,000 items, but are restricted in terms of the attributes that are provided for each cargo item. Typically, this verification process is complete and reliable only for dimensional (i.e., length, width and height) and weight attributes. If discrepancies are detected the Cargo Agent generates warnings.

The ***Hazard Agent*** verifies the proper placement of hazardous cargo items in reference to the various hazardous material codes and regulations discussed previously. It considers issues such as: Is the cargo item stowed in an acceptable deck location according to its stowage requirements? What are the segregation requirements for the cargo item, taking into account both the type of cargo item (e.g., break-bulk, container, vehicle) and the proximity of any other hazardous cargo items? In the case of containers, the Hazard Agent considers the hazard category of each item in the container in assessing the hazard condition of the container and its location relative to any other hazardous cargo item on the ship.

Operational Performance Assessment

During the more than eight years of ICODES releases and wide-spread military use no quantitative metrics have been collected to compare military ship load-planning from the period prior to the availability of ICODES (i.e., prior to 1997) and after ICODES became the system of record for Army, Marine Corps, and Navy surface load-planning. However, it is generally accepted within the military load-planning community that ICODES has been responsible for a dramatic improvement in decreasing the loading time of ships and berthing costs. In addition, ICODES further proved its utility in unanticipated areas, such as ship selection for the movement of supplies, cargo in-transit visibility, historical analysis of cargo movements, and ship design. The following selected areas of military load-planning operations may serve as indicators of the improvements in operating efficiency and cost savings that have been achieved through the deployment of the ICODES suite of adaptive tools over the past several years.

Load-planning efficiency: Previous to the fielding of ICODES, the creation of a pre-stow plan would often take one load-planner using the DOS-based CODES software at least two days. Once the cargo list had been cleansed, through the laborious manual process of comparing the data pertaining to each cargo item with the official equipment library, often a day long process, the load-planner would copy-and-paste the cargo symbols on the ship deck

drawings. Then other planners with expertise in hazardous cargo stowage, trim and stability, and cargo flow would check the plan, which often took another day. This time consuming cycle would begin again for each time the cargo list was updated, often up to 30 times during the development of a pre-stow plan.

With ICODES, and in particular through its agents (i.e., Cargo, Access, Trim and Stability, Hazard, and Stow Agents), a load-planner is able to create a similar pre-stow plan in about three hours. When updated cargo lists arrive the ICODES *merge* function allows the same plan to be updated within minutes without re-starting the planning process.

Marine Corps cargo specialists have indicated that prior to the availability of ICODES the planning of the equipment for a Marine Expeditionary Unit (MEU) involving 10 to 14 ships would take an Operation Planning Team five to seven days. With ICODES this task has been reduced to about 14 hours.

In-transit visibility: An area of support that did not exist prior to ICODES is the electronic submission of cargo manifests and cargo ship placement reports to the ship personnel and to the Port of Debarkation (POD) staff. This capability has provided visibility of cargo to the ship to assist with in-transit issues, to the POD for off-load-planning and/or load-planning of new loads, and to military administrative personnel for tracking and historically reporting on cargo movements.

At a POD, prior to ICODES, immediately after the arrival of a vessel a cargo survey and meeting would be held to discuss cargo placement and off-loading strategies. With the availability of ICODES documentation this half-day delay is no longer necessary resulting in a significant saving of berthing costs. In addition, the off-load-planning that can now be accomplished with ICODES prior to ship arrival results in substantial labor and off-load space assignment savings.

For ships with multiple ports of loading and discharge, ICODES load-plans are now passed electronically from port to port to determine the effects of the loads and off-loads on the ship and provide a common operating picture. Beyond the port, the Army Logistic Operations Center uses a database of ICODES-generated load-plans to estimate off-load times. In the past this has been a labor intensive operation, often resulting in missed deadlines.

