InterSymp – 2006
18th International Conference on Systems Research, Informatics and Cybernetics
(August 7-12, 2006, Baden-Baden, Germany)

Pre-Conference Proceedings of the Focus Symposium
on
Advances in Intelligent Software Systems

Tuesday, August 8th, 2006

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: 1-894613-71-6

Preface

A very powerful new type of information systems technology is rapidly emerging, driven by government and commercial needs for *expert decision-support* and *knowledge management*. One very apparent result of this technology is increasingly intelligent software systems. Computer programs with collaborative agents that are capable of automatically reasoning about data and the dynamic changes in data that occur in real world decision-making situations are already in use by the military and are now transitioning to the commercial world.

It can be argued that our human view of computer software has been shortsighted in respect to two popular notions: first, that data and information are essentially synonymous terms; and, second, that computer intelligence is largely a misnomer because computers are machines. Neither of these notions is accurate. While we human beings are able to convert data (i.e., numbers and words without relationships) automatically into information due to the experience (i.e., *context*) that is held in our cognitive system, computers do not have the equivalent of a human cognitive system and therefore store data simply as the numbers and words that are entered into the computer. For a computer to interpret data it requires an information structure that provides at least some level of *context*. This can be accomplished utilizing an ontology of objects with characteristics and a rich set of relationships to create a virtual version of real world situations and provide the context within which agent logic can automatically operate.

In the broadest sense an agent is a computer-based program or module of a program that has communication capabilities to external entities and can perform some useful tasks in at least a semi-autonomous fashion. Agent software can range from simple, stand-alone, predetermined applications to the most intelligent, integrated, multi-agent decision-support system that advanced technology can produce today.

There are many types of software agents, ranging from those that emulate symbolic reasoning by processing rules, to highly mathematical pattern matching neural networks, genetic algorithms, and particle swarm optimization techniques. The focus of many of the products of this technology is on ontology-based decision-support systems that utilize agents with symbolic reasoning capabilities. In these systems the reasoning process relies heavily on the rich representation of objects and their relationships provided by the ontology.

The capabilities of ontology-based multi-agent systems are several orders above those of past data-processing systems that were confined to predetermined algorithmic solution sequences. Those systems worked well when the problem in the real world was exactly as predicted during the design and development stages of the software. However, more often than not the problems encountered in the real world did not conform to those predictions. Agent-based programs are able to *adapt* their solution capabilities to a real world problem situation because of the ability of the software agents to reason within the context of the problem situation.

The same capabilities can be applied to the collection and exploitation of the information and knowledge that is generated within an organization (e.g., e-mail messages, telephone calls, minutes of business meetings, and other documents), commonly referred to as *knowledge management*. The automated capture of this wealth of information within an ontology-based system allows software agents to conduct intelligent searches, identify patterns, and specifically assist in the development of plans, the evaluation of alternative courses of action, and monitor changing conditions in collaboration with each other and the human user.

Knowledge is an intellectual facility that allows a person to perform tasks that require an understanding of what has to be accomplished, the formulation of a plan of action, and the skills that are required to undertake the task. It normally involves the acquisition over time of factual information, associations that bind the factual information into more general patterns, principles, rules, and problem solving skills. A person acquires knowledge through experience, formal education, and a life-long process of self-education. Accordingly, knowledge is a commodity that is held within the brain of each individual person. Both the communication of this personal knowledge from one individual to another and the collection of the knowledge as a corporate asset has become a serious concern of organizations, and is commonly referred to as knowledge management.

The reasons for this concern are due to two interrelated factors, namely advances in information technology and increasing performance expectations. Technical advances in the processing and storage capacity of digital computers, together with the linkage of these computers into networks of distributed nodes, have greatly increased the capability of organizations to deliver goods and services. With these increased capabilities have come heightened expectations for quality, accuracy, responsiveness, and capacity.

To meet increasingly more exacting performance requirements, organizations have been forced to frequently reexamine their formal structure, processes, and the ability of its members to collaboratively and expeditiously perform their tasks. Since these tasks depend largely on the knowledge held by individuals, the need to share this implicit knowledge explicitly among groups has become more and more important. Under these circumstances it is natural for organizations to take steps to protect their knowledge assets, by collecting and storing the knowledge of individuals in an explicit form that will make it readily accessible to others. This requirement cannot be satisfied by the storage of data only. In addition to data, the explicit representation of knowledge requires the storage of the mappings that convert data into information to place data into context, and the rules (i.e., business rules) that allow information to be effectively utilized in planning, problem solving, and decision-making processes.

A key characteristic of knowledge is the ability to abstract information and relationships so that they can be applied to problems that are outside of the realm of a person's existing experience. In this respect *abstraction* is a very powerful representational mechanism that allows for the generalization (not to be confused with vagueness as these generalizations are often quite concrete) of basic and sometimes common characteristics.

As information technology permeates all aspects of life and the economy turns decidedly information-centric, wealth is increasingly defined in terms of information-related services and the availability of knowledge. In other words, knowledge has become a commodity that has value far in excess of the manufactured products that represented the yardstick of wealth during the industrial age.

How this new form of human wealth should be effectively utilized and nurtured in commercial and government organizations has in recent years become a major preoccupation of management. The question being asked is: How can we capture and utilize the potentially available knowledge for the benefit of the organization? The phrase "…potentially available" is appropriate, because much of the knowledge is hidden in an overwhelming volume of computer-based data. What is not commonly understood is that the overwhelming nature of the stored data is due to current processing methods rather than necessity. These processing methods have to rely largely on manual tasks because only the human user can provide the necessary context for interpreting the computer-stored data into information and knowledge. If it were possible to capture information (i.e., data with relationships), rather than data, at the point of entry into the computer then there would be sufficient context for computer software to process the information automatically into knowledge. This is not just a desirable capability, but an absolute requirement for the capture and effective utilization of knowledge within any organization. Corporate knowledge also acts as an extension to individual knowledge. The representation and storage of *best practices*, combined with proper navigational tools, contributes to the efficiency of individuals performing tasks for which they may not have sufficient experience.

Jens Pohl, June 2006

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

InterSymp-2006
International Conference on Systems Research, Informatics and Cybernetics

Focus Symposium
on
Advances in Intelligent Software Systems
Tuesday, August 8, 2006

TABLE OF CONTENTS

Elements of Human Decision-Making

Jens Pohl, Ph.D.

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

Abstract

The purpose of this paper is to present some understandings of the human problem-solving activity that we have gained in the Collaborative Agent Design Research Center (CADRC) over the past two decades. Since we feel strongly that the human decision-maker should be an integral component of any computer-based decision-support system, it follows that we would have endeavored to incorporate many of the elements that appear to be important to the user in the design of these systems. The complexity of the human cognitive system is evidenced by the large body of literature that describes problem-solving behavior and the relatively fewer writings that attempt to provide comprehensive explanations of this behavior. Our contributions in this field are confined to the identification of important elements of the problem-solving activity and exploration of how these elements might influence the design of a decision-support system.

Keywords

agents, analysis, communication, computers, context, data, data-centric, decisions, decision-making, decision-support, design, evaluation, information, information-centric, intuition, problem-solving, reasoning, representation, synthesis, visualization

Some Human Problem Solving Characteristics

Human beings are inquisitive creatures by nature who seek explanations for all that they observe and experience in their living environment. While this quest for understanding is central to our success in adapting to a changing and at times unforgiving environment, it is also a major cause for our willingness to accept partial understandings and superficial explanations when the degree of complexity of the problem situation confounds our mental capabilities. In other words, a superficial or partial explanation is considered better than no explanation at all. As flawed as this approach may be, it has helped us to solve difficult problems in stages. By first oversimplifying a problem we are able to develop an initial solution that is later refined as a better understanding of the nature of the problem evolves. Unfortunately, now we have to contend with another characteristic of human beings, our inherent resistance to change and aversion to risk taking. Once we have found an apparently reasonable and workable explanation or solution we tend to lose interest in pursuing its intrinsic shortcomings and increasingly believe in its validity. Whether driven by complacency or lack of confidence, this state of affairs leads to many surprises. We are continuously discovering that what we believed to be true is only partly true or not true at all, because the problem is more complicated than we had previously assumed it to be.

The complexity of problems faced by human society in areas such as management, economics, marketing, engineering design, and environmental preservation, is increasing for several reasons. First, computer-driven information systems have expanded these areas from a local to an increasingly global focus. Even small manufacturers are no longer confined to a regionally localized market for selling their products. The marketing decisions that they have to make must take into account a wide range of factors and a great deal of knowledge that is far removed from the local environment. Second, as the net-centricity of the problem system increases so do the relationships among the various factors. These relationships are difficult to deal with, because they require the decision-maker to consider many factors concurrently. Although the biological operation of the human brain is massively parallel, our conscious reasoning processes are sequential. Simply stated, we have difficulty reasoning about more than two or three variables at any one time. Third, as the scope of problems increases decision-makers suffer simultaneously from two diametrically opposed but related conditions. They tend to be overwhelmed by the shear volume of data that they have to consider, and yet they lack information in many specific areas. To make matters worse, the information tends to change dynamically in largely unpredictable ways

It is therefore not surprising that governments, corporations, businesses, down to the individual person, are increasingly looking to computer-based decision-support systems for assistance. This has placed a great deal of pressure on software developers to rapidly produce applications that will overcome the apparent failings of the human decision-maker. While the expectations have been very high, the delivery has been much more modest. The expectations were simply unrealistic. It was assumed that advances in technology would be simultaneously accompanied by an understanding of how these advances should be applied optimally to assist human endeavors. History suggests that such an a priori assumption is not justified. There are countless examples that would suggest the contrary. For example, the invention of new materials (e.g., plastics) has inevitably been followed by a period of misuse. Whether based on a misunderstanding or lack of knowledge of its intrinsic properties, the new material was typically initially applied in a manner that emulated the material(s) it replaced. In other words, it took some time for the users of the new material to break away from the existing paradigm. A similar situation currently exists in the area of computer-based decision-support systems.

The Rationalistic Tradition

To understand current trends in the evolution of progressively more sophisticated decision-support systems it is important to briefly review the foundations of problem solving methodology from an historical perspective. Epistemology is the study or theory of the origin, nature, methods and limits of knowledge. The dominant epistemology of Western Society has been technical rationalism (i.e., the systematic application of scientific principles to the definition and solution of problems).

The rationalistic approach to a problem situation is to proceed in well defined and largely sequential steps as shown in Figure 1: define the problem; establish general rules that describe the relationships that exist in the problem system; apply the rules to develop a solution; test the validity of the solution; and, repeat all steps until an acceptable solution has been found. This simple view of problem solving suggested a model of sequential decision-making that has retained a dominant position to the present day. With the advent of computers it was readily embraced by 1st Wave software (Figure 2) because of the ease with which it could be translated into decision-support systems utilizing the procedural computer languages that were available at the time.

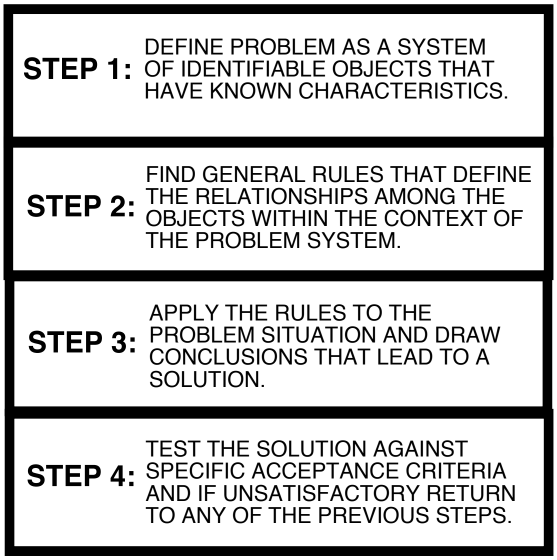


Figure 1: Solution of simple problems

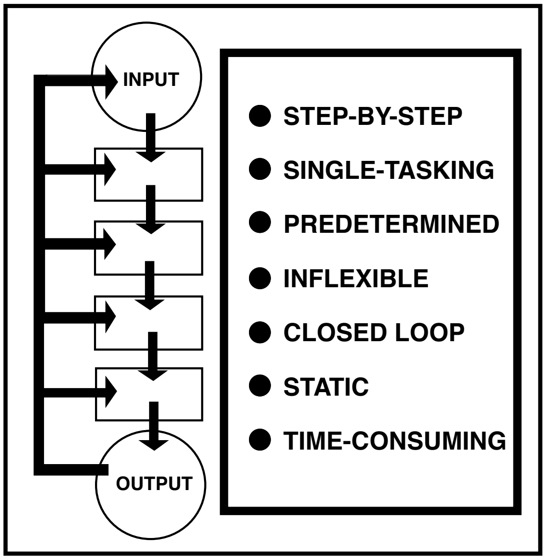


Figure 2: Sequential decision-support

The close correlation between the rationalistic approach and what is commonly referred to as the scientific method, is readily apparent in the series of basic steps that are employed in scientific investigations: observe the phenomenon that requires explanation; formulate a possible explanation; develop a method capable of predicting or generating the observed phenomenon; interpret the results produced by the method; and, repeat all steps until an acceptable explanation of the observed phenomenon has been found. Scientific research typically attempts to establish situations in which observable actions (or reactions) are governed by a small number of variables that can be systematically manipulated. Every effort is made to keep the contrived situation simple, clear and deterministic, so that the results of the simulation can be verified.

However, natural phenomena and real world problems are often very complex involving many related variables. Neither the relationships among the variables nor the variables themselves are normally sufficiently well understood to provide the basis for clear and comprehensive definitions. In other words, problem situations are often too complex to be amenable to an entirely logical and predefined solution approach. Under these circumstances the analytical strategy has been to decompose the whole into component parts, as follows:

> Decompose the problem system into sub-problems.
>
> Study each sub-problem in relative isolation, using the rationalistic approach (Figure 1). If the relationships within the sub-problem domain cannot be clearly defined then decompose the sub-problem further.
>
> Combine the solutions of the sub-problems into a solution of the whole.

Underlying this problem-solving strategy is the implicit assumption that an understanding of parts leads to an understanding of the whole. Under certain conditions this assumption may be valid. However, in many complex problem situations the parts are tightly coupled so that the behavior of the whole depends on the interactions among the parts rather than the internal characteristics of the parts themselves (Bohm 1983, Senge 1993). An analogy can be drawn with the behavior of ants. Each ant has only primitive skills, such as the ability to interpret the scent of another ant and the instinctive drive to search for food, but little if any notion of the purpose or objectives of the ant colony as a whole. In other words, an understanding of the behavior of an individual ant does not necessarily lead to an understanding of the community behavior of the ant colony of which the ant is a part.

Decomposition is a natural extension of the scientific approach to problem solving and has become an integral and essential component of rationalistic methodologies. Nevertheless, it has serious limitations. First, the behavior of the whole usually depends more on the interactions of its parts and less on the intrinsic behavior of each part. Second, the whole is typically a part of a greater whole and to understand the former we have to also understand how it interacts with the greater whole. Third, the definition of what constitutes a part is subject to viewpoint and purpose, and not intrinsic in the nature of the whole. For example, from one perspective a coffee maker may be considered to comprise a bowl, a hotplate, and a percolator. From another perspective it consists of electrical and constructional components, and so on.

Rationalism and decomposition are certainly useful decision-making tools in complex problem situations. However, care must be taken in their application. At the outset it must be recognized that the reflective sense (Schon 1983) and intuition of the decision-maker are at least equally important tools. Second, decomposition must be practiced with restraint so that the complexity of the interactions among parts is not overshadowed by the much simpler behavior of each of the individual parts. Third, it must be understood that the definition of the parts is largely dependent on the objectives and knowledge about the problem that is currently available to the decision-maker. Even relatively minor discoveries about the greater whole, of which the given problem situation forms a part, are likely to have significant impact on the purpose and the objectives of the problem situation itself.

Decision Making in Complex Problem Situations

As shown in Figure 3, there are several characteristics that distinguish a complex problem from a simple problem. First, the problem is likely to involve many related issues or variables. As discussed earlier the relationships among the variables often have more bearing on the problem situation than the variables themselves. Under such tightly coupled conditions it is often not particularly helpful, and may even be misleading, to consider issues in isolation. Second, to confound matters some of the variables may be only partially defined and some may yet to be discovered. In any case, not all of the information that is required for formulating and evaluating alternatives is available. Decisions have to be made on the basis of incomplete information.

Third, complex problem situations are pervaded with dynamic information changes. These changes are related not only to the nature of an individual issue, but also to the context of the problem situation. For example, a change in wind direction during a major brushfire may have a profound impact on the entire nature of the relief operation. Apart from precipitating an immediate re-evaluation of the firefighting strategy, it may require the relocation of firefighters and their equipment, the replanning of evacuation routes, and possibly even the relocation of distribution centers. Certainly, a change in the single factor of wind direction could, due to its many relationships, call into question the very feasibility of the existing course of action (i.e., the firefighting plan). Even under less critical conditions it is not uncommon for the solution objectives to change several times during the decision-making process. This fourth characteristic of complex problem situations is of particular interest. It exemplifies the tight coupling that can exist among certain problem issues, and the degree to which decision-makers must be willing to accommodate fundamental changes in the information that drives the problem situation.

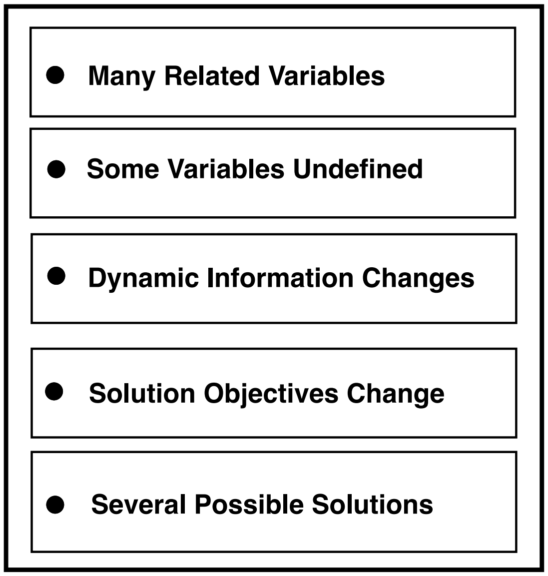


Figure 3: Character of complex problems

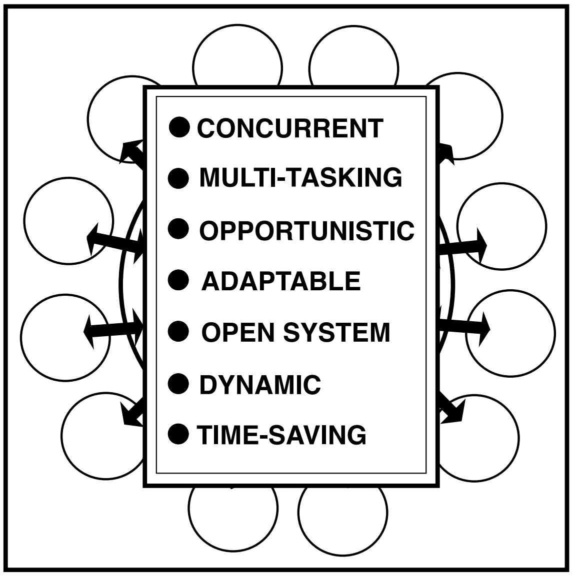


Figure 4: Parallel decision-support

Fifth, complex problems typically have more than one solution (Archea 1987). It is usually fruitless to look for an optimum solution, because there are no static benchmarks available for evaluating optimality. A solution is found to be acceptable if it satisfies certain performance requirements and if it has been determined that the search for alternatives is no longer warranted. Such a determination is often the result of resource constraints (e.g., availability of time, penalty of non-action, or financial resources) rather than a high level of satisfaction with the quality of the proposed solution.

While human decision-making in complex problem situations has so far defied rigorous scientific explanation, we do have knowledge of at least some of the characteristics of the decision-making activity.

> Decision-makers typically define the problem situation in terms of issues that are known to impact the desired outcome. The relative importance of these issues and their relationships to each other change dynamically during the decision-making process. So also do the boundaries of the problem space and the goals and objectives of the desired outcome. In other words, under these circumstances decision-making is an altogether dynamic process in which both the rules that govern the process and the required properties of the end-result are subject to continuous review, refinement and amendment.
>
> The complexity of the decision-making activity does not appear to be due to a high level of difficulty in any one area but the multiple relationships that exist among the many issues that impact the desired outcome. Since a decision in one area will tend to influence several other areas there is a need to consider many factors at the same time. This places a severe burden on the human cognitive system. Although the neurological mechanisms that support conscious thought

> processes are massively parallel, the operation of these reasoning capabilities is largely sequential. Accordingly, decision-makers tend to apply simplification strategies for reducing the complexity of the problem-solving activity. In this regard it becomes readily apparent why 2nd Wave software provides a much more useful architecture for decision-support systems (Figure 4).
>
> Observation of decision-makers in action has drawn attention to the important role played by experience gained in past similar situations, knowledge acquired in the general course of decision-making practice, and expertise contributed by persons who have detailed specialist knowledge in particular problem areas. The dominant emphasis on experience is confirmation of another fundamental aspect of the decision-making activity. Problem-solvers seldom start from first principles. In most cases, the decision-maker builds on existing solutions from previous situations that are in some way related to the problem under consideration. From this viewpoint, the decision-making activity involves the modification, refinement, enhancement and combination of existing solutions into a new hybrid solution that satisfies the requirements of the given problem system. In other words, problem-solving can be described as a process in which relevant elements of past prototype solution models are progressively and collectively molded into a new solution model. Very seldom are new prototype solutions created that do not lean heavily on past prototypes.
>
> Finally, there is a distinctly irrational aspect to decision-making in complex problem situations. Donald Schon refers to a "...reflective conversation with the situation...". (Schon 1983). He argues that decision-makers frequently make value judgments for which they cannot rationally account. Yet, these intuitive judgments often result in conclusions that lead to superior solutions. It would appear that such intuitive capabilities are based on a conceptual understanding of the situation, which allows the problem solver to make knowledge associations at a highly abstract level.

Based on these characteristics the solution of complex problems can be categorized as an information intensive activity that depends for its success largely on the availability of information resources and, in particular, the experience and reasoning skills of the decision-makers. It follows that the quality of the solutions will vary significantly as a function of the problem-solving skills, knowledge, and information resources that can be brought to bear on the solution process. This clearly presents an opportunity for the useful employment of computer-based decision-support systems in which the capabilities of the human decision-maker are complemented with knowledge bases, expert agents, and self-activating conflict identification and monitoring capabilities.

Principal Elements of Decision-Making

Over the past two decades that the CADRC Center has been developing distributed, collaborative decision-support systems some insights have been gained into the nature of the decision-making activity. In particular, we have found it useful to characterize decision-making in terms of six functional elements (Figure 5): ***information***; ***representation***; ***visualization***; ***communication***; ***reasoning***; and, ***intuition***.

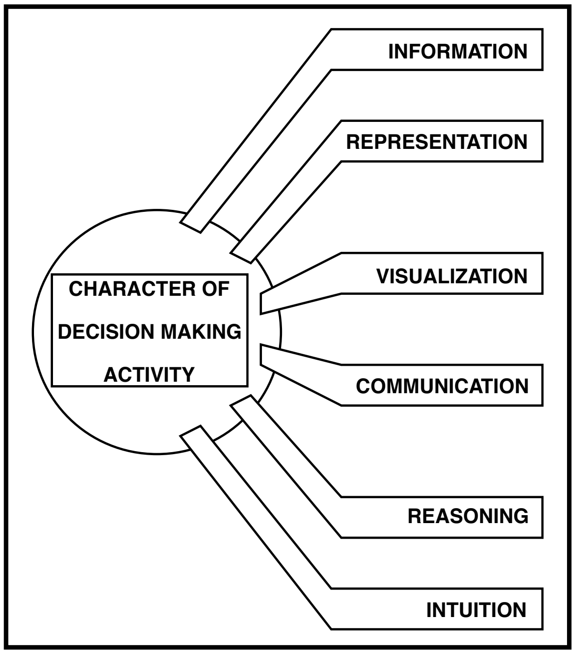


Figure 5: Decision-making elements

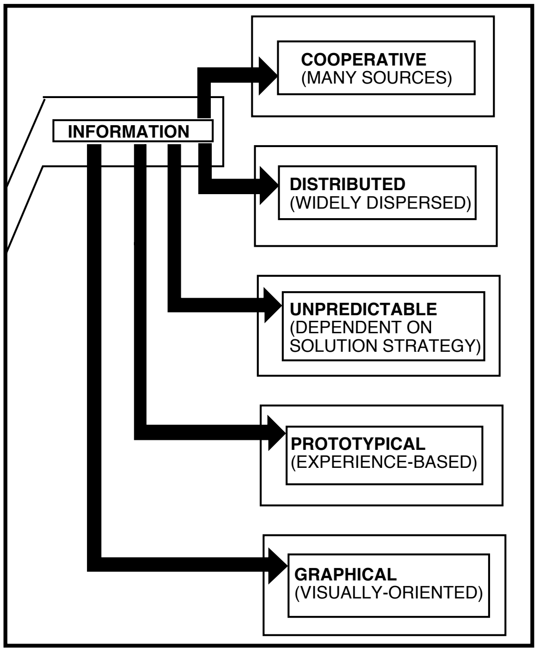


Figure 6: The ***information*** element

The *Information* Element

Decision-making in complex problem situations is a collaborative activity involving many sources of information that are often widely dispersed. Seldom is all of the information required for the solution, or even only a component of the problem, physically located in the immediate vicinity of the decision-maker. In fact, much of the information is likely to reside in remote repositories that can be accessed only through electronic means, the telephone, e-mail, or the temporary relocation of a member of the problem-solving team (Figure 6). If the desired information requires expert advice the services of a consultant may be required in addition to, or instead of, access to an information resource.

The term *information* is used here in the broadest sense to include not only factual data and the progressively more comprehensive and detailed description of the problem system, but also the many knowledge bases that are part of the local and global environment within which the problem situation is constituted. In this regard, we are concerned with the knowledge of the individual members of the problem-solving team, the knowledge of peripheral players (e.g., colleagues, associates and consultants), the collective knowledge of the profession (such as the various engineering professions, the military establishment, or the management profession) and industry, and beyond that those aspects of what might be referred to as global knowledge that impact the problem context.

Typically, the problem specifications (i.e., constraints, criteria, and objectives) evolve with the problem solution as the decision-makers interact with the problem situation. Accordingly, the information requirements of the problem solver are not predictable since the information needed to solve the problem depends largely on the solution strategy adopted (Fischer and Nakakoji 1991). In this respect problem solving is a learning process in which the decision-maker progressively develops a clearer understanding of the problem that is required to be solved.

Much of the information that decision-makers use in the development of a problem solution is gleaned from experience with past projects. In fact, it can be argued that solutions commonly evolve out of the adaptation, refinement and combination of prototypes (Gero et al. 1988). This argument suggests that the more expert human decision-makers are the more they tend to rely on prototypical information in the solution of complex problems. It would appear that the accumulation, categorization and ability to apply prototype knowledge are the fundamental requirements for a human decision-maker to reach the level of *expert* in a particular domain. Based largely on the work of Gero et al. (1988) and Rosenman and Gero (1993) the following techniques used by engineering designers to develop solutions through the manipulation of prototypes can be identified as being universally applicable to other problem domains:

> ***Refinement:*** The prototype can be applied after changes have been made in the values of parameter variables only (i.e., the instance of the prototype is reinterpreted within the acceptable range of the parameter variables).
>
> ***Adaptation:*** Application of the prototype requires changes in the parameters that constitute the description of the prototype instance, based on factors that are internal to the prototype (i.e., a new prototype instance is produced).
>
> ***Combination:*** Application of the prototype requires the importation of parameter variables of other prototypes, producing a new instance of a reinterpreted version of the original prototype.
>
> ***Mutation:*** Application of the prototype requires structural changes to the parameter variables, either through internal manipulations or the importation of parameter variables from external sources (i.e., either a reinterpreted version of the original prototype or a new prototype is produced).
>
> ***Analogy:*** Creation of a new prototype based on a prototype that exists in another context, but displays behavioral properties that appear to be analogous to the application context.

For application purposes in knowledge-based decision-support systems prototypes may be categorized into five main groups based on knowledge content (Schon 1988, Pohl and Myers 1994):

1. **Vertical** prototype knowledge bases that contain typical object descriptions and relationships for a complete problem situation or component thereof. Such a knowledge base may include all of the types that exist in a particular problem setting, for example: an operational template for a particular kind of humanitarian relief mission; a certain type of propulsion unit; or, a building type such as a library, sports stadium, or supermarket.
2. **Horizontal** prototype knowledge bases that contain typical solutions for sub-problems such as commercial procurement practices, construction of a temporary shelter, or techniques for repairing equipment. This kind of knowledge often applies to more than one discipline. For example, the techniques for repairing a truck apply equally to the military as they do to auto-repair shops, engineering concerns, and transportation related organizations.

3. **Domain** prototype knowledge bases that contain guidelines for developing solutions within contributing narrow domains. For example, the range of structural solutions appropriate for the construction of a suspension bridge during a military mission is greatly influenced by the availability of material, the prevailing wind conditions, and the time available for erection. Posed with this design problem military engineers will immediately draw upon a set of rules that guide the design activity.
4. **Exemplar** prototype knowledge bases that describe a specific instance of an object type or solution to a sub-problem. Exemplary prototypes can be instances of vertical or horizontal prototypes, such as a particular building type or a method of welding a certain kind of steel joint that is applied across several disciplines and industries (e.g., building industry and automobile industry). Decision-makers often refer to exemplary prototypes in exploring solution alternatives to sub-problems.
5. **Experiential** knowledge bases that represent the factual prescriptions, strategies and solution conventions employed by the decision-maker in solving similar kinds of problem situations. Such knowledge bases are typically rich in methods and procedures. For example, a particularly memorable experience such as the deciding event in a past business negotiation or the experience of seeing for the first time the magnificent sail-like concrete shell walls of the Sydney Opera House, may provide the basis for a solution method that is applied later to create a similar experience in a new problem situation that may be quite different in most other respects. In other words, experiential prototypes are not bound to a specific type of problem situation. Instead, they represent techniques and methods that can be reproduced in various contexts with similar results. Experiential knowledge is often applied in very subtle ways to guide the solution of sub-problems (e.g., a subterfuge in business merger or take-over negotiations that is designed to mislead a competing party).

The amount of prototypical information is potentially overwhelming. However, the more astute and experienced decision-maker will insist on taking time to assimilate as much information as possible into the problem setting before committing to a solution theme. There is a fear that early committal to a particular solution concept might overlook characteristics of the problem situation that could gain in importance in later stages, when the solution has become too rigid to adapt to desirable changes. This reluctance to come to closure places a major information management burden on the problem solver. Much of the information cannot be specifically structured and prepared for ready access, because the needs of the problem solver cannot be fully anticipated. Every step toward a solution generates new problems and information needs (Simon 1981).

The *Representation* Element

The methods and procedures that decision-makers utilize to solve complex problems rely heavily on their ability to identify, understand and manipulate objects (Figure 7). In this respect, objects are complex symbols that convey meaning by virtue of the explicit and implicit information that they encapsulate within their domain. For example, military strategists develop operational plans by reasoning about terrain, weather conditions, enemy positions, weapon assets, and so on. Each of these objects encapsulates knowledge about its own nature, its relationships with other objects, its behavior within a given environment, what it requires to meet its own performance objectives, and how it might be manipulated by the decision-maker within a given problem scenario (Figure 8). This knowledge is contained in the various representational forms of the object as factual data, relationships, algorithms, rules, exemplar solutions, and prototypes.

The reliance on object representations in reasoning endeavors is deeply rooted in the innately associative nature of the human cognitive system. Information is stored in long-term memory through an indexing system that relies heavily on the forging of association paths. These paths relate not only information that collectively describes the meaning of symbols such as building, car, chair, and tree, but also connect one symbol to another. The symbols themselves are not restricted to the representation of physical objects, but also serve as concept builders. They provide a means for grouping and associating large bodies of information under a single conceptual metaphor. In fact, Lakoff and Johnson (1980) argue that "*...our ordinary conceptual system, in terms of which we both think and act, is fundamentally metaphorical in nature.*" They refer to the influence of various types of metaphorical concepts, such as *"…desirable is up"* (i.e., spatial metaphors) and *"…fight inflation"* (i.e., ontological or human experience metaphors), as the way human beings select and communicate strategies for dealing with everyday events.

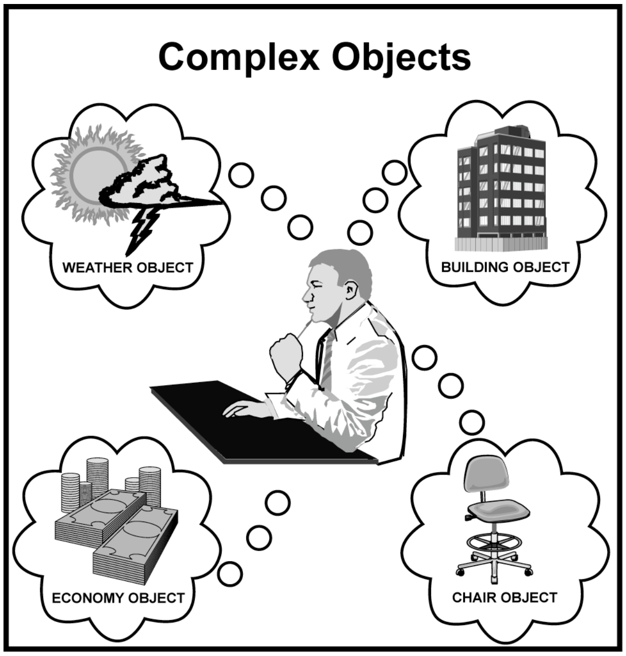


Figure 7: Symbolic reasoning with objects

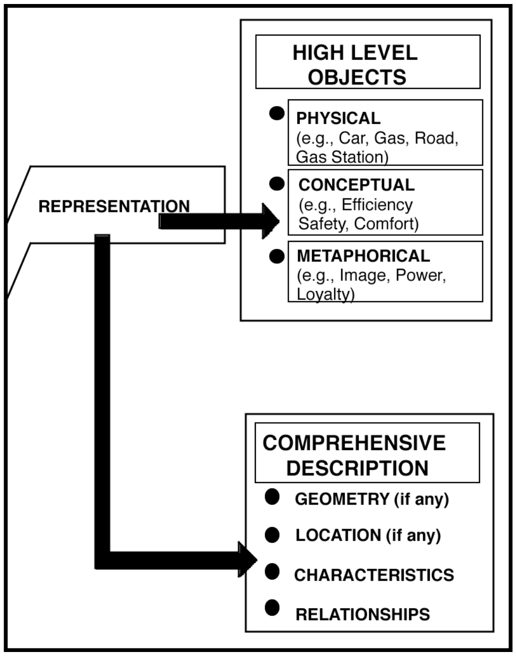


Figure 8: The ***representation*** element

Problem-solvers typically intertwine the factually based aspects of objects with the less precise, but implicitly richer language of metaphorical concepts. This leads to the spontaneous linkage of essentially different objects through the process of analogy. In other words, the decision-maker recognizes similarities between two or more sub-components of apparently unrelated objects and embarks upon an exploration of the discovered object seeking analogies where they may or may not exist. At times these seemingly frivolous pursuits lead to surprising and useful solutions of the problem at hand.

The need for a high level representation is fundamental to all computer-based decision-support systems. It is an essential prerequisite for embedding artificial intelligence in such systems, and forms the basis of any meaningful communication between user and computer. Without a high level representation facility the abilities of the computer to assist the human decision maker are confined to the performance of menial tasks, such as the automatic retrieval and storage of data or the computation of mathematically defined quantities. While even those tasks may be highly productive they cannot support a partnership in which human users and computer-based systems collaborate in a meaningful and intelligent manner in the solution of complex problems.

The term *high level representation* refers to the ability of computer software to process and interpret changes in data within an appropriate context. It is fundamental to the distinction between data-centric and information-centric software. Strictly speaking data are numbers and words without relationships[1]. Software that incorporates an internal representation of data only is often referred to as *data-centric* software. Although the data may be represented as objects the absence of relationships to define the functional purpose of the data inhibits the inclusion of meaningful and reliable automatic reasoning capabilities. Data-centric software, therefore, must largely rely on predefined solutions to predetermined problems, and has little (if any) scope for adapting to real world problems in near real-time.

Information, on the other hand, refers to the combination of data with relationships to provide adequate context for the interpretation of the data. The richer the relationships, the greater the context and the more opportunity for automatic reasoning by software agents. Software that incorporates an internal information model (i.e., ontology) consisting of objects, their characteristics, and the relationships among those objects is often referred to as *information-centric* software. The information model provides a virtual representation of the real world domain under consideration. Since information-centric software has some *understanding* of what it is processing it normally contains tools rather than predefined solutions to predetermined problems. These software tools are commonly referred to as agents that collaborate with each other and the human user(s) to develop solutions to problems in near real-time, as they occur.

The *Visualization* Element

Problem solvers use various visualization media, such as visual imagination, drawings and physical models, to communicate the current state of the evolving solution to themselves and to others (Figure 9). Drawings, in particular, have become intrinsically associated with problem solving. Although the decision-maker can reason about complex problems solely through mental processes, drawings and related physical images are useful and convenient for extending those processes. The failings of the drawing as a vehicle for communicating the full intent of the decision-maker do not apply to the creator of the drawing. To the latter the drawing serves not only as an extension of long-term memory, but also as a visual bridge to its associative indexing structure. In this way, every meaningful part of the drawing is linked to related data and deliberation sequences that together provide an effectively integrated and comprehensive representation of the artifact.

From a technical point of view a great deal of headway has been made over the past two decades in the area of computer-based visualization. However, without high-level representation capabilities even the most sophisticated computer generated images are nothing but hollow shells. If the computer system does not have even the simplest understanding of the nature of the

---

[1] Even though data are often stored in a relational database management system, the relationships that are stored with the data in such a database are structural in nature and do not provide any information on how the data will be used (i.e., the *context* of the data).

objects that are contained in the image then it cannot contribute in any way to the analysis of those objects. On the other hand, visualization in combination with high-level representation becomes the most powerful element of the user-interface of a decision-support system. Under these circumstances, visualization promotes the required level of understanding between the user and the computer as they collaborate in the solution of a problem.

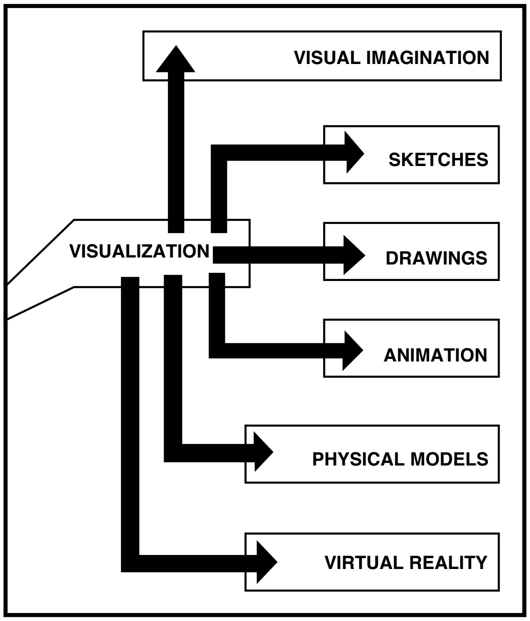


Figure 9: The ***visualization*** element

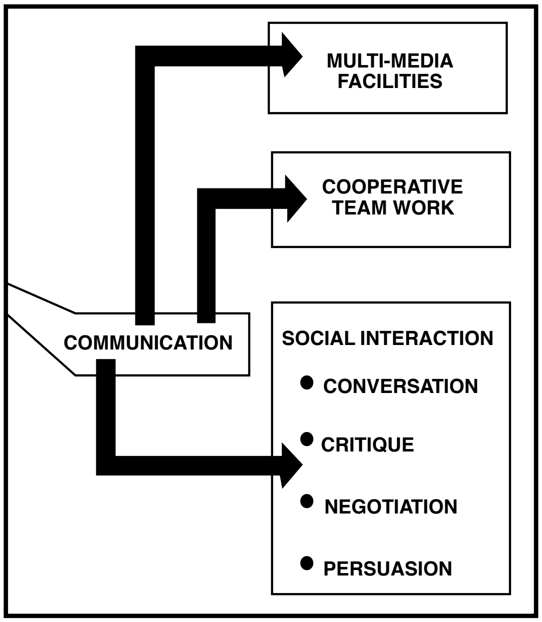


Figure 10: The ***communication*** element

The *Communication* Element

The solution of complex problems is typically undertaken by a team of decision-makers. Each team member contributes within a collaborative decision-making environment that relies heavily on the normal modes of social interaction, such as conversation, critique, negotiation, and persuasion (Figure 10). Two aspects of such an interactive environment are particularly well catered for in computer-based systems. The first aspect relates to the ability of computer-driven communication networks to link together electronically based resources located anywhere on Earth or in space. Technical advances in the communication industry have greatly enhanced the ability of individuals to gain access to remotely distributed information sources, and to interact with each other over vast distances. In fact, connectivity rather than geographical distance has become the principal determinant of communication.

The second aspect is interwoven with the first by relatively recent technological advances that have permitted all types of information to be converted into digital form. Through the use of digital switching facilities modern communication networks are able to transmit telephone conversations and graphical images in the same way as data streams have been sent from one computer to another over the past 40 years.

As a direct result of these advances in communication systems the convenient and timely interaction of all of the members of a widely dispersed problem-solving team is technically assured. It is now incumbent on software developers to produce computer-based decision-support systems that can fully support collaborative teamwork, which is neither geographically nor operationally limited. Such systems will integrate not only computer-based information resources and software agents, but also multiple human agents (i.e., the users) who will collaborate with the computer-based resources in a near real-time interactive environment.

The *Reasoning* Element

Reasoning is central to any decision-making activity. It is the ability to draw deductions and inferences from information within a problem-solving context. The ability of the problem solver to reason effectively depends as much on the availability of information, as it does on an appropriately high level form of object representation (Figure 11). Decision-makers typically define complex problems in terms of issues that are known to impact the desired outcome. The relative importance of these issues and their relationships to each other change dynamically during the decision-making process. So also do the boundaries of the problem space and the goals and objectives of the desired outcome. In other words, the solution of complex problems is an altogether dynamic process in which both the rules that govern the process and the required properties of the end-result are subject to continuous review, refinement and amendment (Reitman 1964 and 1965, Rittel and Weber 1984).

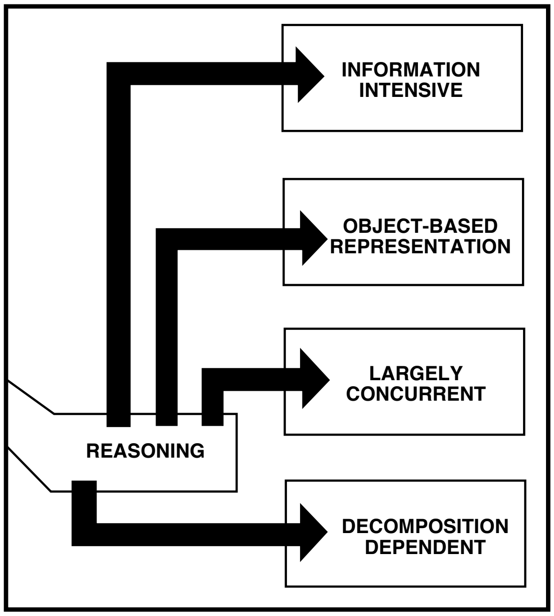


Figure 11: The ***reasoning*** element

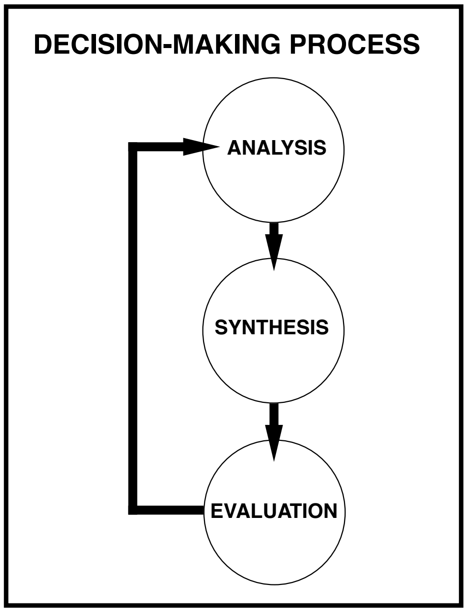


Figure 12: Reasoning methodology

As discussed previously, the complexity of a problem is normally not due to a high degree of difficulty in any one area but the multiple relationships that exist among the many issues that impact the desired outcome. Since a decision in one area will tend to influence several other areas there is a critical need for concurrency. However, the reasoning capabilities of the human problem solver are sequential in nature[2]. Accordingly, decision-makers find it exceedingly difficult to consider more than three or four issues at any one time. In an attempt to deal with the

---

[2] Reasoning is a logical process that proceeds in a step-by-step manner. In this respect reasoning is quite different from intuition, which allows humans to spontaneously come to conclusions that are neither consciously formulated nor explainable at the time of their first appearance.

concurrency requirement several strategies are commonly employed to reduce the complexity of the reasoning process to a manageable level.

> ***Constraint Identification:*** By sifting through the available information the problem-solver hopes to find overriding restrictions and limitations that will eliminate knowledge areas from immediate consideration.
>
> ***Decision Factor Weighting:*** By comparing and evaluating important problem issues in logical groupings, relative to a set of predetermined solution objectives, the decision-maker hopes to identify a smaller number of issues or factors that appear to have greater impact on the final solution. Again, the strategy is to reduce the size of the information base by early elimination of apparently less important considerations.
>
> ***Solution Conceptualization:*** By adopting early in the decision-making process a conceptual solution, the problem-solver is able to pursue a selective evaluation of the available information. Typically, the problem-solver proceeds to subdivide the decision factors into two groups, those that are compatible with the conceptual solution and those that are in conflict. By a process of trial and error, often at a superficial level, the problem-solver develops, adapts, modifies, re-conceives, rejects and, often, forces the preconceived concept into a final solution.

In complex problem situations reasoning proceeds in an iterative fashion through a cycle of *analysis*, *synthesis* and *evaluation* (Figure 12). During the *analysis* stage (Figure 13) the problem-solver interprets and categorizes information to establish the relative importance of issues and to identify compatibilities and incompatibilities among the factors that drive these issues.

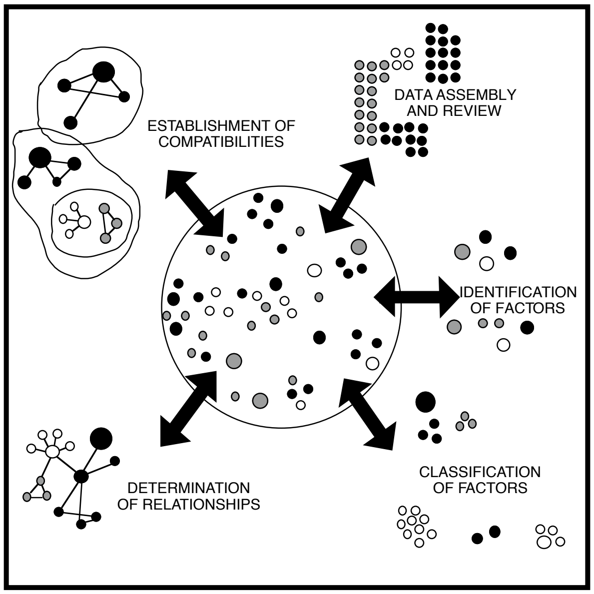


Figure 13: *Analysis* stage of reasoning

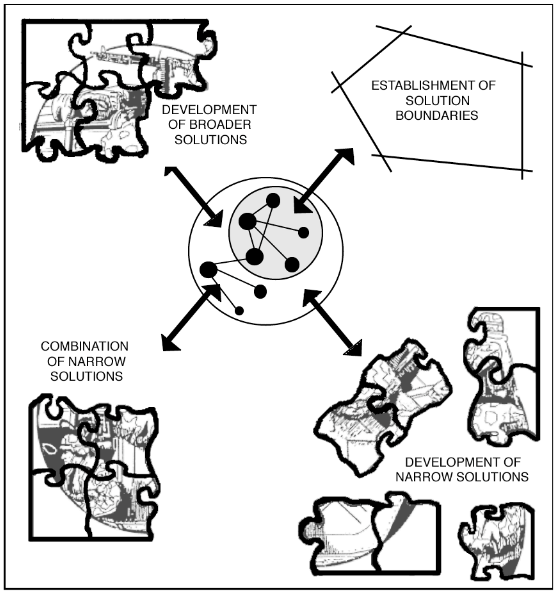


Figure 14: *Synthesis* stage of reasoning

During *synthesis* (Figure 14) solution boundaries and objectives are continuously reexamined as the decision-maker develops narrow solutions to sub-problems and combines these narrow solutions into broader solutions. Initially, these solution attempts are nothing more than trial balloons. Or, stated in more technical terms, explorations based on the development of the relationships among the principal issues and compatible factors identified during the *analysis* stage. Later, as the problem-solving activity progresses, firmer conceptual solution strategies with broader implications emerge. However, even during later cycles the solution strategies tend to be based on a limited number of issues or factors.