Trim and stability analysis: Since the ICODES Trim and Stability Agent utilizes certified formulas for ship trim and stability calculations, the results are not only used by load-planners but also by the ship's crew to confirm ship loading conditions. Because of the trusted quality of the validated ICODES trim and stability analysis, ships are much less prone to unsafe stow configurations and further, sail up to a day earlier than in the pre-ICODES era. The earlier departure of ships leads to fuel savings since ships are able to proceed at reduced speed and still stay on schedule. In addition, ships stowed using the precision and operational knowledge offered by the ICODES system experience decreased port costs associated with berthing and service fees.

Prior to the availability of ICODES ships were often loaded with little concern for the distribution of weight along the ship's perpendicular axis, eventually causing several classes of ships to develop stress factures. The continuous monitoring of the condition of the ship during load-planning has led to better load distributions and the resultant reduction in costly ship repairs.

Reconciliation of planned cargo placement: Using the ICODES Automatic Information Technology (AIT) capabilities, the staging area cargo placement and the ship as-loaded plan is confirmed with hand-held Personal Digital Assistants (PDA), as opposed to manually drawn sketches and tally sheets. Using the ICODES AIT functionality, personnel costs have been reduced to about 20% of the cost of the manual process and the number of port cargo administrative personnel have been reduced by about 50%. With the increasing availability of AIT wireless communications at ports cargo locations are updated automatically to an ICODES computer in the port command center, allowing near real-time visibility of cargo to port administrative personnel and preventing the misplacement of hazardous materials.

Since its first release as a system of record in 1997, the granularity of the cargo data has increased greatly as ICODES moved from Level 4 to Level 6 detail. A typical Army cargo list in 1997 seldom included more than 2,000 individual cargo items. From 2004 onward ICODES has been required to process Marine Corps cargo lists with more than 30,000 individual cargo items. Despite this increase in the volume of data the performance of ICODES, in terms of response time, has continued to reduce as well. The typical performance results shown in Table 1 are based on periodic metrics collected by CDM's ICODES Test Group over the past eight years.

Table 1: Historical ICODES performance metrics

Tested Procedure	V 3.0 (1998)	V 5.0(2001)	V 5.4 (2005)
Create two-ship load-plan with 2,400 normal cargo items	20 min	8 min	1.5 min
Create two-ship load-plan with 1,200 hazardous cargo items	25 min	11 min	2.5 min
Unstow inventory of 2,400 items from two ships	10 min	5 min	1.0 min

ICODES System Architecture

As a KMES®-based system, designed according to service-oriented architecture principles and implemented within the ICDM development environment, ICODES incorporates a three-tier architecture that draws a clear distinction between representation, logic and presentation. It is a multi-agent system based on a knowledge management premise, in the sense that it incorporates an expressive information model consisting of context-oriented objects, their detailed characteristics, and the relationships that associate these objects to each other and the functional capabilities of the application (Diaz et al. 2006).

This internal information model provides the necessary context to enable reasoning agents to collaborate with each other and the human user to collectively evaluate events and generate warnings and alerts. Figure 5 provides a view of the ICODES conceptual architecture. Among other aspects, this diagram clearly illustrates the attention to the separation of concerns (i.e., representation, logic, and presentation) that is embedded in the design. Figure 6 provides an additional view of ICODES, presented in the Unified Modeling Language (UML) symbology (Fowler and Scott 1997), describing the significant components along with their inter-dependencies. This alternative view also emphasizes the attention given to inter-component visibility and the objective of a decoupled architecture.

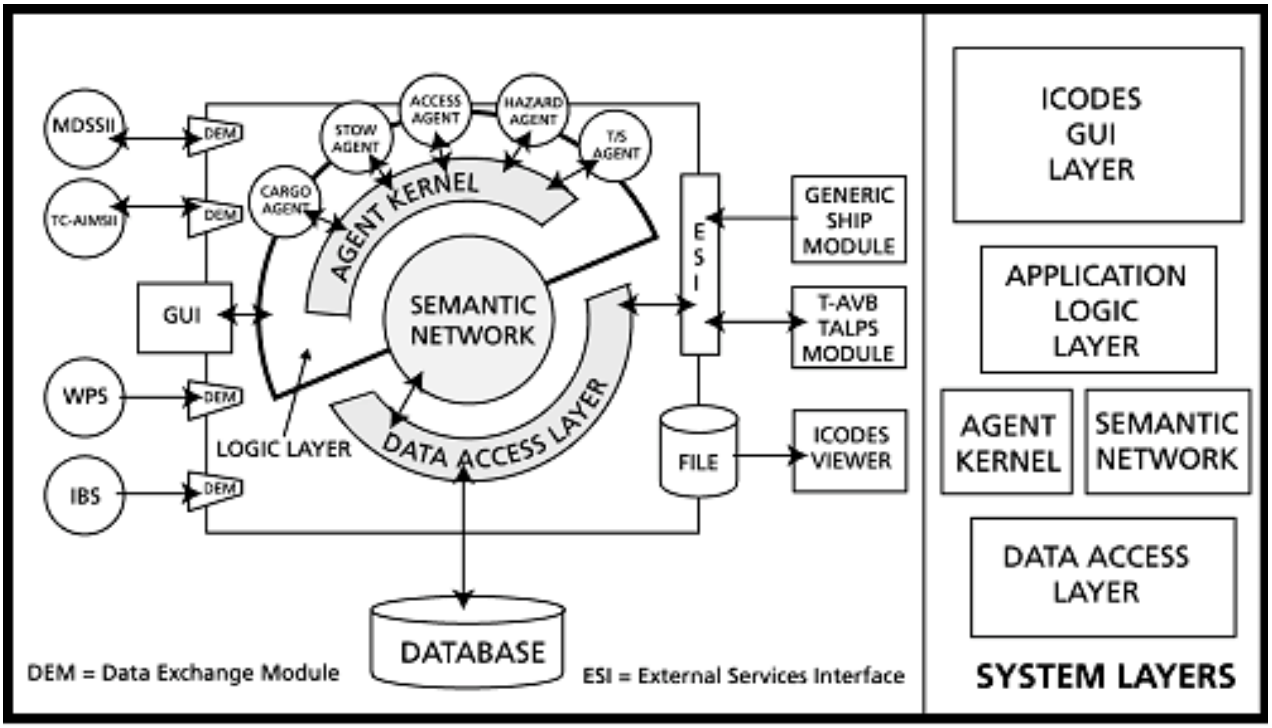


Figure 5: ICODES conceptual architecture

Domain Tier: This tier (also referred to as the semantic network) of the ICODES multi-tiered architecture is essentially comprised of a set of services that govern access to and general lifecycle management of the contextual objects representing the space-planning domain. The scope of this contextual description goes beyond representation of the physical elements that comprise the load-planning environment (i.e., vessels, cargo, etc.) but also include the more intangible, and sometimes much more subtle, concepts and notions that are vital to a complete description of the problem space and evolving solution(s). For example, included in the domain model and housed within this tier are expressive descriptions of the notions of restrictions, violations, recommendations, and accessibility. These are the intangible concepts that empower the ICODES agents with the ability to grasp a deeper understanding of the space-planning activity than would be possible with a data-centric representational paradigm.

The primary service within this tier is the *Semantic Network Service*, which has five related roles. First, it acts as a repository for objects and certain metadata in the semantic network. Second, it provides transaction management for any actions operating over this network of objects. Third, it arranges for the maintenance of the persistent storage of objects. Fourth, it stores dependencies of objects or tagged data on objects housed within the semantic network. Finally, it arranges for notification of interested components (i.e., agents, plug-ins, etc.) when objects in the semantic network change.