During the *evaluation* stage (Figure 15) the decision-makers are forced to test the current solution strategy with all of the known problem issues, some of which may have been considered only superficially or not at all during the formulation of the current solution proposal. This may require the current solution concepts to be modified, extended or altogether replaced. Typically, several solution strategies are possible and none are completely satisfactory. Archea (1987), in his description of the architectural design activity refers to this activity as "... *puzzle-making*", suggesting by implication that the decision-maker utilizes the reasoning cycle more as a method for exploring the problem space than as a decision-making tool for forcing an early solution.

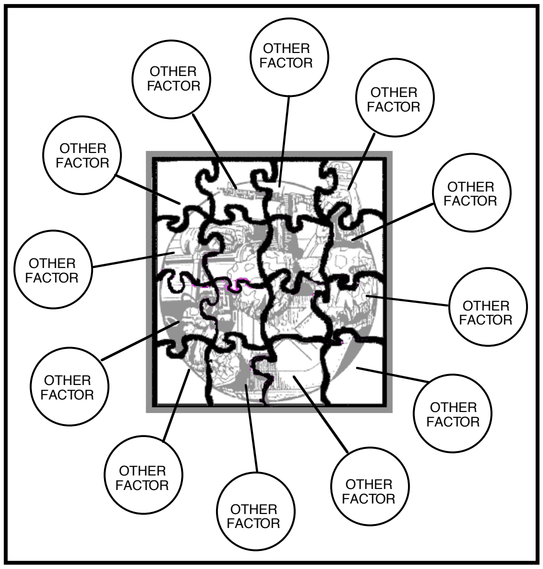


Figure 15: *Evaluation* stage of reasoning

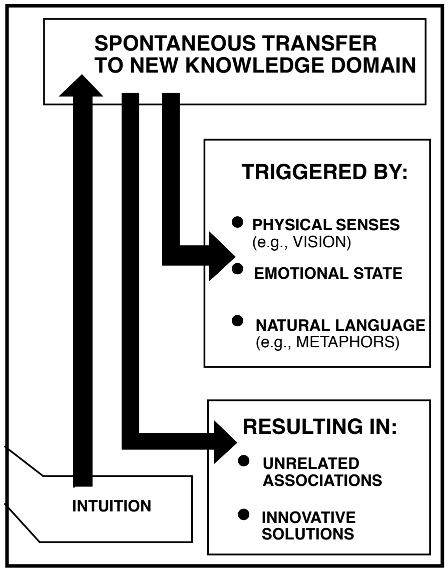


Figure 16: The ***intuition*** element

The *Intuition* Element

Donald Schon (1983 and 1988) has written extensively about the intuitive aspects of decision-making. Although he focused primarily on engineering design as an application area, his views provide valuable insight into the solution of complex problems in general. Design has all of the common characteristics of complex problem situations, and some additional ones such as the desire for solution uniqueness, that make it a prime candidate for computer-based assistance (Pohl et al.1994).

In Schon's (1988) view designers enter into "... *design worlds*" in which they find the objects, rules and prototype knowledge that they apply to the design problem under consideration. The implication is that the designer continuously moves in and out of design worlds that are triggered by internal and external stimuli. While the reasoning process employed by the designer in any particular design world is typically sequential and explicitly logical, the transitions from state to state are governed by deeper physiological and psychological causes. Some of these causes can be explained in terms of associations that the designer perceives between an aspect or element of the current state of the design solution and prototype knowledge that the designer has accumulated through experience. Others may be related to emotional states or environmental stimuli, or interactions of both (Figure 16).

For example, applying Schon's view to the broader area of complex problem solving, a particular aspect of a problem situation may lead to associations in the decision-maker's mind that are logically unrelated to the problem under consideration. However, when the decision-maker pursues and further develops these associations they sometimes lead to unexpected solutions. Typically, the validity of these solutions becomes apparent only after the fact and not while they are being developed. In popular terms we often refer to these solutions as *creative leaps* and label the author as a brilliant strategist. What we easily forget is that many of these intuitions remain unrelated associations and do not lead to any worthwhile result. Nevertheless, the intuitive aspect of decision-making is most important. Even if only a very small percentage of these intuitive associations were to lead to a useful solution, they would still constitute one of the most highly valued decision-making resources.

The reasons for this are twofold. First, the time at which the decision-maker is most willing to entertain intuitive associations normally coincides with a most difficult stage in the problem solving process. Typically, it occurs when an impasse has been reached and no acceptable solution strategy can be found. Under these circumstances intuition may be the only remaining course of action open to the decision-maker. The second reason is particularly relevant if there is a strong competitive element present in the problem situation. For example, during a chess game or during the execution of military operations. Under these circumstances, strategies and solutions triggered by intuitive associations will inevitably introduce an element of surprise that is likely to disadvantage the adversary.

The importance of the *intuition* element itself in decision-making would be sufficient reason to insist on the inclusion of the human decision-maker as an active participant in any computer-based decision system. In designing and developing such systems in the CADRC over the past decade we have come to appreciate the importance of the human-computer partnership concept, as opposed to automation. Whereas in some of our early systems (e.g., ICADS (Pohl et al. 1988) and AEDOT (Pohl et al. 1992)) we included agents that automatically resolve conflicts, today we are increasingly moving away from automatic conflict resolution to conflict detection and explanation. We believe that even apparently mundane conflict situations should be brought to the attention of the human agent. Although the latter may do nothing more than agree with the solution proposed by the computer-based agents, he or she has the opportunity to bring other knowledge to bear on the situation and thereby influence the final determination.

The Human-Computer Partnership

To look upon decision-support systems as partnerships between users and computers, in preference to automation, appears to be a sound approach for at least two reasons. First, the ability of the computer-based components to interact with the user overcomes many of the difficulties, such as representation and the validation of knowledge, that continue to plague the field of machine learning (Forsyth 1989, Thornton 1992, Johnson-Laird 1993). Second, human and computer capabilities are in many respects complementary (Figures 17 and 18). Human capabilities are particularly strong in areas such as communication, symbolic reasoning, conceptualization, learning, and intuition. We are able to store and adapt experience and quickly grasp the overall picture of even fairly chaotic situations. Our ability to match patterns is applicable not only to visual stimuli but also to abstract concepts and intuitive notions. However, although the biological bases of our cognitive abilities are massively parallel, our conscious reasoning capabilities are essentially sequential. Therefore, large volumes of information and multi-faceted decision contexts tend to easily overwhelm human decision-makers.

When such an overload occurs we tend to switch from an analysis mode to an intuitive mode in which we have to rely almost completely on our ability to develop situation awareness through abstraction and conceptualization. While this is our greatest strength it is also potentially our greatest weakness. At this intuitive meta-level we become increasingly vulnerable to emotional influences that are an intrinsic part of our human nature and therefore largely beyond our control.

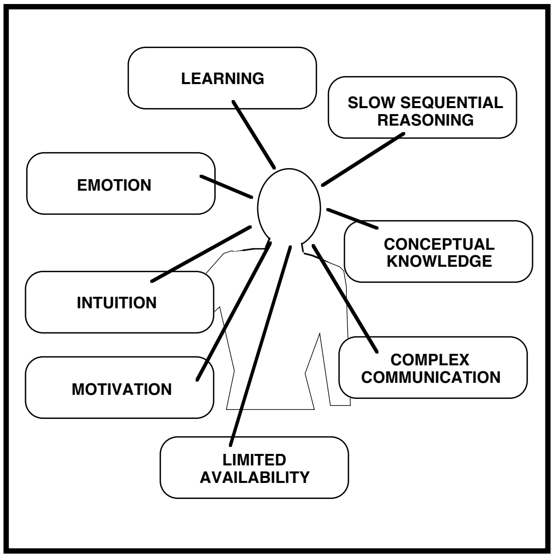


Figure 17: Human abilities and limitations

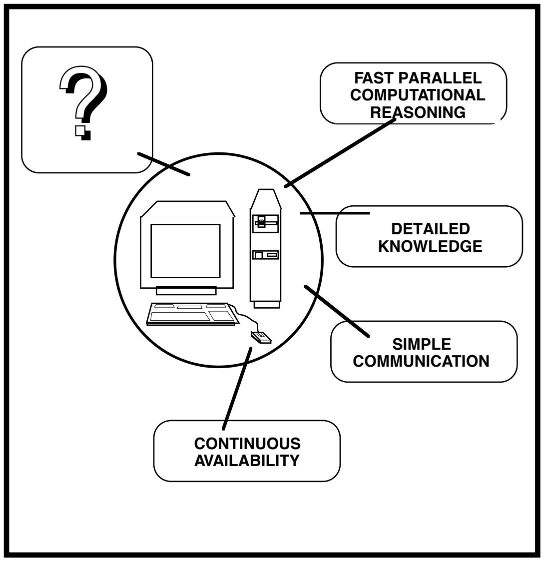


Figure 18: Computer abilities and limitations

The capabilities of the computer are strongest in the areas of parallelism, speed and accuracy (Figure 18). Whereas the human being tends to limit the amount of detailed knowledge by continuously abstracting information to a higher level of understanding, the computer excels in its almost unlimited capacity for storing data. While the human being is prone to making minor mistakes in arithmetic and reading, the computer is always accurate. A slight diversion may be sufficient to disrupt our attention to the degree that we incorrectly add or subtract two numbers. However, if the error is large we are likely to notice that something is wrong further downstream due to our ability to apply conceptual checks and balances. The computer, on the other hand, cannot of its own accord (i.e., at the hardware level) distinguish between a minor mistake and a major error. Both are a malfunction of the entirely predictable behavior of its electronic components. However, at the software level it is possible to provide a layer of automatic reasoning capabilities (i.e., collaborating agents) served by an underlying information model (i.e., ontology). Software with such embedded capabilities is able to draw inferences leading to more sophisticated human-like conclusions.

The differences between the human being and the computer are fundamental. All of the capabilities of the digital computer are derived from the simple building blocks of *0* and *1*. There is no degree of vagueness here, *0* and *1* are precise digital entities and very different from the massively parallel and largely unpredictable interactions of neurons and synapses that drive human behavior. It is not intuitively obvious how to create the high level representations of real world objects (e.g., ship, aircraft, dog, house, power, security, etc.) that appear to be a prerequisite for reasoning and learning capabilities, in a digital computer. While these objects can be fairly easily represented in the computer as superficial visual images (in the case of physical objects such as aircraft, weapons and buildings) and data relationships (in the case of conceptual objects such as power and security) that in itself does not ensure that the computer has any understanding of their real world meaning. These representations are simply combinations of the basic digital building blocks that model, at best, the external shell rather than the internal meaning of the object.

In this respect the term *information-centric* refers to the representation of information in the computer, 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, bound into volumes, folded, and so on. However, all of this is unrelated to the content of the written note on the paper. This content represents the meaning of the sheet of paper. It constitutes the purpose of the paper and governs what we do with the sheet of paper (i.e., its use). In other words, the nature and efficiency of the storage medium is more often than not unrelated to the content or representation that is stored in the medium.

In the same sense, the way in which we store bits (i.e., *0s* and *1s*) in a digital computer is unrelated to the meaning of what we have stored. For a computer to interpret data it requires an information structure that provides at least some level of *context*. This can be accomplished utilizing an ontology of objects with characteristics and a rich set of relationships to create a virtual version of a real world situation. The resultant level of information representation is normally adequate to provide the context within which agent logic can automatically operate.

References

Archea J. (1987); 'Puzzle-Making: What Architects Do When No One Is Looking'; in Kalay (ed.) Principles of Computer-Aided Design: Computability of Design'; Wiley, New York, New York.

Bohm D. (1983); 'Wholeness and the Implicate Order'; Ark Paperbacks, London, England.

Schon D. (1983); 'The Reflective Practitioner: How Professionals Think in Action'; Basic Books.

Fischer G. and K. Nakakoji (1991); 'Making Design Objects Relevant to the Task at Hand'; Proc. AAAI-91 Ninth National Conference on Artificial Intelligence, MIT Press, Cambridge, Massachusetts (pp.67-73).

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

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.

Johnson-Laird P. (1993); ‘Human and Machine Thinking’; Erlbaum, Hillsdale, New Jersey.

Lakoff G. and M. Johnson (1980); 'Metaphors We Live By'; University of Chicago Press, Chicago, Illinois.

Pohl J., A. Chapman, L. Chirica, R. Howell and L. Myers (1988); 'Implementation Strategies for a Prototype ICADS Working Model'; Technical Report, CADRU-02-88, CAD Research Center, Design and Construction Institute, College of Architecture and Environmental Design, Cal Poly, San Luis Obispo, California.

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

Pohl J. and L. Myers (1994); 'A Distributed Cooperative Model for Architectural Design'; Automation in Construction, 3(2-3), July (pp.177-185).

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

Reitman W. (1964); 'Heuristic Decision Procedures, Open Constraints, and the Ill-Defined Problems'; in Shelley and Bryan (eds.) Human Judgments and Optimality, Wiley, New York, new York (pp.282-315).

Reitman W. (1965); 'Cognition and Thought'; Wiley, New York, New York.

Rittel H. and M. Webber (1984); 'Planning Problems are Wicked Problems'; in Cross (ed.) Developments in Design Methodology, Wiley, New York, New York (pp.135-144).

Rosenman M. and J. Gero (1993); 'Creativity in Design Using Design Prototypes', in Gero and Maher (eds.) Modeling Creativity and Knowledge-Based Creative Design, Erlbaum, Hillsdale, New Jersey (pp.111-138).

Schon D. (1988); 'Designing: Rules, Types and Worlds'; Design Studies, 9(3), July (pp.181-190).

Senge P. (1993); 'Transforming the Practice of Management'; Human Resource Development Quarterly, Jossey-Bass, San Francisco, California.

Simon H. (1981); 'The Sciences of the Artificial'; MIT Press, Cambridge, Massachusetts.

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

A GAME OF COLLABORATIVE ARCHITECTURAL DESIGN: THE BIRTH OF ^{c} – HOUSE

A true simulation of a first briefing session

Gianfranco Carrara and Antonio Fioravanti

gianfranco.carrara@uniroma1.it antonio.fioravanti@uniroma1.it

Department of Architettura e Urbanistica per l'Ingegneria
University of Studies of Rome "La Sapienza"

Via Eudossiana, 18
00184 Rome
Italy

Abstract

In Collaborative Design, particularly in architecture, little is known about the effects of the mutual relations among actors in the process and on the process.

And from the outset, the actors, which, how many? And how can a design be recognized as Collaborative? Did it influence the process itself, in the sense that its structure influenced the outcome, or only changed the order in which the actors acted, did it change the final outcome? Are there any settled, invariant, issues in this jelly represented by architectural design?

These issues must be investigated in depth by those who, like ourselves, set out to build a general model simulating the process/product of Collaborative Architectural Design in order to further develop it using modern ICT tools – CollCAAD.

Over time we have examined in depth certain theoretical issues (Carrara and Fioravanti 2001), general approach problems (Carrara et al. 2000), as well as specific sectors and key issues (Carrara and Fioravanti 2002). However, the complex of relations underpinning the actions-reactions of the actors involved is a phenomenon that cannot be studied "in vitro", but must be reproduced "in vivo" and analyzed on the spot.

However, first of all, in order not to be overwhelmed in the simulation of the general model by claims to priority by requests to intervene in the design action by the actors or by mutually exclusive constraints or diverging (partial) goals of the actors, we deemed it useful to construct a "game" – a highly simplified CollCAAD model.

The aim of the game is not just to open up the way to a general model but also to provide a useful teaching tool to illustrate to the student the consequences of his own design choices vis-à-vis the choices made by others, and to accustom him to working as part of a group.

In order to implement this game, which we have called _ –House, we previously simulated a preliminary briefing session for the design of an exhibition pavilion. The aim was to immediately sift the quantities related to the observed phenomenon, identifying the fundamental ones,

discarding the non essential ones and reducing as far as possible the number of actors involved, without reducing the richness of the design dialectics.

A no less important aim of the present article is to concretely and tangibly clarify our conception regarding CollCAAD, as the same concepts and terms often take on different meanings and nuances if the context is changed.

The article consists of a report and commentary of this simulated work session which proved stimulating for the research students present and gratifying for those experiencing it. The final outcomes and reflections kept the game within given conceptual bounds, eliminating any dangerous overflows. The work session may be considered maieutic to the birth of _-House, a game that warrants an ad hoc article in order to be adequately illustrated.

1. Games in Architecture

The games, applying Plato's diairetic method, may be subdivided into games of skill or of strategy. The former are characterized by the ability to perform certain movements and therefore space and time are of importance; velocity is at a premium space/time, s/t; the latter are characterized by detachment from current reality and from the single act; the itinerary, not the time taken is valid; the overall final result is at a premium.

1.1. IMPORTANCE OF GAMES

The architecture game is more a part of the second type in which even when we speak of an 'athletic gesture' by the architect (Motorway Church by Michelucci in Florence, Guggenheim at Bilbao by F.O. Gehry, to mention but a few…) we are always dealing with a work that takes time, and is developed in time. In this second type of game the final result is important to the point of being able to afford a number of secondary losses, as, for example, in chess where it may often be necessary to sacrifice important pieces in order for the purpose of a checkmate, demolishing part of the preceding architecture works.

The study and development of 'games' simulating a complex problem or for the further study of basic concepts describing and guiding a process are extensively used in advanced design companies, in relations among human beings and in behavioural habits. Our very nature, the identification of the ego, arises out of play.

A game always has a "creative" component in so far as in a narrative context, action may be less dependent on logic (Woodbury et al. 2001) and is "joyful" in so far as the consequences of different from usual choices have no effect on reality. From this non penalization, unless in the context of the game itself, there derives a stimulus to engage in new "games" so as to extend the exploration of solution space. In this way, self-learning occurs of the "rules of effective game behaviour" not coded into the initially established "rules of the game" – the strategies.

The simulation games as defined above are played in a conventionally defined "simplified context" in order to focus attention on strategies of the 'moves', on elaborating the thinking and on the logical correctness of behaviour. The architecture games have had both the didactic purpose of placing the student in a "situation"[1] in which he can learn about the consequences of

---

[1] It should be noted that in Collaborative Architectural Design the actor is 'situated' (Gero 1999) or present in a 'condicio' (Carrara and Fioravanti 2002).

his design choices as well as research to study complex relations among operators in the specific case of architectural design[2].

In both cases the aim is to explore cognitive fields that are little known a priori and that are brought to the surface by the exploration itself, to formulate hypotheses based on such exploration and to evaluate them as a function of the objectives adopted and above all to develop new methods for exploring them better. Therefore the ultimate aim is not “the discovery” or the “solution” of something or a problem but the heuristic enhancement of the exploration: a refinement of the strategy to resolve a game.

Their great utility derives from the new strategy being applied to contexts other than the conventional one defined by the game by means of a process of extrapolation.

1.2. ARCHITECTURE GAMES

The research described arises out of several works in this field. The games developed for computer assisted architectural design always contain a plurality of aspects: the simplification of the design process, interactions among participants, difficulty of assembling the components, scoring, aspects of the process to be highlighted, the pictorial nature, likeness and figurability of the representation, the interface between player, the game and the other participants, the “items” to act on.

1.2.1. StringCVE game

For instance, in the work by J. Moloney (Moloney 2002, 2005) the game – StringCVE – refers to the early phases of design and is aimed at project rapid prototyping, mainly as regards the insertion of a building into the landscape, and a swift and pragmatic exchange of information, observations and suggestions among the actors. In order to achieve this, extensive modeling is used of a vast area of the landscape involved (1.8 Km2), the fast and cheap graphics engines of some games[3] to model the terrain, a single type of actor[4] which has a single interface, shared by actors playing the game (the students), the avatars that localize their point of view, a library of material texture, simple prismatic volumes depicting the building and labels on “significant places” – the Notation. These are comments marked on the place of intervention according to the particular perspective view of the actor-student.

Judgment on the projects is delegated to four external ‘critics’ that interact synchronically and contemporaneously among themselves and with the student, at the end of the work. The main advantage of this game played among the actors is the large number of interactions regarding the project deriving from the facility with which the building set in a figuratively realistic context can be manipulated, as well as the dialogue among all the student-actors (homogeneous among themselves) which can occur both synchronously and diachronically.

1.2.2. Cube Game

---

[2] Study by means of simulation of the phenomenon in the simplified context is of use in avoiding taking negligible or misleading parameters into account.

[3] Il Torque of Garage Games Inc, e Deep Server of Right Hemisphere Ltd.

[4] Note that an ‘actor’ is "Any participant in the design process", and it is used in this sense in this paper (Wix 1997).

Another “game” from which we have drawn some grounds for reflexion is the one developed in the courses illustrated by Prof. Y. Kalay at the University of California at Berkeley (Kalay and Jeong 2003). This game – Cube Game – mimes the construction single family homes in an allotment. The player roles are as follows: the client, the architect and the builder. The assessment is based on the market value of the building as conventionally valued. This is a hemi-symmetric situation compared with the preceding one: in the former we have many actors making judgments in a single role, in the latter only three actors with different roles with clearcut responsibilities and carefully defined skills. Each element is controlled by a single actor; there is no overlap of authority. The value of the building is obtained not only from the house itself, but also from its position in the context (streets, natural beauty) and the increased income deriving from advancing urbanization. There is a significant presence of the procedural - temporal-complexity aspect in this game (Carrara e Fioravanti 2004 pp. 427-8).

The three actors have an interface complying with a ‘specific representation’ of their own which depends to a large extent on a model of reality of their own, and therefore three models, as a function of their specific roles. Thus, in his own “specific representation”: the customer, defines the specifications of the home, the number of rooms, their interrelationship and relative arrangement, the overall budget; the architect, designs how to the assemble the simple cubes representing the rooms; the builder, establishes the cost and profit of each type of room. Each actor pursues his own particular goal: the client, the increase in the value of his investment and the satisfaction of his needs; the architect, to respond positively to the demands of many customers and reduce overall design time; the builder, to maximize his own profit. The cubes representing the rooms are interrelated by simple juxtaposition.

The main advantage of this game is the articulation of the process that sheds a clear light on the relations among the actors, the trade-off between different goals (not necessarily divergent or identical), a more explicit and vaster semantics both of the detailed and the overall goals. This second game is aimed at learning insofar as the judgment is continuous; evaluation takes place at every ‘move’ (design proposal); it is not necessary to achieve a predetermined result or respect a time limit in order to be evaluated. This is because learning also amounts to how to attain a goal, by means of which strategies, and what steps are best suited to individual strategies and not just to the goal itself.

1.2.3. Our game

These two examples, two almost completely different types of game, provided the pretext for a different game that has the objective of simulating design process in current practice in order to define a valid architecture project abundantly provided with creative ideas although with the contradictions and misunderstandings typical of the intermediate phases(Carrara and Fioravanti 2005).

The game is focused on the design process as aimed at satisfying the needs of utilization of the building and technical and construction feasibility vis-à-vis given needs. This game is part of the research carried out at the CAADLab of the Department of Architecture and Town Planning for Engineering related to Collaborative Design – CollCAAD . The immediate aim of the game is to provide a useful e-learning tool; the ultimate aim is, through the construction of the latter, to clarify and focus more clearly on the CollCAAD problems of which it represents a useful simplification.

2. The pre-game – a NO-ICT simulation of the game

The importance of games, in particular those concerning architecture, lies in the fact that they can act as a stimulus to creativity (Carrara and Fioravanti 2005) and more precisely, training for creativity. Indeed merely to mention one fundamental psychology and pedagogy study (Schoon 1992) "the more creative solutions generally come from students who are prepared to critically examine a large number of iterations".

This claim is true and valid whenever:

- the actors have already acquired critical capacities or are encouraged to enhance them (by means of the introduction of new criteria for "evaluating the conventional value" during the game);

and

- a large number of permutations of new solutions or proposed design solutions have already been developed and are being developed (in order to be effective they must be evaluated continuously and fluently Moloney 2005 pg 57).

The game that we are developing, as we shall see, takes also these two requirements into account. The first, by changing on the fly the 'weighting' given to the actors' conventional evaluations; the second, by means of "simplified pieces with simplified unions".

2.1. THE ACTORS

In order to define this game we simulated a verbal session of Collaborative Design during the early stages of the definition of the preliminary project, starting from a simple brief, a small pavilion for contemporary art works – Digital Art. Other specifications were (later neglected or somewhat overlooked) opening five days a week, from 10 a.m. to 10 p.m. and temporary exhibitions for about one or two months at most (in fig. 1, above, it is written 5 days/w, 10 – 22 open, 1 – 2 month). We recorded the intense and very fast-moving conversation taking place and photographed the whiteboard where the notes, concepts and ideas we considered essential were jotted down. The work session, which was stimulating for the PhD students present and gratifying for the participants, was intense and lasted less than two hours. Each role was performed by a researcher: but how many and which roles should have been performed? Which roles were better performed by human beings and which by agents-automata using automated procedures? Which building components were significantly important in achieving a correct trade-off among the actors?

The aim of this session was to find the types and subtypes of rules in Collaborative Architectural Design that could then be applied to the game that simulates it. They were identified and classified in the course of the session itself. To facilitate the reader's understanding of the text we will first list the "types of rules": [I] Interface, [K] Knowledge, [G] Game, [V] eValuation, [P] Project. In the text they have been boxed to highlight them.

In order to respond to these numerous questions, we began a briefing session in an 'open' way, from a minimal set of actors, building components, rules and judgments and gradually adding concepts we believed were essential during the session itself.

These minimal sets of concepts were defined both by each actor and collectively by the team.

The interactions by means of which these concepts were defined were noted using felt-tip pens on two whiteboards reproduced in figs. 1 and 2.

They simulate other media and systems in order to communicate. Likewise, oral communication requires physical supports: the air, the voice, hearing and the brain; as well as of non material supports: culture.

Already from this consideration alone the importance of a physical interface appears quite evident. It would have to accommodate two different information types: formally represented structured information (with a semantics defined collectively for all the actors and with as many semantics as there are homogeneous subgroups thereof); and informal information arising out of the verbal dialogue among the actors (which can then conveniently be rendered formal or not, by means of agreement between two interlocutors). Therefore the first rule:

[I1], first Interface rule: physical means and protocol for structured and non structured information.

Verbal dialogue is possible only in the presence of an agreement on Common Concepts, i.e. presupposes a Common Knowledge Base grounded on shared semantics.

[K1], first Knowledge rule: to have a minimum amount of shared semantics.

2.1.1. The Client

It was decided that the actor who starts the game should be the client or commissioning agent – **C** – and that he should play an active part like all the other actors during the performance of the game itself. There is no external jury to set the parameters at the beginning of the game and to check them at the end; here the evaluation by the actors, or automatically by I.A.s, is continuous and takes place whenever the Shared Project is modified.

How is alternation among the actors achieved? In two ways: through the “milestones”, when all the actors are obliged to intervene and to formulate their own judgment and assent; and in the interval between these when judgment or formulation of new design projects is optional.

[G1], first Game rule: how the actors intervene.

[V1], first rule of eValuation: criteria for assigning a value.

Each of the actors has predefined his own objectives as a function of which the project is evaluated from his own point of view This means that the project is evaluated several times in the course of the work, using different “yardsticks” depending on the actors,.

The positive consequences are as follows:

- the student, who is stimulated to learn, propose, evaluate and ‘create’ individually, is here encouraged to act in accordance with collective, synergistic principles, as in professional practice;
- the critical and self-critical spirit is refined, as recommended at the beginning of the section by Schoon, as each actor is subjected to continual global verifications with regard to the entire brief, and to personal ones with regard to his own goals; and sometimes to change his own evaluation criteria as also his own design objectives.

[G2], second Game rule: work as a group.

Like the other actors, the customer may change his design ideas in the course of the game and thus his design evaluations.

> [V2], second eValuation rule: vary one's own design criteria and that of one's own group.

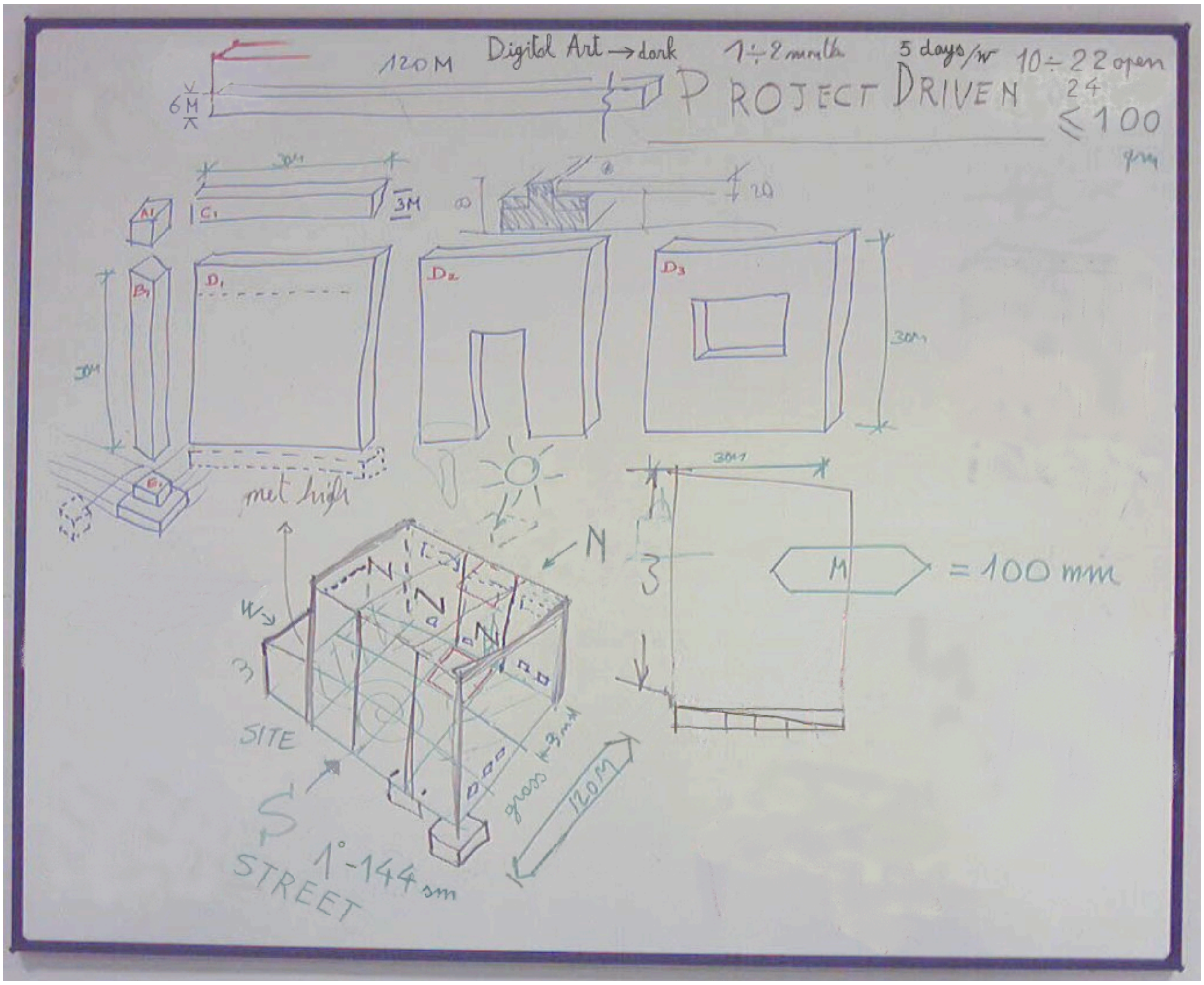


Figure 1. An example of a hardware Shared Design Workspace model = physical representation + actors' shared brain part outside the picture. Step 1, components, specifications and context (in the broad sense).

2.1.2. The Architect

After the customer – **C** – an architect – **A** – is needed to qualitatively and quantitatively spell out the customer's requirements. Under the rapid succession of demands from **A** to **C**, the brief takes shape:

> [G3], third Game rule: one move -> two actors involved – trade-off between two actors.

about 100 sm2, and the activities performed must agree with the internal heights (‘met height' written on the whiteboard fig. 1), internal height for works of art 7.05 m., other uses 3 m, display space must not be extended so that max dimension = 12.0 m.

[P1], first Project rule: the specifications[5].

The dialogue between **C** and **A**, aside from the verbal exchange, also takes place through a knowledge of a common, not technical, sense – the sketch – which is present in practically all actor relations, it is a *figurative* and *representational* knowledge of reality.

[K2], second Knowledge rule: the sketch.

The architect proposed a rectangular plan consisting of 3 x 4 modules having a 3.0 m. side length, so as to obtain 108 m2, 9 modules on top, the exhibition room and 3 below where the entrance and the offices are. Where is this? Location SITE – (written on the whiteboard in fig. 1) near a STREET, in this way further specifications are added.

[P2], second Project rule: the possibility of adding new project rules.

Consequently, the side of the building at right angles to the street means that the site configuration cannot be greater than 12.0 m, also because the customer does not want an extended display area that follows an itinerary but a rather compact area ([P1], see…). All **A**’s elements on whiteboard are coloured blue.

[I2], second Interface rule: SDW and PDW.

The whiteboard is the means used to render explicit the shared data of the project: it represents the physical transposition of the Shared Design Workspace, SDW. In it both the collectively owned data – the Common Project – and that part of the data belonging to an individual actor that he has decided to share as he has identified them as being useful to the group, appear and are therefore shared for the first time. The concepts outlined above point to a need a separation the private design workspace from the shared one.

The latter are ‘shared’ as knowledge although often it is private like intellectual property (Carrara et al. 2002).

A will use a set of envelope elements or conventional elements known to him, such as: full wall, wall with door, wall with window, etc. For the sake of simplicity he wants all these elements to be modular, with dimensions in multiples of 0.3 m. Every actor has his own items coded in a Knowledge Base, and it does not mean they are all used in every project.

[K3], third Knowledge rule: abacus of components.

This rule emerged for the first time during this CollCAAD simulation, and was contributed by **A**, but is true also for all the other actors; they too have their own “abacus of components” [K^i 3] con {i actor | i=1,…,n}. Knowledge is actually much vaster and during the simulation we will

---

[5] Each project-game can have different specifications – the context –usually in the broad sense, see note 1.

see other examples of this, without altering the nature of the type of rule [K3].

2.1.3. Structural engineer

The engineer comes into action in structure **S**, sketching [K2] in black (to distinguish himself from **A** [I2]) on the whiteboard the elements needed to define the bearing structure: a series of frames side by side that, in order to leave free space in the display area has no intermediate supports and uses beams running from one side of the building to the other.

But already at this point, although by means of different representations in the two actors (volumetric and superficial for **A**, mainly linear for **S**) these actors need:

- a connecting link between their representations, a highest common denominator of the measures;

and

- a simplification and reduction of the number of building elements.

To simplify and unify the different components, this leads to the same Euclidean space having parallelepiped elements and a modular dimensional coordination, in the case in question we adopted the International System of measures, IS, having a base size M = 0.10 m. The building elements introduced by S will be plinths, pillars, beams, floor slabs.

[I3], third Interface rule: the protocol.

This interface is the first "technical interface" that we find among specialists; indeed it is "the" interface among actors": it is the highest common denominator among the representations specific to the specialists; it is an integral part of the Common Project drawn up by the actors.

S sets out the bearing structure of the large display room using beams running transversal to the building of a length of 3 X 3 m = 9 m. It is clear that the beams cannot be included in the thickness range of **A's** default module up to now for all the 'pieces', namely 3 M = 0.3 m: these have a different variability. **S** therefore adds to the abacus of the 'common pieces' a new section of his own abacus of components, [K3], 'different' element compared with those used so far: a beam of a height equal to 10 M = 1.0 m. We now have a different application of rule [K3]: to add new pieces in the SDW, so this is a new type of rule.

[K4], fourth Knowledge rule: to be able to add new Knowledge rules.

The use of 1.0 m beams leads to conflict with **A** [G3] who does not want the ceiling of the exhibition room to be crossed by a series of visible beams; the trade-off between the two needs is achieved by virtue of a common interface: the geometrical section.

These few steps reflect the great importance of the geometry and its representation in developing an architecture project as it is not a simple representation tool but a tool for modeling reality and for design verification.

The geometry of the beam must be visible to **S**, as well as to **A.** Its principal representation is the axis lines and sections of beams and pillar, but not the spaces below. In this way S gains fresh awareness of **A's** knowledge as far as the exhibition room space is concerned so proposes moving the beams to the covering, at the extrados.

2.1.4. Energy engineering

At this stage of the project we decide that, in order to achieve the goals set out in the game[6] construction, the co-presence of a fourth actor – the energy engineer – E, is necessary, while, on the contrary, the presence of a building actor is not required – as in the *Cube Game* – as the goals are different.

With regard to the geometry being defined, side by side with what is being elaborated by **A** and **S**, **E** asks **C** and **A** to define the orientation[7] of the exhibition pavilion and the climate aspects in the project; **E** checks the openings planned by **A** in the lower part of the exhibition room on the East and North sides (see fig. 1) and, as an alternative, hypothesized as larger and high on the North and West walls.

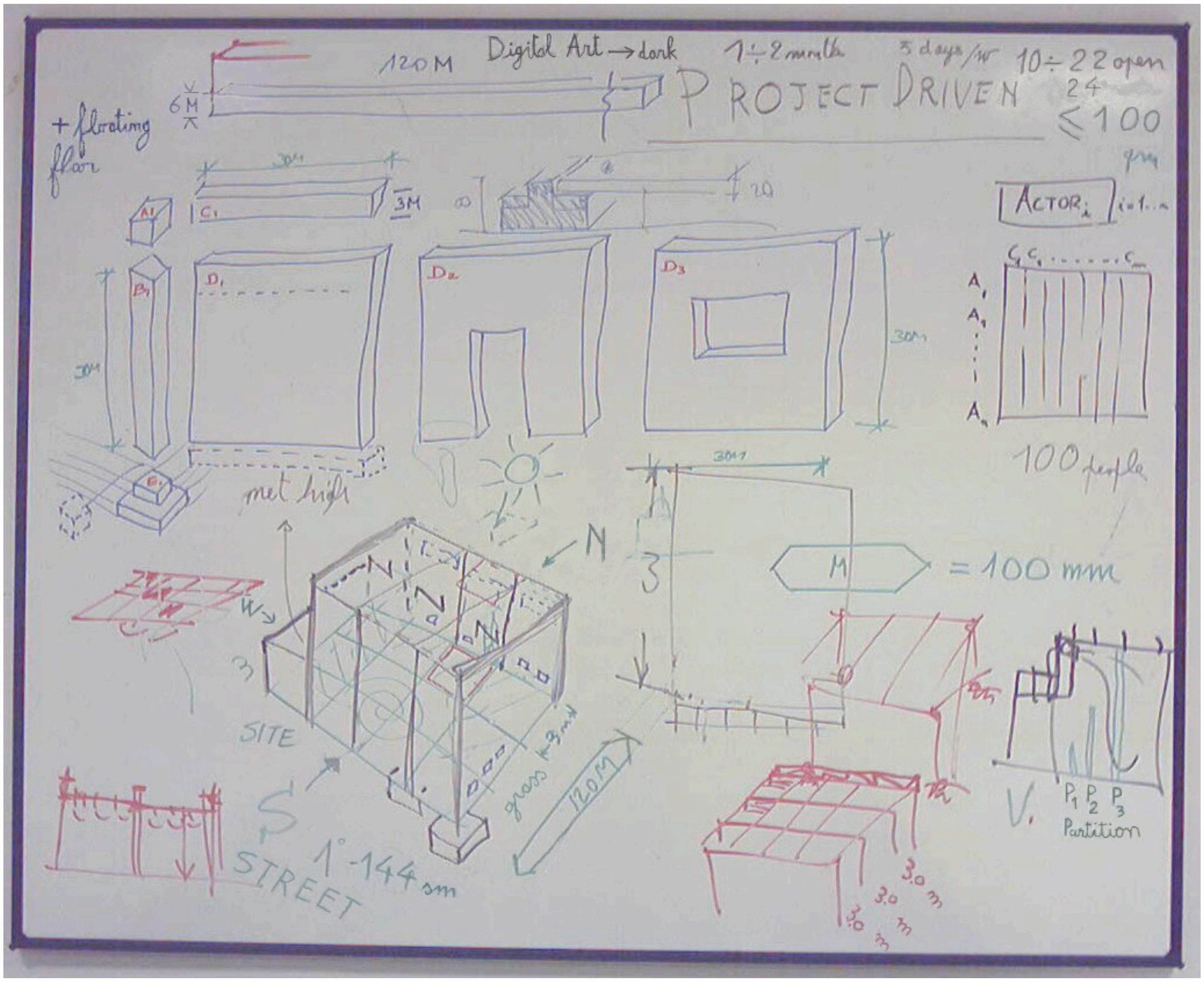


Figure 2. An example of a hardware Shared Design Workspace model = physical representation + actors' shared brain part outside the picture. Step 2, several PDWs and the *audience*.

---

[6] In an exhibition pavilion to display digital works of art environmental quality in terms of air purity, acoustics and lighting is of great importance. For the sake of extreme simplification we have made only elementary considerations concerning natural lighting.

[7] One of the new requisites, i.e. the northern exposure, is symbolized with the letter N, that is an iconic symbol visible in many sketches on the whiteboard in figs. 2 and 4.

It is important to identify who is in conflict with whom for a specific problem [8].

[G4], fourth Game rule, who is involved in what.

E immediately rejects the hypothesis of high windows on the West; **E** proposes, as clearly shown by his section sketches in red, skylights with an apparatus beneath them to filter and control the light.. One important parameter for **E** is the ratio between the building's surface area and its volume (**E** writes on the whiteboard S/V, fig. 3).

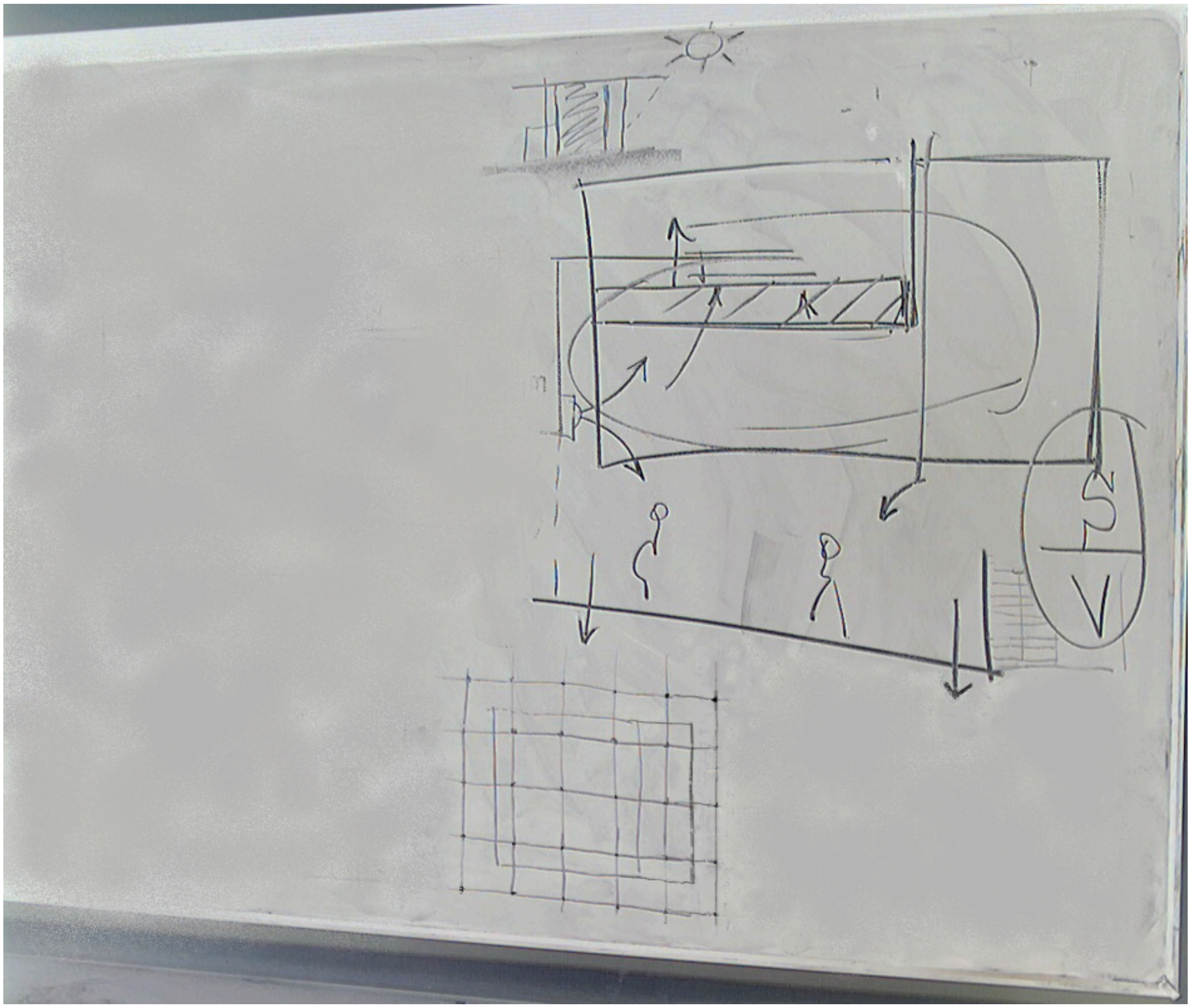


Figure 3. An example of a hardware Shared Design Workspace model = physical representation + actors' shared brain part outside the picture. Step 3, alternative layout plant, structure and HVAC system.

---

[8] The concept of the set of actors involved at a given time in a given problem – the *audience* – has already been the subject of one of our problications, i.e. Carrara e Fioravanti 2004. This concept is referred to in fig. 1 on the right with the sketch of a table; the actors – A^i – on the lines, the constraints on the columns – C^j .

This is another case in which an actor shares part of his knowledge with others.

These physical quantities (translated into the geometric model of the Common Project) are also significant for **A**. **E** grants **A** the sharing of own knowledge concerning these concepts, and of his/her conventional algorithms needed to assess them. The aim of shearing is to solve at least the simpler problems asynchronously without the intervention of human actors, but with the aid of the Intelligent Assistant present in the respective KBs.

[K5], fifth Knowledge rule: sharing.

Only at this stage **E** does interact with **A** and does a trade-off session on the Surface Area/Volume ratio begin [G3].

E proposes the input of conditioned air from above on one side of the exhibition room through conduits and machinery installed on top of the lower wing of the building.

By so doing, however, **E** imposes extra constraints on **S**: increase permanent loading of the roof slab on that part of the building. In this way rules [P2] and [P1] are applied.

If the building stood in a seismic zone there would be horizontal loads concentrated half way up the pillars of the exhibition hall which would entail redesigning the structure and/or the structural layout and/or the entire shape of the building, an endless series of subordinates would be opened up and so we shall neglect this eventuality with a view to simplicity implicit in the game.

So far we have only had conflicts due to data driven processes, that is, by inputting values, lead to consequences of interest to someone else: it is a constraint decision for several characteristics that has repercussions on another actor, due to preceding choices.

Afterwards, the pressure from two important inputs may cause significant ‘perturbations’ (Carrara and Fioravanti, 2005 page 298) in the design process: **C** is not satisfied with the roughly outlined and featureless appearance of the building. **S** autonomously proposes covering the building with a framework outside the envelope which obviously entails beams longer than the maximum size of the exhibition pavilion. In this way he puts forward solutions that do not pertain strictly to his sphere of activity but belonging to those of **A** and **E**. He does not limit himself to exploring his own possible solutions given by the parameters imposed on him by others, but expands the frontiers of his PDW, of the other PDWs, and thus of the SDW.

In this case we can no longer speak logically, in the strict sense, of "data driven" constraints, as while the project solution put forward so far “is acceptable” in the sense of satisfying all the initial requirements, now two actors autonomously propose to change (vis-à-vis the previously agreed rules): one the approach to the problem; the other, the “pieces” in the game. It is not a “data driven” process, but an “actor driven” one. They play an active role in introducing specific new design specifications [P2] and new components and knowledge [K4].

At the same time **A** describes to **C** an autonomously conceived idea to make the large exhibition room polyfunctional by means of movable partitions. **C** accepts and praises the idea as it will increase the revenue produced by the pavilion when the exhibition activity is absent.