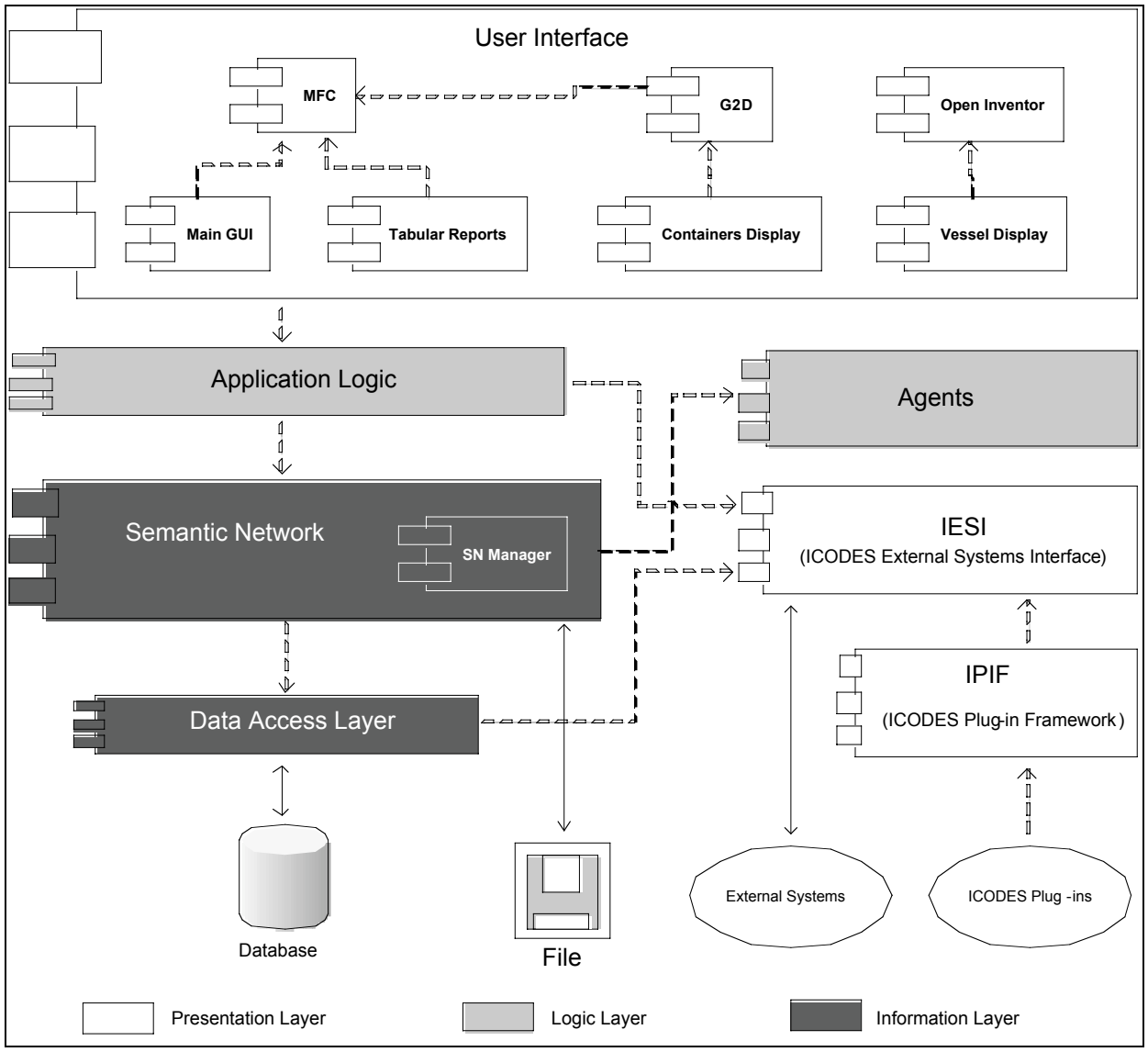


Figure 6: Key component dependencies

The Semantic Network Service engages the *Subscription Service*. This service provides a robust mechanism permitting semantic network clients to register interests in events occurring within the semantic network. The Subscription Service allows subscribers to be notified of events including object creation (i.e., objects being added to the semantic network), object destruction, and object modification. Employing a call-back design pattern, the subscriber provides a method or function to be called when the particular event of interest occurs. When the Subscription Service determines that the specific event has occurred, appropriate notification is sent to all interested parties. As an added means of efficiency, such notification may include state information regarding the event, thus obviating the need for the receiver to issue a set of follow-on queries to examine the details of the event.

The Domain Tier also houses a *Transaction Service* that maintains the overall integrity and consistency of the ICODES information environment. At the heart of this service is the notion of a transaction, as taken from the world of database systems. Any change to the semantic network occurs within the context of a transaction, although that transaction is sometimes provided implicitly. Multiple transactions may be active simultaneously, although no object may take part in more than one transaction at any given time.

Upon completion of a transaction, the Transaction Service employs the *Persistence Service*, which is responsible for reflecting the affects of this transaction within the persistent form of the semantic network. This involves the addition of new objects to storage, the updating of persistent storage to reflect changes made in modified objects, and the removal of objects that have been removed from the semantic network. Although not restricted to any particular vendor, ICODES currently uses the Microsoft SQL Desktop Engine database for its persistence needs. Further, ICODES is flexible enough to actually use more than one form of storage concurrently.

The primary interface offered to Persistence Service clients takes the form of the Independent Data Access Layer (IDAL). This interface is used for both saving an object's state to a data store as well as sharing objects across processes. IDAL provides a *handler* class that defines the interface for persistence of data throughout the ICODES system.

The Semantic Network Manager is capable of attaching tagged data and/or objects to any existing object in the semantic network in the form of *Attachments*. The attached data benefit from the normal lifecycle management offered by the semantic network and are therefore available to any internal or external component (e.g., agent, plug-in, etc.). The first of these two types of attachable elements, tagged data, is textual in nature and although it imparts no contextual information, has presumably a degree of meaning to the contributing component. The tagged data are given a name by the contributor requesting the housing of the attachment. Once named, any component may then retrieve the attached element by providing the name and the identifier of the semantic network object to which the element is attached.

However, far more powerful than attachment of meaningless text is the ability to attach a contextual object. Since an attached object becomes dependent on the object it is attached to (i.e., is owned by), the Semantic Network Manager will automatically remove the attached object if the object it is attached to is removed from the semantic network. This allows a component to create objects in the semantic network that are to be integrated with a load-plan without being concerned about cleaning up objects that are no longer relevant. This ability to add external, non-native types of content to the semantic network is yet another mechanism empowering ICODES with the ability to support the evolving needs of the user with minimal to no impact on the existing code base.

Logic Tier: This portion of the ICODES three-tier architecture not only houses the standard application-level logic typically found in such a tier but also contains the community of decision-support agents that provide the analytical depth empowering the ICODES space planning environment. The agents described in a previous section of this paper collaborate in an indirect fashion to assist in the development of efficient and correct load-plans. A brief

discussion of how this community of agents works together to formulate violation-free load-plans in both the assisted stow and the manual stow modes, follows.

The main function of the agent community operating within the *Assisted Stow* mode is to automatically find a valid stow location (i.e., free of any issues relating to accessibility, hazardous materials, trim and stability, etc.) for cargo items within some space. This is achieved through a round-robin style of agent collaboration. Once configured with user preferences and restrictions, the Assisted Stow capability takes advantage of the Cargo Agent by filtering out cargo items that are not deemed valid for stow. In most cases cargo items are filtered out because of missing information relating to dimension or weight. The next step is to locate an empty space on the vessel where the cargo item can be placed without overlapping other cargo items. However, the search for an empty space is constrained by factors such as the weight of the cargo item. For example, the heaviest items in the cargo list should be placed near the bottom of the ship for reasons of overall stability.

This candidate location is then presented for evaluation to both the Access Agent and the Hazardous Material Agent. In accordance with their particular domain expertise, if either agent finds an issue with the candidate placement the Assisted Stow capability nominates an alternative location and the evaluation repeats. This process is repeated until a violation-free location is found or the expanse of possible stow locations is exhausted. In either case, the Assisted Stow capability continues onto the next cargo item to be stowed.

In contrast to the collaborative assessment model applied in Assisted Stow, the *Manual Stow* (i.e., User Stow) mode of operation instructs the agents to function independently of one another. In other words, as the user places, or templates, a cargo item within a stow space each agent reacts concurrently identifying any outstanding issues incurred by such placement in accordance with their individual domain of expertise. Any such violations or warning are presented to the user in the form of agent reports comprised of a concise depiction of the issue along with any possible resolutions the agent may be able to offer. However, regardless of the severity of the issue, the user makes the final determination of how, and even if, the issue is to be resolved.