Proposals that are autonomous versus outside constraints represent the seal of approval of Collaborative Design vis-à-vis Cooperative Design (Carrara et al. 1997). In the latter case one is limited to cooperating with the work of others by attempting to avoid conflicts, accepting compromises, optimizing ones’ own process in a well defined domain of several authorities; in the former one is "concerned" also with the good of other actors, constraints and goals are introduced in the domains of others, and one often deliberatively goes against the initial constraints (see the case of **S** when beams longer than the maximum length of the building are proposed, figs. 3 and 4, top center).

E thus points out that the planned installation is compatible with partitions P_1 and P_2 (fig. 2, bottom right), but not with the P_3 type partitions that reach ceiling height and impede air conditioning in parts of the exhibition room.

A and **C** declare the constraint of polyfunctionality of the exhibition room to be non negotiable.

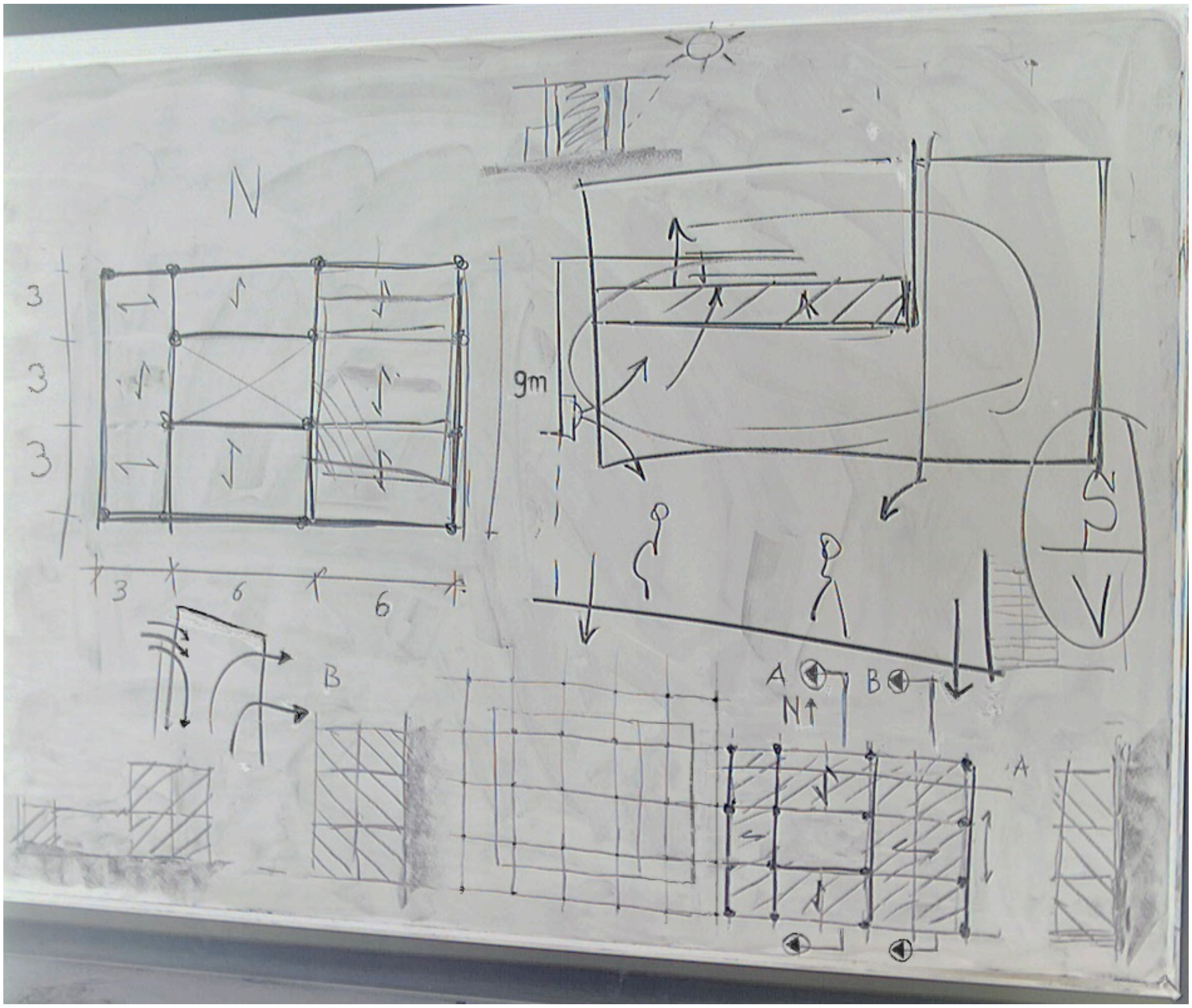


Figure 4. A hardware Shared Design Workspace model = physical representation + actors' shared brain part outside the picture. Step 4, more architectural, structural, plant design solutions.

[G5], fifth Game rule: negotiability of constraints.

This is a typical example of a constraint introduced in the course of the design process that is non predictable at the outset of the brief and, since, as observed during the game, the design depends on the data, actors and the timing of the actions undertaken, we can conclude by saying: it is by doing it that it is guided. As shown on the whiteboard it is a clear example of Project Driven Design, fig. 1.

The creative ideas emerging in the course of the process of designing a specific project in a given context are *unique* and represent the *preponderant*[9] *parameters* of every project.

So **A** changes the starting layout and proposes a building with a courtyard with the same number, 12, of square modules as the preceding one, fig. 3 symmetrical plan in the lower centre. **C** likes the idea but criticizes the lack of an exhibition room and the fact that the building has become a place of exhibition with a compulsory itinerary and the mixing of exhibition area and service area. In this ring-shaped distribution, **S** imposes the structure with all the beams on the outer perimeter in order to avoid having visible beams inside the pavilion, and he points out the low cost resulting from the relative small floor slab span. We thus discover that there were constraints in one or more KBs that affect not only his own PDW but also the SDW. In this case the cost of the structure had been imposed by **S** from the beginning of the game without informing the others of this at the outset. This is a particular example of rule K3.

Due to the criticism by **C** of the 1st ring shaped layout design solution, **A** proposes a more dynamic, less rigid, solution, restoring the exhibition room to the East, although slightly smaller sizes due to the 4x2 basic modules compared with the original 3 x 3 modules. The solution shown in fig. 4 below, center, as a variant of the "annular version" was rejected as the modules amount to 16 in number, i.e. greater than the 14 modules representing a non negotiable constraint set by **C** in a confidential fashion at the beginning of the brief. In this case there was a constraint not defined explicitly by one of the actors, since it was unimportant to him, but during the course of the design process it could become decisive. This is another special example of rule [K3]: there may be constraints not made explicit at the outset.

This may happen for two reasons: either they are intentionally not revealed by the actor at the outset, or they are implicit, not formalized by the actor as considered by him to be part of the Common KB. Indeed part of an actor's knowledge is rendered explicit immediately inasmuch as both for privacy and because the actor did not even think it necessary to render it explicit as it was obvious for him. But it often happens that what he considers obvious is for others a novelty, with conspicuous misunderstandings and therefore the actor needs to make it explicit, formalize it and publish it.

A therefore proposes to reduce the width of the room to 3 x 2 basic modules, or reduce the courtyard from 2 x 2 basic modules to 2 x 1 (fig. 4 left and bottom right).

The total number of modules is 13, more than the 12 initially laid down, but less than the non negotiable maximum number of 14. The sections for this 2nd central courtyard arrangement are set out on fig. 4 below: on the left, the longitudinal one; in the middle, the transversal one "B" on the exhibition hall; on the right, the transversal one "A" on the courtyard. **S** changes the layout of the beams to a longitudinal direction to reduce the costs of the floor slab and thus of the building, thereby balancing out the greater cost of the additional module.

It may be noted that the constraint imposed by **C** is not respected by A and transitively falls

---

[9] The creative ideas are predominant because they characterize a project by given it meaning, quality and policy decisions.

on on **S**; this is an example of “net-constraint” (Carrara et al. 1997): the constraint refers to the network of relations among actors. The constraint may refer to the whole team, not to one or two individual actors.

[G6], sixth Game rule: one move -> many actors involved – trade-off among several actors.

In this way the exhibition hall is again traversed by visible beams that **A** does not want. **S** proposed to change the direction of the beams and the floor slab (fig. 4 bottom right plan) at the price of an increase in costs. This leads to a three-way negotiation (**C**, **A**, **S**) as **E**, in this phase, has performed his task and it is not considered necessary to involve him again in order to reduce plant costs.

Here we have a second type of net-constraint that refers not so much to one technical solution or another, or the increased cost of one and the decreased cost of the other specification item as to the overall conception of the planned work. The overall evaluation must be made using criteria different from the initial ones.

From this case we infer that the evaluations (the scores in the case of the game) assigned by each actor to the project have coefficients (the ‘weights’), varying in accordance with their utilization or with the “context” in which they operates.

In this case, as it is an exhibition pavilion, priority is given to the attractive appearance, the pleasure of frequenting highly quality spaces, and so the actors may agree to assign a greater “weight” to **A’s** evaluation and therefore prefer the solution s/he proposes.

3. The definition of the game: _ – house

3.1. CONSIDERATION OF THE PRE-GAME

The simulation performed by the actors in the first briefing session revealed numerous invariants[10] regarding Collaborative Architectural Design and the development of a game. These will be useful both in e-learning programs and in the study of the mechanisms of design interactions and iterations.

The first invariant. *From the outset all the actors–disciplines involved must be present, or at least, the more significant ones*.

In our case we cannot drop below four actors without the model changing from simple to simplistic.

Also the idea of replacing the builder in respect to the *Cube Game* (1.1.2 section) with **S** and **E** is correct, as the research topic is related more to the technical and building aspect than to the industrial and financial aspect. The need to immediately involve the principal professional figures from the outset is due to the presence of net-constraints (Carrara et al. 1997) that they delegate to the activity and choices of several actors, often the whole group.

---

[10] We consider a phenomenon or a characteristic of a structured set as *invariant* , somewhat like Lévi-Strauss when, with changing boundary conditions, the reciprocal relations of the set and the context mean that this phenomenon or characteristic vary according to compensatory relations.

Second invariant. For the construction of the game it is necessary *to calibrate the topic to avoid it becoming too vast. A corollary to this need is that the "items" at the outset be severely reduced in number*, so that, although as seen in the simulation during the design process new ones may be added, their number will nevertheless remain limited.

The third invariant. *Architectural design as such, in so far as it is related to the 'uniqueness' of the building in a unique 'condicio'* (Carrara and Fioravanti 2002) *normally required, also in simplified contexts, the invention and definition of new components*.

Indeed, initially envisaged by **S** in the simulation was an excessive number of components for structural design, such as plinths, nodes, slabs, 'T' beams (fig. 1) which were then not utilized; while it was necessary to introduce a 9.00 m long beam 0.90 m high, and then a 6.00 m long and 0.60 m high beam.

Fourth invariant. *In the resolution of conflicts it is often necessary to redefine the range of solutions confined at the outset by the presence of over- stringent constraints*.

Of course, in the case of the game, this possibility will be limited to special cases.

In the course of the simulation we observed that on several occasions, in order to break the deadlock, we had to overrule and sometimes modify several initial specifications, e.g. surface area (from 100 to 110 sm, and thence from 110 sm to 117 sm), obtaining the advantage of architectural quality (central courtyard, building volumes better balanced and expressive of their designated use); the air conditioning plant (that is changed from the type with input from a single-side to a two-side type) with the advantage of being able to fractionate the exhibition room.

Often it is not only necessary to change the project limits, and thus to intervene on the PDWs and SDW, but also to redefine the field of operations among the various actors, as when for **E** we simplified its tasks, no longer air conditioning with minimal engineering problems but only: exposition with respect to the cardinal points, energy balance and water distribution system. Or when it is necessary to introduce new professional figures, such as the environmental actor in the study of important interventions or the localization of problems.

Fifth invariant. *The gradual closer focusing on design objectives could lead to objectives that are quite different from the starting ones*.

Modification of specifications thus led not so much to a "softening" of a number of parameters, merely to ultimately attain the much-desired approbation of all the actors, as to overall advantages such a greater aesthetic appeal, and thus the capacity to attract more visitors, as well as a stronger emotional involvement, thus to facilitate memory of the visit for advertising promotion; fractionability and polyfunctionality, thus greater return on investment.

On closer view it is seen to entail recognizing the decisive importance of evaluation factors that were not even taken into consideration at the beginning of the game as they were not well known. This involves the redefinition and/or introduction of new design objectives.

Il game must therefore make the SDW and PDW paradigm effective.

To achieve this paradigm, three conditions must occur in reality: the three-dimensional geometric consistency of the construction, the shareability of data and knowledge, the intelligibility of the data.

These conditions are present in the game in simplified form and thus give rise to as many invariants.

Sixth invariant. *The importance of a system of 'common coordinates' in which to "fix the items" and render their positions certain in order to control their incompenetrability in the SDW, while allowing the possibility of a 'private coordinate' system, the PDWs.*

Since the different actors are working each in his own PDW and thus defining their own project and their own geometry of the building and architectural components, it may happen that one or more of these conflicts volumetrically with others because often the actors design the same components. This aspect is aggravated by the fact that the actors use a symbolic, conventional (and simplified) representation of reality for the geometrical modeling of the parts concerning their contribution to the project, such as, just by way of example, the lines of the axis of a structural beam with regard to its volume[11].

Therefore, in order to identify and then avoid problems of inconsistency of the geometrical data of the objects making up the construction, it is necessary to dispose of a single geometric model of it to which all can refer. As in the simulation of the briefing session we were able to represent the ideas of all the actors on a single SDW consisting of a whiteboard where it was possible to ascertain geometrical inconsistencies, in reality it would be necessary to define a SDW capable of containing all the components designed by the actors in a space of "reference coordinates" common to all, and use it to ascertain their consistency.

Seventh invariant. *The interface must be able to make available the "Common Data" specific to the whole group and the 'Shared Data', that are part of the data and knowledge specific to an actor who decides to share them.*

Obviously the actor also works on his "Private Data", which are not shared with others.

Other aspects of the preliminary briefing session that have been glossed over, as it were, and which exist outside the whiteboard, are those linked to intellectual property and to the privacy of the shared data.

Without clear-cut regulations and the assignment of intellectual property no Collaborative Design can be successful (Carrara et al. 2004).

Eighth and the final, but not the least important, invariant is the semantics of shared components and of concepts. *In order to be able to exchange information and knowledge it is necessary to have Shared Knowledge* and from this knowledge, through the Perspective/Filter mechanism, "to translate" the semanteme of a universe of one actor to another.

Another important connotation of the game is its *recursiveness*. Indeed when a new rule of the Game – [G] – claims that there may be new game rules [G]; or when a knowledge rule – [K] – defines that the old components are made by others and/or new components we have the application of a mechanism of logic, i.e. 'conceptual recursiveness'. In this way, the boundaries of a knowledge domain are extended, scaled down and redefined while thanks to the high degree of polymorphism of the paradigm SDW/PDWs it is possible to structure the same classes of objects in different ways in different domains. This means that we can explore other problems typical of collaborative architectural design through techniques of I.A.

---

[11] In this article we have overlooked the problems arising due to the use of application programs that have different geometric structures that would make it more difficult to achieve an interoperability among the programs themselves.

4. Conclusions towards c-House

To this end, and also thanks to the contribution from the study of previous games, we would like to give *a number of definite directions* to our game, that we have called c-House, in order to respond to the aims illustrated earlier.

Unlike the *StringCVE* game, wishing to take into consideration the technical and construction aspects and the consequent problems it is necessary to have several actors – building design professionals. Among these we shall not consider landscape architects who lie outside our objective. For the number of actors involved reference is made to section 2.1 and the first invariant of section 3. 2

As *the number of actors having different goals grows*, the game becomes increasingly realistic – *communications and conflicts and suggestions and proposals arise* as a result of these goals, *overall feasibility is pursued*, *there is a greater likelihood that creative options will emerge*. the precondition for this is to have the ability to *interface [I]*.

Again unlike the first game there will be no external judgment at the end of the game but, since the aim is to simulate a design team following the Collaborative Design paradigm, *the eValuations [V] are given by all the actors at each phase*. For the sake of simplicity, several evaluations may be performed automatically by means of agents – the evaluators – for example those regarding costs [P], without using quantity surveyors. We will not have an implicit semantics of information exchanged among actors, which is adequate only when we are in the presence of a single actor, but an *explicit semantics [K]*. Just as, owing to the plurality of actors, the evaluation (the game point score) must be *made explicit*, and no longer synchronically, but at least for some *asynchronically*, in different phases. This is all because we have a plurality of Private Design Workspaces – PDWs – in which to develop hypotheses and evaluations, which are asynchronic with respect to the Shared Design Workspace – SDW. Consequently, *interaction by the actors with the project* can no longer be chaotic, but *must be subject to process rules [G]*.

With respect to the second game – *Cube Game* – attention is focused more on architectural design so that the *professional figure of the Energy-Plant Engineer is introduced*; we shall instead neglect all hypotheses regarding the evaluation of real estate income; we shall *explicitly and formally define the semantics of the "concepts"* not on in the PDW but also in the SDW; likewise we shall explicitly and formally evaluate the judgments[12] not only in the PDW but also in the SDW; lastly, let us seek to fill in the gap of *being able to perform overall project evaluations within their own PDW*, without necessarily having to make any modifications to the SDW.

In our case the modifications to set off the controls can take place in a *'test' environment* which in no way passes on these modifications to the shared and accepted environment.

4.1. THE RULES

The "rules" discovered in the briefing session were classified under five different headings: I-Interface, K-Knowledge, G-Game, V-eValuation, P-Project. They are numbered in order of appearance, not in order of logical necessity. Indeed, it may very well happen that "afterwards" it is discovered that pre-existing to and subsumed one rule another might be necessary. For example, rule [I2] (PDW/SDW) pre-exists rule [K2] (sketch): without a PDW and a mechanism

---

[12] The scores achieved in the game.

for its publication into the SDW, the actor cannot enable the actors to participate in his own sketch. This observation can only occur empirically, after the premises of some rules have come to light.

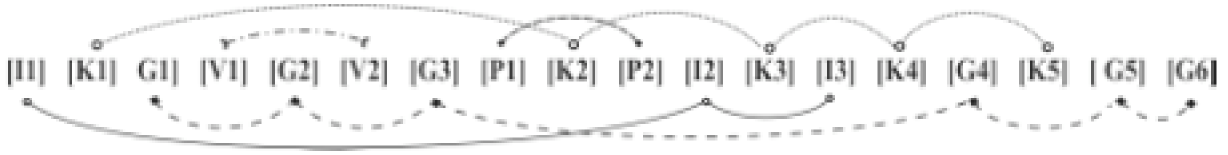


Figure 5. Sequence of rules in the pre-game session.

The five types of rules are considered in abstract: we refer to categories of prototypes in a "universal" sense and not contingency related. For example, the concept of 'evaluation of a project' is essential for the briefing regardless of the existence or non existence of project specifications. It is always true that these specifications are, as we saw earlier, a premise for the expression of the evaluation but, and here we have the counter inference, the absolute evaluation could be the result of other considerations and not be dependent on specifications.

The five types are fundamentally irreducible. They cannot be reduced, from a logical point of view, to a lesser number of type or even "ad unum".

Without in any way detracting from the possibility that in information science these types may be "structurally" [13] reduced to a lesser variety, as indeed we think would be desirable in the structural logic of the information science object-agent.

The game rule subtypes, [G], are disclosed at the beginning and towards the end as, in order to begin the process needs rules and at the end, insofar as it is a (complex phenomenon) CollCAAD it requires processing rules, that cannot be identified a priori.

The knowledge rules, [K], display an interesting behaviour: they gather "harmonically" in a geometric progression in order to specify the entities: semantemes, sketch, components, metacomponents, shareability.

After this progression towards explicit expression the need is no longer felt for other knowledge rule subtypes: they are merely repeated in new instances of the same.

As regards the eValuation rules, [V], they are first specified and very shortly after rendered more elastic (as the process varies), although they are essentially fixed around the beginning of the session and other subtypes are no longer introduced.

The Project rules, [P], are situated halfway through the discovery of the rules, like the [V] rules they consist of a specification subtype and a variation possibility subtype.

Lastly the Interface rules, [I], are physically indispensable at the beginning (physical means) and then specified after the middle of the sessions when the interactions come fully into play and the interface is called upon to decide other important questions, like coherence/incoherence ones.

Running quickly through the list of rules emerging and identified in the course of the

---

[13] In the sense of structuralism – a current of thought which interprets anthropology – types which have similar logical-formal and relational structures.

simulation of a preliminary briefing session we observe that the game rules predominate over all the other rules as, having simplified that actors, the components and the required knowledge as far as possible, what prevails is rightly the aspect of the game regulation: its process.

Of course, the number of rule subtypes in no way corresponds to the quantity of knowledge contained in them, for instance, each time the project is changed on the basis of a trade-off rule [G5], clearly the knowledge of one or more actors may be changed although all these modifications are merely new occurrences of [K3]. That is to say, a clear distinction must be made between the numerosity (the cardinality of the instantiation of the type) with which a type of rule is applied and the number of subtypes of a given rule type.

This rapid overview shows how the objective of studying the relations of a CollCAAD process by means of simplified simulation succeeded in identifying a number of rules governing this process. These "universal" rules will go to make up the foundations on which the implementation of a game – c-House – is based, taking into account the observations made during the preliminary briefing session. The game will be to a large extent conventional and 'simple' as regards the geometry [P], the participating actors, the construction components [K], their characteristics, the criteria of evaluation [V], the usual windows interface [I]. All this should serve to focus research, on the one hand, on the specific project logic of the actors-professions; and on the other, on the interactions among the actors [G].

5. Acknowledgements

The research was partially funded by University of Studies of Rome "La Sapienza", Athenaeum Research Project 2002: "A model for collaborative integrated architectural design". The author who, in the simulation of the briefing session played the part of **A**, wishes to thank the other participants in the "game": PhD D. Raccah – **S**, Profs. C. Cianfrini and M. Corcione – **E** and my friend Prof. Y. Kalay – **C**.

References

Carrara G and Fioravanti, A: 2005, Creative architectural design boosted by ICT enhanced incubator, *in* JS Gero and R Sosa (eds), *Computational and Cognitive Models of Creative Design*, 6th International Roundtable Conference, Heron Island, 10-14 December 2005, pp. 275-300.

Carrara, G, Fioravanti A: 2004, How to construct an audience in Collaborative Design - The Relationship among which Actors in the Design Process, *in* B Rudiger, B Tournay and H Orbaek, (eds), *Architecture in the Network Society*, 22nd eCAADe Congress, 15 - 18 September 2004, pp. 426-434.

Carrara G, Fioravanti A and Nanni U: 2004, Knowledge Sharing, not MetaKnowledge. How to join a collaborative design Process and safely share one's knowledge, *in* J Pohl (ed.) *Intelligent Software System for the New Infostructure*, 16th Focus Symposium and InterSymp-2004 Conference Proceedings, Baden-Baden, 30 July 2004, pp. 105-118.

Carrara G and Fioravanti A: 2002, 'Private Space' and 'Shared Space' Dialectics in Collaborative Architectural Design, *in* J Pohl (ed.) *Collaborative Decision-Support Systems*, 14th Focus

Symposium and InterSymp-2002 Conference Proceedings, Baden-Baden, 30 July 2002, pp. 27-44.

Carrara G. and Fioravanti A: 2001, A Theoretical model of shared distributed Knowledge bases for Collaborative Architectural Design, *in* JS Gero and K Hori (eds), *Strategic Knowledge and Concept Formation III* Conference Proceedings, Sydney, 17-18 December 2001, pp. 129-144.

Carrara, G., Fioravanti, A., and Novembri, G.: 2000, A framework for an Architectural Collaborative Design, *in* D Donath (ed.), *Promise and Reality. State of the Art versus State of practice in computing for the design and planning processes*, Proceedings of 18th eCAADe 2000 Conference, Weimar, 22-24 June 2000, pp. 57-60.

Carrara, G, Fioravanti, A, and Novembri, G: 1997, An Intelligent Assistant for architectural design studio, in B Martens (eds.), Challenges of the Future, 15th eCAADe 1997 Conference, Wien, a Web page in http://info.tuwien.ac.at/ecaade/proc/carrara/carrara.htm

Kalay YE, Jeong, Y: 2003, A collaborative Design Simulation Game, *International Journal of Architectural Computing*, **1**(4).

Kalay YE, Jeong, Y: 2003, Collaborative Design Simulation Game, Proceedings of the 8th CAADRIA Conference, Bangkok Thailand 18-20 October 2003, pp. 745-758

Lee AWK: 2001, Design Computing Education Software Development Integration of Scaffolding Strategies and Multiple Representations, *in* K Koszewski and S Wrona (eds), *[Designe-ducation]* Connecting the Real and the Virtual, 20th eCAADe Conference Proceedings, Warsaw 18-20 September 2002, pp. 140-143.

Moloney, J: 2002, StringCVE Collaborative Virtual Environment software developed from a game engine, *in* K Koszewski and S Wrona (eds), *[Designe-ducation] Connecting the Real and the Virtual*, 20th eCAADe Conference Proceedings, Warsaw 18-20 September 2002, pp. 522-525.

Moloney, J: 2005, Game Engine and Virtual Design Studios – Technology and Pedagogy, *in* JP Duarte, G Ducla-Soares and A Zita Sampaio (eds), *Digital Design: the Quest of New Paradigms*, 23rd eCAADe Conference Proceedings, Lisbon 21-24 September 2005, pp. 55-62.

Reffat RM and Gero JS: 1999, Situatedness: A New Dimension for Learning Systems in Design, *Architectural Computing from Turing to 2000*, 17th eCAADe Conference Proceedings, Liverpool, 15-17 September 1999, pp. 252-261.

Schoon, I: 1992, Creative achievement in architecture: A psychological study - (Psychological Studies), Leiden University Press.

Wix J: 1997, ISO 10303 Part 106, BCCM (Building Construction Core Model) /T200 draft.

Woodbury, RF, Shannon, SJ and Sterk, TD: 2001, What works in a design game? - Supported by student reactions to being made to play, *in* JS Gero, S Chase and M Rosenman (eds), 6th CAADRIA 2001, Sydney, 19-21 April 2001, pp. 411-420.

Woodbury, RF, Wyeld, TG, Shannon, SJ, Roberts, IW, Radford, A, Burry, M, Skates, H, Ham, J, and Datta, S: 2001, The Summer Game, *in* H Penttilä (ed.), *Architectural Information Management*, 19th eCAADe 2001, Helsinki, 29-31 August 2001, pp.293-297.

Multi-Agent System for Airspace Control in the Combat Zone

V. A. Skormin

Center for Advanced Information Technologies
Binghamton University, Binghamton NY, U.S.A.

M. Pechoucek

Gerstner Laboratory, Faculty of Electrical Engineering
Czech Technical University, Prague, Czech Republic

T. E. Busch

Air Force Research Laboratory,
Rome NY, U.S.A.

Abstract

Successful airspace control is one of the key factors maximizing the effectiveness of military air operations. It includes long and short-term planning and control that utilizes large and dynamic databases, and constitutes a combination of resource allocation, routing, scheduling, and deconfliction tasks performed repeatedly under information uncertainty to accommodate for the continuously arriving new information reflecting the dynamics of the battlefield. Combined with large volumes of data to be analyzed and stringent time requirements, these tasks place heavy burden on personnel, leading to costly inefficiencies. Modern computing technologies are capable of expanding the share of airspace control functions performed by computers resulting in numerically justified decisions that will enhance planning and reduce pressure on its personnel without freeing them from the ultimate responsibility. Computer-based planning, scheduling, and control are based on a mathematical formulation of the entire problem. Due to the high complexity, the problem solution is to be decomposed and its particular subsets are obtained in a decentralized, but coordinated fashion. This approach is best served by the “multi-agent” system technology that is deployed as the computational engine behind the airspace control system described herein. The paper features the distributed coordination mechanisms based on collective decision-making (voting) and sharing complex social knowledge (individual flight plans and aircraft status), design and implementation of the computational model, design and development of the algorithms and required knowledge structures for distributed coordination, the agent architecture and specific agents responsible for data collection/updating and planning/scheduling/deconfliction tasks, and the required visualization technology.

Key words

airspace control; air traffic; planning; multi-agent systems; software agents; deconfliction.

Introduction

Successful airspace control is one of the key factors maximizing the effectiveness of air operations and the entire combat operation. It includes long and short-term planning and scheduling, and real-time control. These functions utilize a large and dynamic database and constitute a very complex combination of resource allocation, space allocation, routing, scheduling, deconfliction, and control tasks that are to be solved under information uncertainty and rigid time constraint. In addition, these tasks must be solved periodically to accommodate for the continuously arriving new information reflecting the dynamics of the battlefield operation. The Airspace Control Authority (ACA) has highly trained personnel utilizing their knowledge, experience, and intuition to perform all necessary ACA functions. However, the complexity of the tasks, large volumes of data to be analyzed and stringent time requirements, place a heavy burden on individual planners that combined with the entire scope of issues labeled as "human factor," may adversely affect the quality and timeliness of ACA decisions. Consequently, these could be among the factors limiting the success of the air campaign and preventing advanced military equipment and personnel to utilize their potential to the fullest. Availability of modern computing technologies creates the conditions when the share of ACA functions performed by computers could be expanded resulting in numerically justified decisions and reduced pressure on its personnel. This effort is aimed at the development of a computer-based system technology that will enhance ACA operation providing ever-increasing support to its personnel without freeing them from the ultimate responsibility.

An air traffic control system of the future is visualized as a fully decentralized, automated computer-based system. Such a system would allow for the utilization of the capabilities of personnel, equipment, and munitions to their full potential, as well as maximum safety of the air operation. A battlefield environment is highly dynamic. The decentralization results in the most flexible air operation control system that can easily accommodate for the rapidly changing situations providing that the necessary information is obtained and processed in a timely fashion. The latter could be achieved only by computer-based, fully automatic data acquisition and decision support. These considerations are implemented in the system for air traffic control in the battlefield zone presented herein with the goal of the maximum utilization of rapidly changing data and providing timely decision support to personnel of ACA (Wickens et al., 1998).

The solution to an air traffic control/planning problem is sought in the spatial, functional, and time domains. The *spatial aspect* of the problem deals with the geographical map, coordinates of the air bases, targets, airborne refueling stations and hazardous areas, and the utilization of airspace. It results in the definition of rational and safe routes for particular aircraft connecting the base of original deployment to the target or multiple targets, and to the designated landing base, when necessary through refueling areas. The f*unctional aspect* of the problem addresses the task assignment to particular pilots/aircraft, weapons-to-targets assignment, and the logistics of the entire air operation. The *time-domain aspect* of the problem includes the scheduling of the operation of particular aircraft at the take-off and landing stages, in-air refueling, and engaging targets. It could be seen that every aspect of the planning process is dictated by the tactical and intelligence information, affected by weather, and is consistent with the technical characteristics of aircraft, weapons, and targets. The system is expected to compile better-than-average air traffic control plans that will be presented for approval to ACA personnel in the enhanced, user-friendly format, which would complete the planning stage of the process.

The computer-based system represents the "big picture" of the airspace control problem in the *functional-domain*, *spatial-domain* and *time-domain* based on a mathematical model. Particular entities of the model "live" in the simulation environment and as such, obey laws of mechanics and aerodynamics, engage in communication among themselves and with the mission control, expend fuel and ammunition, experience various hazards, sustain battlefield damage, etc., and ultimately provide invaluable feedback for the enforcement of particular considerations and rules of engagement, detection of conflicts and deconflicting, and assessment of the "goodness" of the planning decisions. The system obtains numerical solutions of the particular subsets of the airspace control problem and coordinates "local," independently obtained solutions, thus resulting in conflict-free, long and short-term plans and schedules. The intermediate solutions are coordinated and deconflicted with the enforcement of specific considerations, and when possible, optimized. The entire solution task is visualized as an ongoing, iterative process driven by continuously updated databases reflecting the battlefield dynamics and newly arrived data. The capability of incorporation of human expertise presented in a formalized and intuitive fashion, and accommodation of new rules, considerations, and conditions is viewed as an important feature of the system. It could be further enhanced by inclusion of statistical analysis tools, and supervised and unsupervised learning capabilities.

Operation of the system includes planning and execution stages. At the planning stage, the plan of the entire air operation utilizing time-invariant data, such as geographical and performance characteristics of the aircraft, and a priory given information, such as the initial description of the air operation, is established. The execution stage addresses the effect of all factors preventing the implementation of the accepted plan of air operation, as well as the possible deviations from the plan. In order to assure the completion and overall success of the operation, the proposed system has the capability of rapid re-planning (deconfliction) achieved at the lowest possible cost. This process must employ some collaboration/negotiation between the involved entities. It facilitates the control of the air operation and is accomplished by providing updated assignments to individual pilots in a timely fashion. Consequently, the proposed system should perform the data acquisition task on a continuous basis and utilize reliable and secure communication channels with individual aircraft, as well as successful visualization techniques.

In many ways, the realization of the above capabilities is well served by the implementation of multi-agent system technology that has been successfully deployed for a number of large-scale software engineering projects for industrial and military applications (Wooldridge and Jennings, 1995) and specifically for airspace control (Tomlin et al., 1997), (Hill et al., 2005). Recent advancements in multi-agent system technology provide the means for the development of fully automated planning, scheduling, and operation control systems for complex, multivariable processes exhibiting hybrid (both continuous and discrete) behavior. Modern multi-agent systems emulate complex collaboration and information exchange processes taking place within a group of human experts engaged in finding a compromise solution of complex problems. Driven by mathematically justified procedures and utilizing high-speed computers, these systems consistently generate better-than-average and very prompt solutions that could be continuously updated on the basis of most recent information available. Unsurprisingly, multi-agent system technology has been chosen for the development of the ACA system featured in this paper.

Specifics of the airspace control problem

Solution of the airspace control problem results in an Airspace Control Plan (ACP) that allocates critical battlefield resources, equipment, space, and time reflecting

- Rules of engagement and disposition of air defense weapon systems,

- Air, land and maritime situations in the area of responsibility such as existing equipment limitations, electronic warfare, and C4 requirements that may adversely affect adherence to the ACP,

- Anticipated restricted area based on initial deployment of friendly forces and bases,

- Existing air traffic control areas, base defense zones, controlled or uncontrolled airspace, and overflight of neutral nations,

- Mission profiles, combat radii, and identification capability of aircraft operating in the area of responsibility,

- Enemy air defense weapons capabilities, deployment, and electronic attack/deception capabilities,

- Emergency procedures for aircraft experiencing difficulties,

- Procedures for day, night, and adverse weather conditions,

- Procedures for en route and terminal area air traffic control procedures for aircraft transitioning to and from the battle area that complement planned combat requirements,

- Procedures to support surge operations requiring high volumes of air traffic,

- Enemy offensive air capabilities, vulnerability of defensive counter aircraft to enemy surface-to-air missiles and vulnerability of friendly surface-based air defenses to enemy long-range artillery (Airspace…, 2005).

It important that a straight-forward attempt to plan/schedule the missions unavoidably requires that the following issues be addressed:

1. *Traffic hazards* i.e. potential conflicts with other objects on the surface or in flight such as other aircraft, missile launches, or other potential hazards characterized by the number, type, position, and intent available via surveillance.

2. *Current en route weather hazards* including hail, icing, turbulence, high winds associated with thunderstorm activity, thunderstorm activity over oceanic airspace, wind shear and microburst alerts, intensive precipitation, and areas of low visibility and tornadoes. This information is available from the Global Weather Information System.

3. *Rational airspace utilization* due to the fact that the value of the airspace for all users becomes increasingly critical as military operations, commercial operations, general aviation, rocket launches, and artillery shells compete for airspace. Airspace use/availability information

is dynamic; it allows utilizing available airspace to enhance flight operations for both mission and economic priorities.

4. *Aircraft-to-airspace separation* ensures that aircraft maintain a safe distance from special use airspace, such as hazardous and warning areas defined via intelligence and surveillance data and regulatory publications and specific control instructions. Separation standards ensure that aircraft remain at an appropriate minimum distance from such areas.

5. *Aircraft-to-aircraft en route separation* in airspace ensures that a safe distance is maintained between aircraft. Separation standards are defined for the different aircraft operating environments. They separate aircraft using standard rules for vertical, lateral, and longitudinal separation. When potential conflicts exist, an air traffic planner evaluates the situation and develops conflict resolution alternatives. Special rules exist for aircraft to aircraft separation services in oceanic airspace.

6. *Aircraft-to-aircraft separation in terminal airspace* ensures that a safe distance is maintained between aircraft. Within terminal airspace, requirements for separation vary by airspace Class. There are standard rules for vertical, lateral, and longitudinal separation methods. When potential conflicts exist, an air traffic planner evaluates the situation and develops conflict resolution alternatives.

7. *Aircraft-to-terrain/obstacle separation* that ensures that aircraft maintains a safe distance from terrain and obstacles.

8. *Current Surface Separation* that prevents taxi conflicts and runway incursions.

Consequently, the planning process constitutes a number of parallel, semi-autonomous tasks, utilizing common, continuously updated databases that are aimed at the detection and resolution of the conflicts. The solution process is typically decentralized and results in “local” solutions reflecting “local” criteria and constraints that must be coordinated in the interests of the overall solution (FAA 2005).

Existing practices of air traffic control

The air traffic control system is a vast network of people and equipment that ensures the safe operation of commercial and private aircraft. Air traffic controllers coordinate the movement of air traffic to make certain that planes stay a safe distance apart. Their immediate concern is safety, but controllers must also direct planes efficiently to minimize delays. Some regulate airport traffic through designated airspaces; others regulate arrivals and departures.

Although *airport tower controllers* or *terminal controllers* watch over all planes traveling through the airport’s airspace, their main responsibility is to organize the flow of aircraft into and out of the airport. Relying on radar and visual observation, they closely monitor each plane to ensure a safe distance between all aircraft and to guide pilots between the hangar or ramp and the end of the airport’s airspace. In addition, controllers keep pilots informed about changes in weather conditions such as wind shear, a sudden change in the velocity or direction of the wind, that can cause the pilot to lose control of the aircraft.

During arrival or departure, several controllers direct each plane. As a plane approaches a base, the pilot radios ahead to inform the terminal of the plane's presence. The controller in the radar room, just beneath the control tower, has a copy of the plane's flight plan and already has observed the plane on radar. If the path is clear, the controller directs the pilot to a runway; otherwise, the plane is fitted into a traffic pattern with other aircraft waiting to land. As the plane nears the runway, the pilot is asked to contact the tower. There, another controller, who also is watching the plane on radar, monitors the aircraft the last mile or so to the runway, delaying any departures that would interfere with the plane's landing. Once the plane has landed, a ground controller in the tower directs it along the taxiways to its assigned gate. The ground controller usually works entirely by sight and/or relies on radar information if visibility is very poor.

The procedure is reversed for departures. The ground controller directs the plane to the proper runway. The local controller then informs the pilot about conditions at the airport, such as weather, speed and direction of wind, and visibility. The local controller also issues runway clearance for the pilot to take off. Once in the air, the plane is guided out of the airbase's airspace by the departure controller.

After each plane departs, airbase tower controllers notify *enroute controllers* who will now take charge. Nationally, there are 20 air route traffic control centers located around the country, each employing 300 to 700 controllers, with more than 150 on duty during peak hours at the busiest facilities. Airplanes usually fly along designated routes; each center is assigned a certain airspace containing many different routes. Enroute controllers work in teams of up to three members, depending on how heavy traffic is; each team is responsible for a section of the center's airspace. A team, as exemplified by commercial aviation, might be responsible for all planes that are between 30 and 100 miles north of an airport and flying at an altitude between 6,000 and 18,000 feet.

To prepare for planes about to enter the team's airspace, the radar associate controller organizes flight plans coming off a printer. If two planes are scheduled to enter the team's airspace at nearly the same time, location, and altitude, this controller may arrange with the preceding control unit for one plane to change its flight path. The previous unit may have been another team at the same or an adjacent center, or a departure controller at a neighboring terminal. As a plane approaches a team's airspace, the radar controller accepts responsibility for the plane from the previous controlling unit. The controller also delegates responsibility for the plane to the next controlling unit when the plane leaves the team's airspace.

The radar controller, who is the senior team member, observes the planes in the team's airspace on radar and communicates with the pilots when necessary. Radar controllers warn pilots about nearby planes, bad weather conditions, and other potential hazards. Two planes on a collision course will be directed around each other. If a pilot wants to change altitude in search of better flying conditions, the controller will check to determine that no other planes will be along the proposed path. As the flight progresses, the team responsible for the aircraft notifies the next team in charge of the airspace ahead. Through team coordination, the plane arrives safely at its destination.

Both tower and enroute controllers usually control several planes at a time; often, they have to make quick decisions about completely different activities. For example, a controller might direct a plane on its landing approach and at the same time provide pilots entering the airport's airspace with information about conditions at the airport. While instructing these pilots, the controller also might observe other planes in the vicinity, such as those in a holding pattern waiting for permission to land, to ensure that they remain well separated.

In addition to airbase towers and enroute centers, air traffic controllers also work in flight service stations operated at more than 100 locations nationally. These *flight service specialists* provide pilots with information on the station's particular area, including terrain, preflight and inflight weather information, suggested routes, and other information important to the safety of a flight. Flight service specialists help pilots in emergency situations and initiate and coordinate searches for missing or overdue aircraft. However, they are not involved in actively managing air traffic.

Some national air traffic controllers work at the FAA's Air Traffic Control Systems Command Center in Herndon, VA, where they oversee the entire system. They look for situations that will create bottlenecks or other problems in the system, then respond with a management plan for traffic into and out of the troubled sector. The objective is to keep traffic levels in the trouble spots manageable for the controllers working at enroute centers.

The FAA has implemented an automated air traffic control system, called the National Airspace System (NAS) Architecture. The NAS Architecture is a long-term strategic plan that will allow controllers to more efficiently deal with the demands of increased air traffic. It encompasses the replacement of aging equipment and the introduction of new systems, technologies, and procedures to enhance safety and security and support future aviation growth. The NAS Architecture facilitates continuing discussion of modernization between the FAA and the aviation community (Nolan, 1990).

While the above description primarily reflects the operation of commercial aviation, it provides sufficient detail for the purpose of this project.

Multi-agent planning and execution processes

In the nearest future, the advanced methods of computer science and artificial intelligence will play a pivotal role in air traffic control of military and civilian as well as manned and unmanned aerial vehicles. We have been investigating the use of agent based technology and the multi-agent algorithms for deployment in this specific application domain.

Multi-agent system is a collection of loosely coupled autonomous programs that perform collective behavior and collective decision making by means of interaction, negotiation, cooperation but also methods of teamwork, competition or social dominance. Multi-agent system domain provides a wide selection of ready to use COTS or open source integration platforms as well as various techniques and algorithms suitable for different coordination tasks.

The use of this highly innovative technology is appropriate in the situations where the data required for decision making are not available centrally. As air traffic control domain needs to move to less human driven problem and a problem more suited for automated decision making, we expect that substantial amount of computation and data maintenance will be onboard of the aircraft. Similarly, the future air-traffic operation (especially in battle-field or surveillance operations) will require techniques implementing safe, fast and robust deconfliction algorithms and would allow for other replanning scenarios in highly dynamic and unpredictable environment.

This expectation leads to investigation of a highly decentralized decision making systems that will make an important use of the available multi-agent technologies. Operation of a multi-agent air traffic control system is supposed encapsulate the following 3 decision making phases:

Data acquisition

Time-invariant data includes geographical information (digital map); performance characteristics of aircraft, primarily operational speed ranges and fuel burn rates given for various Standard Configuration Loads (STL); coordinates of the friendly airbases; and airspace design (aircraft separation) criteria that could be defined for various visibility conditions (i.e., sizes of the air corridors and tunnels, and communication, alert, safety, and collision zones around aircraft). Airspace design criteria are established based on the capability of aircraft to accurately fly and maintain pressure altitudes in higher altitude cruise and based on the capability of the aircraft and the relationship to separation criteria in lower altitude situations. Airspace design criteria for flight objects for a special use (hazardous/restricted) airspace activity include the time duration and volume of airspace around the trajectory required to execute the mission. This addresses dynamic airspace restrictions with variable separation for security, military operations, remotely operated aircraft, and reusable launch vehicles. *Time-varying data* includes the plan of the air operation that designates targets for particular aircraft (pilots) and assigns weapons to target and defines the NET (not earlier than) and NLT (not later than) times for particular target; weather-related information; coordinates and status of particular targets; and hazardous areas, also known as special use airspace (areas defended by SAMs, areas occupied by flying artillery shells, rocket launches, etc.). It could be seen that this information reflects the battlefield dynamics, i.e., changing goals of the air operation, neutralization of targets and detection of new targets and hazardous areas, changing weather conditions, etc. Finally, *reported data* represents the current status of the particular aircraft, such as payload, technical status, available fuel status, and actual aircraft position.

Initial planning

The first step of the initial planning process begins with establishing a logical time schedule for the neutralization of particular targets that constitute a subset of the air operation plan. This is followed by assigning aircraft/weapons to targets, selection of take-off airbases, and the bases where aircraft are to return after the completion of the mission. Temporal coordinates of the rest of the node points are to be calculated based on the average speed of the aircraft and the geometrical distance between the appropriate locations. At the next step of the initial planning, all intermediate points of the aircraft paths are to be calculated by interpolation, assuming that the node points are connected by straight lines in the four-dimensional space. The number of intermediate points is defined according to some chosen time step and average speed of the aircraft.

Deconfliction

The flight plans that are results form the initial planning process may contain possible conflicts and collision situations. Collision avoidance is not solved during the initial planning process due to high computational requirements related to this process and due to high dynamics of expected flight traffic. The detection and resolution of the conflicts criteria utilized in this process are defined based on the capability of aircraft to accurately fly and maintain required altitudes. Criteria for flight objects for a special use (hazardous/restricted) airspace activity include the time duration and volume of airspace around the trajectory required to execute the mission. This addresses dynamic airspace restrictions with variable separation for security, military operations, remotely operated aircraft, and reusable launch vehicles. It should be emphasized that detection and resolution of the conflicts takes into consideration weather conditions, time of the air operation, and the geographical region that dictates the size of the air corridors, air tunnels, communication, and alarm and danger zones surrounding aircraft.

The deconfliction process can be physically embedded in the initial planning phase or fight execution phase, described below. If deconfliction is to be executed during initial planning it needs to be implemented on top of a multi-agent simulation of the flight-plans elaborated during the initial planning process. Possible collisions will be resolved by the multi-agent deconflcition methods and log of the resulting operation will provide the final non-conflicting plans. More natural alternative is to implemented deconfliction within the flight execution process. The aircraft would follow their mission plan and carry out deconfliction process up in the air. This concept is referred to as *free-flight* and is particularly suitably for unmanned aerial vehicle operation.

Flight execution

The execution stage addresses the effect of all factors preventing the implementation of the accepted plan of air operation. These factors include unexpected changes in weather conditions, damage sustained by particular aircraft, actual fuel status, newly detected targets and hazardous zones, failure to neutralize targets according to the plan, failure to follow the required schedule, failure to stay within the designated corridor/tunnel, etc. It could be seen that in addition to making the goals of air operation unattainable, these factors can result in additional conflicts. In order to minimize the effect of these factors on the completion and overall success of the operation, the proposed system has the capability of rapid re-planning (deconfliction) achieved at the lowest possible cost. This process must employ some collaboration/negotiation between the involved entities. It facilitates the control of the air operation and is accomplished by providing updated assignments to individual pilots in a timely fashion. Unlike the initial planning, conflict resolution at this stage implies a decision process that takes into account when reported (real) data on technical status of the involved aircraft is available, amount of fuel on board, and the aircraft position.

Status of the Implementation

The agent-based air traffic based on the architecture listed in the previous section has been designed recently and has been implemented on top of the ***A*-globe** multi-agent platform (Sislak et al., 2005) and has been presented at (Pechoucek et. al, 2006). The system features technology for agent based flight modeling, air-traffic planning mechanism for a single plane, rule-based and utility based deconfliction mechanisms (See Fig. 1.). Specific negotiation-based conflict resolution procedures have been developed and implemented in multi-agent environment originally suggested by (Schulz et al. 1997), (Tomlin et al. 1997), and further developed for airspace deconfliction by (Pechoucek et al. 2006). The deconfliction mechanism is distributed by its nature that allows for addressing the high volume of computations associated with the solution of this problem.

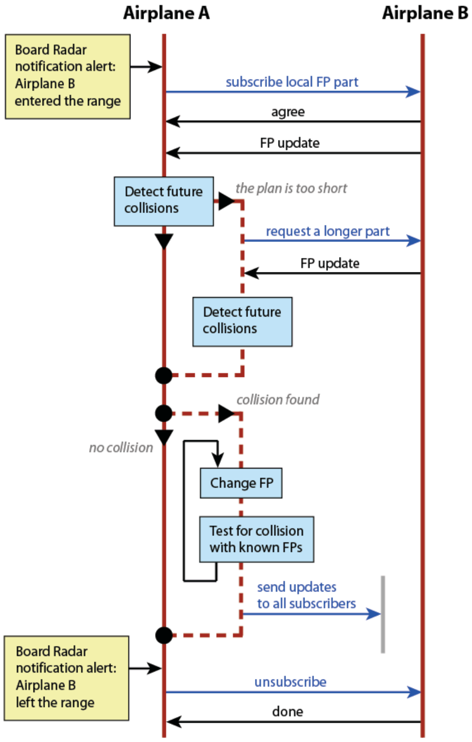
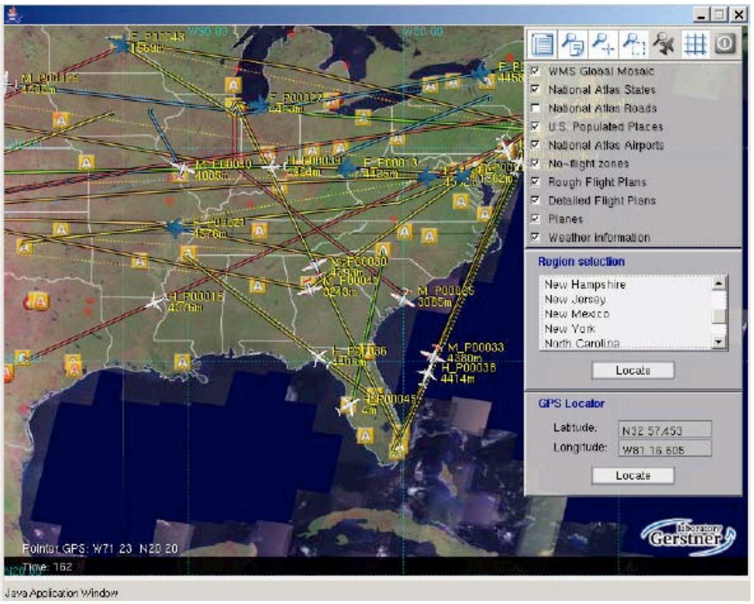


Fig.1. Deconfliction negotiation protocol

Currently, massive scalability tests are under development. The developed system also provides 3-dimensional and web-accessible 2-dimensional presentation (GUI) layer (see Fig. 2). The system performs data-fusion on top of various data from freely available data-sources that have been integrated in the system (e.g. mosaic of Landsat7 images, USGS geographical data, GNIS name-related data, but also almost real-time data from the airport traffic monitors of major U.S. airports).



☑ WMS Global Mosaic
☑ National Atlas States
☐ National Atlas Roads
☑ U.S. Populated Places
☑ National Atlas Airports
☑ No-flight zones
☑ Rough Flight Plans
☑ Detailed Flight Plans
☑ Planes
☑ Weather information

Region selection

New Hampshire
New Jersey
New Mexico
New York
North Carolina

Latitude: N32 57.453
Longitude: W81 16.605

Fig.2. Presentation layer

Acknowledgement

The authors are grateful to AFRL and OSD for funding this project and to John A. Beyerle for valuable suggestions.

References

Marik, V., Pechoucek, M., Stepankova, O. (2001). Social Knowledge in Multi-Agent Systems, In (M. Luck et. al, eds.) *Multi-Agent Systems and Applications*, LNAI 2086, Springer-Verlag, Heidelberg, 2001, pp. 211-245.

Pechoucek, M., Marik, V., Barta, J. (2002). A knowledge based Approach to Coalition Formation. In: *IEEE Intelligent Systems*. vol. 17, no. 3, IEEE Computer Society, New York, pp. 17-25.

Pechoucek, M., Marik, V., and Stepankova, O. (2001). Towards Reducing Communication Traffic in Multi-Agent Systems. In: *Journal of Applied System Studies*, Cambridge International Science Publishing, No.1, Cambridge, UK.

Pechoucek, M., Marik, V., Stepankova, O., Macurek, F., Tichy, P. (1999). Meta-agent: A Workflow Mediator in Multi-Agent Systems. In: *Intelligent Workflow and Process Management: The New Frontier for AI in Business*. Vol.1. San Francisco: Morgan Kaufmann Publishers.

Riha, A., Pechoucek, M., Vokrinek, J., Marik, V. (2002). ExPlanTech: Exploitation of Agent-based Technology in Production Planning, In: *Multi-Agent Systems and Application II*, (Marik, V. et. al., eds.), LNAI 2322, Springer Verlag.

Sandholm, T. (1999). Distributed Rational Decision Making, In: (Weiss, G. ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pp. 201-258, MIT Press.

Tambe, M. (1997). Towards Flexible Teamwork. In: *Journal of Artificial Intelligence Research*, AI Access Foundation and Morgan Kaufman Publishers, Vol. 7, pp. 83-124.

Wooldridge, M., Jennings, N. (1995). Intelligent Agents: Theory and Practice. In: *The Knowledge Engineering Review*, Vol. 10, No. 2, pp. 115-152.

"Airspace Control in the Combat Zone," Air Force Doctrine Document 2-1.7, 13 July 2005.

Wickens, C.D., Mavor, A.S., Parasuraman, R., McGee, J.P. (1998). *The Future of Air Traffic Control: Human Operators and Automation*, The National Academies Press.

Smith, P. G. (Smith, P. G. ed.), "Concept of Operations in the National Airspace System in 2005", FAA, 1999.

FAA List of all Operational Improvements in the NAS Architecture 5 database, Oct 2005.

Nolan, M. S. (1990). *Fundamentals of Air Traffic Control*, Wadsworth Inc.

Tomlin, C., Pappas, G., and Sastry, S. (1997). Conflict resolution for air traffic management: A case study in multi-agent hybrid systems. *IEEE Transactions on Automatic Control*.

Zlotkin, G. and Rosenschein, J. S. (1989). Negotiation and task sharing among autonomous agents in cooperative domains. In: N. S. Sridharan, editor, *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 912–917, San Mateo, CA, Morgan Kaufmann.

Sislak, D., Rehak, M., Pechoucek, M., Rollo, M., and Pavlicek, D. (2005). A-**globe**: Agent development platform with inaccessibility and mobility support. In: *Software Agent-Based Applications,Platforms and Development Kits*, pp. 21-46, Berlin, Birkhauser Verlag.

Pechoucek, M., Sislak, D., Pavlicek, D., and Uller, M. (2006b): Autonomous Agents for Air-Traffic Deconfliction. In: *Proc. AAMAS-06 – Industry Track*, ACM Press, pp. 1498-1505

Hill, J. C., Johnson, F. R., Archibald, J. K., Frost, R. L., and Stirling, W. C. (2005). A cooperative multi-agent approach to free flight. In *AAMAS*, pp. 1083-1090.

TRANSWAY: Planning with the Tabu Search Algorithm

J. Nibecker, D. Taylor, R. Chambers, H. Larsen, K. Cudworth, C. Warren, M. Porczak, and J. Pohl

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

Abstract

Military deployment and distribution responsibilities call for intelligent collaborative tools in support of strategic and operational planning functions involving the sustainment and movement of military forces. The sustainment requirement is generated at the operational level and is dynamic. It is composed of shifting priorities responding to changes in commander's intent and changes in the operational situation.

The TRANSWAY software application is designed as a set of intelligent collaborative tools supporting operators performing planning and re-planning tasks in a dynamically changing decision-making environment. TRANSWAY includes several agents with strategic and operational planning and re-planning capabilities. The principal agent is based on the Tabu Search algorithm, with the intent of finding an optimum plan for the delivery of supplies from multiple origins, through multiple routes, with different kinds of conveyances, to multiple destinations, within specified time and resource constraints.

The TRANSWAY System Architecture

The TRANSWAY system has a three-tier, service-oriented architecture, implemented using the Integrated Cooperative Decision Making (ICDM) ontology-based software development framework and the Hibernate object/relational persistence and query service. Figure 2.1 provides an illustration of the key components within each of these tiers (i.e., *presentation*, *information*, and *logic* tiers).

TRANSWAY incorporates an internal information model (i.e., ontology) consisting of objects, their characteristics, and the relationships among those objects. The information model is a virtual representation of the real world domain under consideration and is designed to provide adequate context for software agents (typically rule-based) to reason about the current state of the virtual environment. Since information-centric software has some 'understanding' of what it is processing it normally contains tools rather than predefined solutions to predetermined problems. These tools are commonly software agents that collaborate with each other and the human user(s) to develop solutions to problems in near real-time as they occur. Communication between information-centric applications is greatly facilitated since only the changes in information need to be transmitted. This is made possible by the fact that the object, its characteristics and its relationships are already known by the receiving application.

The *presentation* tier interfaces with human operators through a Graphic User-Interface (GUI) comprised of a menu system, map display, agent display, and various reports. The main TRANSWAY GUI is based on the Generic Space Generator (GSG) framework employing Java Bean technology and offering high performance map and graphics management. The map display supports a variety of map formats (e.g., CADRG, satellite imagery, etc.) and provides standard map interaction functionality (i.e., zoom, pan, highlight, layer management, etc.). In addition, due to its objectified nature theater and operational entities (e.g., tracks, operations centers, routes, planned activities, etc.) can be presented within the map display and interrogated through direct operator interaction. The agent display shows various concerns and recommendations generated by the agents for the operator to inspect.

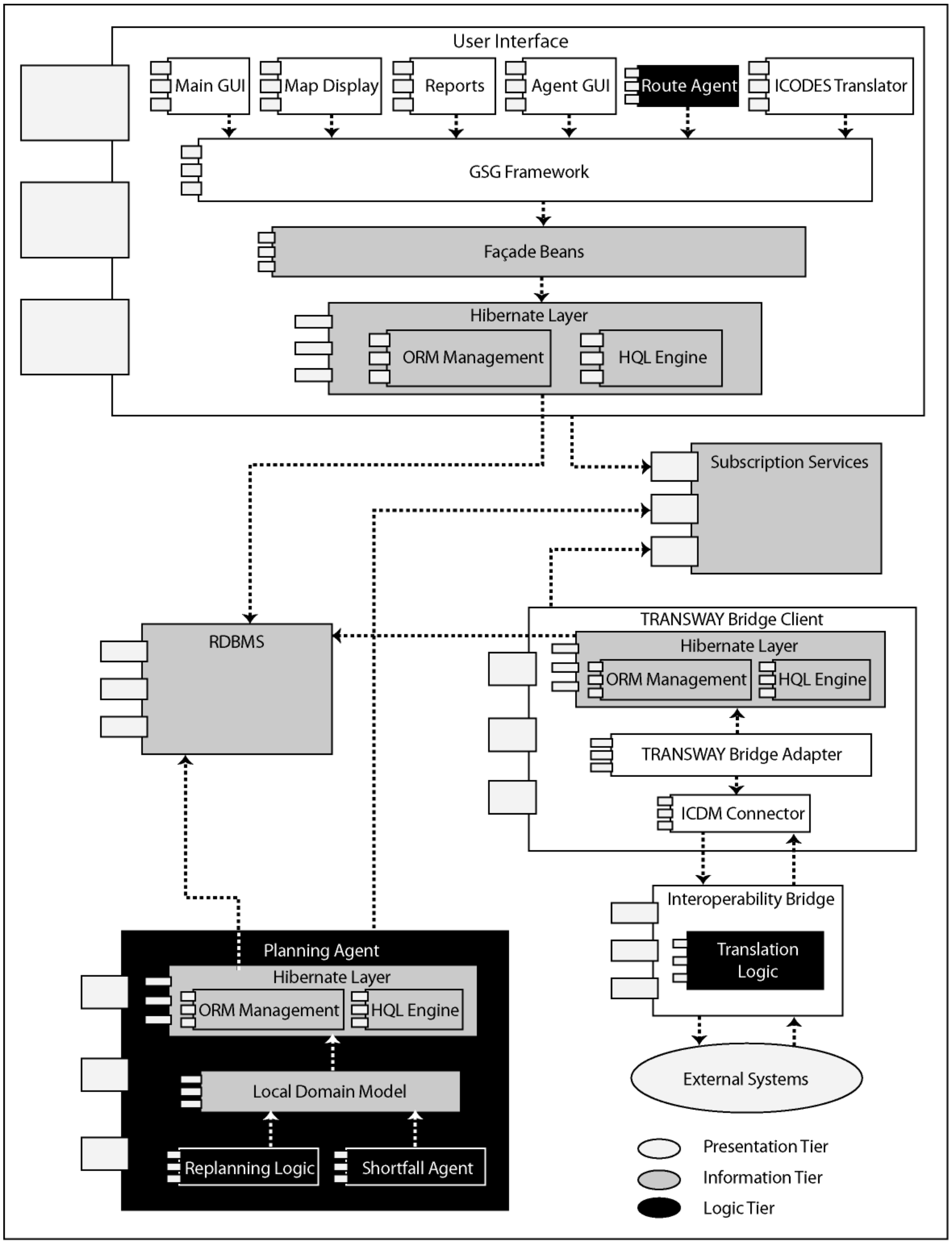


Figure 2.1: The TRANSWAY system architecture

Presentation and interaction with external systems is provided through the ICDM Interoperability Bridge supporting complex translation among potentially disparate system representations

(Leighton et al. 2004). Such translations can be specified as Extensible Style-sheet Language Transforms (XSLT) or rule-based logic. The underlying interaction metaphor supported by the Interoperability Bridge is that of *remote service calls* issues between bridge clients (i.e., interoperating systems).

The *information* tier utilizes an information-based ontology that provides relationship-rich descriptions of the concepts, notions, and entities relevant to the domains over which the system operates. These information-centric descriptions form the means by which intelligent decision-support agents analyze the evolving common operational picture. To support high degrees of extensibility, flexibility, referential integrity, and representational accuracy the TRANSWAY ontology employs numerous well-established analysis patterns such as operational-knowledge separation, contextual roles, and so on (Fowler 2003, Fowler 1997, Fowler and Scott 1997) as the basis for many of the concepts and entities it represents.

Information within the TRANSWAY system is persisted in a standard Relational Database Management System. To support the object-oriented nature inherent in the ontology structure a Hibernate object-to-relational mapping (ORM) layer is inserted within each client. It is through this object access layer that clients (e.g., GUI, agents, Interoperability Bridge) interact with the ontology. Collaboration among system entities is empowered through the use of the ICDM *Subscription Service* to register ontology-based interests and the Hibernate Query Language (HQL) facilities. Using these two mechanisms, TRANSWAY clients employ a decoupled collaboration model interacting with other parts of the system via the changes that occur in the ontology. This type of interaction model parallels the well-established *blackboard architecture* prominent in artificial intelligence-oriented systems. A further advantage of this type of decoupled collaborative architecture is that since clients need not know of each other's existence it is possible to attach and detach clients based on evolving system and operational needs.

The *logic* tier is comprised of technologies derived from both the artificial intelligence (AI) and operations research disciplines, in the form of software agents. The agents take the form of Java applications or other AI-based languages that collaborate via the information tier in accordance with a standard *blackboard* model. Agents provide the reasoning capabilities in TRANSWAY in several forms. Planning agents utilize proven planning algorithms that produce quality plans according to set criteria. Other monitoring agents utilize symbolic reasoning to recognize complex patterns representing specific situations that require the attention of the operator.

On the symbolic reasoning side, rule-based agents are employed to analyze theater and operational context providing alerts and recommendations (e.g., entire plans, or reacting to changing circumstances, or alternative actions that can be incorporated into existing plans). Another type of agent employed in the TRANSWAY system is based on the Tabu Search algorithm (Karaboga and Pham 2000, Glover and Laguna 1997). Unlike symbolic reasoning, the Tabu approach evolves toward solutions to complex problems (i.e., scheduling, etc.) by applying an extended greedy search algorithm that employs forms of adaptive memory to avoid premature isolation in local optima with respect to the effective solution space. By employing two historically disparate technologies the TRANSWAY agents take advantage of the precision and definability of symbolic reasoning and the performance of a greedy search, while minimizing each of their respective limitations.

To aid in development and management of decision-support systems such as TRANSWAY, the ICDM toolkit provides framework generation tools capable of automatically processing the UML representation of an ontology into a platform specific implementation (Leighton et al. 2004). The ability to quickly and iteratively move from model to implementation promotes a development environment where agility to changing requirements and evolving knowledge acquisition are significantly improved over more manual approaches.

The Underlying Ontology

The representation of data and its interpretation for decision-support systems must be complex by necessity due to the very nature of the decision-support process. This complexity may be defined either in the interpretation of the data or it may be placed in the data representation itself. By placing the complexity in the data representation, less work is required to be performed to interpret the data. Additionally, this complex representation may more accurately reflect the real nature of the problem to be analyzed and may in fact more directly represent the knowledge that is proposed to be captured.

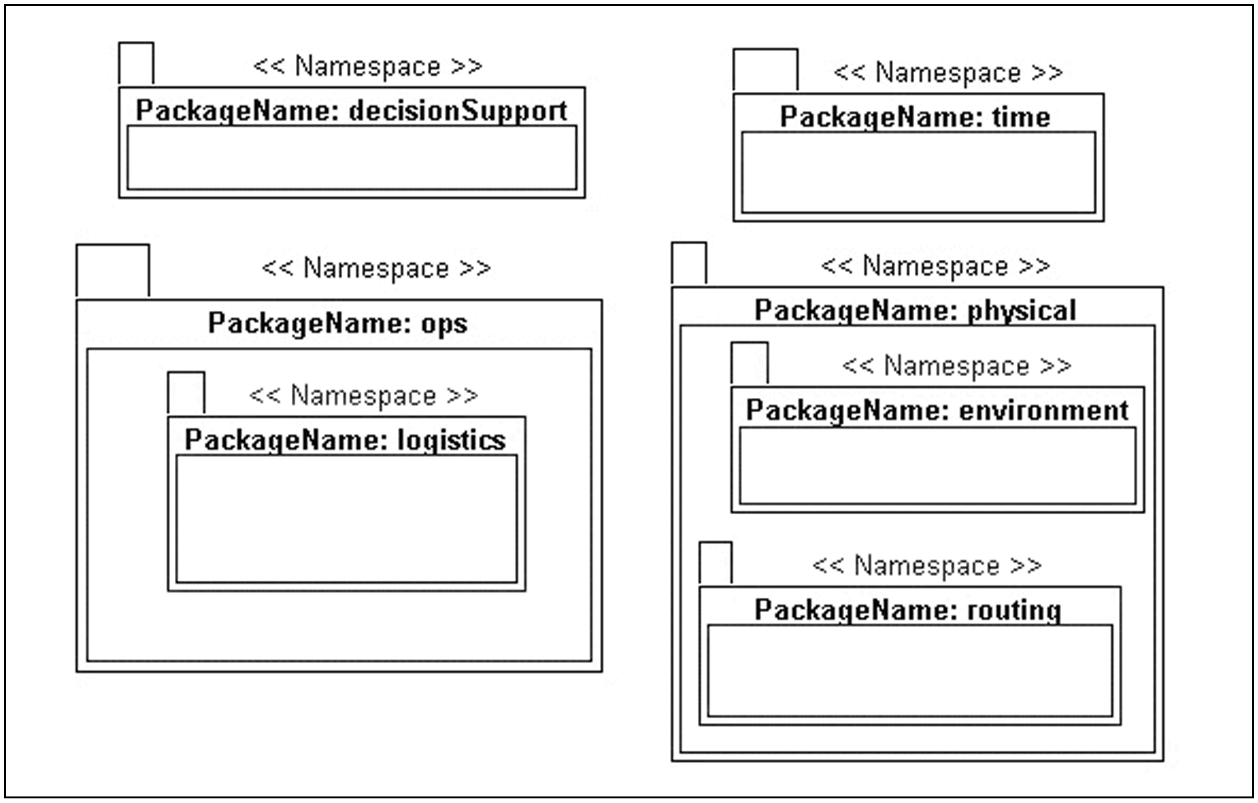


Figure 5.1: TRANSWAY ontology domains

An ontology can be characterized as an explicit specification of a conceptualization. The term is borrowed from philosophy, where an ontology is a systematic account of existence. For a software application, what "exists" is that which can be represented. When the information and knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them represents all the information and knowledge that can be known in the context of the applications that employ them. In such an ontology, definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms.

The TRANSWAY ontology is divided into logical domains that can be described using the Unified Modeling Language (UML) methodology (Figure 5.1). These domains, or namespaces, are indicated by UML package symbols and named accordingly. Within each domain exist definitions of the various concepts and entities relevant to the representation and analysis of key aspects of each domain. Classes located within package symbols are defined within that domain. These classes may relate to classes defined in other domains through either *inheritance* or *associations*. In both cases, referenced classes are identified by their symbols existing outside the primary package symbol with some type of relationship symbol connecting them to package elements. Domains themselves may be related to each other in either a sibling or parent/child relationship. Such connections are an indication of the particular scope and inter-domain visibility. Following are brief textual descriptions and UML-based illustrations describing each domain. The names of the classes currently supported by TRANSWAY and some typical class descriptions are included in the Appendix.

The Tabu Agents

The current version of TRANSWAY includes several agents built around the Tabu search algorithm (Karaboga and Pham 2000, Glover and Laguna 1997). Tabu Search is a local search method for exploring a solution space (OpenTS 2005). It is best suited for combinatorial solution spaces where a certain combination of atomic entities is considered a solution.

The TRANSWAY agents need to be highly responsive to system events, so that they can adjust their plan generation strategies dynamically as the user makes changes to the visual environment. For example, if a route becomes unavailable due to weather or an enemy threat the agents should be informed of the disabled route and respond appropriately. A common practice for supporting this level of responsiveness in a Java development environment is to use Java Beans. A Java Bean provides a strategy for event-driven programming. By encapsulating all of the properties of an object into a bean and notifying listeners when properties change it is possible to create the necessary event-driven environment.

Since the TRANSWAY system incorporates many small agents that perform specific computational tasks, threading and synchronization required particular attention. Often several of these computational tasks need to be performed in parallel or, more accurately stated, cannot be performed serially. An example of this requirement for concurrency is the need for one agent to monitor the current demand for supplies, while another agent continually calculates the all-pairs shortest path algorithm.

Separation of Trip and Plan Generation: The literature describes many different approaches to combinatorial problems of the type encountered in trip routing (Talbi 2002). Based on a review of this literature it was decided early on in the design of the TRANSWAY agents to treat trip and plan generation as separate problems. It was noted that most of the approaches cited in the literature utilize not one but several strategies for solving the combinatorial problem. While the different strategies are normally domain specific, the commonality that appears to exist among most of the approaches is to limit the search space of the problem by taking advantage of the known constraints of the system. This criterion was adopted as an important design feature of the TRANSWAY Tabu agents, to limit the number of trips produced so that the combinations of trips that make up a better (i.e., more optimal) plan can be found more quickly.

Selection of Search Methodology: After the separation of trip and plan generation the planning part becomes primarily a search problem. As new trips are generated they need to be considered as possible components of a recommended plan. However, even with the limitation of the search space through the application of constraints, the combination of generated trips into valid plans is likely to be time consuming. It was therefore decided that the TRANSWAY user should be provided with some means for controlling the number of plans generated by the agents. In the current version of TRANSWAY this is accomplished by allowing the user to set a time limit at the beginning of the plan generation process, and by allowing the user to terminate the search process at will. Several different search methods were considered, as follows:

> *Simulated Annealing:* This method is essentially a simulation of the annealing process in metals. A temperature value that simulates a cooling effect much like annealing is defined. This value eventually becomes cold enough to force the searching to find a close local optimum.
>
> *Genetic Algorithms:* This method involves breeding solutions and applying random mutations to evolve a population of 'best fit' solutions.
>
> *Constraint Logic Programming:* This method involves using a search algorithm with discrete domains to find values that satisfy the given constraints (e.g., backward chaining).
>
> *Tabu Search:* This method is based on the concept that new solutions should not revisit portions of the solution space previously considered.

The Tabu Search method was selected because it is particularly suitable for the type of vehicle routing and scheduling problem encountered by TRANSWAY (Crino 2002). However, there was still a need to translate the mathematical representation of the Tabu search algorithm into the object-oriented environment of the TRANSWAY architecture. For example, in the case of trip representation, each trip contains a reference to a conveyance object and a list of 'trip legs' representing each journey that the conveyance will embark on, together with its associated cargo.

Another theoretical notion that required translation was the concept of a *move* (Crino 2002). In the Tabu environment a move is typically defined as replacing one trip in the solution with another trip. However, a trip cannot be replaced by just any other trip. Crino (2002) uses the conveyance as a convenient identifier, so that one trip can be replaced by another trip if they share the same conveyance. This is not acceptable in the case of TRANSWAY because conveyances should be able to make more than one trip. Therefore, in TRANSWAY trips are identified by the degree to which the demand for supplies is satisfied. Accordingly, a set of trips can be replaced by another set of trips that satisfies all or a subset of the demands.

Tabu Search Strategies: In the TRANSWAY implementation the Tabu agent attempts to find the best combination of trips that together form reasonable planning recommendations. The trips in this case are the atomic entities. The Tabu agent tries to add or remove trips during each iteration of the algorithm based on several strategies. It will first attempt to add trips to the current solution. If it cannot add more trips to its current solution it will remove trips and begin again.

One fundamental aspect of a Tabu search is the use of adaptive memory. By maintaining a list of taboo choices the Tabu agent is capable of diversifying its approach through the combinatorial solution space. When Tabu examines the various choices or trips that can be added to the current plan it first checks the taboo list to see if that solution has already been examined and chooses the best non-taboo option as the new incumbent solution. This approach allows the algorithm to search through a large combination of trips, while considering solutions that hold the most promise relatively quickly.

Using the Tabu agent TRANSWAY is able to find reasonable plans in a short amount of time and more optimal plans if it is allowed to continue running. Once some ending criterion has been reached the algorithm will stop and report the best solution that has been found. In the current version of TRANSWAY reporting occurs on a continuous basis as better and better solutions are found. The user may stop the search at any time.

Principal Design Components: The implementation of the Tabu algorithm in TRANSWAY can be best described in terms of two principal design components, namely *services* and *agents*. In respect to *services*, an event manager receives events from the TRANSWAY ontology through the ICDM-based subscription service. Agents acting as listeners are able to register interest in these events, which are treated as services. The following *services* have been implemented in the current version of TRANSWAY:

Request Service: This service maintains the locations, quantities, priorities, time windows, and types of supplies requested.

Conveyance Service: This service maintains the current locations and capabilities of all of the conveyances within the AOR.

Supply Service: This service maintains the locations, quantities, and types of supplies available.

Routing Service: This service listens to changes within the graph-like structure of nodes and route segments. A shortest path matrix is maintained for each type of route traversal such as air, water, and land. Accordingly, agents are able to ask the routing service whether one or more routes exist between two nodes and, if yes: What is the shortest route? Agents may also ask the routing agent to compute shortest routes based on a maximum range between refueling stops.

Several kinds of *agents* with different functional responsibilities have been implemented in TRANSWAY to collaboratively develop strategic planning solutions, as follows:

Generic Trip Generation Agents: These agents generate a set of all possible trips that satisfy all of the business rule constraints. In this regard a generic trip is composed of a vehicle traveling to a supply depot, picking up supplies, delivering those supplies to another location, and returning to its home base. However: a conveyance cannot exceed its range without refueling; a conveyance must travel on a route of its traversal type; a conveyance should try and take the shortest path when available; and, an impediment may cause the need for alternate routes.

Convoy Building Agent: This agent is responsible for constructing convoys out of trucks. The convoy then acts as another conveyance for the other agents to work with.

Advanced Trip Generation Agents: These agents take the single trips that have been generated and determine whether combining two or more of these trips could lead to greater efficiency. For example, two trips could be combined when they use the same conveyance and their time constraints are compatible.

The conveyance scheduling and routing problem falls into a class of problems that are *NP-complete*. This means that these problems grow in complexity quite fast, and it is unreasonable to try and examine every possible solution to a sizable scenario. The Tabu algorithm addresses this problem by providing good heuristics to guide searching.

A Typical TRANSWAY Scenario

The main TRANSWAY screen (Figure 3.1) is divided into two principal areas. On the left side, moving from the top down, below the main option bar the user will find: three agent icons; objects that may be placed on top of the map (the right side of the screen); a tree-structure that provides quick and convenient access to the data that the system is currently populated with; and, at the bottom a command window for the Tabu agent. On the right side of the screen is a geo-referenced map that allows the user to pan to any part of the world and, subject to the availability of maps, zoom down to street level if desired. Objects representing nodes (e.g., SAAs, APODs, etc.), route segments, impediments, and areas of interest may be moved from the left side of the screen to the right side by simple *click to locate* actions. Alternatively, the user may specify latitude-longitude locations and the selected object will be automatically placed on the map in the correct location. These objects, whether entered by the user or pre-initialized in the system, have attributes that relate to TRANSWAY's internal ontology and provide the necessary context for automated agent actions.

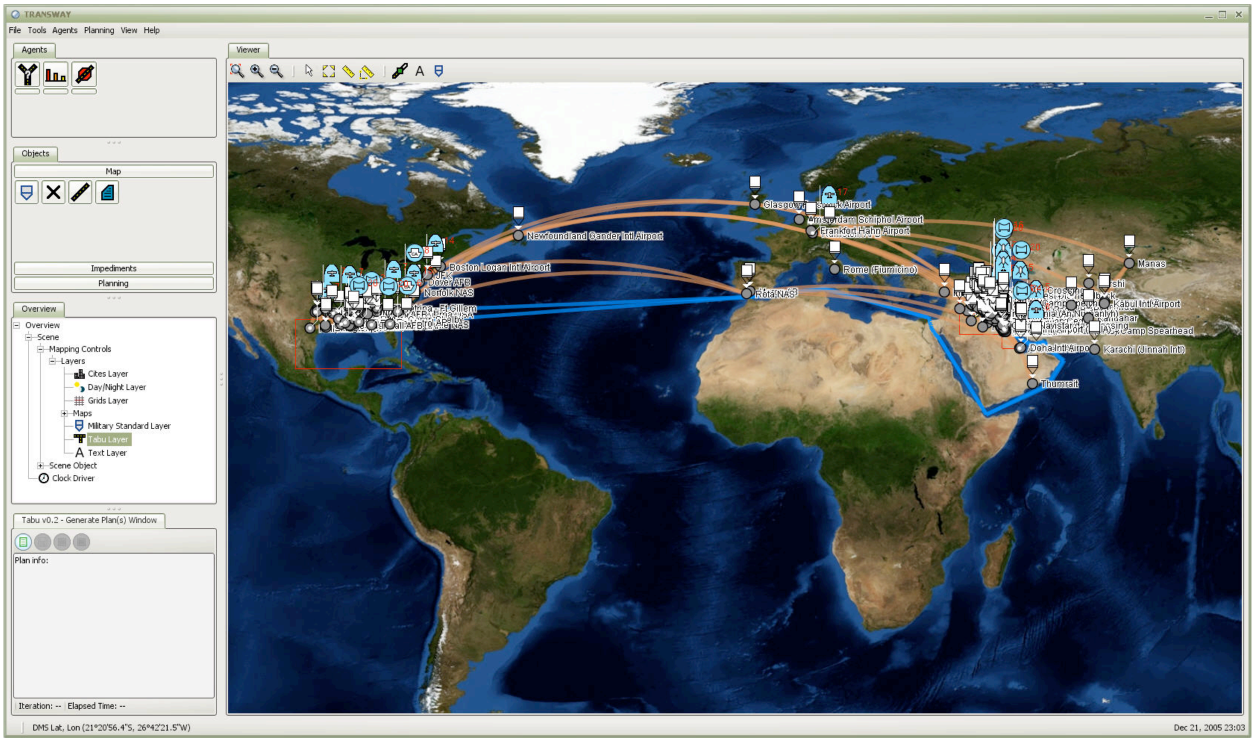
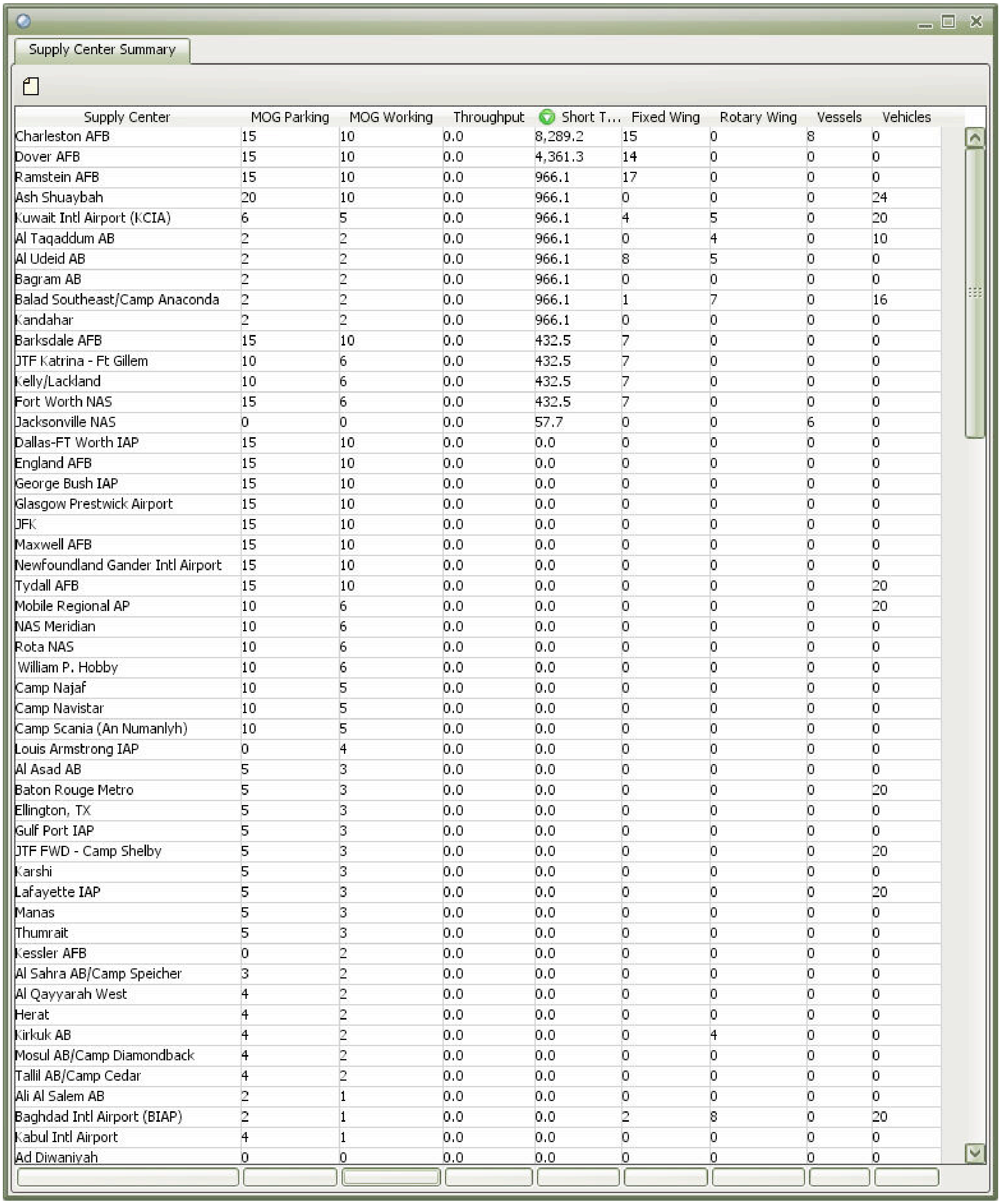


Figure 3.1: Main TRANSWAY screen

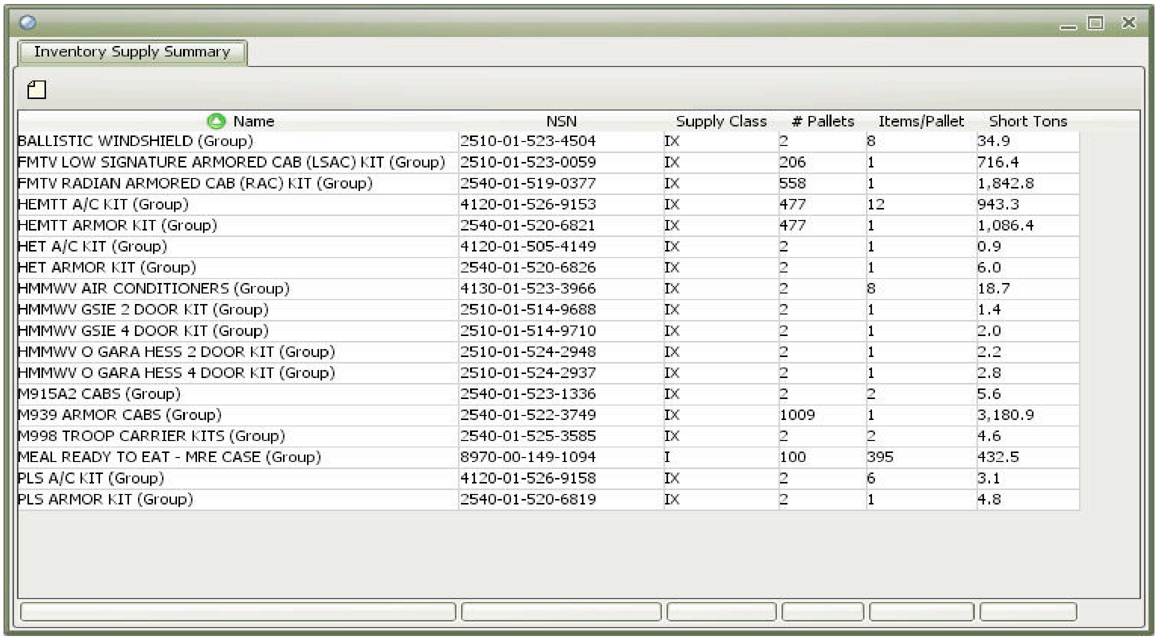
TRANSWAY is by no means limited to the current set of attributes. With the contractual goal of this first version of a prototype system to demonstrate the typical capabilities of an ontology-based multi-agent system, attributes were selected in a fairly generic fashion based on the feedback that the development team received during early demonstrations, perusal of military documents, and in-house experience with other logistic planning systems.



Supply Center	MOG Parking	MOG Working	Throughput	Short T...	Fixed Wing	Rotary Wing	Vessels	Vehicles
Charleston AFB	15	10	0.0	8,289.2	15	0	8	0
Dover AFB	15	10	0.0	4,361.3	14	0	0	0
Ramstein AFB	15	10	0.0	966.1	17	0	0	0
Ash Shuaybah	20	10	0.0	966.1	0	0	0	24
Kuwait Intl Airport (KCIA)	6	5	0.0	966.1	4	5	0	20
Al Taqaddum AB	2	2	0.0	966.1	0	4	0	10
Al Udeid AB	2	2	0.0	966.1	8	5	0	0
Bagram AB	2	2	0.0	966.1	0	0	0	0
Balad Southeast/Camp Anaconda	2	2	0.0	966.1	1	7	0	16
Kandahar	2	2	0.0	966.1	0	0	0	0
Barksdale AFB	15	10	0.0	432.5	7	0	0	0
JTF Katrina - Ft Gillem	10	6	0.0	432.5	7	0	0	0
Kelly/Lackland	10	6	0.0	432.5	7	0	0	0
Fort Worth NAS	15	6	0.0	432.5	7	0	0	0
Jacksonville NAS	0	0	0.0	57.7	0	0	6	0
Dallas-FT Worth IAP	15	10	0.0	0.0	0	0	0	0
England AFB	15	10	0.0	0.0	0	0	0	0
George Bush IAP	15	10	0.0	0.0	0	0	0	0
Glasgow Prestwick Airport	15	10	0.0	0.0	0	0	0	0
JFK	15	10	0.0	0.0	0	0	0	0
Maxwell AFB	15	10	0.0	0.0	0	0	0	0
Newfoundland Gander Intl Airport	15	10	0.0	0.0	0	0	0	0
Tydall AFB	15	10	0.0	0.0	0	0	0	20
Mobile Regional AP	10	6	0.0	0.0	0	0	0	20
NAS Meridian	10	6	0.0	0.0	0	0	0	0
Rota NAS	10	6	0.0	0.0	0	0	0	0
William P. Hobby	10	6	0.0	0.0	0	0	0	0
Camp Najaf	10	5	0.0	0.0	0	0	0	0
Camp Navistar	10	5	0.0	0.0	0	0	0	0
Camp Scania (An Numanlyh)	10	5	0.0	0.0	0	0	0	0
Louis Armstrong IAP	0	4	0.0	0.0	0	0	0	0
Al Asad AB	5	3	0.0	0.0	0	0	0	0
Baton Rouge Metro	5	3	0.0	0.0	0	0	0	20
Ellington, TX	5	3	0.0	0.0	0	0	0	0
Gulf Port IAP	5	3	0.0	0.0	0	0	0	0
JTF FWD - Camp Shelby	5	3	0.0	0.0	0	0	0	20
Karshi	5	3	0.0	0.0	0	0	0	0
Lafayette IAP	5	3	0.0	0.0	0	0	0	20
Manas	5	3	0.0	0.0	0	0	0	0
Thumrait	5	3	0.0	0.0	0	0	0	0
Kessler AFB	0	2	0.0	0.0	0	0	0	0
Al Sahra AB/Camp Speicher	3	2	0.0	0.0	0	0	0	0
Al Qayyarah West	4	2	0.0	0.0	0	0	0	0
Herat	4	2	0.0	0.0	0	0	0	0
Kirkuk AB	4	2	0.0	0.0	0	4	0	0
Mosul AB/Camp Diamondback	4	2	0.0	0.0	0	0	0	0
Tallil AB/Camp Cedar	4	2	0.0	0.0	0	0	0	0
Ali Al Salem AB	2	1	0.0	0.0	0	0	0	0
Baghdad Intl Airport (BIAP)	2	1	0.0	0.0	2	8	0	20
Kabul Intl Airport	4	1	0.0	0.0	0	0	0	0
Ad Diwaniyah	0	0	0.0	0.0	0	0	0	0

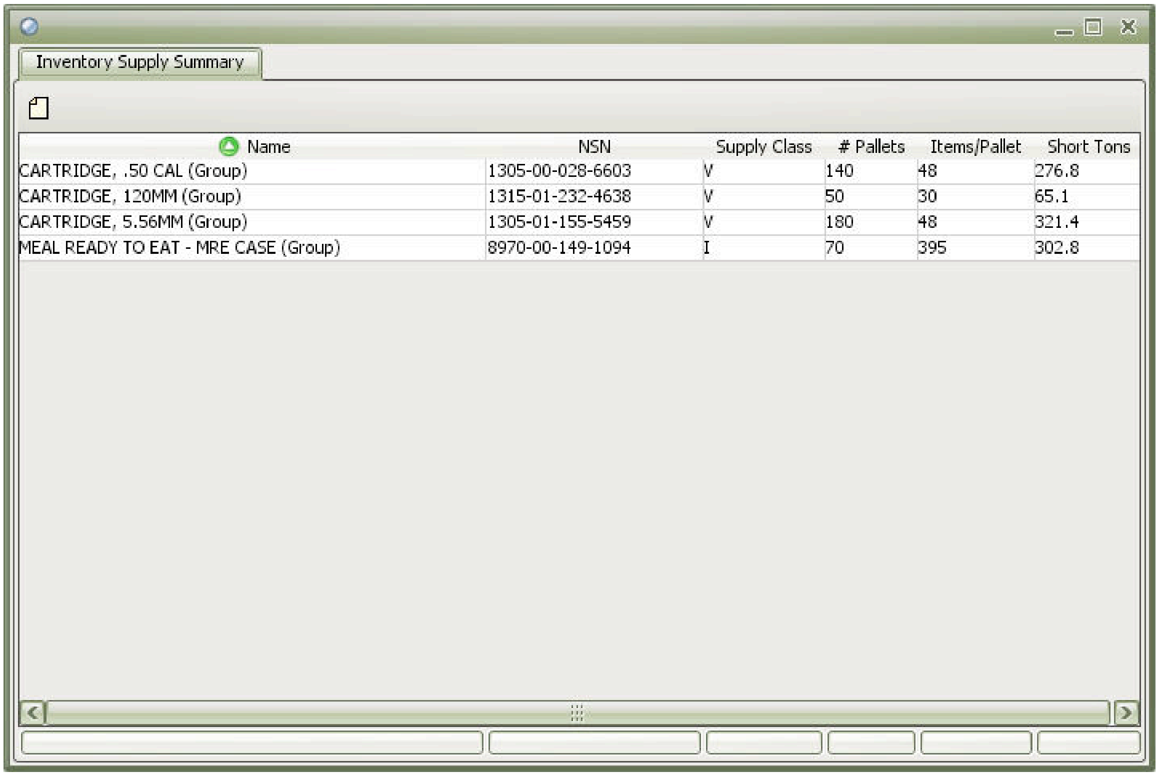
Figure 3.2: Summary of supplies and available conveyances at supply centers

The report shown in Figure 3.2 provides a summary of supplies (short tons) and available conveyances (i.e., fixed wing aircraft, helicopters, ships, and trucks (in convoys)) at most supply centers currently initialized in the system for this particular demonstration scenario. Details of supplies at Charleston and Al Udeid are shown in Figures 3.3 and 3.4 (in terms of supply Class, number of pallets, number of items per pallet, and short tons), respectively.