Presentation Tier: Like most industry-standard applications, ICODES offers a graphical user-interface (GUI) as the primary means of exchanging information with the user. Because ICODES is designed to run on a Windows operating system platform its user-interface adheres to the standard Windows logo compliance design pattern. ICODES offers its presentation capabilities in two forms, standalone and web-based.

The design of the ICODES standalone GUI is comprised of several internally developed as well as off-the-shelf components. Each of these components operates in unison to provide a robust means for interacting with the user. Following is a brief discussion of several of these components.

The *Vessel Display* portion of the ICODES stand-alone user-interface is based on the Open Inventor low level graphics library (Wernecke 1994). Open Inventor is a commercial object-oriented three-dimensional (3D) graphical toolkit built on top of OpenGL (Shreiner 2000). This toolkit uses a programming model based on a 3D Scene database that dramatically simplifies graphics programming. Open Inventor offers the software developer a rich set of objects including cubes, polygons, text, materials, cameras, lights, trackballs, handle boxes,

3D viewers, and editors. Together these elements allow for the development of robust, interactive graphical software applications.

The *Graphics 2D (G2D) Viewer* is a cross-platform implementation of a set of two-dimensional graphics tools. G2D displays its graphical elements in a layered manner. The viewer tool supports drag-and-drop operations as well as the drawing of graphic primitives, including multi-polygons and Bezier curves[4]. The G2D Viewer also supports the vector and matrix mathematics functionality that is required for the use of transforms.

The fundamental G2D viewer architecture essentially consists of three components: the G2D surface that forms the conceptual container in which all displayed information resides; multiple graphical layers displaying the graphical elements comprising the scene; and, the user-interface layer where items are drawn during drag or drawing operations.

In the *Web-Based User-Interface* (or Thin Client) ICODES load-plans are described as documents that represent all of the information available to the ICODES application during the preparation of a load-plan. When a stow-planner uploads a load-plan to the ICODES File Share, the ICODES Thin-Client uses standard XML parsers along with other tools to extract information from the file and place it in a database, thereby making that information available for later use in the user-interface.

The Thin-Client uses a standard SAX[5] parser to process the load-plan file. Instead of building a tree representation of an entire XML file in memory as a DOM[6] parser would do, a SAX parser identifies the individual parts of an XML document as it reads the file and immediately passes those parts to an object that implements the *org.xml.sax.ContentHandler* interface. When the parser identifies, for example, the start of an XML element and processes its attribute list, the parser will call the *ContentHandler.startElement* method, passing the element's name and universal identification (if applicable), and the list of attributes as *name/value* pairs. SAX parsers eliminate the need to parse the entire document before processing can begin, which is important when dealing with notably large XML documents such as ICODES load-plans.

Extended Functional Requirements

After an extensive evaluation of several existing military planning systems in 2007 it was determined by a joint TRANSCOM and JFCOM assessment team that both the functionality and scope of ICODES should be extended to support all military conveyance load-planning and staging requirements. With a release date scheduled for October 2010, it is expected that ICODES GS will incorporate a Collaborative Information Workspace (CIW) that will provide global user-access to a seamless environment of intelligent load-planning and cargo staging tools. These tools must function equally well in net-centric and stand-alone mode, through either a web-based Thin-Client or a Thick-Client user-interface.

---

[4] Developed by Pierre Bézier in the 1970s for CAD/CAM operations, a Bezier curve is a cubic equation that can be used in computer graphics for the construction of non-linear shapes.

[5] SAX is an acronym for Simple API for XML (see: http://www.saxproject.org/).