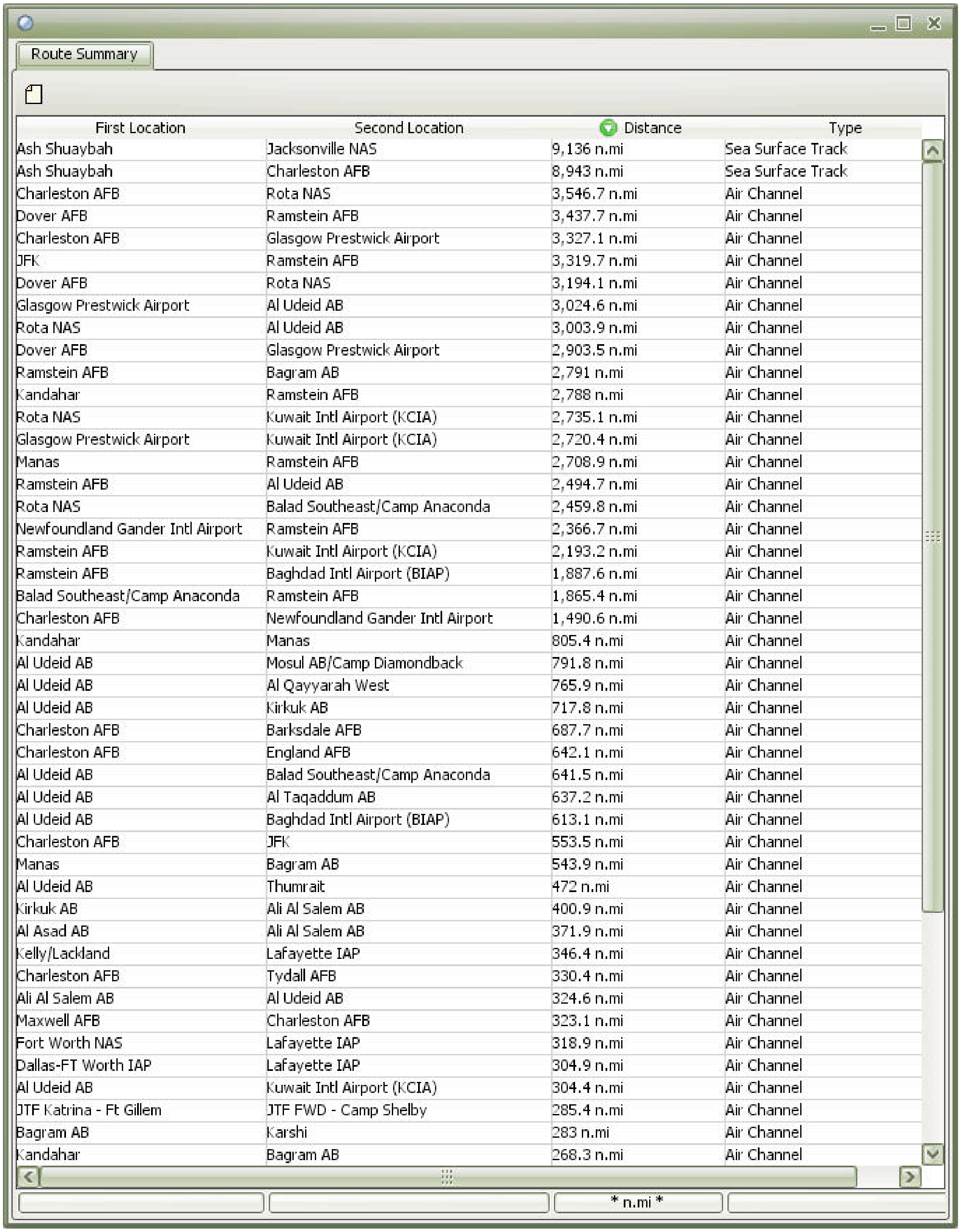
Name	NSN	Supply Class	# Pallets	Items/Pallet	Short Tons
BALLISTIC WINDSHIELD (Group)	2510-01-523-4504	IX	2	8	34.9
FMTV LOW SIGNATURE ARMORED CAB (LSAC) KIT (Group)	2510-01-523-0059	IX	206	1	716.4
FMTV RADIAN ARMORED CAB (RAC) KIT (Group)	2540-01-519-0377	IX	558	1	1,842.8
HEMTT A/C KIT (Group)	4120-01-526-9153	IX	477	12	943.3
HEMTT ARMOR KIT (Group)	2540-01-520-6821	IX	477	1	1,086.4
HET A/C KIT (Group)	4120-01-505-4149	IX	2	1	0.9
HET ARMOR KIT (Group)	2540-01-520-6826	IX	2	1	6.0
HMMWV AIR CONDITIONERS (Group)	4130-01-523-3966	IX	2	8	18.7
HMMWV GSIE 2 DOOR KIT (Group)	2510-01-514-9688	IX	2	1	1.4
HMMWV GSIE 4 DOOR KIT (Group)	2510-01-514-9710	IX	2	1	2.0
HMMWV O GARA HESS 2 DOOR KIT (Group)	2510-01-524-2948	IX	2	1	2.2
HMMWV O GARA HESS 4 DOOR KIT (Group)	2510-01-524-2937	IX	2	1	2.8
M915A2 CABS (Group)	2540-01-523-1336	IX	2	2	5.6
M939 ARMOR CABS (Group)	2540-01-522-3749	IX	1009	1	3,180.9
M998 TROOP CARRIER KITS (Group)	2540-01-525-3585	IX	2	2	4.6
MEAL READY TO EAT - MRE CASE (Group)	8970-00-149-1094	I	100	395	432.5
PLS A/C KIT (Group)	4120-01-526-9158	IX	2	6	3.1
PLS ARMOR KIT (Group)	2540-01-520-6819	IX	2	1	4.8

Figure 3.3: Details of supplies at Charleston



Name	NSN	Supply Class	# Pallets	Items/Pallet	Short Tons
CARTRIDGE, .50 CAL (Group)	1305-00-028-6603	V	140	48	276.8
CARTRIDGE, 120MM (Group)	1315-01-232-4638	V	50	30	65.1
CARTRIDGE, 5.56MM (Group)	1305-01-155-5459	V	180	48	321.4
MEAL READY TO EAT - MRE CASE (Group)	8970-00-149-1094	I	70	395	302.8

Figure 3.4: Details of supplies at Al Udeid

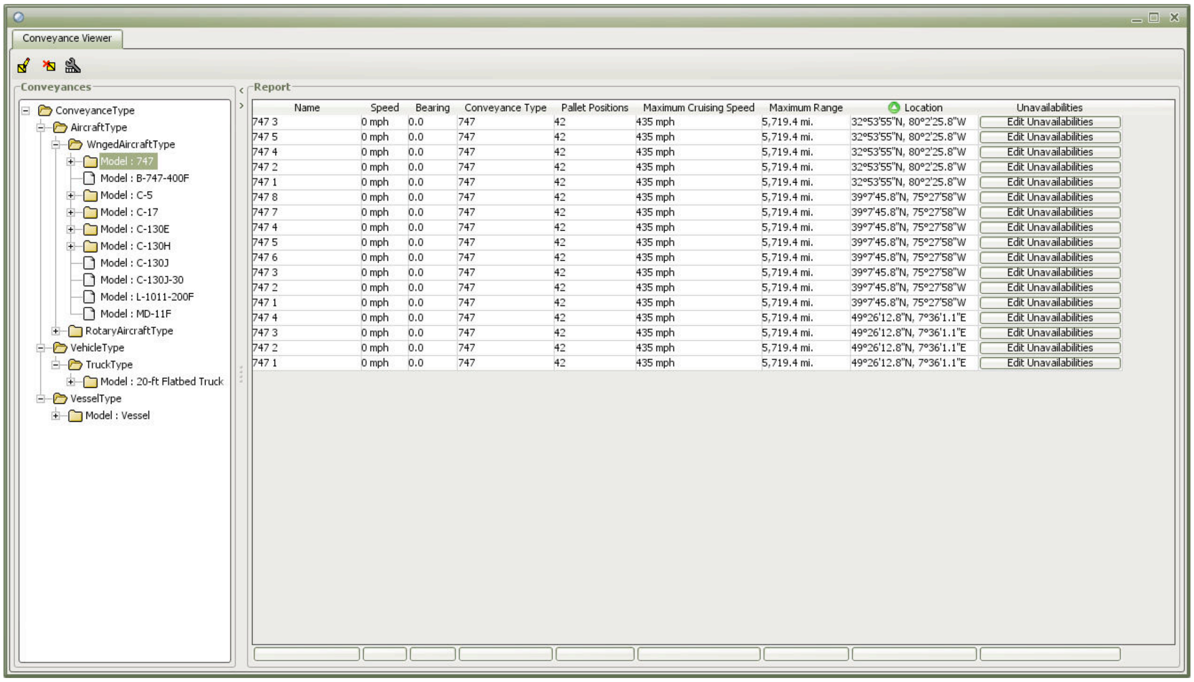


First Location	Second Location	Distance	Type
Ash Shuaybah	Jacksonville NAS	9,136 n.mi	Sea Surface Track
Ash Shuaybah	Charleston AFB	8,943 n.mi	Sea Surface Track
Charleston AFB	Rota NAS	3,546.7 n.mi	Air Channel
Dover AFB	Ramstein AFB	3,437.7 n.mi	Air Channel
Charleston AFB	Glasgow Prestwick Airport	3,327.1 n.mi	Air Channel
JFK	Ramstein AFB	3,319.7 n.mi	Air Channel
Dover AFB	Rota NAS	3,194.1 n.mi	Air Channel
Glasgow Prestwick Airport	Al Udeid AB	3,024.6 n.mi	Air Channel
Rota NAS	Al Udeid AB	3,003.9 n.mi	Air Channel
Dover AFB	Glasgow Prestwick Airport	2,903.5 n.mi	Air Channel
Ramstein AFB	Bagram AB	2,791 n.mi	Air Channel
Kandahar	Ramstein AFB	2,788 n.mi	Air Channel
Rota NAS	Kuwait Intl Airport (KCIA)	2,735.1 n.mi	Air Channel
Glasgow Prestwick Airport	Kuwait Intl Airport (KCIA)	2,720.4 n.mi	Air Channel
Manas	Ramstein AFB	2,708.9 n.mi	Air Channel
Ramstein AFB	Al Udeid AB	2,494.7 n.mi	Air Channel
Rota NAS	Balad Southeast/Camp Anaconda	2,459.8 n.mi	Air Channel
Newfoundland Gander Intl Airport	Ramstein AFB	2,366.7 n.mi	Air Channel
Ramstein AFB	Kuwait Intl Airport (KCIA)	2,193.2 n.mi	Air Channel
Ramstein AFB	Baghdad Intl Airport (BIAP)	1,887.6 n.mi	Air Channel
Balad Southeast/Camp Anaconda	Ramstein AFB	1,865.4 n.mi	Air Channel
Charleston AFB	Newfoundland Gander Intl Airport	1,490.6 n.mi	Air Channel
Kandahar	Manas	805.4 n.mi	Air Channel
Al Udeid AB	Mosul AB/Camp Diamondback	791.8 n.mi	Air Channel
Al Udeid AB	Al Qayyarah West	765.9 n.mi	Air Channel
Al Udeid AB	Kirkuk AB	717.8 n.mi	Air Channel
Charleston AFB	Barksdale AFB	687.7 n.mi	Air Channel
Charleston AFB	England AFB	642.1 n.mi	Air Channel
Al Udeid AB	Balad Southeast/Camp Anaconda	641.5 n.mi	Air Channel
Al Udeid AB	Al Taqaddum AB	637.2 n.mi	Air Channel
Al Udeid AB	Baghdad Intl Airport (BIAP)	613.1 n.mi	Air Channel
Charleston AFB	JFK	553.5 n.mi	Air Channel
Manas	Bagram AB	543.9 n.mi	Air Channel
Al Udeid AB	Thumrait	472 n.mi	Air Channel
Kirkuk AB	Ali Al Salem AB	400.9 n.mi	Air Channel
Al Asad AB	Ali Al Salem AB	371.9 n.mi	Air Channel
Kelly/Lackland	Lafayette IAP	346.4 n.mi	Air Channel
Charleston AFB	Tydall AFB	330.4 n.mi	Air Channel
Ali Al Salem AB	Al Udeid AB	324.6 n.mi	Air Channel
Maxwell AFB	Charleston AFB	323.1 n.mi	Air Channel
Fort Worth NAS	Lafayette IAP	318.9 n.mi	Air Channel
Dallas-FT Worth IAP	Lafayette IAP	304.9 n.mi	Air Channel
Al Udeid AB	Kuwait Intl Airport (KCIA)	304.4 n.mi	Air Channel
JTF Katrina - Ft Gillem	JTF FWD - Camp Shelby	285.4 n.mi	Air Channel
Bagram AB	Karshi	283 n.mi	Air Channel
Kandahar	Bagram AB	268.3 n.mi	Air Channel

Figure 3.5: Summary report of air channels and sea routes

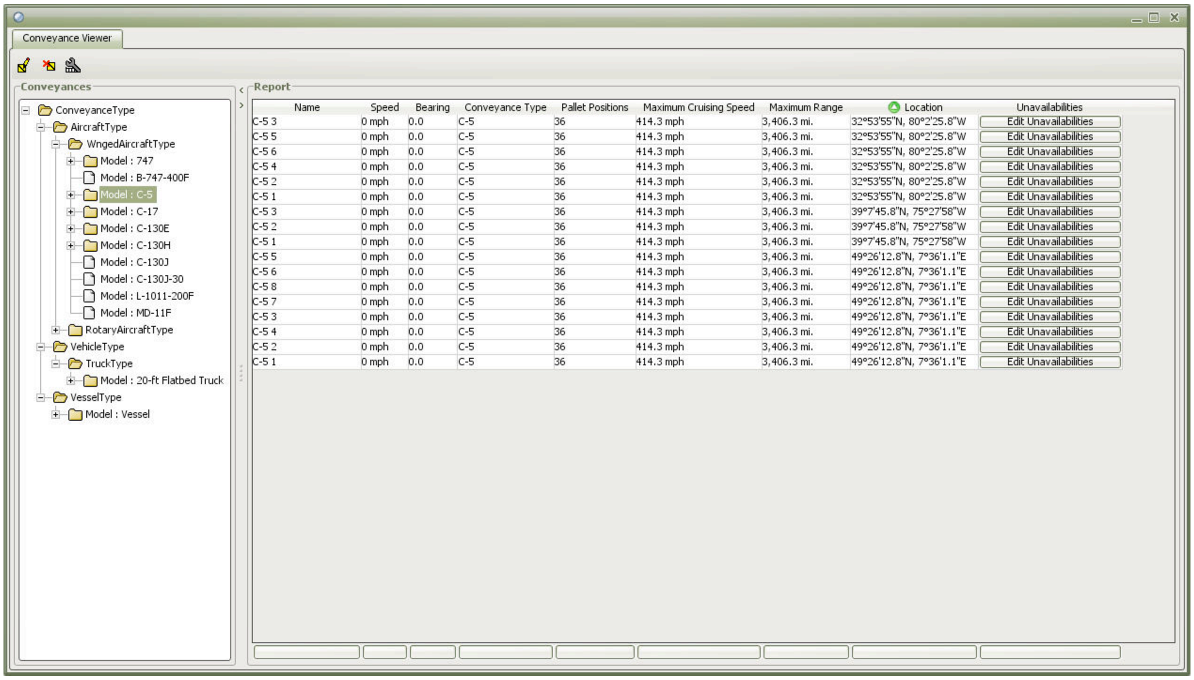
Figure 3.5 provides information about the air channels and sea routes that the system has been initialized with for this particular demonstration scenario. In each case the two end-points and the distance in nautical miles is indicated.

Detailed information about the current compliment of conveyances can be obtained by selecting the appropriate report. Typical examples for various fixed wing aircraft, trucks and ships are shown in Figures 3.6 to 3.11, below. The reason that the *speed* and *bearing* attributes in each table are zero is because the conveyances are not currently in-transit.



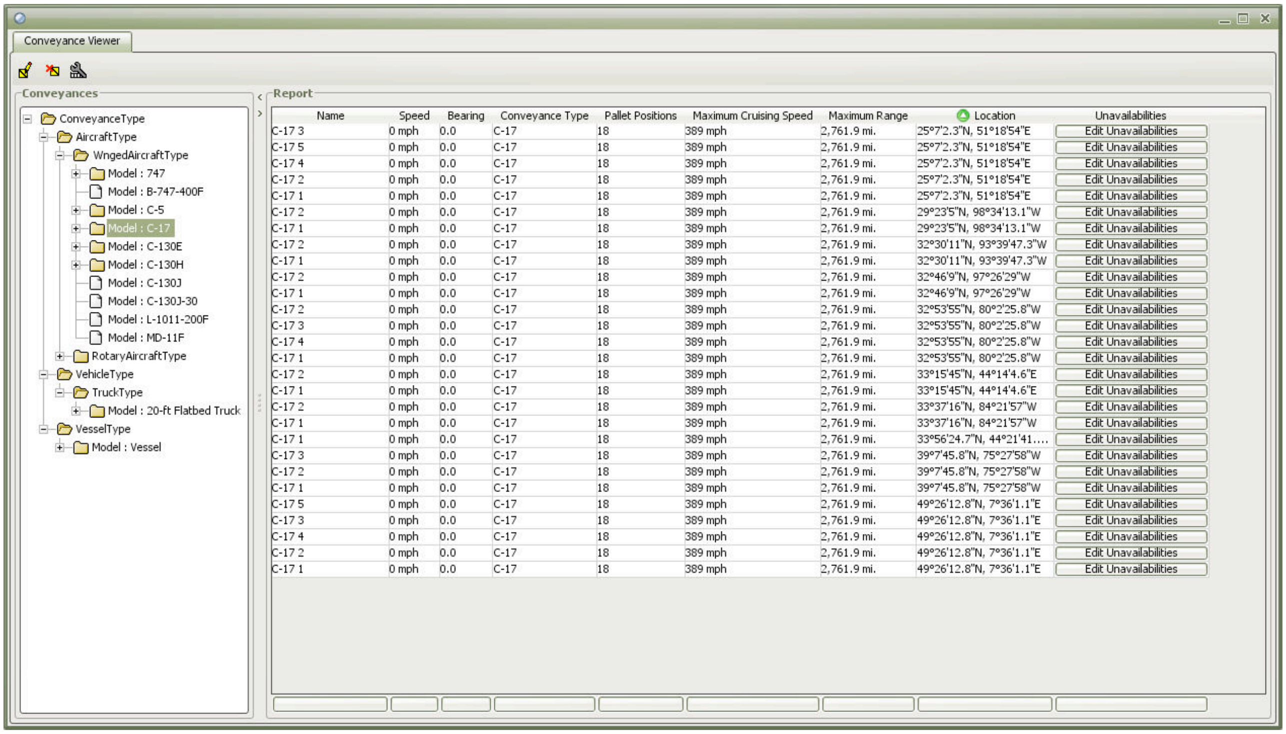
Name	Speed	Bearing	Conveyance Type	Pallet Positions	Maximum Cruising Speed	Maximum Range	Location	Unavailabilities
747 3	0 mph	0.0	747	42	435 mph	5,719.4 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
747 5	0 mph	0.0	747	42	435 mph	5,719.4 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
747 4	0 mph	0.0	747	42	435 mph	5,719.4 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
747 2	0 mph	0.0	747	42	435 mph	5,719.4 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
747 1	0 mph	0.0	747	42	435 mph	5,719.4 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
747 8	0 mph	0.0	747	42	435 mph	5,719.4 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
747 7	0 mph	0.0	747	42	435 mph	5,719.4 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
747 4	0 mph	0.0	747	42	435 mph	5,719.4 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
747 5	0 mph	0.0	747	42	435 mph	5,719.4 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
747 6	0 mph	0.0	747	42	435 mph	5,719.4 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
747 3	0 mph	0.0	747	42	435 mph	5,719.4 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
747 2	0 mph	0.0	747	42	435 mph	5,719.4 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
747 1	0 mph	0.0	747	42	435 mph	5,719.4 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
747 4	0 mph	0.0	747	42	435 mph	5,719.4 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
747 3	0 mph	0.0	747	42	435 mph	5,719.4 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
747 2	0 mph	0.0	747	42	435 mph	5,719.4 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
747 1	0 mph	0.0	747	42	435 mph	5,719.4 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities

Figure 3.6: Boeing 747 aircraft attributes



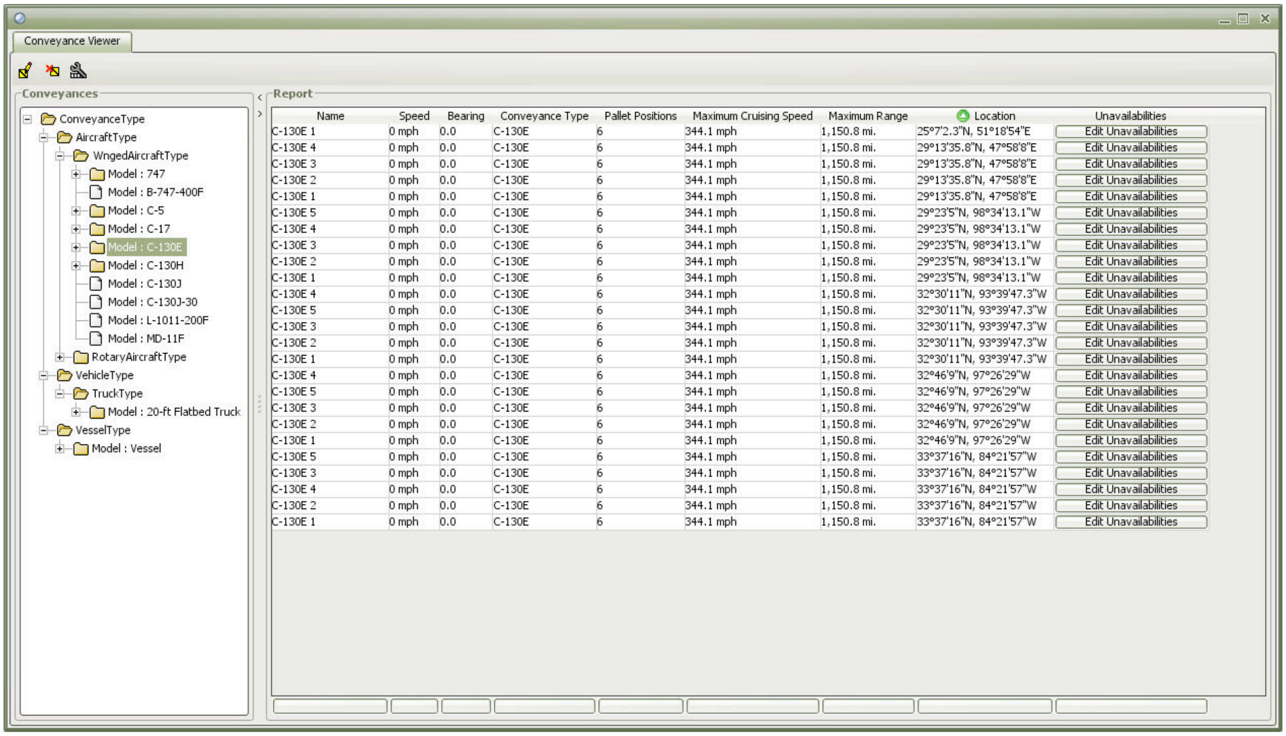
Name	Speed	Bearing	Conveyance Type	Pallet Positions	Maximum Cruising Speed	Maximum Range	Location	Unavailabilities
C-5 3	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
C-5 5	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
C-5 6	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
C-5 4	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
C-5 2	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
C-5 1	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
C-5 3	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
C-5 2	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
C-5 1	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
C-5 5	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
C-5 6	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
C-5 8	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
C-5 7	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
C-5 3	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
C-5 4	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
C-5 2	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
C-5 1	0 mph	0.0	C-5	36	414.3 mph	3,406.3 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities

Figure 3.7: C5 aircraft attributes



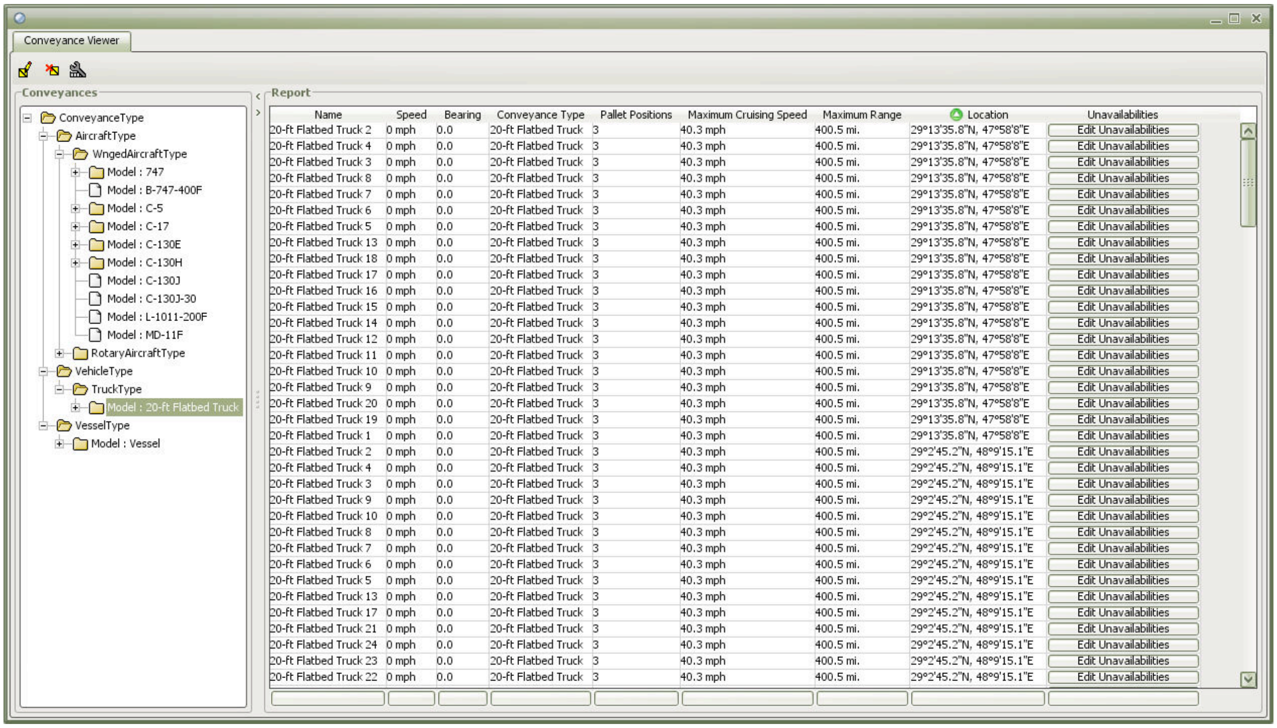
Name	Speed	Bearing	Conveyance Type	Pallet Positions	Maximum Cruising Speed	Maximum Range	Location	Unavailabilities
C-17 3	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	25°7'2.3"N, 51°18'54"E	Edit Unavailabilities
C-17 5	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	25°7'2.3"N, 51°18'54"E	Edit Unavailabilities
C-17 4	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	25°7'2.3"N, 51°18'54"E	Edit Unavailabilities
C-17 2	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	25°7'2.3"N, 51°18'54"E	Edit Unavailabilities
C-17 1	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	25°7'2.3"N, 51°18'54"E	Edit Unavailabilities
C-17 2	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	29°23'5"N, 98°34'13.1"W	Edit Unavailabilities
C-17 1	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	29°23'5"N, 98°34'13.1"W	Edit Unavailabilities
C-17 2	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	32°30'11"N, 93°39'47.3"W	Edit Unavailabilities
C-17 1	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	32°30'11"N, 93°39'47.3"W	Edit Unavailabilities
C-17 2	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	32°46'9"N, 97°26'29"W	Edit Unavailabilities
C-17 1	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	32°46'9"N, 97°26'29"W	Edit Unavailabilities
C-17 2	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
C-17 3	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
C-17 4	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
C-17 1	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
C-17 2	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	33°15'45"N, 44°14'4.6"E	Edit Unavailabilities
C-17 1	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	33°15'45"N, 44°14'4.6"E	Edit Unavailabilities
C-17 2	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	33°37'16"N, 84°21'57"W	Edit Unavailabilities
C-17 1	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	33°37'16"N, 84°21'57"W	Edit Unavailabilities
C-17 1	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	33°56'24.7"N, 44°21'41....	Edit Unavailabilities
C-17 3	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
C-17 2	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
C-17 1	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	39°7'45.8"N, 75°27'58"W	Edit Unavailabilities
C-17 5	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
C-17 3	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
C-17 4	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
C-17 2	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities
C-17 1	0 mph	0.0	C-17	18	389 mph	2,761.9 mi.	49°26'12.8"N, 7°36'1.1"E	Edit Unavailabilities

Figure 3.8: C17 aircraft attributes



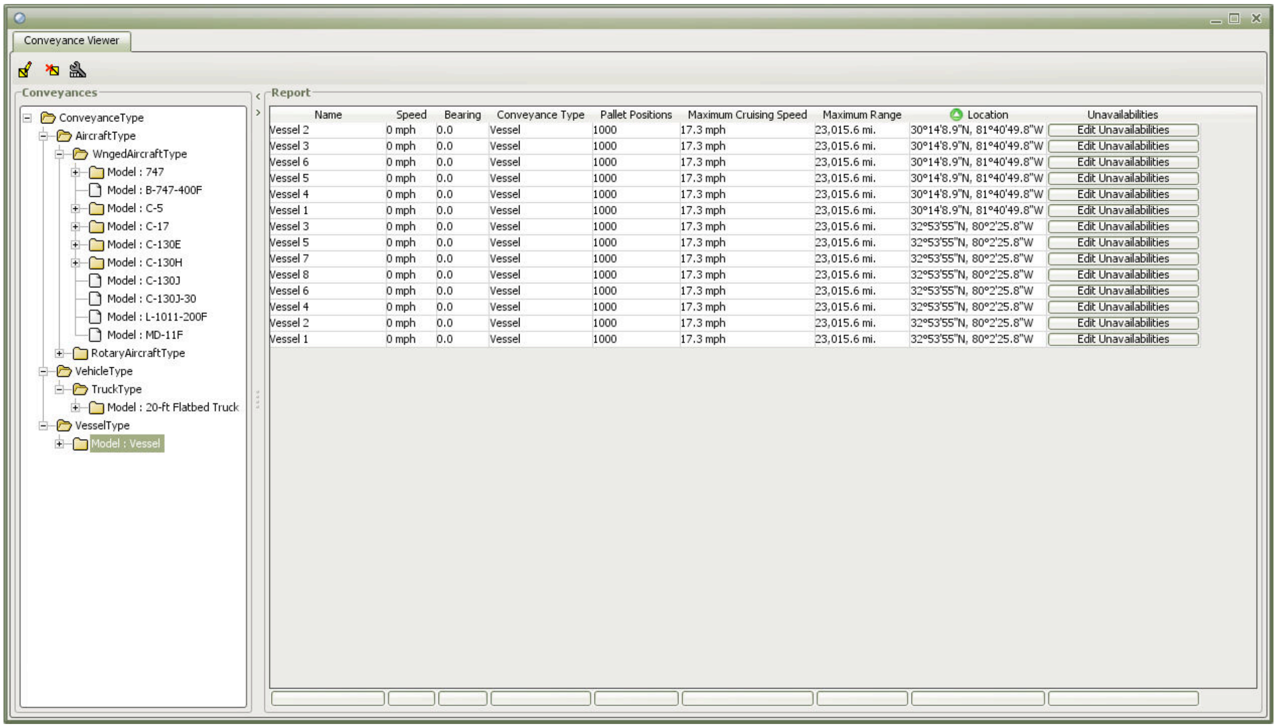
Name	Speed	Bearing	Conveyance Type	Pallet Positions	Maximum Cruising Speed	Maximum Range	Location	Unavailabilities
C-130E 1	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	25°7'2.3"N, 51°18'54"E	Edit Unavailabilities
C-130E 4	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
C-130E 3	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
C-130E 2	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
C-130E 1	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
C-130E 5	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	29°23'5"N, 98°34'13.1"W	Edit Unavailabilities
C-130E 4	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	29°23'5"N, 98°34'13.1"W	Edit Unavailabilities
C-130E 3	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	29°23'5"N, 98°34'13.1"W	Edit Unavailabilities
C-130E 2	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	29°23'5"N, 98°34'13.1"W	Edit Unavailabilities
C-130E 1	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	29°23'5"N, 98°34'13.1"W	Edit Unavailabilities
C-130E 4	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	32°30'11"N, 93°39'47.3"W	Edit Unavailabilities
C-130E 5	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	32°30'11"N, 93°39'47.3"W	Edit Unavailabilities
C-130E 3	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	32°30'11"N, 93°39'47.3"W	Edit Unavailabilities
C-130E 2	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	32°30'11"N, 93°39'47.3"W	Edit Unavailabilities
C-130E 1	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	32°30'11"N, 93°39'47.3"W	Edit Unavailabilities
C-130E 4	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	32°46'9"N, 97°26'29"W	Edit Unavailabilities
C-130E 5	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	32°46'9"N, 97°26'29"W	Edit Unavailabilities
C-130E 3	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	32°46'9"N, 97°26'29"W	Edit Unavailabilities
C-130E 2	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	32°46'9"N, 97°26'29"W	Edit Unavailabilities
C-130E 1	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	32°46'9"N, 97°26'29"W	Edit Unavailabilities
C-130E 5	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	33°37'16"N, 84°21'57"W	Edit Unavailabilities
C-130E 3	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	33°37'16"N, 84°21'57"W	Edit Unavailabilities
C-130E 4	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	33°37'16"N, 84°21'57"W	Edit Unavailabilities
C-130E 2	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	33°37'16"N, 84°21'57"W	Edit Unavailabilities
C-130E 1	0 mph	0.0	C-130E	6	344.1 mph	1,150.8 mi.	33°37'16"N, 84°21'57"W	Edit Unavailabilities

Figure 3.9: C130 aircraft attributes



Name	Speed	Bearing	Conveyance Type	Pallet Positions	Maximum Cruising Speed	Maximum Range	Location	Unavailabilities
20-ft Flatbed Truck 2	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 4	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 3	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 8	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 7	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 6	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 5	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 13	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 18	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 17	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 16	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 15	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 14	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 12	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 11	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 10	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 9	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 20	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 19	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 1	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°13'35.8"N, 47°58'8"E	Edit Unavailabilities
20-ft Flatbed Truck 2	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 4	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 3	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 9	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 10	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 8	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 7	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 6	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 5	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 13	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 17	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 21	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 24	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 23	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities
20-ft Flatbed Truck 22	0 mph	0.0	20-ft Flatbed Truck	3	40.3 mph	400.5 mi.	29°2'45.2"N, 48°9'15.1"E	Edit Unavailabilities

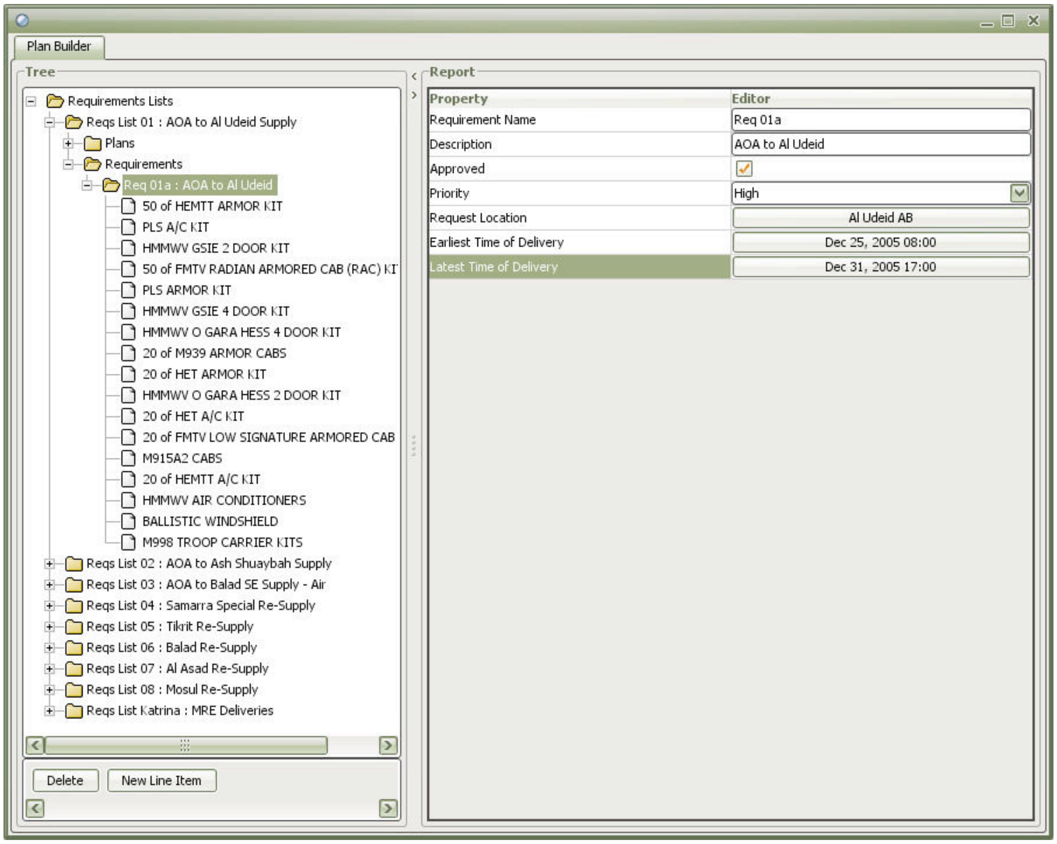
Figure 3.10: Truck convoy attributes



Name	Speed	Bearing	Conveyance Type	Pallet Positions	Maximum Cruising Speed	Maximum Range	Location	Unavailabilities
Vessel 2	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	30°14'8.9"N, 81°40'49.8"W	Edit Unavailabilities
Vessel 3	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	30°14'8.9"N, 81°40'49.8"W	Edit Unavailabilities
Vessel 6	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	30°14'8.9"N, 81°40'49.8"W	Edit Unavailabilities
Vessel 5	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	30°14'8.9"N, 81°40'49.8"W	Edit Unavailabilities
Vessel 4	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	30°14'8.9"N, 81°40'49.8"W	Edit Unavailabilities
Vessel 1	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	30°14'8.9"N, 81°40'49.8"W	Edit Unavailabilities
Vessel 3	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
Vessel 5	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
Vessel 7	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
Vessel 8	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
Vessel 6	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
Vessel 4	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
Vessel 2	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities
Vessel 1	0 mph	0.0	Vessel	1000	17.3 mph	23,015.6 mi.	32°53'55"N, 80°2'25.8"W	Edit Unavailabilities

Figure 3.11: Typical ship attributes

A typical request for *add on armor* is shown in Figure 3.12. It requires deliver to Al Udeid, with a *high* priority and an earliest and latest time for delivery window of 25 to 31 December 2005.



- Requirements Lists
  - Reqs List 01 : AOA to Al Udeid Supply
    - Plans
    - Requirements
      - Req 01a : AOA to Al Udeid
        - 50 of HEMTT ARMOR KIT
        - PLS A/C KIT
        - HMMWV GSIE 2 DOOR KIT
        - 50 of FMTV RADIAN ARMORED CAB (RAC) KIT
        - PLS ARMOR KIT
        - HMMWV GSIE 4 DOOR KIT
        - HMMWV O GARA HESS 4 DOOR KIT
        - 20 of M939 ARMOR CABS
        - 20 of HET ARMOR KIT
        - HMMWV O GARA HESS 2 DOOR KIT
        - 20 of HET A/C KIT
        - 20 of FMTV LOW SIGNATURE ARMORED CAB
        - M915A2 CABS
        - 20 of HEMTT A/C KIT
        - HMMWV AIR CONDITIONERS
        - BALLISTIC WINDSHIELD
        - M998 TROOP CARRIER KITS
  - Reqs List 02 : AOA to Ash Shuaybah Supply
  - Reqs List 03 : AOA to Balad SE Supply - Air
  - Reqs List 04 : Samarra Special Re-Supply
  - Reqs List 05 : Tikrit Re-Supply
  - Reqs List 06 : Balad Re-Supply
  - Reqs List 07 : Al Asad Re-Supply
  - Reqs List 08 : Mosul Re-Supply
  - Reqs List Katrina : MRE Deliveries

Property	Editor
Requirement Name	Req 01a
Description	AOA to Al Udeid
Approved	☑
Priority	High
Request Location	Al Udeid AB
Earliest Time of Delivery	Dec 25, 2005 08:00
Latest Time of Delivery	Dec 31, 2005 17:00

Figure 3.12: Add-on-Armor (AOR) request for delivery to Al Udeid

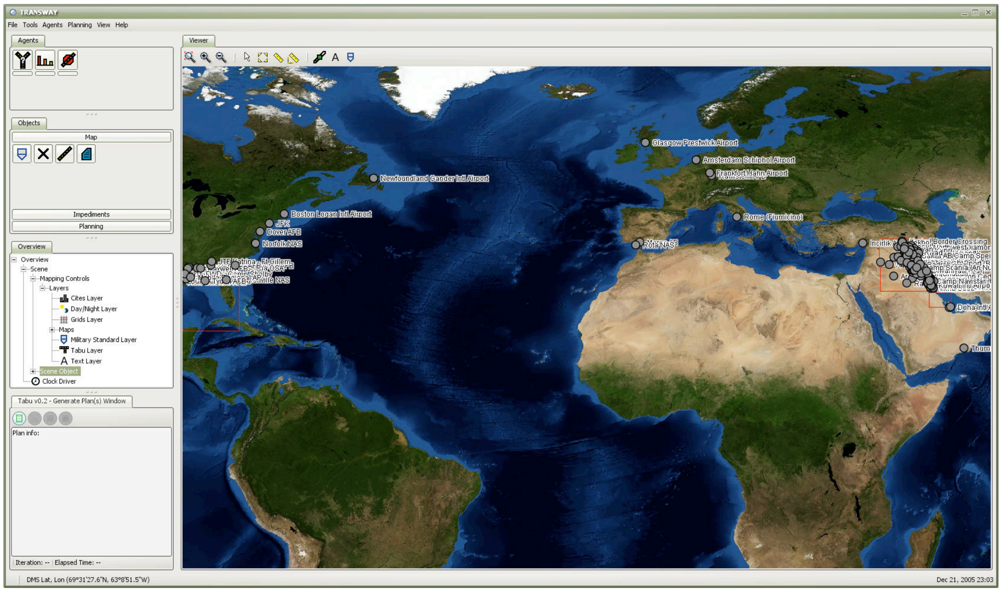
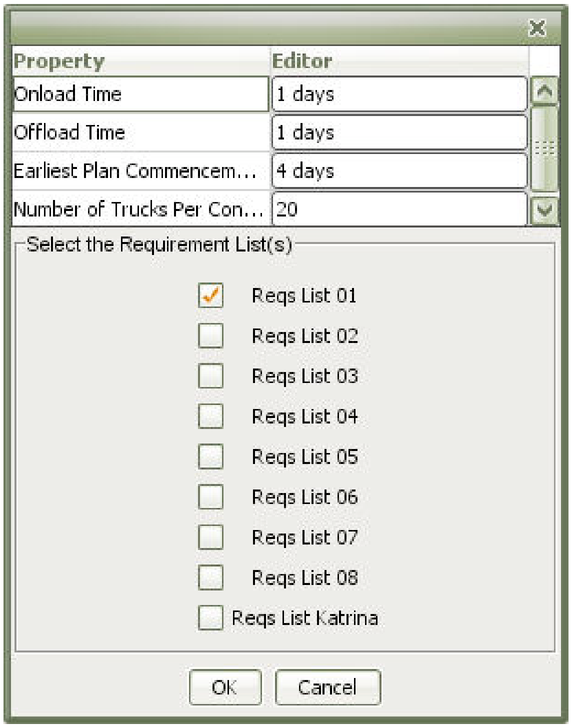


Figure 3.13: User zooms in on map to reduce clutter



Property	Editor
Onload Time	1 days
Offload Time	1 days
Earliest Plan Commencem...	4 days
Number of Trucks Per Con...	20

Select the Requirement List(s)

☑ Reqs List 01
☐ Reqs List 02
☐ Reqs List 03
☐ Reqs List 04
☐ Reqs List 05
☐ Reqs List 06
☐ Reqs List 07
☐ Reqs List 08
☐ Reqs List Katrina

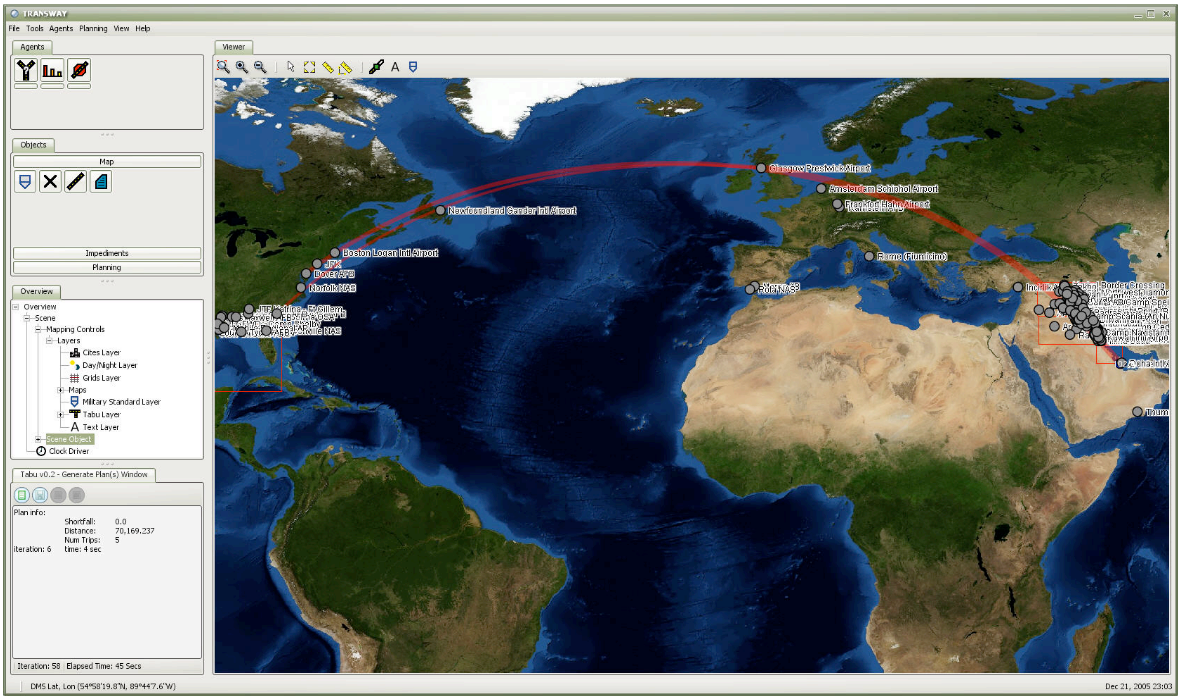
Figure 3.14: Tabu agent interface

Searching Options

☑ Run until stopped.

Allowed Planning Time(sec) 60 seconds

Figure 3.15: Control of search duration



Tabu v0.2 - Generate Plan(s) Window

Plan info:
Shortfall: 0.0
Distance: 70,169.237
Num Trips: 5
iteration: 6 time: 4 sec

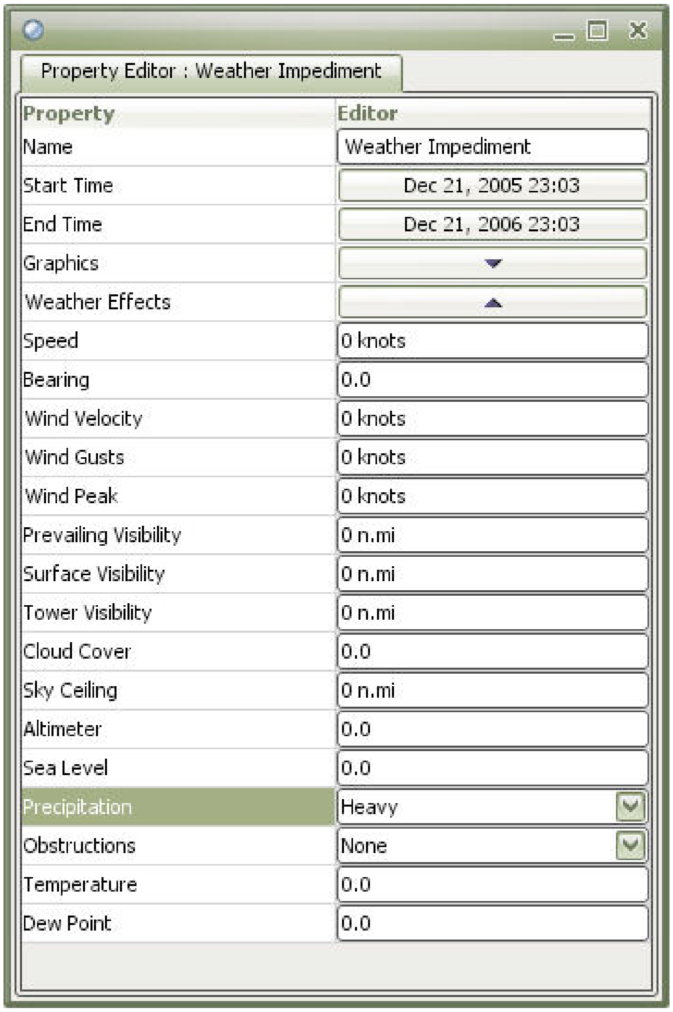
Iteration: 58 | Elapsed Time: 45 Secs

DMS Lat, Lon (54°58'19.8"N, 89°44'7.6"W)

Dec 21, 2005 23:03

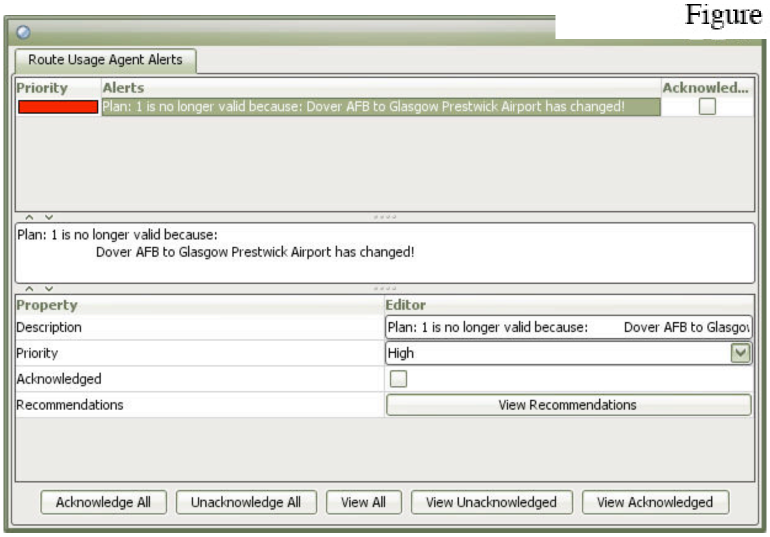
Figure 3.16: Completed first plan showing routes

To fulfill the request for the shipment of *add-on-armor* to Al Udeid (Figure 3.12) the user activates the Tabu agent and selects the appropriate *requirement* from the displayed Requirement Lists (Figure 3.14). In this case the Al Udeid *requirement* is Requirement List 1. Since the Tabu agent has the ability to continue its search for an optimum delivery plan even after it has found a way of satisfying the *requirement*, the user has the option of either setting a maximum time for the planning activity (Figure 3.15) or allowing the agent to continue until all alternatives have been explored. Of course it is not expected that the user would ever want to wait for that length of time and therefore the option for the user to simply stop the agent is available. In future versions of TRANSWAY, particularly if the Tabu agent were to be implemented in an opportunistic mode (i.e., in a manner that would activate the planning process without user involvement as soon as the conditions on which an existing plan were originally based have changed), it would be a relatively simple matter to restrict the extensiveness of the search for an optimum plan. For example, the search could be automatically aborted if after either a specified period of time or a given number of generated plans no better plan has been found.