[6] DOM is an acronym for Document Object Model (see: http://www.w3.org/DOM/)

Initial explorations have indicated that the technology is now available to implement a Thin-Client with full interactive graphics functionality at an acceptable performance level. For example, in ICODES the user must have the ability to drag-and-drop the graphical symbol representing a cargo item located in the graphical display of a ship (in plan view) from one deck at the top of the screen to another deck at the bottom of the screen without any noticeable response delay. Until recently such an action would have required the redisplay of the entire graphics screen with the attendant severe performance penalty due to communication bandwidth limitations. Open source and commercial tools are now available that allow such a web-based user-interface to be designed so that only those portions of the graphic display that have been affected by the user's action need to be refreshed.

The existing ICODES Master Vessel Library that currently includes over 300 objectified ships (i.e., mostly commercial vessels leased by the military for the transportation of supplies) will need to be greatly extended to also include aircraft, train cars, trucks, and marshalling yards. The resulting Shared Object Library is being designed as a set of generic services that can objectify, store and retrieve the graphical representation of any kind of two-dimensional storage space together with its characteristics. It is expected that the size of the new Shared Object Library will be at least one order of magnitude larger than the current Master Vessel Library.

The CIW, designed as an Information Management Framework (IMF), will form the core of the ICODES GS service-oriented architecture implementation. In this role the IMF must provide the base services responsible for the discovery, integration, persistence, and exchange of data, as well as associated mediation and security functions. While these services are largely hidden from the users, they enable the users to perform tasks that require access to multiple data sources in a seamless fashion. Typical examples of the functions performed by these services include: *distribution* (subscription, prioritization and synchronization); *validation* (data cleansing and mapping); *discovery* (metadata management); *persistence* (archiving, indexing and aggregation); *security* (authentication, intrusion detection and virus prevention); *monitoring* (performance measurement, evaluation and optimization); and so on.

Accessible to the users through the IMF will be the tools that they require to perform their tasks. These include the various agents that assist in the load-planning process, the Thin-Client and Thick-Client user-interfaces, user-controlled data access and cleansing tools, and several utilities. Based on service-oriented architecture (SOA) principles these tools function as services and depending on individual requirements incorporate internal representations of context and agents capable of automatically reasoning about data changes within that context. The degree of intelligence and automation embedded in each of these services depends on the functional requirements. For example, the Assisted-Stow Agent will have a fairly high degree of intelligence because of the complexity of the load-planning problem and the dynamic nature of the operational environment.

It is in the nature of an information-centric system environment that the operators are shielded not only from the internal system services of the IMF, but also from the physical location, configuration requirements and data access requirements of the functional services that the operators utilize to perform their tasks. After signing on through a single login entry point, it should be transparent to the operator whether the required capability (i.e., tool) is a single service or several services, and whether it is an external legacy application linked to the IMF by an

intelligent interoperability bridge service or an internal service incorporating artificial intelligence methodologies.

Conclusion

The ICODES application currently provides a comprehensive tool-set of software agents to assist the cargo specialist in the development of ship load-plans for military deployments. It is one of the earliest military examples of information-centric software that incorporates an internal, relationship-rich information model to provide context for the reasoning functions of collaborative software agents. Over the next two years ICODES will be extended in functionality to serve as a toolset for the load-planning of all types of conveyances (i.e., aircraft, trains, and trucks) and assembly areas. At the same time the ability of its underlying SOA-based design will be severely tested as ICODES scales from a standalone application to a global environment of integrated intelligent services.

As an ICDM-based application, ICODES adheres to three notions that are fundamental to its decision-assistance capabilities.

1. ICODES processes information (i.e., data with relationships) as opposed to legacy systems that normally process data only (even though the data may be in the form of objects with characteristics). The key to the assistance capabilities of ICODES is that the system has some understanding of the information that it is processing. In the internal Semantic Network cargo items are described in terms of characteristics that relate each item to hazard, trim and stability, accessibility, and ship configuration, constraints. This internal information model provides context for the automatic reasoning capabilities of software agents.
2. ICODES is a collection of powerful collaborative tools, not a library of predefined solutions. This overcomes the deficiencies of legacy systems in which built-in solutions to predetermined problems often differ significantly from the complex operational situations encountered in the real world. In this respect ICODES is a collaborative decision-support system in which the operator interacts with computer-based agents (i.e., decision making tools) to solve problems that cannot be precisely or easily predetermined.
3. ICODES incorporates agents that are able to reason about the characteristics and the relationships of cargo items, the internal configurations of conveyances and the constraints that must be considered during the development of load-plans. Although these agents are decoupled (i.e., do not know about each others existence) they are able to indirectly collaborate through a data blackboard and subscription services, as they assist the user throughout the load-planning process.