Property	Editor
Name	Weather Impediment
Start Time	Dec 21, 2005 23:03
End Time	Dec 21, 2006 23:03
Graphics	▼
Weather Effects	▲
Speed	0 knots
Bearing	0.0
Wind Velocity	0 knots
Wind Gusts	0 knots
Wind Peak	0 knots
Prevailing Visibility	0 n.mi
Surface Visibility	0 n.mi
Tower Visibility	0 n.mi
Cloud Cover	0.0
Sky Ceiling	0 n.mi
Altimeter	0.0
Sea Level	0.0
Precipitation	Heavy
Obstructions	None
Temperature	0.0
Dew Point	0.0

Figure 3.17: Weather impediment



Priority	Alerts	Acknowled...
🟥	Plan: 1 is no longer valid because: Dover AFB to Glasgow Prestwick Airport has changed!	☐

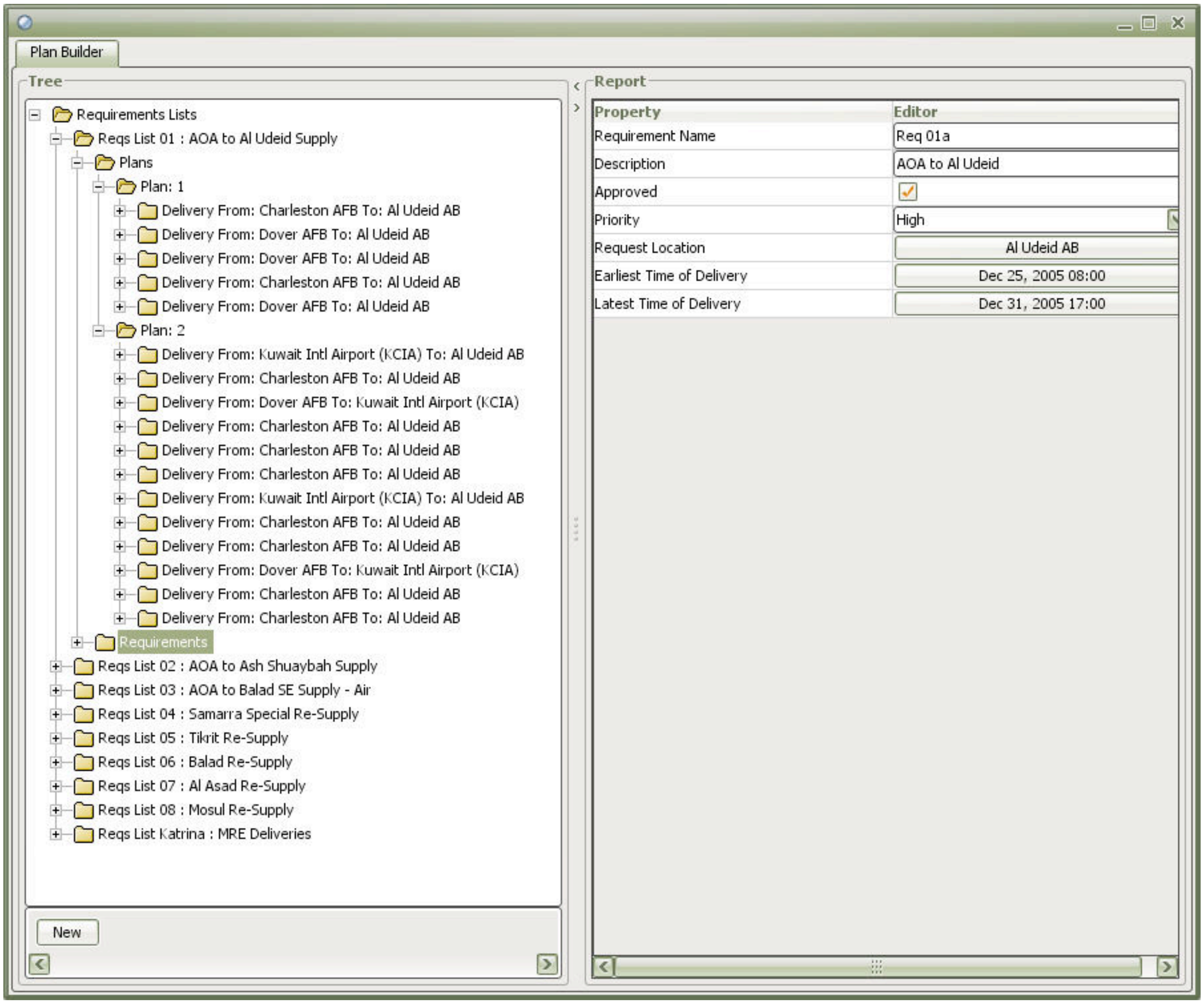
Plan: 1 is no longer valid because:
Dover AFB to Glasgow Prestwick Airport has changed!

Property	Editor
Description	Plan: 1 is no longer valid because: Dover AFB to Glasgo
Priority	High
Acknowledged	☐
Recommendations	View Recommendations

Figure 3.18: Impediment agent alert

For the completed plan the route is shown in Figure 3.16 by means of a red line. Next the user enters an impediment in the form of an adverse weather report that essentially eliminates

Glasgow as a refueling stop (Figure 3.17). Immediately, the Impediment agent alerts the user and suggests that re-planning is in order (Figure 3.18). Again, also in the case of impediments, this first version of TRANSWAY provides only one type of generic impediment (i.e., a weather condition), with the objective of demonstrating the kinds of causes that would require re-planning that could be easily implemented in subsequent versions of the system, based on user preferences and priorities.

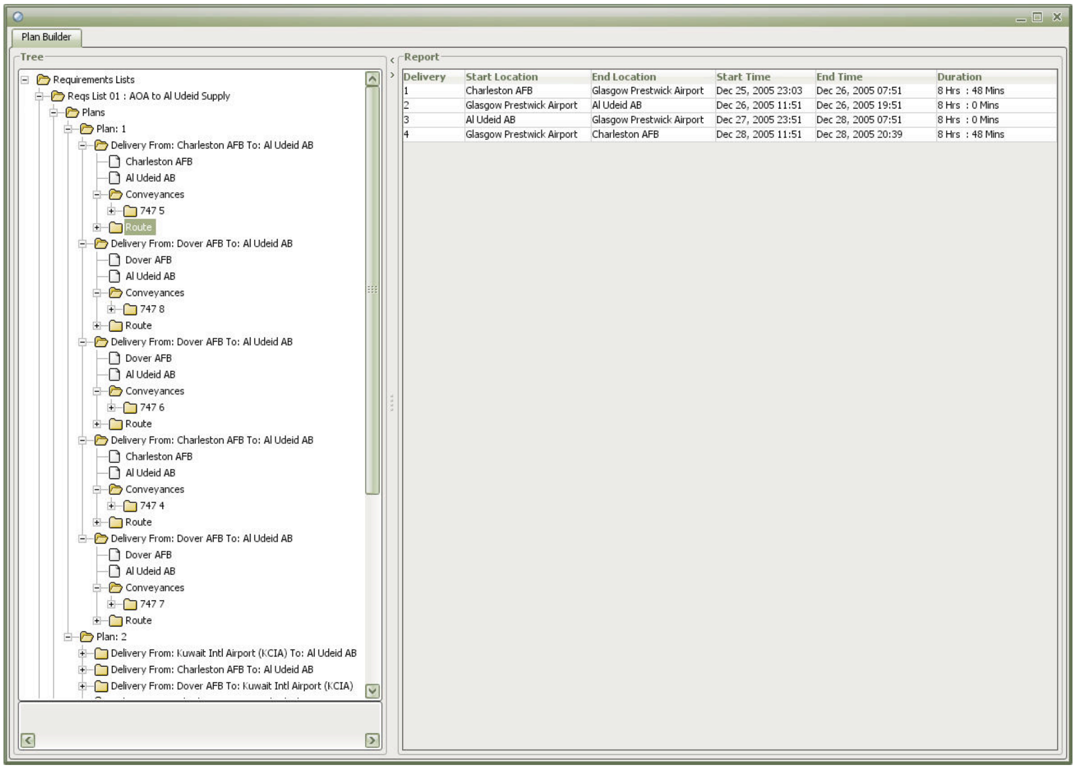


- Requirements Lists
  - Reqs List 01 : AOA to Al Udeid Supply
    - Plans
      - Plan: 1
        - Delivery From: Charleston AFB To: Al Udeid AB
        - Delivery From: Dover AFB To: Al Udeid AB
        - Delivery From: Dover AFB To: Al Udeid AB
        - Delivery From: Charleston AFB To: Al Udeid AB
        - Delivery From: Dover AFB To: Al Udeid AB
      - Plan: 2
        - Delivery From: Kuwait Intl Airport (KCIA) To: Al Udeid AB
        - Delivery From: Charleston AFB To: Al Udeid AB
        - Delivery From: Dover AFB To: Kuwait Intl Airport (KCIA)
        - Delivery From: Charleston AFB To: Al Udeid AB
        - Delivery From: Charleston AFB To: Al Udeid AB
        - Delivery From: Charleston AFB To: Al Udeid AB
        - Delivery From: Kuwait Intl Airport (KCIA) To: Al Udeid AB
        - Delivery From: Charleston AFB To: Al Udeid AB
        - Delivery From: Charleston AFB To: Al Udeid AB
        - Delivery From: Dover AFB To: Kuwait Intl Airport (KCIA)
        - Delivery From: Charleston AFB To: Al Udeid AB
        - Delivery From: Charleston AFB To: Al Udeid AB
    - Requirements
  - Reqs List 02 : AOA to Ash Shuaybah Supply
  - Reqs List 03 : AOA to Balad SE Supply - Air
  - Reqs List 04 : Samarra Special Re-Supply
  - Reqs List 05 : Tikrit Re-Supply
  - Reqs List 06 : Balad Re-Supply
  - Reqs List 07 : Al Asad Re-Supply
  - Reqs List 08 : Mosul Re-Supply
  - Reqs List Katrina : MRE Deliveries

Property	Editor
Requirement Name	Req 01a
Description	AOA to Al Udeid
Approved	☑
Priority	High
Request Location	Al Udeid AB
Earliest Time of Delivery	Dec 25, 2005 08:00
Latest Time of Delivery	Dec 31, 2005 17:00

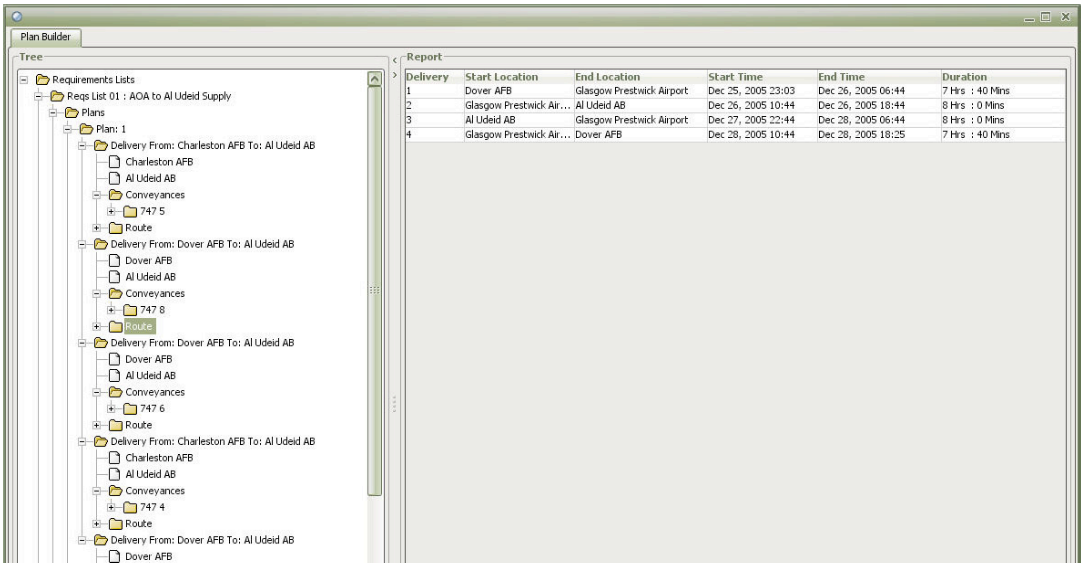
Figure 3.19: Summary of deliveries for the first and second plans

To initiate a re-planning action the user proceeds in the same manner as described previously for the generation of the first plan (Figures 3.14 to 3.16). The user will notice that during the generation of each plan the routes that are being explored by the Tabu agent are dynamically indicated on the map display. Temporarily displayed green lines indicate drop-off points that are being considered. Red lines indicate actual delivery routes with the thickness of the red line providing a proportional indication of the volume of supplies being transported along that particular route. Summary lists of the deliveries involved in both plans are shown in Figure 3.19. Even thought this first test-bed version of TRANSWAY is purposely limited in scope it does allow the user to explore the details of each delivery plan (i.e., start and end locations, conveyances and routes used, start and end times, and duration of each trip), as shown in Figures 3.20 to 3.23.



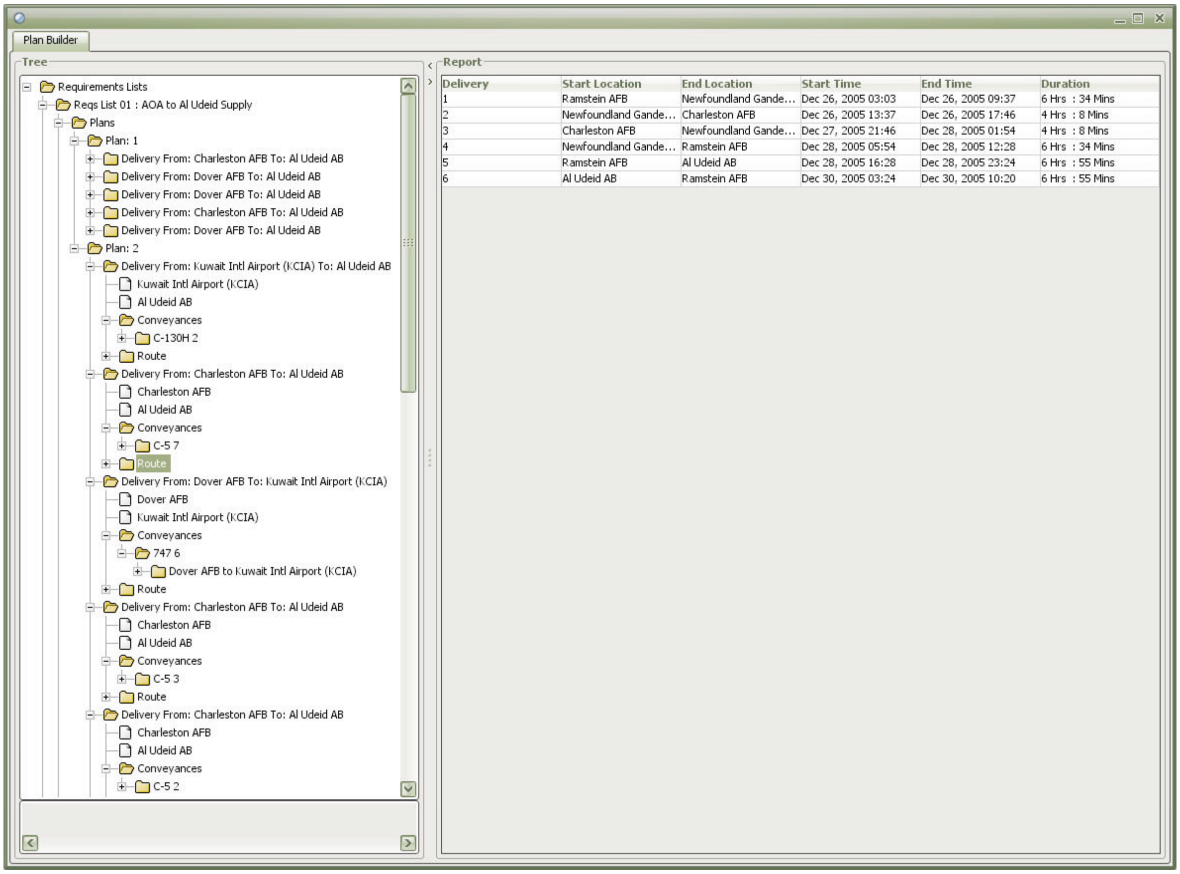
Delivery	Start Location	End Location	Start Time	End Time	Duration
1	Charleston AFB	Glasgow Prestwick Airport	Dec 25, 2005 23:03	Dec 26, 2005 07:51	8 Hrs : 48 Mins
2	Glasgow Prestwick Airport	Al Udeid AB	Dec 26, 2005 11:51	Dec 26, 2005 19:51	8 Hrs : 0 Mins
3	Al Udeid AB	Glasgow Prestwick Airport	Dec 27, 2005 23:51	Dec 28, 2005 07:51	8 Hrs : 0 Mins
4	Glasgow Prestwick Airport	Charleston AFB	Dec 28, 2005 11:51	Dec 28, 2005 20:39	8 Hrs : 48 Mins

Figure 3.20: Typical drill-down details of the first plan



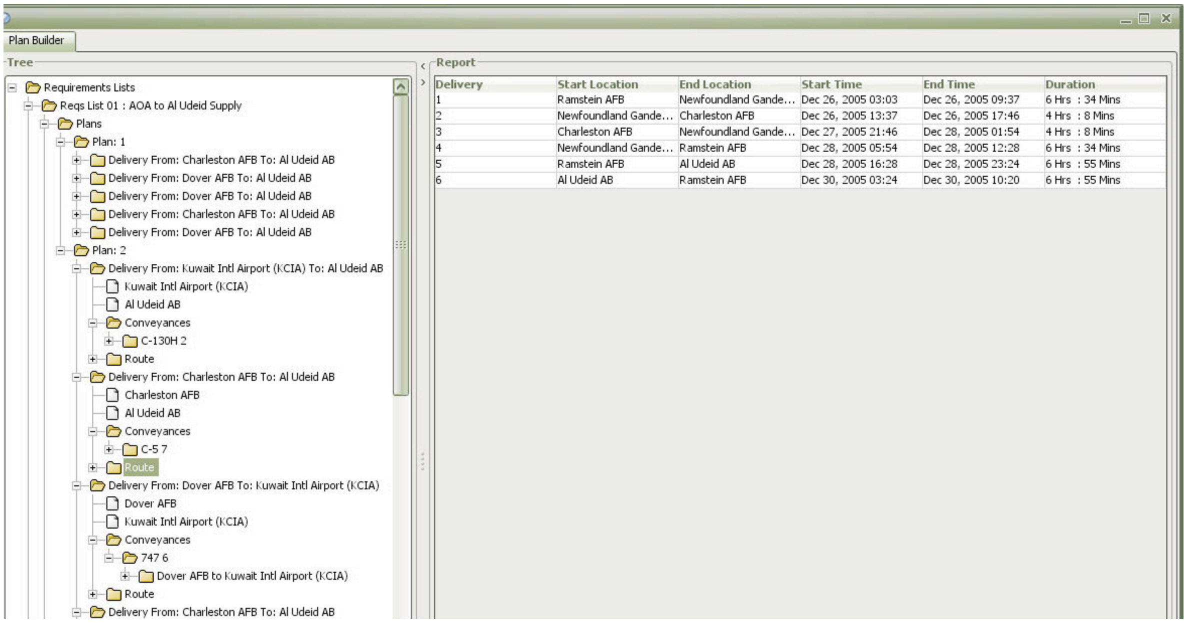
Delivery	Start Location	End Location	Start Time	End Time	Duration
1	Dover AFB	Glasgow Prestwick Airport	Dec 25, 2005 23:03	Dec 26, 2005 06:44	7 Hrs : 40 Mins
2	Glasgow Prestwick Air…	Al Udeid AB	Dec 26, 2005 10:44	Dec 26, 2005 18:44	8 Hrs : 0 Mins
3	Al Udeid AB	Glasgow Prestwick Airport	Dec 27, 2005 22:44	Dec 28, 2005 06:44	8 Hrs : 0 Mins
4	Glasgow Prestwick Air…	Dover AFB	Dec 28, 2005 10:44	Dec 28, 2005 18:25	7 Hrs : 40 Mins

Figure 3.21: Typical drill-down details of the first plan



Delivery	Start Location	End Location	Start Time	End Time	Duration
1	Ramstein AFB	Newfoundland Gande…	Dec 26, 2005 03:03	Dec 26, 2005 09:37	6 Hrs : 34 Mins
2	Newfoundland Gande…	Charleston AFB	Dec 26, 2005 13:37	Dec 26, 2005 17:46	4 Hrs : 8 Mins
3	Charleston AFB	Newfoundland Gande…	Dec 27, 2005 21:46	Dec 28, 2005 01:54	4 Hrs : 8 Mins
4	Newfoundland Gande…	Ramstein AFB	Dec 28, 2005 05:54	Dec 28, 2005 12:28	6 Hrs : 34 Mins
5	Ramstein AFB	Al Udeid AB	Dec 28, 2005 16:28	Dec 28, 2005 23:24	6 Hrs : 55 Mins
6	Al Udeid AB	Ramstein AFB	Dec 30, 2005 03:24	Dec 30, 2005 10:20	6 Hrs : 55 Mins

Figure 3.22: Typical drill-down details of the second plan



Delivery	Start Location	End Location	Start Time	End Time	Duration
1	Ramstein AFB	Newfoundland Gande…	Dec 26, 2005 03:03	Dec 26, 2005 09:37	6 Hrs : 34 Mins
2	Newfoundland Gande…	Charleston AFB	Dec 26, 2005 13:37	Dec 26, 2005 17:46	4 Hrs : 8 Mins
3	Charleston AFB	Newfoundland Gande…	Dec 27, 2005 21:46	Dec 28, 2005 01:54	4 Hrs : 8 Mins
4	Newfoundland Gande…	Ramstein AFB	Dec 28, 2005 05:54	Dec 28, 2005 12:28	6 Hrs : 34 Mins
5	Ramstein AFB	Al Udeid AB	Dec 28, 2005 16:28	Dec 28, 2005 23:24	6 Hrs : 55 Mins
6	Al Udeid AB	Ramstein AFB	Dec 30, 2005 03:24	Dec 30, 2005 10:20	6 Hrs : 55 Mins

Figure 3.23: Typical drill-down details of the second plan

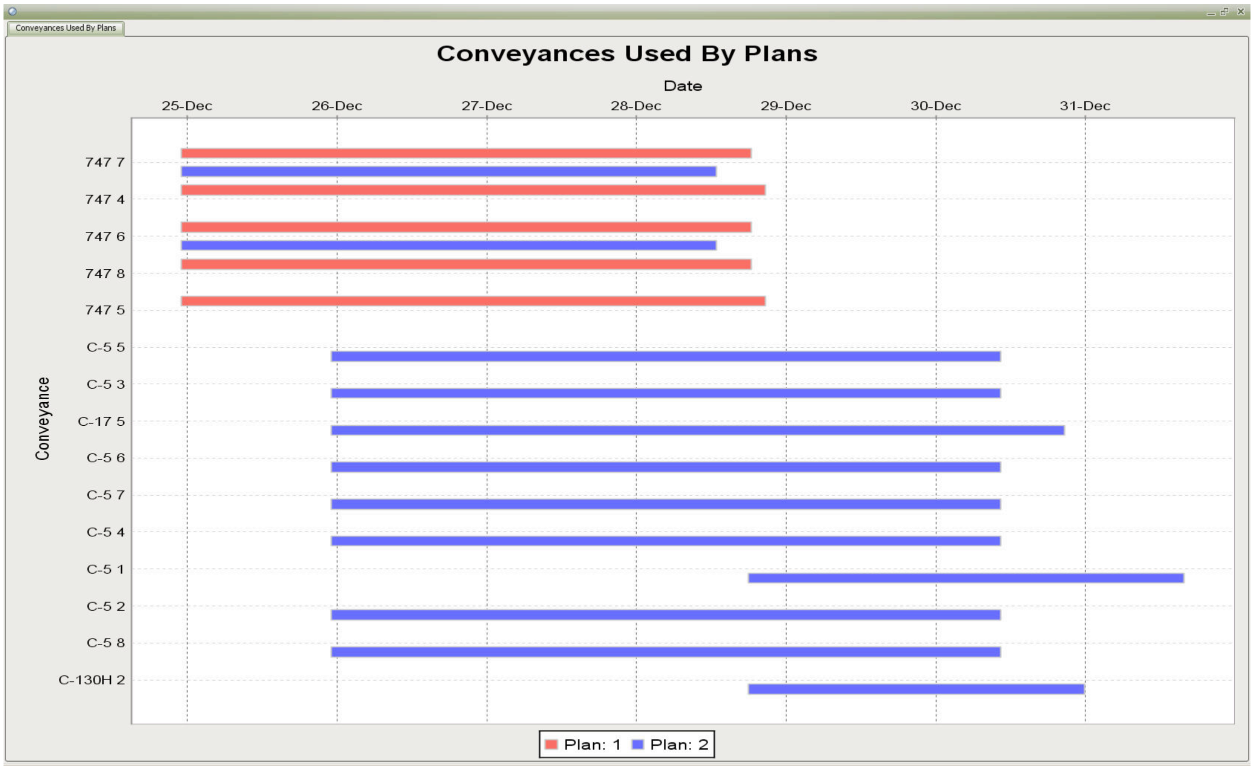


Figure 3.24: Comparison of conveyances needed in support of the first and second plans

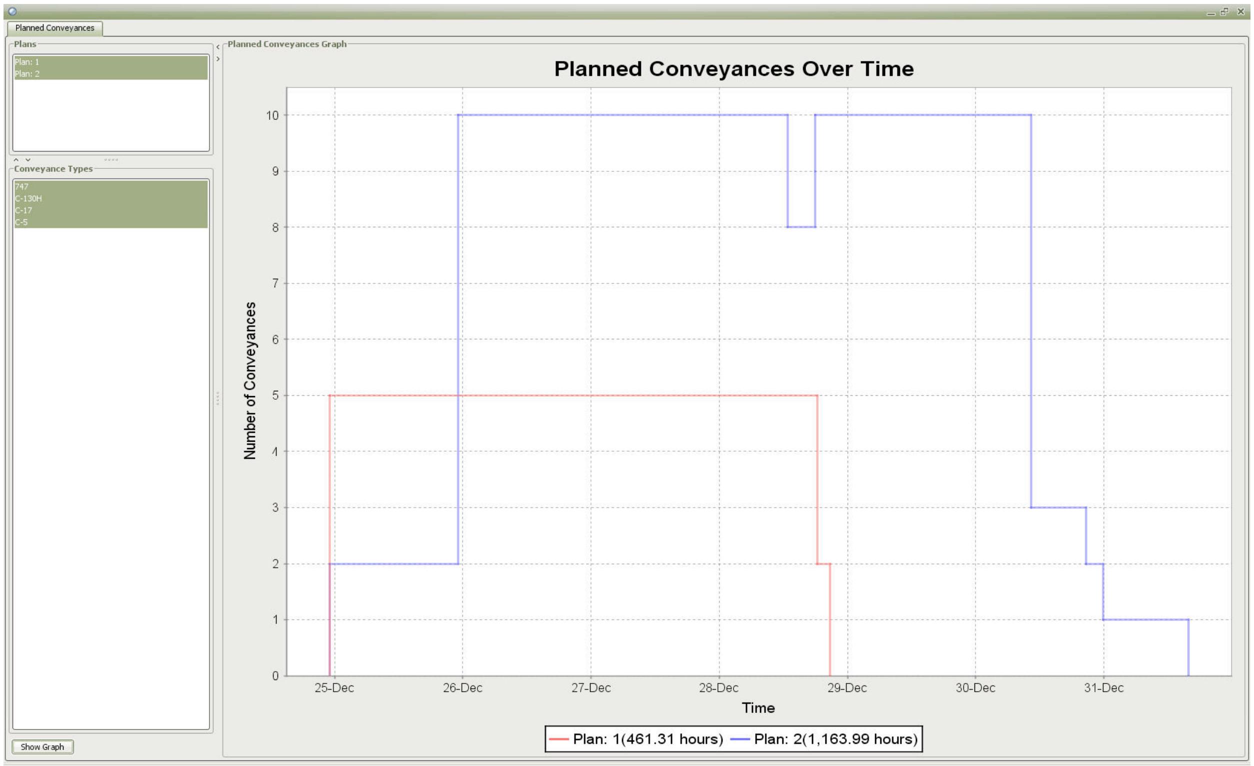


Figure 3.25: Comparison of overall lift requirements for the first and second plans

Apart from the ability of the user to drill down into the details of each delivery plan there are a number of comparative graphical reports available, such as the utilization of specific conveyances by each plan shown in Figure 3.24 and the number of conveyances that are required to support each plan over time shown in Figure 3.25.

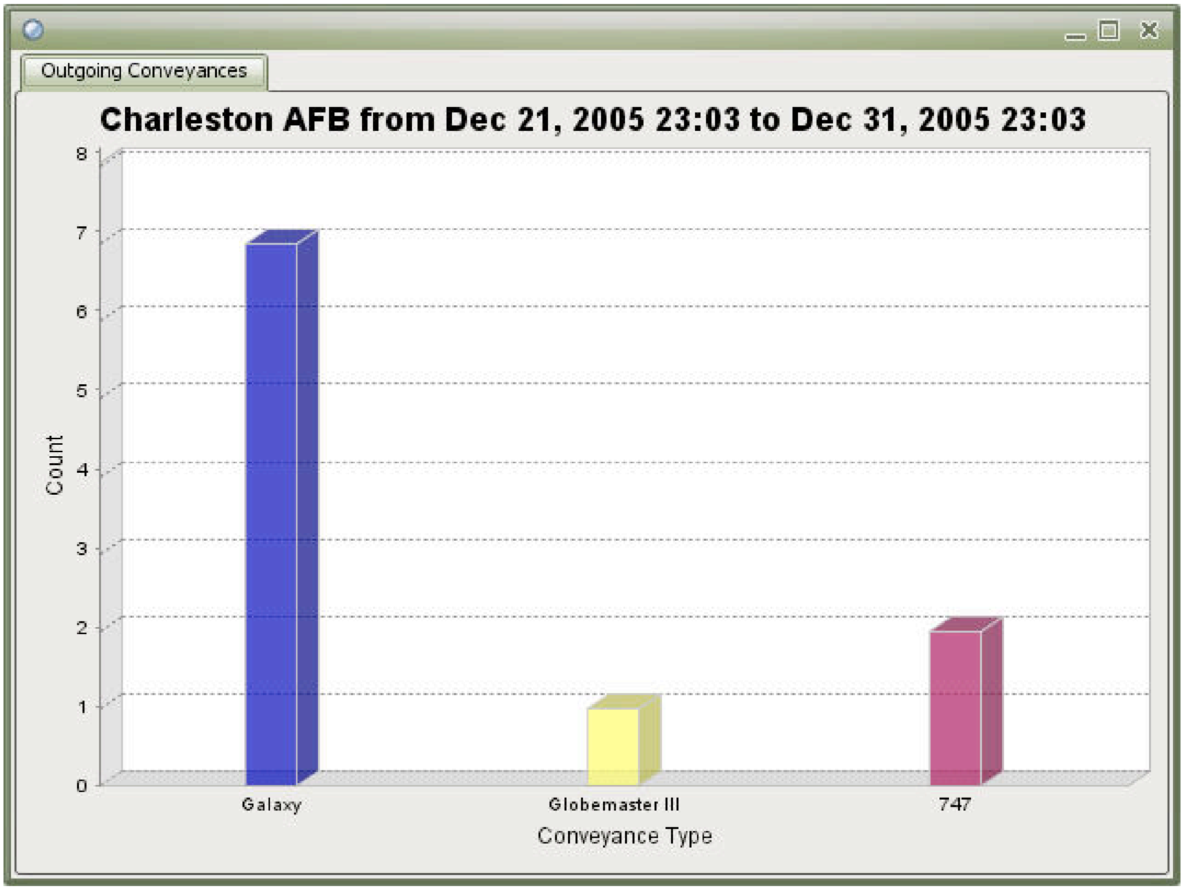


Figure 3.26: Departures from Charleston by conveyance type

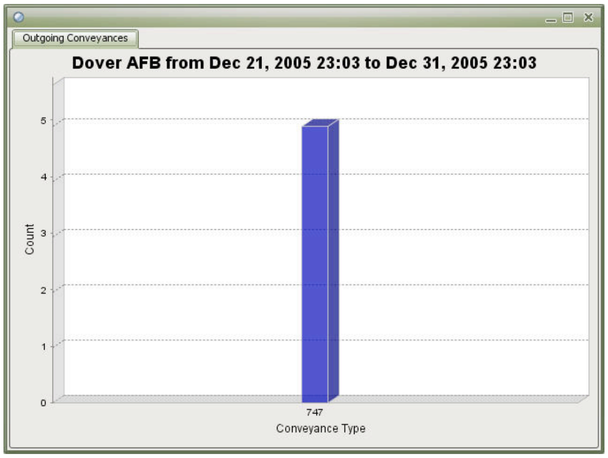


Figure 3.27: Departures from Dover

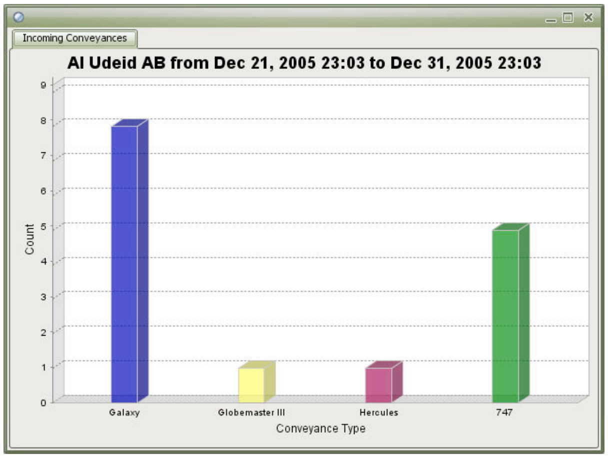


Figure 3.28: Departures from Al Udeid

Figures 3.26 to 3.28 show examples of conveyance departures from the Charleston, Dover and Al Udeid APODs, respectively. Similar reports are available for cargo transfers by date (Figures 3.29 to 3.30) in terms of what was lifted yesterday, the current inventory, and what is planned to be lifted during the next 72 hours. In this way the user is able to determine the expected volume of shipments from any particular APOD on a daily basis. The dates selected for the example bar chart reports shown in Figures 3.29 and 3.30 are December 23 to 26, 2005.

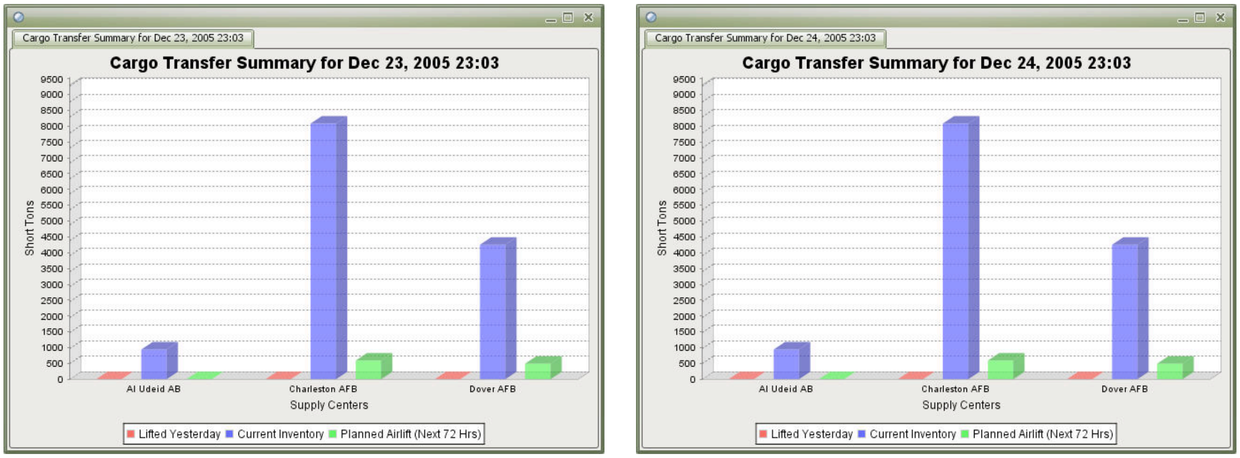


Figure 3.29: Typical cargo transfer history, status, and 72-hour projections

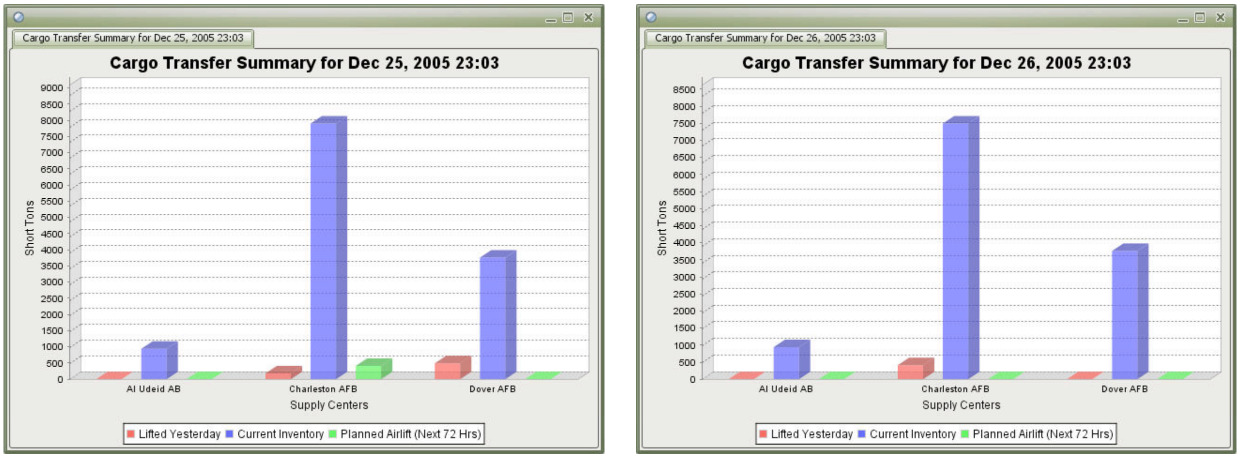


Figure 3.30: Typical cargo transfer history, status, and 72-hour projections

Again, these reports are intended to be examples of the kind of information that can be made available by TRANSWAY. The development team will be guided by feedback from users in future development cycles. The reporting capabilities of the system can be easily extended in any direction within the constraints of data availability.

References

Bauer C. and G. King (2005); 'Hibernate in Action'; Second Edition, Manning Publications, Greenwich, Connecticut.

Crino J. R. (Maj. USA) (2002); ‘A Group Theoretic Tabu Search Methodology for Solving the Theater Distribution Vehicle Routing and Scheduling Problem’; Ph.D. thesis, Air Force Office of Scientific Research, Arlington, Virginia.

Fowler M. (2003); ‘Patterns of Enterprise Application Architecture’; Addison-Wesley, Reading, Massachusetts.

Fowler M (1997); ‘Analysis Patterns: Reusable Object Models’; Addison-Wesley, Reading, Massachusetts.

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

Friedman-Hill E. (2003); ‘Jess in Action’; Manning Publications, Greenwich, Connecticut.

Genesereth M. and N. Nilsson (1987); ‘Logical Foundations of Artificial Intelligence’; Morgan Kaufmann, San Mateo, California.

Glover F. (1986); ‘Future Paths for Integer Programming and Links to Artificial Intelligence’; Computers and Operations Research, Vol 13 (pp.533-5490.

Glover F. and M. Laguna (1997); ‘Tabu Search’; Kluwer, Norwell, Massachusetts.

Hibernate API on-line: http://www.hibernate.org/hib_docs/v3/api.

Hibernate Web site: http://www.hibernate.org.

Karaboga D. and D. Pham (2000); ‘Intelligent Optimization Techniques, Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks’; Springer Verlag, Berlin, Germany.

Leighton R., L. Vempati, A. Davis, M. Porczak and J. Pohl (2004); ‘The ICDM Development Toolkit : Technical Description’ ; Technical Report (CDM-18-04), CDM Technologies, Inc., San Luis Obispo, California.

NASA (1992); ‘CLIPS 6.0 Reference Manual’; Software Technologies Branch, Lyndon B. Johnson Space Center, Houston, Texas.

OpenTS (2005); Web site: http://www.coin-or.org/OpenTS/docs/index.html

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.

Stroustrup B. (1987); ‘The C++ Programming Language’; Addison-Wesley, Reading, Massachusetts.

Talbi E. G. (2002); ‘A Taxonomy of Hybrid Metaheuristics’; Journal of Heuristics, 8 (5), September (pp.541-564).

USTC (2003); ‘Data Management Handbook’; USTRANSCOM (J6), Scott Air Force Base, Illinois.

Wright R. and M. Sweet (2000); ‘OpenGL®: Super Bible’; Waite Group Press, Indianapolis, Indiana.

Tabu Search for Optimization of Military Supply Distribution

Ben Weber
bgweber@gmail.com
CDM Technologies, Inc.
2975 McMillan Ave. Suite 272
San Luis Obispo, CA 93401

Brett Bojduj
bbojduj@calpoly.edu
Department of Computer Science
California Polytechnic State University
San Luis Obispo, CA 93407

Abstract

The dynamic and non-hierarchical nature of the military domain presents a challenge for traditional supply chain optimization. Flow networks and simulation techniques have been applied to the military distribution problem, but are unable to provide time-definite delivery to customers. Recently, optimization approaches have been independently applied towards strategic and operational levels of planning. However, decomposing military distribution into separate problems forces optimization techniques to utilize imprecise data. The size of the military distribution problem has prevented optimization techniques from providing end-to-end planning capabilities. This paper presents a Tabu Search algorithm for simultaneously solving strategic and operational levels of planning. The algorithm uses partial-order planning to separate the optimization process from the constraint verification process. The problem is reduced to a tractable computation by representing scenarios as two-tier systems and only permitting transshipments between different tiers. The results verify that the presented algorithm discovers higher quality solutions than simulation for simultaneously solving strategic and operational levels of planning.

Keywords: agents, logistics, military, operational planning, strategic planning, scheduling, simulation, supply chain optimization, Tabu Search

1. Introduction

Traditional supply chain techniques have been shown to produce inefficient plans when applied in the military domain. Serious shortcomings have illustrated the need for improved logistical processes in military operations such as Desert Storm (Kaminski, 1995) and Allied Force (Brooks, 2000). More recently, the shortfall of add-on-armor in Iraq confirmed supply chain problems (Bowman, 2003). It is necessary to discard the “just-in-case inventory” approach (Schrady, 1999) and move to a rapid and reliable transportation process that provides time-definite delivery to customers (Crino et al, 2002).

The military domain presents a challenge for supply chain optimization, because problems are dynamic and non-hierarchical. Problems in the military domain are more dynamic than problems in the commercial domain, because military problems operate in hostile environments. The military distribution problem may include several theaters of operation, which change as a plan is executed. Threats and unforeseen contingencies may cause a plan to become invalid. If hostile forces attack a convoy, then an alternate plan must be generated to distribute supplies to units. Therefore, the military requires a tool for efficiently responding to changes in a scenario.

The military distribution problem has been represented hierarchically by traditional planning techniques, such as flow networks and simulation. Modeling the problem hierarchically allows the problem to be decomposed into smaller problems, but fails to accurately represent the problem. The military distribution problem does not fit this structure for two reasons: the destinations of supplies are unknown before planning; and, supplies may be delivered from multiple theaters of operation. Therefore, it is necessary to discard the hierarchical approach and utilize an ad-hoc structure.

Military supply distribution is divided into strategic, operational and tactical levels of planning. Operational planning consists of the allocation of supplies and personnel between different locations within a theater. A theater is defined as a geographical area of operation outside of the continental United States under the responsibility of a commander (Crino et al, 2002). Strategic planning consists of the distribution of supplies, personnel, and transportation assets between different theaters of operation. The tactical level of planning specifies the movement of supplies from locations within a theater to individual units. This paper considers only the strategic and operational levels of planning.

The military distribution problem consists of the end-to-end distribution of supplies and personnel between geographical areas of operation. The objectives are to minimize shortfall and minimize the cost required to execute plans. A planning system must meet the following requirements: end-to-end planning; routing and scheduling of transportation assets at the strategic and operational levels; and, time-definite deliveries.

2. Related Work

Current planning techniques include planning by hand, flow networks, simulation, and optimization. Planning by hand is feasible for small scenarios. However, solutions generated by Tabu Search are significantly superior to those obtained by hand for large problem sets (Semet and Taillard, 1993). Planning by hand is unsuitable for the military distribution problem, because scenarios are subject to frequent change.

Problems in the commercial domain are often represented hierarchically and analyzed using supply chain techniques to trace the throughput of each node (Beamon, 1998). Supply chain techniques allow planners to identify problems in the distribution system and react accordingly. Flow networks are a supply chain technique that has been applied to the military domain (McKinzie and Barnes, 2004), providing a tool for analyzing the throughput of hubs in a scenario. Supply chain techniques maximize the throughput of individual nodes, but are unable to provide time-definite deliveries to customers.

Simulations are ruled-based models for solving the military distribution problem (Wu et al, 2003). Simulation models attempt to model military scenarios as accurately as possible. Therefore, plans generated by a simulation are valid for real-world scenarios. Rule-based models are an effective technique for satisfying constraints, but fail to optimize the utilization of resources. Simulations provide a tool for efficiently generating feasible plans. However, simulation models applied to end-to-end planning have been unable to prescribe routing and scheduling of transportation assets at the operational level (Crino et al, 2002).

Current research has utilized optimization techniques for solving the military distribution problem. Most attention has focused on the use of Tabu Search to optimize the utilization of resources. Optimization techniques using Tabu Search have demonstrated the ability to schedule transportation assets at the operational level and provide time-definite deliveries to customers. Tabu Search has been applied towards planning at the strategic level (Barnes et al, 2004) and planning at the operational level (Crino et al, 2002). However, the size of the military distribution problem has prevented optimization techniques from providing end-to-end planning capabilities.

3. Tabu Search Approach

This paper presents a Tabu Search algorithm for simultaneously solving strategic and operational levels of planning. Tabu Search combines greedy heuristics and memory structures to effectively traverse through solution spaces (Glover and Laguna, 1997). The algorithm uses partial-order planning to separate the optimization process from the constraint verification process. The problem is reduced to a tractable size by representing problems as two-tier systems. Also, the search incorporates additional heuristics to improve performance. The result is an algorithm that quickly converges to feasible solutions.

The algorithm uses partial-order planning to achieve a combination of simulation and optimization, and utilizes an approximated objective function that separates the intelligent component of the algorithm from the constraint-checking component. The intelligence of the algorithm is represented as an objective function, used by the search. However, it may be impractical to consider all necessary constraints through an objective function. Therefore, the search considers only those constraints that directly affect the quality of a solution. The algorithm uses an objective function to determine which deliveries to consider for addition to the current plan. Once a candidate delivery has been selected, the constraint portion of the algorithm determines when the delivery should be scheduled and verifies the delivery against a set of rules.

This hybrid approach offers two benefits, but there is a tradeoff. The first benefit is faster iterations compared to a pure optimization approach, since constraint verification is applied to a single delivery each iteration. The second benefit is that the planning agent can handle additional constraints without modification of the optimization portion of the algorithm. However, the optimization component may select a move that violates constraints, because the objective function does not consider all constraints. Therefore, the optimization component may select a poor quality move that degrades the quality of the solution. This problem is resolved using feedback from resulting solutions. The search is informed if a selected delivery improves or degrades the quality of a solution.

The algorithm represents scenarios as two-tier systems. Deliveries at the strategic level represent the top tier, while deliveries at the operational level represent the bottom tier. The search alternates between two modes. In the first mode, the search schedules deliveries for the bottom tier and ignores transshipments. In the second mode, the planning agent schedules deliveries for the top tier and allows transshipments between the different tiers. The planning agent limits deliveries to a maximum of two transshipments and only a single top-tier delivery is permitted.

This structure prevents the algorithm from solving all possible scenarios, but greatly reduces the permutations of deliveries considered by the algorithm.

The Tabu Search algorithm uses the following heuristics to improve the performance of the search: scheduling heuristic; pick-up heuristic; and, removal heuristic. The scheduling heuristic determines when to schedule deliveries by trying to schedule all deliveries as early as possible. However, if a conveyance does not have an opening for an additional delivery, then the scheduling heuristic ejects deliveries that deliver to customers with later request times until an opening is available. The pick-up heuristic is used to estimate the cost to transport supplies from within a theater to locations used for transshipments. It enables the algorithm to consider transportation assets at the operational level when planning deliveries at the strategic level. The last heuristic is the removal heuristic, which deterministically removes deliveries from the solution. The heuristic verifies that customers are supplied from the closest possible supply location. If a customer is not satisfied from the closest supply location, then deliveries currently using the location are ejected from the solution.