The advantages of an information-centric software system have been evidenced in three areas by the performance of ICODES in the field over the past three years. First, if all necessary data are available ICODES is capable of automatically generating the load-plans of four medium-size ships in around two hours. This is a significant improvement in load-planning speed over the legacy application that it replaced. The predecessor application typically required two person-days for the development of a single load-plan. Second, the assistance capabilities of the

ICODES agents elevate the performance of a novice load-planner to at least an acceptable level. This is an important consideration in view of the attrition rate of military cargo specialists during the past decade. The performance of an expert load-planner, on the other hand, is raised to an exceptionally high productivity level. Third, the ability of ICODES to continuously evaluate the evolving load-plan in respect to accessibility, hazardous material, and trim and stability conditions, has greatly increased the quality and accuracy of the resulting load-plan.

References

Bradshaw J. M. (ed) (1997); 'Software Agents'; AAAI Press / The MIT Press, Massachusetts, (pp. 3-11).

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.

Fowler M and K Scott (1997); 'UML Distilled: Applying the Standard Object Modeling Language'; Addison-Wesley, Reading, Massachusetts.

Mowbray T. and R. Zahavi (1995); 'The Essential CORBA: Systems Integration Using Distributed Objects'; Wiley, New York, New York.

MTMC/ICODES (2002); 'ICODES Version 5.2: USMC Basic Training Manual' and 'ICODES Version 5.2: Advanced Training Manual'; Military Traffic Management Command (MTMC), US Army.

Myers M. and J. Pohl (1994); 'ICDM: Integrated Cooperative Decision Making – in Practice'; 6th IEEE International Conference on Tools with Artificial Intelligence, New Orleans, Louisiana, November 6-9.

Pohl J., A. Chapman, K. Pohl, J. Primrose and A. Wozniak (1992); 'Decision-Support Systems: Notions, Prototypes, and In-Use Applications'; Technical Report CADRU-11-97, Collaborative Agent Design Research Center, Design and Construction Institute, College of Architecture and Environmental Design, Cal Poly, San Luis Obispo, California, January.

Pohl K. (2001); 'Perspective Filters as a Means for Interoperability Among Information-Centric Decision-Support Systems'; Office of Naval Research (ONR) Workshop hosted by the Collaborative Agent Design Research Center (CADRC), Cal Poly (San Luis Obispo, CA) in Quantico, Virginia, June 5-7.

Pohl J., K. Pohl, R. Leighton, M. Zang, S. Gollery and M. Porczak (2004); 'The ICDM Development Toolkit: Purpose and Overview'; Technical Report CDM-16-04, CDM Technologies, Inc., San Luis Obispo, California, May.

Pohl J. (2007); 'Knowledge Management Enterprise Services (KMES): Concepts and Implementation Principles'; Pre-Conference Proceedings, Focus Symposium on Representation of Context in Software, InterSymp-2007, Baden-Baden, Germany, 31 July.

Pohl K. (2002); 'The Underlying Design Principles of the ICDM Development Toolkit'; Preconference Proceedings, InterSymp-2002, Focus Symposium on Collaborative Decision-Support Systems, Baden-Baden, Germany, July 29-August 30.

Shreiner D. (ed.) (2000); 'Open GL Reference Manual'; Addison-Wesley, Menlo Park, California.

Wernecke J. (1994); 'The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor'; Addison-Wesley, Menlo Park, California.

Wooldridge M. and N. Jennings (1995); 'Intelligent Agents: Theory and Practice'; The Knowledge Engineering Review, 10(2), (pp. 115-152).