4. Results

The performance of the Tabu Search algorithm was tested in six scenarios against a simulation model. The scenarios were designed to test various aspects of the Tabu Search algorithm and demonstrate the non-hierarchical nature of the military supply distribution problem. The 'Greedy' and 'Theater' scenarios demonstrate the strengths of the heuristics. The 'Non-hierarchical' and 'Multi-theater' scenarios demonstrate the ability of the Tabu Search algorithm to solve multi-theater problems. Finally, the 'Cargo' and 'World' scenarios validate the ability of the Tabu Search algorithm to solve large problems. A summary of results for the testing scenarios is listed in Table 1. Shortfall represents unmet customer demand. Tabu Search was able to satisfy more customer demand than the simulation for all of the scenarios tested. Also, Tabu Search converged to solutions in less than two minutes for the largest scenarios. Even though the travel distances required for the Tabu Search solutions are consistently longer than the distances for the simulation solutions, it should be noted that the differential reduces to less than 7% with increasing complexity and size of the problem space. Furthermore, the Tabu Search algorithm was able to accomplish all deliveries, except for the World scenario, while the simulation solutions resulted in significant shortfalls. The results verify the feasibility of optimization techniques for strategic and operational levels of planning and demonstrate that Tabu Search outperforms current simulation techniques.

	Simulation Shortfall	Simulation Distance	Tabu Search Shortfall	Tabu Search Distance
Greedy	1	100	0	600
Theater	20	6055	0	7407
Non-hierarchical	42	23206	0	32127
Multi-theater	16	40254	0	41745
Cargo	656	2422633	0	2525579
World	508	4564128	17	4859124

Table 1 - Scenario results

5. Conclusion

Current supply chain techniques are unable to meet the requirements for the military distribution problem. Optimization techniques attempt to find optimal solutions and therefore cannot solve large-scale scenarios. This paper presents a Tabu Search algorithm that sacrifices optimality for practical run-time computation by separating the optimization component from the constraint-checking component and limiting the combinations of feasible transshipments. The Tabu Search algorithm combines the strengths of simulation and optimization through the use of partial-order planning. The performance of the Tabu Search algorithm was tested in six scenarios against a simulation model. Tabu Search discovered superior solutions on all problem sets and all solutions were found in less than two minutes. The results verify the ability of Tabu Search to solve at the strategic and operational levels of planning.

References

Barnes, J. W., Wiley, V. D., Moore, J. T., & D. M. Ryer (2004); Solving the Aerial Fleet Refueling Problem using Group Theoretic Tabu Search; Mathematical and Computer Modeling, Vol. 39 (pp. 617-640)

Beamon, B. M. (1998); Supply chain design and analysis: Models and methods; International Journal of Production Economics, Vol. 55 (pp. 281-294)

Bowman, S. (2003); Iraq: U.S. Military Operations; Congressional Research Service Report for Congress

Brooks, R. (2000); Position Paper on the Theater Distribution Management System (TDMS) Initiative; Headquarters, United States Air Force, Washington D.C.

Crino, J., Moore, J. T., Barnes, J. W., and W. P. Nanry (2002); Solving the theater distribution vehicle routing and scheduling problem using group theoretic Tabu Search (Technical Report); Air Force Institute of Technology, University of Texas at Austin

Glover, F., and M. Laguna (1997); Tabu Search; Kluwer Academic Publishers

Kaminski, P. G. (1995); The Revolution in Defense Logistics, Keynote address of the USAD for Acquisition and Technology, Alexandria, Virginia

McKinzie, K., and J. W. Barnes (2004); A Review of Strategic Mobility Models Supporting the Defense Transportation System; Mathematical and Computer Modeling, Vol. 3

Schrady, D. (1999); Combatant Logistics Command and Control for the Joint Force Commander, Navy report

Semet, F., and E. Taillard (1993); Solving real-life vehicle routing problems efficiently using tabu search; Annals of Operations Research, Vol. 41 (pp. 469-481)

Wu, T. T., Powell, W. B., and A. Whisman (2003); The optimizing simulator: an intelligent analysis tool for the military airlift problem (Technical Report); Department of Operations Research and Financial Engineering, Princeton University

InfoTagSim: A Wireless Sensor Network Simulator[*]

James De Leon and Khang Duong

Computer Science Department
California Polytechnic State University
San Luis Obispo, CA 93407
{jedeleon, kduong}@calpoly.edu

June 6, 2006

1 Abstract

Wireless sensor networks have been a popular research topic in recent years due to advances in low power radio frequency (RF) transceivers and embedded microcontrollers. The study reported in this paper focuses on localization, a process by which sensor nodes gather data to aid in accurately estimating their geographic positions. Specifically, the objective is to use non-GPS localization to allow for automated location-mapping, identification, and management of cargo items on military ships and loading areas. The high resource cost in designing, implementing, and testing a full-scale, active RFID hardware system poses a significant limitation for this area of systems research. Accordingly it is proposed to design and implement a simulator to determine the feasibility of such a system.

InfoTagSim is a simulator that analyzes the behavior and performance of wireless sensor networks that utilize localization techniques. InfoTagSim was designed with the following key functionalities in mind: (i) to facilitate the development of new protocols; (ii) to allow for scalability of networks; (iii) to model node hardware characteristics; (iv) to analyze overall system behavior and performance; and, (v) to test the accuracy of the localization algorithm. The paper describes the approach, design, and implementation of InfoTagSim and the results of simulation, which vary in scale, propagation methods, and system characteristics. Furthermore, the accuracy of the localization algorithm, which is based on network node peer-to-peer communication, connectivity range restraints, and computational geometry, is evaluated.

2 Introduction

For those involved in the movement and storage of large quantities of items, much focus lies in solving the problem of inventory management. After completing the planning and storing stages, checking the consistency of the stocked items against the plan forms the bulk of inventory management and also showcases why many seek to improve the tasks involved. Essentially, those tasks consume a great deal of time and energy, and often is tedious work for those involved. Through the use of technology, researchers and industry

---

[*] Research supported by CDM Technologies, Inc., San Luis Obispo, California, USA.

have been able to chip away at the problem, introducing ideas like barcodes, databases, and RFID devices to help automate many of the tasks.

For the military, inventory management poses as complex a logistics problem as stocking a supermarket. However, much of the inventory management deals with military ships and staging areas. This involves the stowing, transportation, and rapid deployment of "…large numbers of tracked and wheeled vehicles, weapons systems, ammunition, power generating and communication facilities, food supplies, and other equipment" [10]. To handle such large tasks, the military funds many research paths, some along the lines of industry, and others along the lines of future technology.

One such example involves a software tool designed to automate the planning process of inventory management, called the Integrated Computerized Deployment System (ICODES). Through the use of intelligent software agents, ICODES currently creates stow plans that factor in hazard material placement, trim and stability, and accessibility [4]. While the software reduces time spent in the planning stage, it still does not alleviate the time-consuming task of checking the actual placement of cargo items on a ship against the plan. Last minute cargo changes, misplaced items, delays in ship arrival, and more can lead to changes in the actual stow locations. This in turn requires that an individual or group of individuals walk around the stow area verifying the original stow plan and checking for inconsistencies. Any inconsistencies then require input into the software to keep the plan up to date.

Due to the software shortcomings, the military also funds other types of research, with the primary goal being the automation of the consistency-checking stage of inventory management. One such research path involves the creation of a location map of *as-loaded* cargo using technology commonly found in industry. Barcode technology coupled with hand-held scanners allows a cargo specialist to walk up to a cargo item, scan and automatically identify the item, and manually place the item into an ICODES plan [3]. This software, named Automated Identification Technology (AIT), validates actual cargo locations against the stow plan. AIT-based processes aim to increase the efficiency and speed of obtaining an *as-loaded* plan, thereby improving in-transit visibility [3]. While an improvement over existing solutions, AIT still exhibits considerable limitations. Tests in the field reveal difficulties in scanning barcodes in direct sunlight and in obtaining barcode data on cargo objects too tall to reach or in other inaccessible locations. In addition to the line-of-sight requirement, this solution still requires a significant amount of time and labor. A person or team must manually scan and place each and every item on the ship, and a military cargo vessel may carry as many as 4000 items.

Another research path involving the consistency-checking stage of inventory management involves passive RFID tags. While a logical successor to barcode technology, current RFID technology focuses on supply chain management and lacks the granularity needed to form location maps of cargo items [11]. At best, passive RFID tags can help place an item in a certain room or hold (through the use of an interrogator at the entrance), but not much more. Further, passive tags possess limited memory and cannot store information sent by a transceiver.

Continuing the same line of research, the next logical step involves the use of active RFID tags. Active RFID tags have longer ranges than both barcode scanners and passive RFID tags. They operate autonomously and can communicate with other tags within range. With those added capabilities, RFID tags can possibly provide a fine enough granularity to perform location estimation of the items associated with the tags. However, limitations such as bandwidth efficiency and battery power may prove to be considerable obstacles to overcome. Still, active RFID tags provide enough potential to make them a worthy research path to follow.

2.1 Motivation

A research project currently undertaken at CDM Technologies uses active RFID tags to perform location estimation. The InfoTags project attempts to augment both ICODES and AIT with the objective of providing a complete inventory management system that is resident in one package. While still in the proof-of-concept phase, the InfoTags project currently consists of a small number of actual tags and a software program that performs location estimation. The overall goal of the InfoTags project is to further automate the consistency-checking stage of inventory management to the point where only a few items need to be scanned to collect information about every other item in the area.

Although the design and testing of a small set of prototype tags has proved useful for gaining an understanding of the operating characteristics of individual tags and the efficacy of physical test facilities (e.g., obtaining transmission ranges), the high cost of these devices poses a serious limitation. First, this route provides little insight into the performance of a full-scale sensor network implementation. Information about propagation times over large networks and overall feasibility remain unknowns. Second, prior to the construction and assembly of the hardware, the sensor network communication protocol requires rigorous testing to ensure proper operation and acceptable performance. Creating hundreds of tags with a poorly designed and tested network communication protocol can prove too costly for a small company. To address these issues, it is proposed to design and implement a wireless sensor network simulator, referred to as InfoTagSim.

3 Overview

InfoTagSim exists specifically to test wireless sensor network protocols with a central focus on localization. Localization is the process by which sensor nodes gather data to aid in accurately estimating their geographic positions [12]. Position estimation is a key element in sensor network applications that:

- Determine the origins of events.
- Perform self-discovery of network coverage and topology.
- Attempt to map physical objects to coordinate locations.

This paper investigates the third motivation in an attempt to increase the speed and operational efficiency of stow planning on military cargo ships.

One solution for localization is the Global Position System (GPS) [1], which uses satellites to find the position of a connected object. Applications can be found in navigation systems in cars, hand-held devices used during outdoor expeditions, and even cell phones. If each of the tags in the wireless sensor network were to be equipped with a GPS receiver, the tags would then possess the capability to obtain their position coordinates from communication between GPS satellites. However, several reasons prove GPS to be an inviable technology for stow planning operations. GPS receivers only work outdoors. The military needs to obtain position estimates from items within cargo holds. Also, the technology is currently too expensive to outfit active RFID tags at the required scale. Lastly, the power constraints of a tag may rule out the use of a GPS receiver. The above reasons prompted this research team to forgo a GPS implementation in favor of localization methods based on network node peer-to-peer communication, connectivity range restraints, and computational geometry.

The work of Doherty, Pister, and El Ghaoui [6] largely influenced the particular implementation of localization used in the InfoTags project. In their research paper, they propose modeling peer-to-peer communication as a set of geometric constraints that yield position estimates for unknown node locations. In other words, one can turn connectivity data gathered by the tags into geometric computations yielding relative positions. If two tags can establish a connection, then the distance between the two tags is, at maximum, the length of the broadcast radius. Conversely, if two tags never receive each other's transmissions, the separation distance between them is at minimum the length of the broadcast radius. With a complete set of connectivity information, it is possible to apply computational geometry to calculate tag position estimates.

3.1 Connectivity Data

The singular goal of the networked sensors in the InfoTags project is to obtain connectivity data. They achieve this by updating and maintaining adjacency tables kept in the tags' external memory. When two tags establish a connection, they are referred to as *neighbors*. Alternately, this connection is referred to as an *adjacency pairing*.

Each individual tag keeps both its own record of adjacencies (i.e., who its neighbors are) and a record of *second hand* adjacency data (i.e., the neighbors of other tags). When a tag receives a data packet, it first obtains the identification number of the packet's originator. For example, if Tag A receives a packet from Tag B, Tag A will first mark Tag B as a neighbor in its adjacency table. Then, Tag A records the contents of the received packet (i.e., Tag B's adjacency data) into its memory. For example, if the packet contains data that recognizes Tag C and Tag B as neighbors, Tag A will record this adjacency pairing in its own table. To reiterate, Tag B is essentially sharing a portion of its adjacency data with Tag A, which will record these adjacency pairings in its own tables. Tags connected in the same cluster eventually obtain all the adjacency data information of all the tags in the cluster. Ideally, after a certain period of time, every tag in the cluster should hold identical adjacency tables. This design allows any one of the tags in a cluster that are interrogated (by a PDA with a RF transceiver) to supply the same adjacency tables, thus requiring only one tag to be interrogated. As expected, the tags completely out of range from all other tags necessarily possess empty adjacency tables.

However, this condition should rarely occur in a ship stowing environment where objects are typically packed closely together.

3.2 Broadcast Protocols

Obtaining adjacency tables forms a crucial step to achieving localization. Network protocols must be carefully designed to disseminate adjacency data to all nodes with the following criteria:

- Keeping network overhead to a minimum,
- Avoiding node crosstalk (i.e., packet corruption due to two broadcasts occurring simultaneously).
- Minimizing the time requirements for adjacency data to propagate to all nodes.
- Maximizing energy efficiency.

Each of these criteria elements interrelates. A protocol can be designed to tolerate a certain level of crosstalk in return for the faster propagation of data. Reducing network overhead could also realize gains in the energy efficiency of tags. Moreover, allowing for higher levels of network traffic may increase node crosstalk while decreasing the energy efficiency.

In order to keep network overhead to a minimum, the protocols were designed to operate exclusively in broadcast mode. When a tag transmits a data packet, any node within the transmission vicinity of the originating tag receives the transmitted packet. Since the essential function of the tags aims to seek out and share information with adjacent tags, broadcast mode serves as the logical choice. To reduce network overhead, tags send no acknowledgment packets back to the packet originators.

Since the specified protocols operate purely in broadcast mode, contention of the wireless medium must be considered as a design issue. Crosstalk occurs when two nodes within the same vicinity broadcast their packets simultaneously. The simultaneous broadcast situation can corrupt a packet for any receiving nodes in that vicinity. Proper error checking through the use of cyclic redundancy checks (CRCs) allows a receiving tag to reject a corrupted packet. Error checking means that the receiver cannot obtain the data from either transmitter since the information is rejected, which also means that the transmitting nodes have wasted energy in transmitting a rejected packet. The worst-case scenario involves frequent packet collisions, slowing propagation of adjacency data considerably since the rejected packets require another transmission. With more broadcasts required to complete the propagation of adjacency data to all tags, network overhead increases as well. The question arises: How much crosstalk can be tolerated before reaching unacceptable levels in the completion time of data propagation?

The above discussion prompted the design of two simple broadcast protocols:

- **Time-sharing protocol**: The time-sharing protocol avoids crosstalk altogether by limiting access to the wireless medium and eliminating contention. The simulator assigns every node a time-share (i.e., a period in which it owns exclusive rights to broadcasting). The major drawback of this implementation lies in the increase of tags in the system. The more tags in the system, the slower the propagation of data. The simulator permits a tag to broadcast a packet only once before waiting for every other tag in the system to broadcast. In an example case of 300 tags in the wireless sensor network, a tag must wait 299 broadcast intervals before the simulator permits the tag to broadcast again. Data propagation speed can be improved by increasing the length of the packet transmitted, thereby sending more adjacency data to its neighbors, as well as by reducing the broadcast interval. This protocol is used as a base protocol to be compared to others that are designed for better performance.

- **Pseudo-random transmission protocol**: Instead of assigning time-shares, the simulator allows each tag to broadcast at random intervals. Setting a minimum interval and a maximum interval bounds the transmission intervals. For instance, if the minimum is set to 1 second and the maximum is set to 10 seconds, the tag first generates a random number between 1 and 10. The tag then waits in receive mode for the number of seconds of the randomly generated number. At the end of that period, the tag transmits its adjacency data. The tag then waits for the broadcast interval before choosing another random number and waiting in receive mode. The process then starts over again. Compared to the time-share protocol, the pseudo-random protocol achieves a higher frequency of transmissions in a given period, thereby decreasing the length of time for the completion of the data propagation sequence. However, packet collisions occur frequently in this protocol. Because the simulator randomly generates time between transmissions, two or more tags can possibly broadcast simultaneously. This can lead to crosstalk if the tags lie within transmission range of each other.

  Packet collisions can be tolerated in the pseudo-random protocol because:

  1. Completion time of data propagation improves considerably over the time-sharing system.
  2. Packet collisions do not adversely affect network overhead and energy efficiency.

  The completion time is expected to decrease considerably, especially as the number of tags in the network increases. Overall, the pseudo-random algorithm is expected to offer better performance in comparison to the time-sharing protocol in large-scale systems, with packet collisions a minimally obtrusive data propagation factor.

4 Design Goals

During the initial design process of the simulator, the research team brainstormed in their endeavor to have the design of the simulator address the following criteria that were considered very important:

- To facilitate the development of new protocols.
- To allow for scalability of networks.
- To model node hardware characteristics.

- To analyze overall system behavior and performance.
- To test the accuracy of the localization algorithm.

As discussed in the previous sections, new protocols can be written and tailored to specific applications. In the case of military ship stowing, the wireless sensor network functions to solve localization of cargo objects. After designing new protocols, InfoTagSim can be used to test and analyze these protocols for functionality, accuracy, and performance. Analysis of the outcome of simulations should help to easily detect design errors. For instance, inconsistencies in the final adjacency tables of tags could mean that the data did not propagate correctly. This is a telltale sign of a flaw in the protocol design. Protocols deemed too slow in forming complete adjacency tables can be modified and retested.

Another major objective for InfoTagSim involves the ability to test wireless sensor networks at a real-world scale. Limitations in time, research funding, hardware expertise, and personnel resources preclude the research team from undertaking actual hardware testing on the scale needed to accommodate typical stow planning environments. Normally, as many as 4000 vehicles, crates, and other equipment is required to be stowed on one military ship. InfoTagSim should allow those quantities to be simulated. Additionally, cargo holds vary in shape and dimensions. Cargo staging areas, depending on physical dimensions and cargo type, may contain more than 600 individual cargo items waiting to be loaded onto a ship. Conversely, some cargo holds onboard ships have a capacity of only several dozen items. With InfoTagSim, the research team is able to simulate sensor networks of 500 nodes just as easily as sensor networks of 50 nodes, with the ability to set the physical dimensions to best represent an actual cargo hold or staging area.

The third design goal of InfoTagSim revolves around the objective of a hardware-independent simulator. In other words, the research team wanted the ability to modify hardware (i.e., active RFID tag) characteristics that affect the outcome of the simulation. InfoTagSim can aid the hardware designer in component selection by allowing hardware characteristics to be modeled and used in the simulation. If the designer currently uses commodity RF transceivers, he or she can obtain hardware characteristics from the data sheets, or can extract the data from hardware experiments on individual transceivers. Currently, InfoTagSim supports simulator variables of broadcast radii (i.e., the transmission distance outward from the tags) and also attempts to model RF radiation patterns of a transceiver as a set of probabilities. Further discussion on these simulator variables follows in subsequent sections.

Another design goal centers on allowing the research team to analyze overall system behavior and performance. Full access is provided to the simulation results in the form of output text to a console screen, visual output on a map, and output similar to the console in log files. The data documented in the console and log files include:

- Node coordinates,
- Specific times of broadcasts,
- The identification number of the transmitting tag,
- Identification numbers of receiving tags,

- Packet data,
- Time elapsed until completion of data propagation, and
- Occurrence of packet collisions.

The simulation console screen and log file seek to allow the researcher to gain a better understanding of how a specific protocol behaves in a given system environment.

The final design goal revolves around the localization algorithm and testing its capabilities and accuracy. In other words, after running the simulator, the algorithm tester can be applied to the results of the simulator to test the accuracy of location estimation and completeness. The algorithm does this by gathering all the adjacency data from a randomly selected node and performing computational geometry on the resulting adjacency sets to estimate node locations. InfoTagSim includes the localization algorithm tester as a runtime option. It is capable of displaying the resulting coordinates on a map shown beside the simulator results, and performing comparisons of the results to the actual node locations. The intention is to fine-tune the localization algorithm to a point where it performs fairly well with both large and small datasets.

5 Implementation and Usage

Subject to the design goal of testing the localization algorithm, it was decided to write InfoTagSim in the C++ programming language. A simple graphic user-interface (GUI) was added to the application to allow the user to modify simulator environment variables and to visualize the node placement on the map. Users can save and load the locations of a simulation run and can view log files created by the simulator during program execution.

The first screen displays a dialog box with edit boxes for the user to enter information for modifying simulation environment variables. Here, the user can enter in the number of tags, broadcast probabilities, broadcast range, broadcast intervals, the network protocol and more. Further discussion on the simulation variables follows in the next sub-section.

After a user enters all the desired changes, the user then clicks on the *Begin Simulation...* button to bring up another dialog box. This dialog box displays two maps, one for the simulation, and one for the algorithm test. Beneath that lies a console that displays all relevant information pertaining to the simulation and algorithm test runs. Buttons for constructing tags, starting the simulation, testing the algorithm, saving locations, and loading locations lie between the upper and lower portions of the dialog.

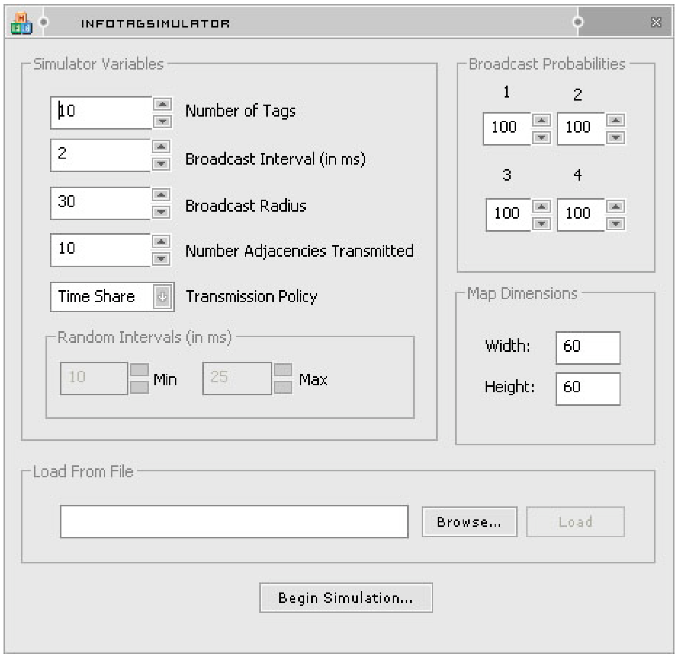
5.1 Simulation Parameters

Figure 1 illustrates the graphical user interface that allows the user to set parameters and configuration options before running the simulation. The following provides a brief description of the simulation parameters available.

- *Number of tags*. The number of tags simulated in the wireless sensor network.
- *Broadcast interval*. The broadcast interval sets how often a broadcast occurs. For the time-sharing protocol, for instance, if the broadcast interval is set to 10

seconds, one broadcast will occur every 10 seconds. If there are 3 tags in the system, Tag A will broadcast at 0 seconds, Tag B will broadcast at 10 seconds, and Tag C will broadcast at 20 seconds. At 30 seconds, the time-sharing cycle is restarted and Tag A broadcasts again. For the pseudo-random protocol, the broadcast interval causes the tag to wait for the assigned interval before selecting a random time to start a broadcast.

- *Broadcast radius*. The broadcast radius specifies the transmission range of the tags in feet. This parameter models the particular RF transceiver used in the wireless sensor network in terms of transmission power.



Simulator Variables

10 Number of Tags
2 Broadcast Interval (in ms)
30 Broadcast Radius
10 Number Adjacencies Transmitted
Time Share Transmission Policy

Random Intervals (in ms)

10 Min
25 Max

Broadcast Probabilities

1: 100
2: 100
3: 100
4: 100

Map Dimensions

Width: 60
Height: 60

Figure 1: Screenshot of InfoTags configuration options

- *Number of adjacencies transmitted*. A user may set the number of adjacency pairs transmitted, which directly correlates to the maximum length of packets. A higher number of adjacencies transmitted reduces the number of times a tag must broadcast before sharing its entire adjacency table with its neighbors.
- *Transmission policy*. Can be either the time-sharing or the pseudo-random transmission protocol. Section 2.3 describes these protocols in more detail.
- *Random intervals*. Used exclusively for operation of the pseudo-random protocol. When a random transmission interval is generated in a tag, the number falls between the minimum and maximum intervals specified.
- *Map dimensions*. The width and height dimensions of the test area (in feet). Currently, InfoTagSim models a rectangular room but later iterations may allow for rooms of varying shapes and sizes.
- *Broadcast probabilities*. These probabilities roughly model variations in RF radiation patterns in commodity RF transceivers. Ideally, RF radiation

patterns radiate in a circular form. However, depending on the specific RF transceiver used, this may not be the case [7]. The tag is divided into quadrants, with the probabilities for each respective quadrant set to match the radiation pattern of that area. Failing the probability test results in a dropped packet.

5.2 Running InfoTagSim

After setting the simulation parameters, InfoTagSim allows the user to construct a random placement of tags and run the simulation to completion. Completion occurs when all adjacency data have been discovered and proliferated to each of the tags in the cluster. How can completion be adapted in the simulator? Finding the solution to this question remains perhaps the most difficult part of implementing InfoTagSim. Once the application establishes node locations and the simulator executes, the tags begin establishing connections and frequently update their adjacency tables with new data. As time progresses, the tags receive fewer and fewer packets that do not contain redundant information (i.e., the adjacency data that a tag already contains in its adjacency table). Based on an arbitrary design decision the simulator stops after every one of the tags receives no new information for 20 consecutive packet receptions. If a tag's redundant packet count is 19, but then in its next reception contains an adjacency it does not already have recorded, the redundant packet count resets to 0. The simulator stops when all the tags in the cluster reach the redundant packet count.

While the simulator runs, the console is continuously updated in respect to: the iteration (i.e., time slice); which tags are transmitting; which tags are receiving; and, the tags that have rejected packages. Once the simulator stops, the user is informed of the total time taken and the total number of broadcasts.

6 Experimental Results

In the first round of experiments, the broadcast probabilities were varied to model different radiation patterns in RF transceivers. Both the time-sharing protocol and the pseudo-random protocol were tested. If the broadcast probability was set to 100, then each of the four quadrants were set to 100. This process was repeated for probabilities of 90 through 50, decrementing by 10 for each set of runs. For both protocols, the number of tags in the system was 10 for every run. Results are shown in Figures 2 and 3. Figure 2 tells us that, in order to reach completion of data propagation in the system, both protocols require an increasing number of transmissions. This is due to the increasing likelihood of dropped packets. Figure 3 shows that the time of completion increases as the probability of successful transmissions decreases. Both protocols appear to be similarly affected by decreasing probabilities. An interesting result of Figure 3 is that the pseudo-random algorithm takes significantly less time before data is propagated to all tags. In one set of runs, the pseudo-random algorithm was on average of 72% faster than the time-sharing algorithm.

In the next experiment, the total number of transmissions is observed in runs of increasing tag quantities. The starting number of tags is 10 and the final set of runs has

50 tags, incrementing the tag count by 10 for each set of runs. The simulations take place in 60 ft by 60 ft room. The tags were set to have broadcast radii of 20 feet. Figure 4 displays the results.

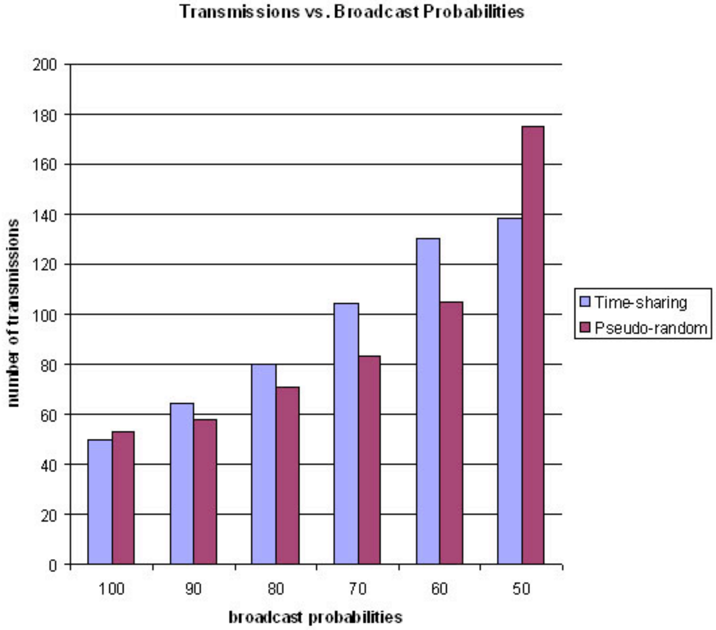


Figure 2: This graph shows total number of broadcasts at different probabilities of successful transmission.

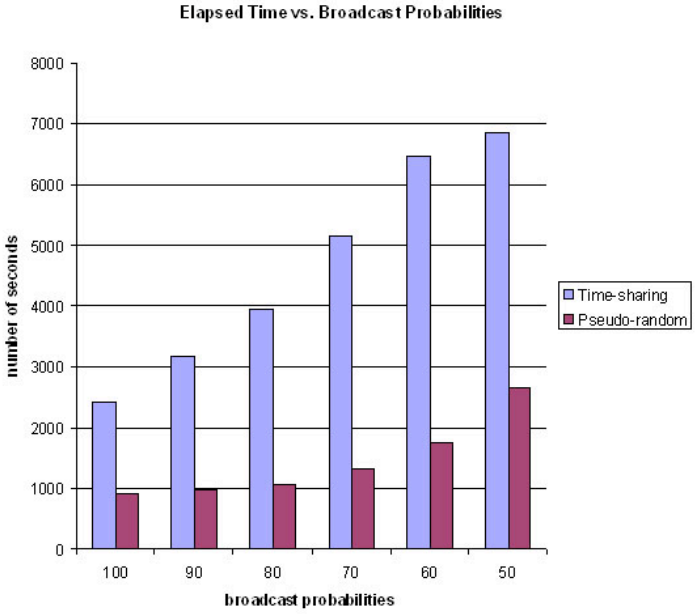


Figure 3: This graph shows total number of broadcasts vs. the number of seconds required to transmit

The time-sharing algorithm requires an average of 49% less broadcasts before completion of data propagation. This discrepancy is due to packet collisions occurring (i.e., each packet collision requires an eventual re-broadcast of the data that was not propagated). However, while the pseudo-random algorithm lacks network efficiency, it offers a substantial advantage in speed of propagation. Figure 5 shows the time elapsed toward the completion of data propagation with increasing tag quantities. With every 10 tags added to the simulation, the time-sharing algorithm exhibits an almost exponential increase in time requirements while the pseudo-random algorithm posts only modest linear increases.

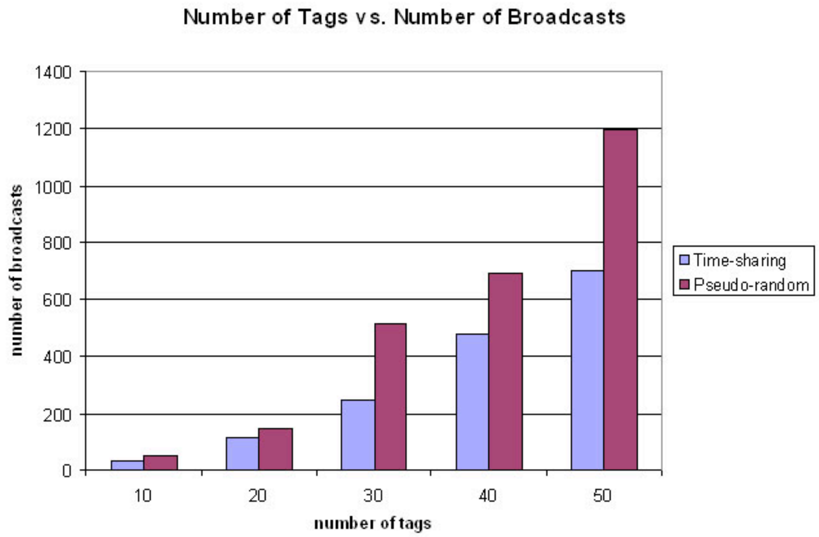


Figure 4: This graph shows the number of tags vs. the number of broadcasts.

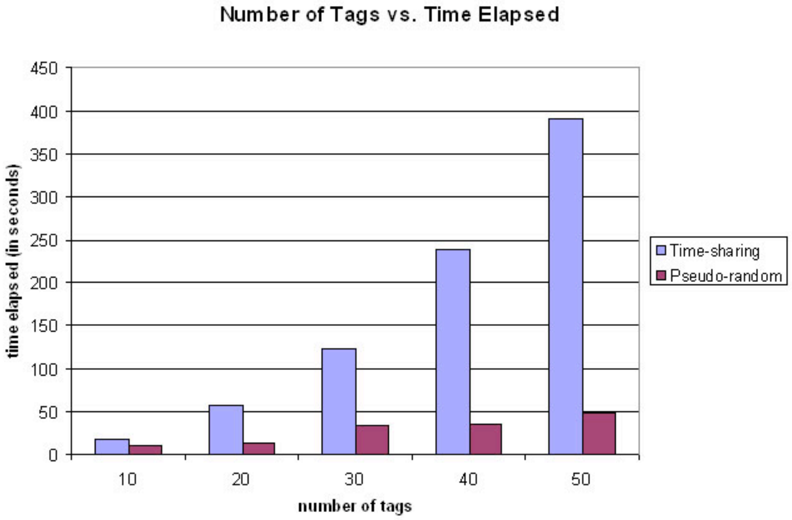


Figure 5: This graph shows the number of tags vs. the time elapsed in seconds

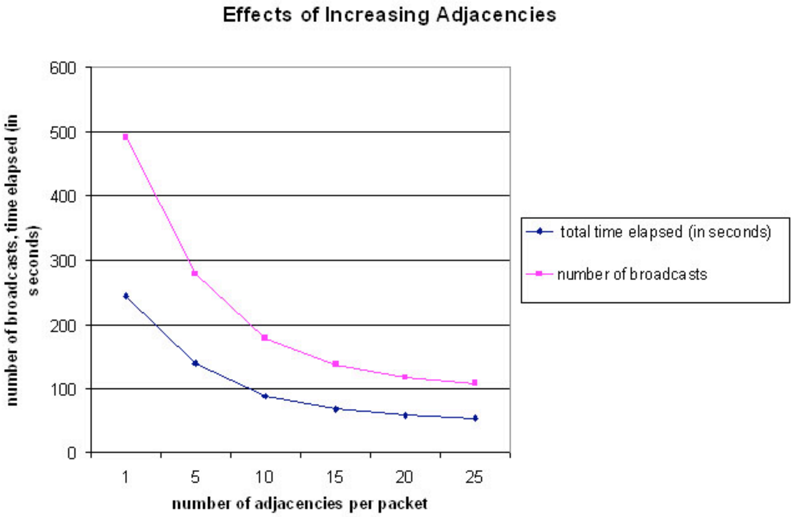
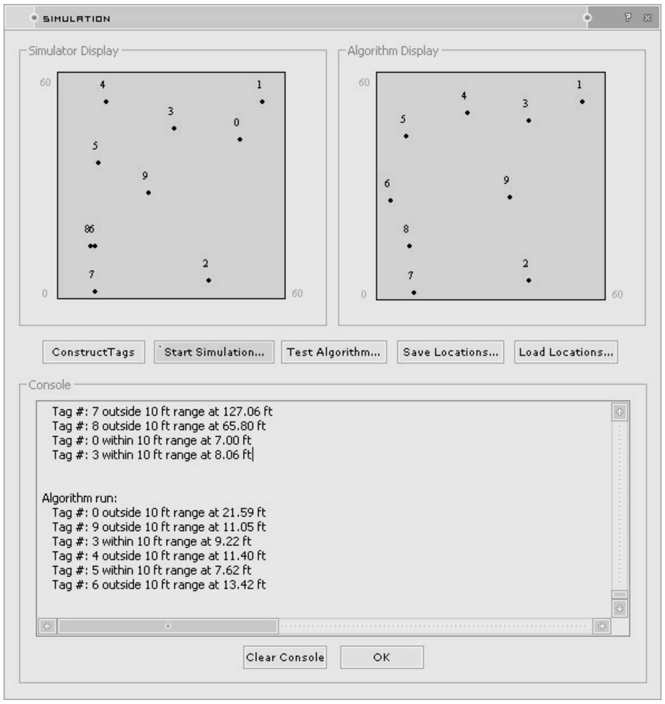


Figure 6: This graph shows the effects the number of adjacencies allowed per packet has on the number of broadcasts and the time elapsed. Note that this graph applies only to the time-sharing protocol.

Many transceivers have specific buffer limitations in which to receive packet data. It was decided to implement packet length requirements in InfoTagSim at a higher level with a number of adjacencies per packet. The objective was to investigate the effect of completion time with increasing adjacencies per run. For this experiment, 10 tags were simulated while the adjacencies per packet were increased.

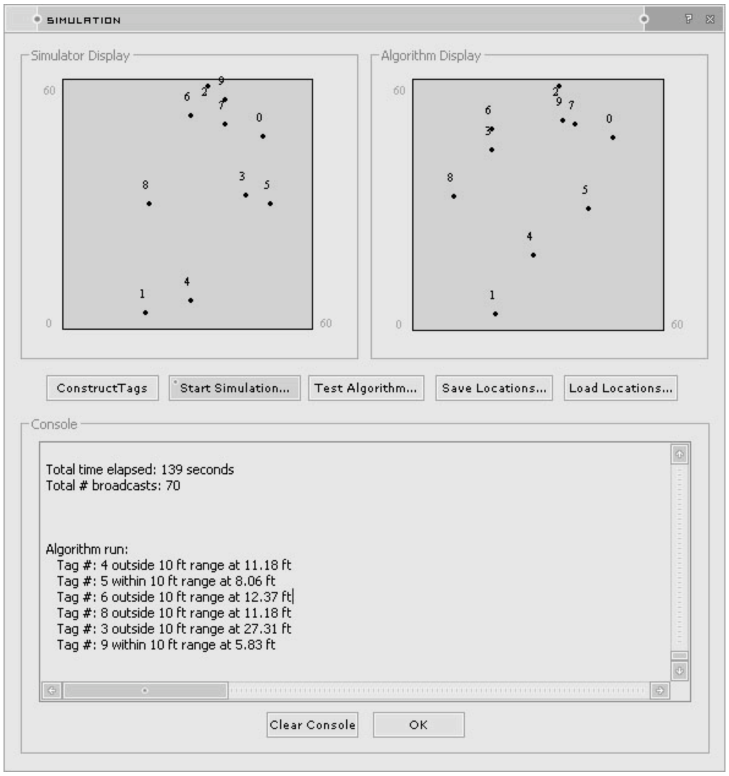


```
  Tag #: 7 outside 10 ft range at 127.06 ft
  Tag #: 8 outside 10 ft range at 65.80 ft
  Tag #: 0 within 10 ft range at 7.00 ft
  Tag #: 3 within 10 ft range at 8.06 ft


Algorithm run:
  Tag #: 0 outside 10 ft range at 21.59 ft
  Tag #: 9 outside 10 ft range at 11.05 ft
  Tag #: 3 within 10 ft range at 9.22 ft
  Tag #: 4 outside 10 ft range at 11.40 ft
  Tag #: 5 within 10 ft range at 7.62 ft
  Tag #: 6 outside 10 ft range at 13.42 ft
```

Figure 7: This screenshot showcases a run of the algorithm tester alongside the simulation results. The side-by-side comparison provides visual confirmation about the accuracy of the algorithm location estimation versus the actual simulation results.

Figure 6 shows the results of this experiment. At first, with an increasing number of adjacencies, the completion time improves considerably. However, the completion time improvement eventually tapers off. This is explained by the fact that going from 20 adjacency transmissions per packet to 25 does not make much a difference for a 10-tag system (which, at most, would have 81 adjacencies if all tags are within range of each other). This means that a tag would only require two or three transmissions before its entire adjacency table is propagated through the network.



```
Total time elapsed: 139 seconds
Total # broadcasts: 70



Algorithm run:
  Tag #: 4 outside 10 ft range at 11.18 ft
  Tag #: 5 within 10 ft range at 8.06 ft
  Tag #: 6 outside 10 ft range at 12.37 ft
  Tag #: 8 outside 10 ft range at 11.18 ft
  Tag #: 3 outside 10 ft range at 27.31 ft
  Tag #: 9 within 10 ft range at 5.83 ft
```

Figure 8: This second screenshot shows another algorithm test run after a simulation run.

As stated in the Design Goals section, it was considered essential for InfoTagSim to support the localization algorithms used to estimate node positions. Two of the position estimates from random placements of 10 tags are included in Figures 7 and 8. The visualization for the estimated positions is placed side-by-side with the map of actual geographic coordinates. The room dimensions are 60 ft by 60 ft and the adjacency set is proliferated by tags using the time-share protocol. The position estimates came out mostly accurate (i.e., an accuracy of within 10 meters of actual position in this particular set up). Most of the estimated positions came out to around 12 feet from the actual location. This suggests that the algorithm for estimating location is at least promising. However, it is recognized that the algorithm will require additional work to estimate tag positions in large scale, densely packed wireless sensor networks.

7 Related Work

Many researchers have addressed the localization problem with varying methods. Niculescu and Nath have devised a localization computational model called Ad Hoc Positioning System (APS) that uses distance vector propagation, GPS triangulation, and signal strength measurements to estimate node positions [8]. In their work, they simulate 100 tags and report results in which estimated tag positions are no less than one radio hop away from their actual geographic positions.

Bulusu, Heidemann, and Estrin use fixed beacon signals to solve localization [2]. They place beacons with known locations at the four corners of their 10m by 10m test area. These beacons transmit a signal periodically to a randomly placed tag in the area. The position of the tag is estimated after analysis of collected tag data. Their results yielded a 90% accuracy rate in estimating data points within one third of the separation distance.

He, Huang, Blum, Stankovic, and Abdelzaher propose range-free localization schemes for coarse-grained applications [5]. They propose APIT, a scheme where high-powered beacons are placed to form triangular regions around nodes with unknown locations. A theoretical method called Point-In-Triangulation is used to calculate node location within or in the proximity of the triangular region. The authors have simulated up to two-dozen tags. They promote range-free schemes as cost-effective solutions that avoid inherent problems with range-based schemes (i.e., multi-path, fading, irregular radio patterns).

Park, Savvides, and Srivastava developed SensorSim, a simulation framework for modeling and analyzing wireless sensor networks [9]. Their simulator proposal possesses similar end goals to the work reported by the authors of this paper (i.e., scalability, development of protocols, and modeling power usage). A very intriguing feature of their simulator is the support for interactions between real and simulated nodes, coined as hybrid simulation.

8 Conclusions and Future Work

In this paper, the authors have described the design and implementation of InfoTagSim, a simulator for wireless sensor networks that specializes in localization. The simulation was exercised to analyze two simple broadcast protocols, namely: a time-sharing broadcast protocol; and, a pseudo-random broadcast protocol. Various simulation parameters were tested and their effects on the system noted. In addition, the position estimates of the localization algorithm were compared with the actual geographic coordinates. The localization algorithm shows promising results on small data sets. While a significant portion of InfoTagSim was completed and tested, several shortcomings and desirable future extensions were noted, as follows:

- **Item Dimensions:** InfoTagSim currently simulates wireless networks at the node level. The simulator essentially treats nodes as equals and thus, the nodes inhabit the same amount of physical space. In future versions of InfoTags, physical cargo will map to each node. Each object assigned to a node should then contain particular item dimensions retrieved from a database of known items (used for ICODES). For example, an object specified as an Apache Helicopter should contain the particular physical dimensions of an Apache Helicopter, thus allowing the simulator to visualize the object's dimensions in the display. Factoring item dimensions into the localization algorithm can also produce a more accurate simulation and allow for a better visualization of an actual cargo hold.

In a like manner, as discussed in an earlier section, assigning space dimensions beyond that of a rectangle also improves the capabilities of the simulator. Loading up of the same Scalable Vector Graphics (SVG) files used in ICODES allows the simulator to mimic actual cargo holds and stow areas.

- **Time Synchronization**: To further the realism of the simulation, the next step requires InfoTagSim to also simulate the effects of time synchronization. In the current InfoTagSim application, nodes assume an idealized time infrastructure. Each tag runs off the same master clock, which means that time measurements do not vary from node to node. Further, no methods currently exist in InfoTagSim to implement synchronization across the nodes. In a time-sharing broadcast protocol, time synchronization becomes a crucial element in ensuring the exclusivity of broadcast rights for a given node. Drift amongst nodes in a time-sharing broadcast system could allow for crosstalk to occur, thus further delaying the completion of data gathering for localization.
- **Power Consumption**: The nature of wireless sensor networks requires much study in the area of energy efficiency. Using less energy translates to a longer useful lifetime of the sensor network. Different modes of operation of a sensor node incur different energy costs. Further, the specific type of hardware used (i.e., microcontrollers, transceivers, sensors) also dictates the amount of energy used. Eventually, InfoTagSim could be improved to perform more robust metrics in respect to those factors that affect power consumption and thus possess the capability to support more elaborate network protocols in which, for instance, the tags enter a sleep state after receiving no new packets (i.e., an indication that all cargo items have settled into a stationary position). The current version of InfoTagSim offers only a few primitive metrics to indirectly account for power consumption, namely: total transmissions; average number of node transmissions; and, elapsed time before completion of localization data-gathering tasks.
- **Received Signal Strength Indicator**: Many RF transceivers possess the ability to measure received signal strength. This mostly, but not always, serves as a crude way to estimate the distance between two connected nodes. The reception of a packet of weak signal strength indicates a longer distance connection of two nodes than the reception of a packet exhibiting a strong signal strength. With localization as a primary goal, one can design elaborate network protocols that account for received signal strength. InfoTagSim, which currently uses a binary outcome (i.e.,

connected or not connected) method to detect neighboring nodes, could benefit greatly from received signal strength functionality. Allowing for more elaborate broadcast protocols that utilize received signal strength could further optimize localization techniques and thereby improve the accuracy of the position estimates.

It should be noted that in order for received signal strength to be effective in localization methods, the experimenter must have knowledge of the correspondence of signal strength values to set distances (i.e., for the specific RF hardware used). This correspondence of values to distances requires new simulator parameters within InfoTagSim falling under the category of hardware characteristics. Still, the reliability of this correspondence can depend on objects and barriers that can alter or inhibit transmission signal radiation patterns, which may consequently provide false signal strength data to the receiving node. Also, InfoTagSim must assume a high level of consistency in both transmission power and received signal strength across all network nodes. In other words, tags must be calibrated uniformly so that received signal strength values correspond to set distances regardless of the transmitting or receiving node.

The authors expect that the aforementioned future improvements will increase the robustness of the simulation and increase InfoTagSim's value as a tool for the accurate analysis of new wireless sensor network designs that specialize in localization.

9 Acknowledgements

The authors would like to thank Pat Wong of CDM Technologies, who provided the vision and support for this undertaking. The authors would also like to thank Dr. Michael Haungs, Dr. Jens Pohl, and Professor Art Chapman for their input, guidance, and support.

10 References

[1] B. Hofmann-Wellenho, H. L., and J.Collins. *Global Positioning System: Theory and Practice*, fourth ed. Springer Verlag, 1997.

[2] Bulusu, N., Heidemann, J., and Estrin, D. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine 7*, 5 (October 2000), 8_34.

[3] CADRC. Cadrc Currents Fall 2005. http://www.cadrc.calpoly.edu/pdf/Currents_Fall2005.pdf, 2005.

[4] CDM Technologies. ICODES Website. http://www.cdmtech.com/web/guest/products/icodes, 2006.

[5] He, T., Huang, C., Blum, B. M., Stankovic, J. A., and Abdelzaher, T. Range-free localization schemes for large scale sensor networks. In *MobiCom '03: Proceedings*

of the 9th annual international conference on Mobile computing and networking (New York, NY, USA, 2003), ACM Press, pp. 81_95.

[6] L. Doherty, L. E. G., and Pister, S. J. Convex position estimation in wireless sensor networks. In *IEEE Infocom* (April 2001), vol. 3, pp. 1655_1663.

[7] Maxstream. *RF Radiation Patterns in Maxstream XBee Transceivers*. http://www.maxstream.net/support/knowledgebase/_les/XSTAN019a_XBeeAntennas.pdf.

[8] Niculescu, D., and Nath, B. Ad hoc positioning system (aps) using aoa, 2003.

[9] Park, S., Savvides, A., and Srivastava, M. B. Sensorsim: a simulation framework for sensor networks. In *MSWIM '00: Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems* (New York, NY, USA, 2000), ACM Press, pp. 104_111.

[10] Pohl, J. *ICODES: Ship loading with service-agents*. http://www.cadrc.calpoly.edu/pdf/total_cadru/chapter%206.pdf, 1997.

[11] Savi. Savi to develop RFID tracking for cargo containers. http://sanjose.bizjournals.com/sanjose/stories/2005/04/18/daily37.html, 2005.

[12] Savvides, A., Han, C.-C., and Strivastava, M. B. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking* (New York, NY, USA, 2001), ACM Press, pp. 166_179.

Collaborative Matrix Control, Monitoring and Management Systems for Infocyber Channels, Gates and Tracks

John F. Halldane

Environmental Systems Consultancy. USA

Overview :

CADRC presents a need for an intelligent internet and alternatives for vertical stovepipe systems and horizontal coded data exchange protocols. A matrix management system of inline and crossline management intersects in nodes for monitoring and control. This was outlined in 2002 for decentralising homeland security. Then we discussed "cues" for sensing un-usual activity in security contingency planning at those nodes in 2003. Shells of security were identified in terms of the product, process or context being impaired, impeded or threatened. We now discuss the the issues and solutions in applying matrix systems in overlays for a responsive intelligent internet.

Duplicate independent "channels" are needed for "intelligent infocyber" at specific nodes with independent sensing and protocols. This is for reasons of security, authorization, monitoring, automation, proprietary interest, equipment warranty, insurance, safety, fault detection, verification, confirmation, "truth" of infocyber, censorship, legal litigation, inbreeding and a delegation of responsibility for the systems to work. For this, channels must use separate "tracks" of duplicated overlays that "couple" at appropriate nodes. Channels may be in any mode including internet webs, e-mail, fax, voice over, and telephone, radio, gps, tv, cable, reports, public warning,... to verify infocyber activity. The nodes have "dedicated couplers" for sensing and observing specified cues in any form of appropriate coupling with "tracks". This avoids the problems of exchange protocols and hardwiring. We should look beyond the internet alone towards independent interworking "collaborative channels" which can both verify and confirm the truths of infocyber. In turn, "gates" at infocyber nodes form collaborative matrix control, monitoring and management systems. We introduce "psychophysical" and "cultural" protocol concepts for the hard, remote and perceptual exchange at gates.
Examples are provided from experience in developing Automated Integrated Monitoring Systems, AIMS, for facility management.

Protocol Issues and Infocyber Exchange :

Protocols are the conventions governing the **exchange of informatic and cybernetic flows or "infocyber"**. They are **agreed ways and codes to communicate** between authorized parties in secure connections in **any mode** for a very specific purpose.

Experience with simultaneous automated control and monitoring of building services equipment in facilities management requires systems to be independently coupled. Let me cite some examples which I believe will illustrate the same general issues for the infocyber universe.

a) With an absorption chiller we wanted to stage the regenerator temperature to interface with solar collectors and diesel cogeneration. The chiller manufacturer agreed, but they would not provide the warranty because the automatic control for the the auxiliary equipment could override the chiller control which might compromise the efficiency of the manufacturer's components. The lesson; **each manufacturer's equipment must stand alone and operated according to their specifications**.

b) Parallel wiring of sensors between the automatic control system of a chiller and an automated energy monitoring system would avoid duplicating the sensors. This required sharing the manufacturer's proprietary protocol to exchange the sensor output. However, suddenly the manufacturer changed the protocol so we had no access to the infocyber. It became a legal proprietary issue because our monitoring could also trace the profile of equipment failure in evidence for the owner. Consequently we had to duplicate some 20 sensors on each chiller for our independent energy Automated Integrated Monitoring-Management System, AIMS, for monitoring, control and management (Chiang and Kandiah 1999). **Control and monitoring systems must be independent in separate channels and never be coupled nor share exchange protocols**.

c) A building was remotely monitored realtime over a telephone line interface. In a demonstration I noted the chiller current was very erratic indicating it was failing. The plant engineer was telephoned to shut it down. This saved a quarter million in repairs which was greater than the cost of the monitoring equipment. A management "expert system" is needed in a **parallel channel to independently scan for "un-usual" infocyber** and to activate appropriate action automatically.

d) We considered operating an oil refinery remotely in a foreign country over the internet. However, message stacking, busy connections, security issues, foreign protocols and uncertain link performance precluded that idea. With **dedicated channels in other modes** it might be possible, such as with cable, radio or satellite.

e) Cast aluminium blades of a 40kW 1.5m dia axial exhaust fan broke after a week of use. A government lab misinterpretted xray tests suggesting poor casting by the manufacturer. Independently I had a bending test done on the blade which verified bending failure rather than casting. Also, in their factory I ran a fan under various asymmetrical load conditions, measured the air velocity vectors, current-voltage and more significantly traced the current fluctuations on an oscilloscope. The installation contractor had slung the fan too low under the beams which made the bends too sharp into the fan. This loaded the fan asymmetrically and subjected each blade to a 20c/s pulsing load equivalent to a metre high stack of A4 paper at the tip. Fatigue failure with this load is about a week for aluminium castings. The sinusoidal current trace had a 10% harmonic over it with a frequency corresponding to the number of blades; this I used to estimate the bending forces on the blades. It is important to **verify and confirm fault sequence histories through a variety of independent channels**.

f) A highbay, high intensity discharge HID lighting fixture caught fire and the installer replaced the whole installation to appease the client. To prevent further litigation I traced the fault to a capacitor and broke it apart for inspection. There were dry joints from poor soldering which would overheat the junction of the conductors and cause the fire. This is a component fault by a separate manufacturer. Now who is responsible? A component manufacturer has no idea how their component is to be used, a fixture designer has no control of the component reliability, the installer only has responsibility for it working safely, the insurance companies taking on the probable risks, an operator using the equipment inappropriately, an owner failing to observe maintenance and warranty provisions, a fire detection system, sprinkler activation,.. or do we blame the acts of God? In this case a poorly manufactured component is a probable event for both component and fixture manufacturer and they should allow for this. The fire is also a predictable probable event in terms of the facility where the risks are covered by property insurance. In contingency planning our heuristic models depend on discovering chance sporadic events to prepare for the possible "what if" situations. In emergency planning our probability models predict probable events from known likely situations (Halldane 2002). It suggests that **all parties share proportional responsibility for contingent events**, but **designated parties should share the delegated responsibilities for probable events**.

g) An hermetically sealed motor of a chiller compressor exploded in the plantroom of a Singapore mall. It happened a month after the warranty expiration in true Murphy law fashion. Moisture had entered the refrigerant which also cools the motor then shorted the electrical windings. Tests showed the shell tubes were pitted with corrosion. As a consultant I traced the fault sequence. As part of an energy conserving contract the chiller had been replaced with a seemingly more efficient American chiller at a discounted cost. Unfortunately it was oversized for the job and ran under partial load with poor performance. This ruptured the shaft seals to allow air moisture into the system. The paper cartridge moisture trap was supposed to be changed every six months but this was never done because the pipes were too difficult to uncouple for maintenance. So this broke the warranty for the chiller. The tube corrosion was likely from nickel impurities in the copper which is banned in the US. Manufacturers often dump their banned stock overseas in unsuspecting markets. The chiller performance never saved energy which invalidated the energy contract but the contract had no clause for failure nor to restore it to the original condition, so the owner still had to pay for repairs. Facility management is only responsible from when systems become operational and not for the design. What a mess of multiple party responsibility, litigation and a delegation of claims. However, the fault sequences were probable and predictable but simply not accepted by the parties. A lesson for us is that the **vested interests of the parties involved must be identified and independently monitored** by a **responsible embracing authority to plan-incorporate-finance-design-build-operate-manage each project**; a "pifdbom". This goes beyond the customary bounds of responsible management and ownership.

Intelligence as Truth in a Thinking System :

Intelligence can be considered as "truth" process to interpret meaningful infocyber, to "think" within certain situations, to learn a "culture", then to respond appropriately in civilized ways. It seems the internet is at a culture building stage with libraries of reposited infocyber in catalogues of situations. There is a huge question of what the internet should learn and what to divulge, between certain parties, within specific products, processes or contexts? These are the broader issues of censorship and security. Even in lawful democratic societies the legal constraints need interpretation and are often ambiguous, contradictory or unknown. Also **law is based on past precedent probable situations which makes it ineffectual and exceedingly slow in anticipating future contingent scenarios**. Law is bound by the constitutions of a country, state and community for its authority. Our system models should anticipate the consequences of infocyber in behavioral actions and cognitive thinking. Those models should distinguish the **truths of infocyber**, be more **heuristic**, **learn the cultural opportunities and constraints**, then **form channels for appropriate civil responses**.

The danger of these learning systems is "**inbreeding**" where detrimental legends and fallacies are perpetuated as a culture within the system. It is often a product of inline dominant single channel infocyber flow. We have this in our education system where teachers teach teachers to teach **in order to perpetuate itself for its own sake**. Thus the consequences of what is taught needs to be coupled to its infocyber so that it can be modified for the real professional world, then appropriately applied for the benefit of our communities. This is why I have an applied professional approach to teaching mathematics and physics rather than a traditional purely academic one. Societies, associations, institutions, clubs, religious groups, the law, ... thrive on their self-learning systems. Truths within those organizations become their beliefs. Problems arise when those beliefs are misapplied outside the products, processes and contexts of what they do. **Truths of each infocyber channel should be independent yet accountable within the broader context of related channels they could be coupled to**.

Truth leads to its own consequence. Consequences can be observed, sensed and recognized as fact. So **truth is a test or an evaluation of something** to determine whether that something exists and performs as anticipated **for its own good**. In our case it is the usefulness of the combined infocyber, channels and parties involved. Does the use of such a system perform as intended for its benefit? Please note that we do not need a contrary argument such as untrue, fiction, false or failure since those mechanisms can lead to quite different consequences away from the truth we are testing. We should avoid circular logic, circumlocution or simply "beating about the bush". Perhaps the **worst case is the null hypothesis in statistics** where you disprove a thing is by chance in order to accept a contrary alternative hypothesis as true. Then they go on to say the logic does not prove a causal relationship anyway! It is far **easier to reason in statistics that two things can be the same according to an agreed sameness criterion**, so now we do not need to discuss different or alternative scenarios. (Halldane 1989) Also I dislike double negatives; for example in neurophysiology, what is "not uninhibited"?

We can think of **truth as a gradation** in "shades of truth" from none, through a threshold to trueness as a self-evident observation of the consequences. This **avoids semantic differentials** with opposing antithetical true-false statements that psychologists like to use. Many questions do not have an opposite argument. So we could ask how much is the truthfulness in accordance with the facts and how reliable or conforming is the truth in a given circumstance.

To Verify and Confirm the Shades of Truth :

Testing for truth may be by induction or deduction in a traditional logic of premise-argument-conclusion. With induction we believe the conclusion is true by a reasonableness of its premise within the context of its conclusion. This leads to heuristic models for contingent operations and in scientific discovery. With deduction we accept the premise and believe the argument is true which then predicts a probable conclusion. This is the basis of all our predictive models in design. In turn the logic is used to **both verify and confirm the truth of consequences.**

To verify a flow of infocyber we test the input-output with knowns to see if the product and processing is correct. Basically we **verify a single channel is true by a deductive test within that system**. We click on a website or link, the internet-computer deduces a result and up comes a page on the screen; the consequence is assumed to be true. **Verification checks whether the operation works but not the meaning or content** of that infocyber.

To confirm a flow of infocyber we **test the content for its meaningfulness** as a product, then within its process and context. Fundamentally we confirm **together with other related independent multiple channels that the consequence is true by inductive and or deductive tests coupled to the original channel**. This leads us to tracking "tracks" which are far more demanding and responsive.

Responsive Thinking Systems :

The recent (Dec 2004) tsunami in the Indian Ocean is an example of our desperate need to verify and confirm the shades of truth in a responsive thinking system. The cyber-data indicated a deep sea earthquake over 4.5 on the Richter Scale. Earthquake and sea wave info-data followed through various communication channels from Banda Aceh in northern Sumatra near the epicenter. About an hour later (450km,280mile) at Phuket in Thailand the tide receded followed by huge sea waves. Later there were news reports of devastation and survivor experiences from all around the coasts of the Indian Ocean.

Tsunami are more common in the Pacific Ocean and locations in Japan are far more prepared with international warning systems, evacuation plans and in some cases wave walls. In **verifying infocyber** the question is whether the ocean floor movement will lead to a displacement sufficient to create the deep pressure wave and whether coastal tides will recede then build excessively in sufficient time for any appropriate response. **Realtime cyber-data could be available** from seismographs, deep sea pressure gages and tide level gages. But these would need to be linked in **realtime channels to some "thinking center" to verify, confirm and respond** accordingly. In **confirming infocyber we need fast independent sightings** of coastal quakes, excessive tide recessions and coastal wave formation. This has often been done through local telephone, radio stations, mobile TV and amateur "ham" radio communications. The most important point is the **consequence of these truths**. What happens when a tsunami of such speed, height and volume of water affects a coastal community. Communities must be prepared for probable emergencies and likely contingencies. They must know the local signs of trouble and take appropriate action. Now monsoons, typhoons, hurricanes, storms, flooding, tornadoes, drought, fires,... are probable, predictable almost annual events. Yet communities **rebuild in the same old way for the same devastating consequences**. **Insurance encourages this** as their concept is only to reinstate rather than to improve (Halldane 2003,p33).

Consequences need consideration as a continuum in time. The **resulting havoc often suggests a redesign or elimination of system components**. In a cyber world viruses lead to antiviruses and firewalls or to choose systems that are less vulnerable. Links with their exchange protocols, for instance, are very vulnerable so perhaps exchanges could be replaced with referals to uncouple the links. In the tsunami case I have planned and designed "freeway with town enterprises". Here the freeway is bunded to save the mangroves that protect the coastline, it avoids coastal flooding by the daily tides and now it may mitigate some of the wave damage (Halldane and Khan 2001). Sidesway design of buildings helps both in earthquake, wind and landslope design. We can economize our **design by combining the consequences of various likely events**.

The "thinking" parts of our systems have often been in parallel independent people-thinking channels. If an automated system goes wrong then people are brought in to diagnose and correct the faults. We also have series dependent people-interactive channels highlighted in video games that anticipate or react to moves. There are smart expert systems preprogramed to verify and react to infocyber. Our future **intelligent internet needs also to confirm infocyber** as part of the thinking process. To do this it **must go outside of itself to other independent channels** in order to confirm its truth. Now that is a real challenge especially for news channels with their duplicated links, inbred traditions, the on-the-scene correspondent, spontaneous opinions, and marketing hype. Generally the sides of issues are presented but rarely resolved, possibly because of the entertaining variety of consequences within such a diversity of contexts. So we would **need to specify both consequence and context** for our thinking intelligent channels to have truthful infocyber.

Collaborative Matrix Control, Monitoring and Management :
Gates and Tracks : Psychophysical and Cultural Protocols :

Collaboration is the act of people with their systems working together for the benefit of the parties involved. The Internet is not owned, patented nor copywrited; so who is responsible for it working? Networking was created by the US military for their communications. It was then opened to universities and developed under the National Science Foundation who in 1995 privatized it to create the World Wide Web, www, protocols. Voluntary groups of non-profit member based organizations such as the Internet Society, ISOC, and the International Corporation of Assigned Names and Numbers, ICANN, consolidated standards. The US Dept Commerce in 1998 recognized ICANN which led to Regional Internet Registries for domain names in the familiar .com .org .net ... Public network access is now through Internet Service Providers, ISP, of private companies or government regulated monopolies in various countries. The internet can be in any mode from telephone, cable to radio in order to interface with connected computers which respond through a Transmission Control Protocol / Internet Protocol, TCP / IP, and File Transfer Protocol, FTP, Hyper Text Transfer Protocol, http, and Uniform Resource Locators, URL, all in open software. University networking is being preserved through Internet 2.

Security, censorship and personal information are key areas for concern. We have found that **voluntary member organizations**, such as in societies like the medical and engineering professions, **do not police themselves well according to their ethics**. It is an inbreeding problem where members are unwilling to sanction other members and their program sponsors. Identity theft, junk mail, spam, violent games and pornography are rampant. A Communications Decency Act 1996 and a Children's Protection Act 1998 failed to pass in the US because of constitutional "freedom" law. The Homeland Security Act 2002 allows the ISP to reveal subscriber information and trace e-mails to government agents. Already there are cases of ISP selling subscriber information to credit and marketing groups. A legal approach tends to be after-the-fact, post-corrective and constraining for business.

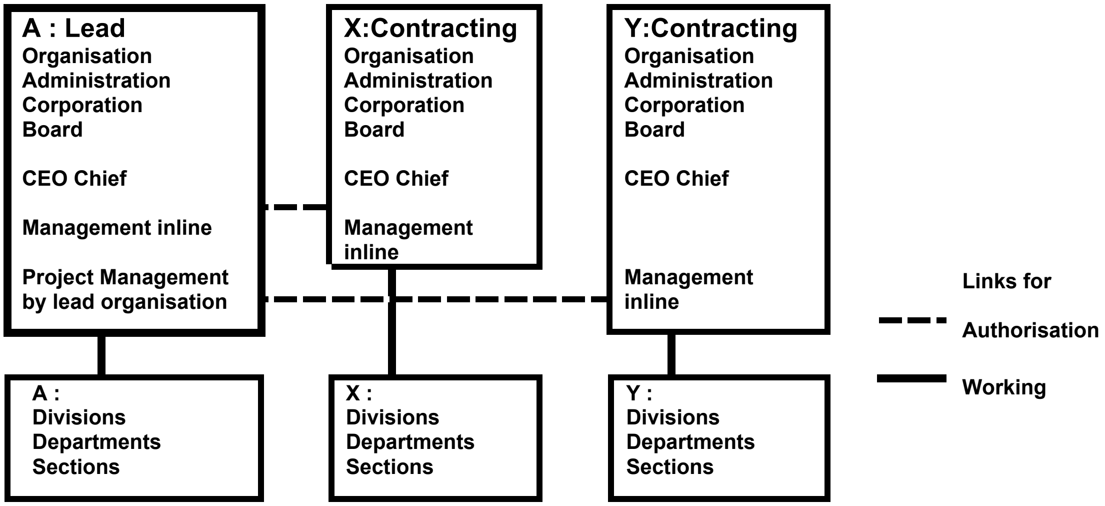
Governance is a government authorized management approach to control the networking of computer uses. Many liken this to censorship or big brother looking over your shoulder or an invasion of personal freedoms. Unfortunately most of our governance models are slow legalistic authoritarian bureaucratic inline management promulgated by lawyers. It is the vertical stovepipe system CADRC wants to avoid.

Collaborative matrix approaches offer **integrated combinations of "inline" organization with "crossline" project management principles** intersecting in **coupled nodes for infocyber control, monitoring and management**. Issues and solutions are set out in the Management Chart (Halldane 2002). This is readily modified to a **Lead Collaborative** with specific **Programs**, possibly according the domains of use and levels of security, with authorized **Nodes** in specified coupled protocols. The objective of each **nodal coupling would depend on its consequence in the control, monitoring and management** of the systems involved.

A **Lead Collaborative** may be through an international oversight commission with technology, country, provider and user representation and control. The goals would be to create and maintain an effective, secure working system within its authorized programs. There is an ethical responsibility for an intelligent internet to encourage truthful, wholesome infocyber by promulgating ethical standards and by monitoring the consequences. But an internet is only one part of a communication system and indeed we have found other independent channels are necessary to confirm truth. So our **collaborative matrix needs to provide the product, process and context for assessing truth**. A liberal legal approach favors an "anything goes", "caveat emptor", let the buyer beware attitude. Here the parties bear no responsibility unless they agree to a guarantee of performance in some warranty. Now a **systems designer can only work from opportunities** that can and should be done, rather than from the constraints of what could or should not be done.

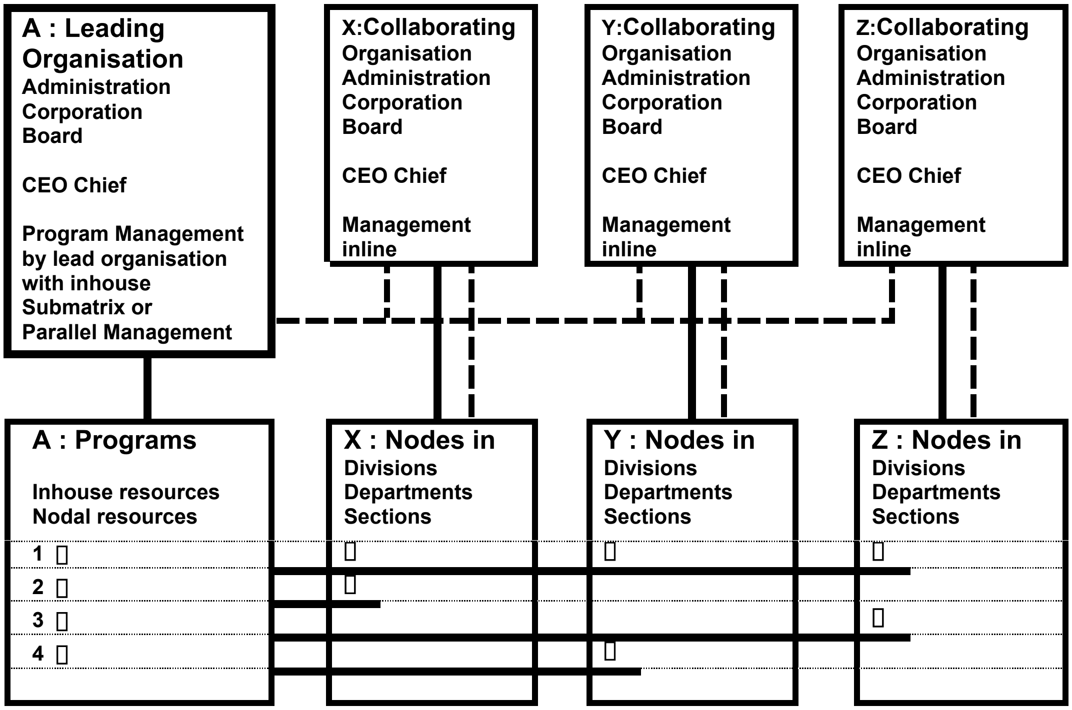
Gates are the type of coupling for Node Control. **Hard coupled gates** link dependent channels with shared protocol exchanges to replicate infocyber. **Remote gates** observe the physical channel activity in independent channels, which is characteristic of capacitively or inductively coupled detectors and remote measuring instruments. **Perceptual** gates rely on channels for people's sensory and cognitive observations, especially those of the consequences.

" Inline" Management of Conventional Organisations . . . . .



Issues : With "inline" the upper management executive instigates contracts with other organisations and authorises actions in a top-down way. At a working level within the divisions of those organisations there is a reiterative brainstorming for alternative solutions but they can not get their heads together because of the bottleneck of communications upline and over to the lead organisation. Final decisions should only go upline because upper management does not have the time, skills, nor resources at that level to coordinate an overflow of information needed in decision-making. Further, at working levels ideas are changing from time to time which inline managers are unable to tolerate.

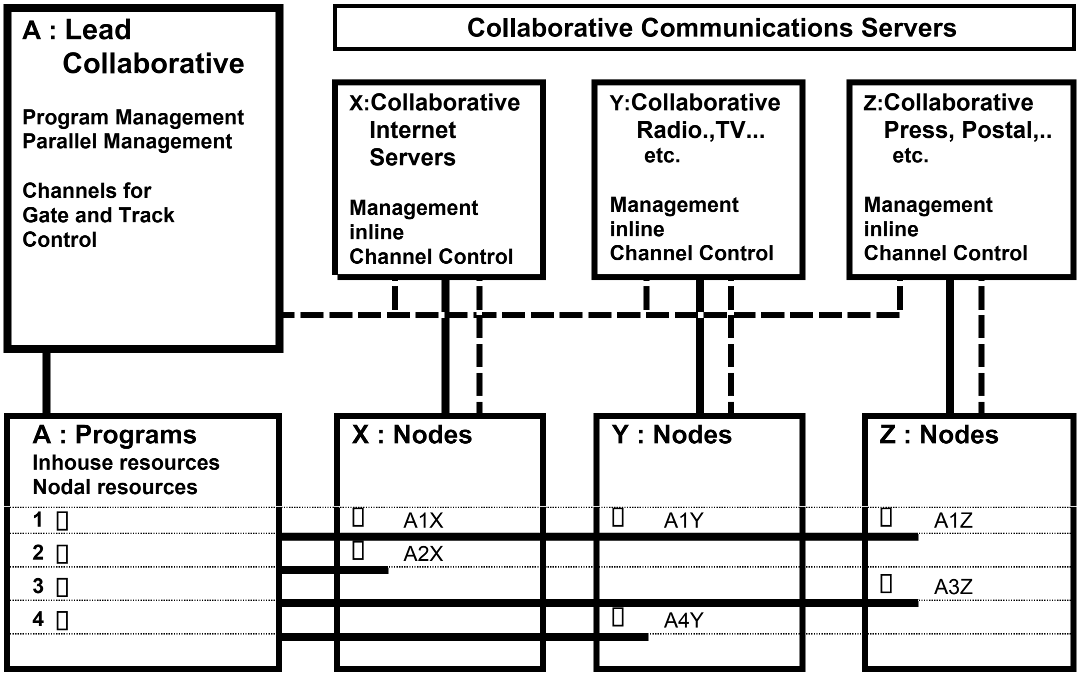
" Crossline" Matrix Management for Programs and Projects . . . . .



Solutions: Matrix Program Management, authorised by upper management executives, identifies the interdisciplinary team to work with, the Nodal Managers in each organisation, arrangements for inhouse duties and time, contracts and allocation of resources by all parties. Crossline Managers coordinate activity, brainstorm ideas, take initiative action, improvise where necessary, undertake investigations, testing, feasibility analysis, report up their lines with final decisions or recommendations for endorsement and record. The Program and Nodal Management Team must be competent, capable, enthusiastic,... acting with trusted authority from all the organisations involved. This way the lead organisation becomes an umbrella for program managers with all the resources decentralised in an effective collaborative administrative matrix and communications network.

From: Homeland security through a decentralised matrix management. John F. Halldane. 4th Annual ONR Workshop on Collaborative Design-Support Systems. Office of Naval Research. Virgina. Sept. 18-19, 2002 JFHalldane 1999, 2002

Matrix for Collaborative Control, Monitoring and Management



Security Shell Criteria. . . *for protection and vulnerability*

Shell	***A***	**Impaired Product :**	**Irreversible damage . hazardous condition . destroyed .**	**Probability : . . .Models**
Shell	***B***	**Impaired Process :**	**Irreversible stoppage . vitals inactive . resources out . facility closure .**	
Shell	***C***	**Impaired Context :**	**Irreversible damage to supporting context and collaboration . invasion .**	
Shell	***D***	**Impeded Product :**	**Reversible damage . strikes . delays .**	
Shell	***E***	**Impeded Process :**	**Reversible stoppage . breakdowns . walkouts . sabotage . supply faults . bypass components .**	
Shell	***F***	**Impeded Context :**	**Reversible damage to supporting context and collaboration . demonstrations . utility interruption**	
Shell	***G***	**Threatened Product :**	**Contingency to product impairment, impediment boycott . weather conditions .**	**Heuristic : :**
Shell	***H***	**Threatened Process :**	**Contingency to process impairment, impediment . union dispute . sabotage . maintenance . safety .**	
Shell	***I***	**Threatened Context :**	**Contingency to context impairment, impediment war . depression .**	

Source: John F Halldane "Security Contingency Planning Matrix" 15th InterSymp 2003 p31.

Hard gates share and verify infocyber through exchange protocols. Remote and perceptual gates are usually in different modes of communication from that of an internet channel so they are valuable in confirming infocyber through the consequences of user activity. The lead collaboratives could be **authorized to control these gates** in order to **detect un-usual activity** of a channel and to respond appropriately according to the suggested **Security Shell Criteria** that threatens, impedes or impairs the product, process or context of the consequences.

Tracks are independent channels for tracking infocyber in monitoring a product, process or context. An analogy is in tidal river channels with tracks beside and between those channels. A person walking a track may observe a saltwater backflow or bore and alert local officials by telephone of the un-usual phenomenon. Our lesson here is that observers must be **aware and knowledgable of the un-usual cues** in particular situations and are then **capable of communicating those observations** to modify the consequences. It brings us back to the "truth" and "culture" for intelligent thinking systems. Tracks also need to work through gates for nodal control. What are the protocols to do this? Hard and remote coupled gates have a variety of software for those shared protocol exchanges and are indeed the theme for this conference.

However, a growing concern is the **"psychophysical" exchange** between **non-physical cognitive thinking minds and their physical sensory behavioral environment in tracking infocyber**. A previous paper (Ref#2) outlined the thresholds, performance, impedance and impairment of our perceptions in tracking un-usual sensory cues. Now I believe we need to work on the **cognitive and behavioral consequences within acceptable cultural protocols for world wide communications**. Such cultural protocols should use their diversity of expression rather be confined to some world standard... so long as those **cultural protocols are compatible and collaborative**. A matrix approach is a way to organise such a mission.

Acknowledgement :

It is my pleasure to have the opportunity to contribute to the Focus Symposium on Collaborative Decision-Support Systems through the Collaborative Agent Design Research Center in San Luis Obispo, California.

I truly thank my colleague Dr Jens Pohl of CADRC for this participation. With sorrow we reflect the passing of our colleague Prof. Day Ding a year ago. He brought me into Architectural Science at Sydney which opened many a world wide adventure.

References :

Chiang, A. and Kandiah K. (1999); **Using Automated Integrated Monitoring Systems (AIMS) Extending the Building Management System (BMS)**. ISI-SEACEUM-5,1999, Paper 13, p317-358. Describes the real time "Electric Eye" system by Supersymmetry and four energy saving case studies..

Halldane, J.F. (2002); **Homeland security through a decentralised matrix management.** Corollary to 4th Annual ONR Workshop on Collaborative Design-Support Systems. Office of Naval Research. Hosted by CalPoly San Luis Obispo, Collaborative Agent Design Research Center. Quantico. Virgina. Sept. 18-19, 2002 Issues of threat, breach, reaction, recovery, countermeasures. Contingency planning anticipating with chance heuristic sporadic models. Emergency planning with after-the-fact probability models. Matrix management in across the inline in rows of parallel programs.

Halldane, J.F. (2003); **Security Contingency Planning Matrix: Cues . perception. psychophysical systems . performance criteria . surety .... for life-quality and environment.** Inter Symp 2003. 15 International Conference on: Systems Research, Informatics and Cybernetics: Focus Symposium. Collaborative Decision-Support Systems. July 29 2003 . Baden-Baden. Germany. July 8 - August 2, 2003

Halldane, J.F. (1997); **Facility management systems for new generations of highrise: Coping with change and simplifying statistics**. ISI-CTBUH-TBM 2000 Paper 37b.. Conquest of vertical Space in the 21st Century, Int. Fed. Highrise Structures, Council for Tall Buildings and Urban Habitat. The Concrete Society, UK, The Paragon Hotel, London. 8-10 Oct 1997. Facility management systems, automated integrated monitoring systems, building reviews, simplified statictics for new maintenance patterns and utility energy management.

Halldane, J.F. (1989); **Design integration for minimal energy and cost**. Handbook. Elsevier Applied Science Publishers, UK, June 1989. Includes the sameness criterion and the middle-third concept for statistics.

Halldane, J.F. and Khan, M. (2001); **Freeway Enterprise Development: Demonstration of sustainable development.** Regional Seminar on Sustainable Urban Planning and Environmental Management in ASEAN. HSF,ROAP,UNEP,UI. Hotel Bumi Karsa Bidakara, Jakarta. 7-9 Nov 2001. Addresses Freeway with Towns development and the building of bunds to hold back tidal waters and to save the mangroved coast. Background in ghost town issues.

Halldane, J.F. (2006); **Collaborative Matrix Control, Monitoring and Management Systems for Infocyber Channels, Gates and Tracks.** John F. Halldane. Inter Symp 2006. 18 International Conference on: Systems Research, Informatics and Cybernetics: Advances in Intelligent Software Systems . Baden-Baden. Germany. August 8, 2006 .

The Theory of SK-Languages as a Powerful Framework for Constructing Formal Languages for Electronic Business Communication

V.A. Fomichov

Faculty of Applied Mathematics,
Moscow State Institute of Electronics and Mathematics (Technical University),
109028 Moscow, Russia
and Department of Information Technologies
K.E.Tsiolkovsky Russian State Technological University - "MATI",
121552 Moscow, Russia
vdrfom@aha.ru

Abstract

The subject of the paper is a new class of formal languages called standard knowledge languages (SK-languages) as a powerful and flexible tool for the realization of electronic business communication, in particular, for building contracts concluded by computer intelligent agents and representing contents of arbitrary e-negotiations. The definition of SK-languages is a part of a mathematical model describing a system consisting of such 10 operations on structured meanings (SMs) of natural language texts (NL-texts) that, using primitive conceptual items as "blocks", it is possible to build SMs of, probably, arbitrary NL-texts. This means that a class of languages is determined being convenient for building semantic descriptions of arbitrary goods, services, and contracts. The principal advantages of SK-languages in comparison with Discourse Representation Theory, Theory of Conceptual Graphs, and Episodic Logic concern representing complicated goals and destinations of things, definitions of concepts, compound definitions of sets, and meanings of discourses with the references to the meaning of a phrase or larger part of discourse.

Keywords

e-commerce; electronic contracting; e-negotiations; natural language processing; semantic representation; standard K-languages

Introduction

During last years, the field of electronic commerce (e-commerce) has been attracting the attention of many researchers in the world. It is underlined in (Hasselbring and Weigand 2001) that if the messages in the filed of e-commerce are to be processed automatically, the meaning must be formalized This opinion coincides with the idea stated in the paper (Kimbrough and

Moore 1997) about the necessity off developing logical-semantic foundations of constructing formal languages for business communication.

In the field of e-commerce two interrelated subfields of researches have emerged called e-negotiations and electronic contracting. The collection of central problems faced by the researchers in these subfields includes the creation of formal languages for constructing records of commercial e-negotiations carried out by computer intelligent agents and forming contracts concluded in the course of such negotiations. These problems can be considered as important particular cases of the general problem of constructing formal languages for business communication.

The analysis shows that the records of commercial negotiations and contracts potentially can be formed with the help of expressive means of natural language (NL) used for the construction of arbitrary NL-texts pertaining to medicine, technology, law, etc.

In particular, the texts from such documents may include: (a) questions with interrogative words; (b) questions with the answer "Yes" or No"; (c) infinitives with dependent words expressing goals ("to sell 30 boxes with apples") or destinations of things; (d) constructions formed out of infinitives with dependent words by means of the logical connectives "and", "or", "not" and expressing compound designations of goals or destinations of things;
(e) complicated designations of sets ("a consignment consisting of 50 boxes with oranges");
(f) fragments where the logical connectives "and", "or" join not the designations of assertions but the designations of objects ("the product A is distributed by the firms B1, B2, …, BN");
(g) explanations of the terms being unknown to an applied intelligent system (because the firms invent and produce new products);
(h) fragments containing the references to the meanings of phrases or larger fragments of a discourse ("this proposal", "that order", etc.); (i) the designations of the functions whose arguments and/or values may be the sets of objects ("the staff of the firm A", "the suppliers of the firm A", "the number of the suppliers of the firm A").

It follows from the above said that the problem is complicated, and it is necessary to use for its solution the most broadly applicable theories (ideally, universal) of representing meanings of NL-texts provided by mathematical linguistics and mathematical computer science.

The papers (Fomichov 1996, 2002a, 2002b) describe two versions of the basic mathematical model of the theory of K-calculuses and K-languages (the KCL-theory); it considerably expanded of the stock of formal tools destined for representing in a formal way the meanings (or contents, or semantic structure) of NL-texts. This basic model includes the definition of a new class of formal languages called standard knowledge languages (standard K-languages, SK-languages). A mathematical model was created describing a system of 10 partial operations on structured meanings (SMs) of NL-texts and, in particular, determining the class of SK-languages. The descriptions of the main features of this model can be found in (Fomichov 2002c, 2004).

The following hypothesis was put forward: 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 by means of these 10 partial operations. The principal advantages of SK-languages in comparison with Discourse Representation Theory, Theory of Conceptual Graphs, and Episodic Logic concern representing complicated goals and destinations of things, definitions of concepts, compound definitions of sets, and meanings of discourses with the references to the meaning of a phrase or larger part of discourse.

This paper continues the line of the papers (Fomichov 1998, 2000) and, in particular, states in more details the main ideas of (Fomichov 2005). The goal is to ground the broad prospects of using the theory of SK-languages for building records of commercial e-negotiations in arbitrary application domains carried out by computer intelligent agents and for forming the contracts concluded in the course of such negotiations. All examples considered in this paper use only the expressive power of restricted SK-languages, completely defined in (Fomichov 1996).

Main Properties of SK-Languages Being Useful for Forming Contracts and Records of e-Negotiations

The analysis shows that the SK-languages posses the expressive possibilities being necessary and sufficient for representing in a formal way the contents of contracts and of the records of commercial negotiations.

In order to illustrate an important part of such possibilities, let’s consider a multi-partner scenario of the interaction of business partners in the course of handling a car damage claim by an insurance company (called AGFIL). The names of the involved parties are Europ Assist, Lee Consulting Services (Lee C.S.), Garages, and Assessors. Europ Assist offers a 24-hour emergency call answering service to the policyholders. Lee C.S. coordinates and manages the operation of the emergency service on a day-to-day level on behalf of AGFIL. Garages are responsible for car repair. Assessors conduct the physical inspections of damaged vehicles and agree repair upon figures with the garages (Xu and Jeusfeld 2003).

The process of a car insurance case can be described as follows. The policyholder phones Europ Assist using a free-phone number to notify a new claim. Europ Assist will register the information, suggest an appropriate garage, and notify AGFIL which will check whether the policy is valid and covers this claim. After AGFIL receives this claim, AGFIL sends the claims details to Lee C.S. AGFIL will send a letter to the policyholder for a completed claim form. Lee C.S. will agree upon repair costs if an assessor is not required for small damages, otherwise an assessor will be assigned. The assessor will check the damaged vehicle and agree upon repair costs with the garage. After receiving an agreement of repairing car from Lee C.S., the garage will then commence repairs. After finishing repairs, the garage will issue an invoice to the Lee C.S., which will check the invoice against the original estimate. Lee C.S. returns all invoices to AGFIL. This firm process the payment. In the whole process, if the claim is found invalid, all contractual parties will be contacted and the process will be stopped.

This scenario provides the possibility to illustrate some properties of SK-languages making them a convenient tool for formally describing contracts.

Property 1. The possibility to build compound designations of concepts.
Example. The concept "a repair invoice" can be represented by the string *invoice * (Theme, certn repair)* of some SK-language, where *certn* is the informational item corresponding to the meaning of the word "certain".

Property 2. The possibility to build compound designations of things and situations.
Example. A concrete repair invoice and a concrete firm called *"Europ Assist"* can be represented by the strings of some SK-language

*certn invoice * (Theme, certn repair : e1) : x1,*
*certn firm1 * (Name1, "Europ Assist") : x3,*

where the strings *e1, x1, x3* are to be considered as the unique marks of a concrete event, concrete invoice, and concrete firm respectively.

Property 3. The possibility to build compound designations of goals.
Example. Let E1 = "The policyholder phones Europ Assist to inform about a car damage". The text E1 mentions the following entities: a policyholder, the concrete firm called "Europ Assist", a concrete car, and a concrete event of the kind "a car damage". Then on the first step of building a K-representation (KR) of E1, i.e. a semantic representation being an expression of some SK-language, the following K-strings will be constructed denoting these entities:

*certn person * (Hold1, certn polis1 : x1) : x2,*
*certn firm1 * (Name1, "Europ Assist") : x3,*
*certn car1: x4, certn damage1 * (Object1, certn car1: x4) :e2 .*

Then E1 may have the following K-representation:

*Situation(e1, phone-communication * (Agent1, certn person * (Hold1, certn polis1 : x1) : x2)(Object2, certn firm1 * (Name1, "Europ Assist") : x3)(Purpose, Inform-transfer * (Them1, certn damage1 * (Object1, certn car1: x4) :e2))).*

Property 4. The existence of the means allowing for representing in a compact way the time and causative relations between the situations.

Property 5. The possibility to construct compact semantic representations of such fragments of sentences which are obtained by means of joining the designations of things, events, concepts or goals with the help of logical connectives AND, OR.
Example. Let E2 = "After receiving a repair invoice from the firm "Lee C.S." and a claim from the policyholder, the company "AGFIL" pays the car repair to the garage". The text E2 mentions two events (dynamic situations) characterized by semantic items *receiving1* and *payment1.* Then a K-representation of E2 can be the expression

(Situation (e1, (receiving1 * (Agent2, certn firm1* (Name1, “AFGIL”) : x1)(Object1, certn invoice * (Theme, certn repair : e2) : x2)(Sender1, certn firm1* (Name1, “Lee C.S.”) : x3) □ receiving1 * (Agent2, x1)(Object1, certn claim1 : x4) (Sender1, certn person * (Hold1, certn policy1 : x5) : x6))) □ Situation (e2, (payment1* (Agent2, x1)(Addressee1, certn garage : x7)(Sum, Cost (e2))) □ Before (e1, e2)) .

Property 6. The existence of the formal means allowing for representing structured meanings of the discourses with the references to the meanings of sentences and larger fragments of the texts.
Example. Let E3 = “The firm “Europ Assist” provides a policyholder with a telephone service; in particular, assigns a garage for repair and informs the company “AGFIL” about a claim of a policyholder. Then E3 may have the following KR:

(Situation (e1, service1 * (Agent2, certn firm1* (Name1, “Europ Assist”) : x1)(Instrument, certn telephone : x2)(Object1, arbitrary person * (Hold1, certn policy1: x3) : x4) : P1 □ Concretization (P1, ((Situation (e2, assigning1 * (Agent2, x1)(Addressee1, x4))( )(Object3, certn garage * (Destination1, repair) : x5)) □ Situation (e3, information-transfer * (Agent2, x1)(Addressee1, certn firm1* (Name1, “AFGIL”) : x6)(Content1, certn claim1 * (Authors, x4) : x7))))) .

The variable *P1* in the constructed formula is a mark of the semantic representation (SR) of the sentence S1 = “The firm “Europ Assist” provides a telephone service to a policyholder”. In the second part of the discourse E3 this mark is used for representing in a compact way the references to the meaning of the sentence S1.

Property 7. The possibility to formally represent the meanings of contractual obligations depending on conditions.
Example. Let E4 = “ The firm “Lee C.S.” assigns an expert for investigating a car during 41 hours after receiving a claim about a car damage if the repair cost doesn’t exceed 500 USD”. The a KR of the text E4 can be the expression

Implies(_ Greater1 (Cost1 (certn repair1 * (Object1, certn car : x1) : e1), 500/USD) , (Situation (e2, assigning1 * (Agent2, certn firm1* (Name1, “Lee C.S.”) : x2)(Person1, certn expert : x3)(Goal1, certn investigation1 * (Object1, x1) : e3)(Moment, t1)) □ _ Greater1 (Difference (t1, t0), 41/hour)) □ Situation (e4, receiving1 * (Agent2, x2)(Object1, certn claim1 * (Theme, certn damage1 * (Object1, x1) : e5))(Time, t0)))) .

Going beyond the scope of the scenario of business interaction discussed above, let’s formulate two additional important properties of SK-languages.

Property 8. The existence of formal means allowing for constructing compound designations of sets as components of semantic representations of NL-texts being records of negotiations or contracts.

Example 1. The set consisting of 12 single rooms in the three-star hotels of Vienna may have a K-representation of the form

*certn set * (Number, 12) (Qualitative-composition, room * (Kind1, single)(Location, any hotel * (Kind2, three-star)(Loc, Vienna)))* .

Example 2. We can designate a concrete planned series of 5 consignments, each consisting of 50 tea services No. 48 and 32 dinner services No. 27, as follows:

*certn set * (Quantity, 5)(Compos1, consignment * (Compos2, (certn set * (Quantity, 50) (Compos1, service1 * (Kind, tea)(No, 48)) ∧ certn: set * (Quantity, 32)(Compos1, service1 * (Kind, dinner)(No, 27))))) : S1* .

Here *S1* is the mark of the planned series of 5 consignments.

Property 9. The possibility to build object-oriented semantic representations of the records of negotiations or contracts, i.e. the expressions of the form

*certn inform-object * (Kind1, concept)(Content1, cont)*$(r_1, u_1)\ldots(r_n, u_n)$,

where *concept* is the designation of the notion "negotiation record" or "contract", *cont* is a K-representation of a document, $r_1, \ldots, r_n$ are the designations of the external characteristics of a document (expressing its metadata, for instance, the data about the authors, date, language, etc.), and $u_1, \ldots, u_n$ are the strings interpreted as the designations of the data associated with a document.

The additional useful properties of SK-languages from the standpoint of building semantic representations (SRs) of contracts and records of negotiations are the possibilities (a) to explicitly indicate thematic roles (or conceptual cases, or semantic cases) in the structure of SRs of NL-texts, (b) to reflect the meanings of the phrases with direct and indirect speech, with the word "a concept", (c) to consider the functions with the arguments and/or values being the sets of objects or concepts (Suppliers, Staff, etc.).

Conclusions

The author of this paper carried out a comparative analysis of the expressive possibilities of SK-languages and of the natural language phenomena reflecting in the structure of commercial contracts and the records of negotiations. A number of the results of this analysis is set forth above. The fulfilled analysis allows for formulating the assumption that the expressive possibilities of SK-languages are sufficient for building with their help the formal representations of contracts and records of commercial negotiations.

During last decade, the most popular approaches to building formal representations of the meanings of NL-texts have been Discourse Representation Theory (Kamp and Reyle 1996); Theory of Conceptual Graphs, represented, in particular, in (Sowa 1999), and Episodic Logic (Hwang and Schubert 1993; Schubert 2000). In fact, Discourse Representation Theory and Theory of Conceptual Graphs are oriented at describing the semantic structure of only sentences and short simple discourses. Episodic Logic studies the structure of only a part of discourses, more exactly, of discourses where the time and causative relationships between the situations (called episodes) are realized.

The authors of Discourse Representation Theory, Theory of Conceptual Graphs, and Episodic Logic don't pose the problem of formally representing the meanings of arbitrary texts pertaining to arbitrary fields of human professional activity: medicine, technology, business, etc. The reason is that, in particular, the expressive possibilities of these theoretical approaches are restricted from the standpoint of modeling the semantic structure of NL-texts of the kinds (a) – (i) listed above in the section "Introduction".

On the contrary, the definition of the class of SK-languages became an answer to the following question: how it would be possible to describe in a mathematical way a system of operations on conceptual structures allowing for building (after a finite number of steps) semantic representations of arbitrarily complicated sentences and discourses from arbitrary application domains, starting from primary informational items.

Thus, the theory of SK-languages opens new prospects of building formal representations of contracts and records of commercial negotiations carried out by computer intelligent agents. Besides, it follows from the fulfilled study that SK-languages provide a unique spectrum of possibilities for representing the results of semantic-syntactic analysis by linguistic processors of the discourses being contracts or the records of commercial negotiations.

References

Fomichov, V.A. (1996); A Mathematical Model for Describing Structured Items of Conceptual Level; Informatica (Slovenia)., Vol. 20 (pp. 5-32)

Fomichov, V.A. (1998); Theory of Restricted K-calculuses as a Comprehensive Framework for Constructing Agent Communication Languages; In: Fomichov, V.A., Zeleznikar, A.P. (eds.): Special Issue on NLP and Multi-Agent Systems, Informatica (Slovenia), Vol. 22 (pp. 451-463)

Fomichov, V.A. (2000); An Ontological Mathematical Framework for Electronic Commerce and Semantically-structured Web; In:.Zhang, Y., Fomichov, V.A., Zeleznikar, A.P. (eds.): Special Issue on Database, Web, and Cooperative Systems. Informatica (Slovenia)., Vol. 24 (pp. 39-49)

Fomichov, V.A. (2002a); Mathematical Foundations of Representing Meanings of Texts for the Development of Linguistic Informational Technologies. Part 1. A Model of the System of Primary Units of Conceptual Level; Informational Technologies (in Russian), No. 10 (pp. 16-25).

Fomichov, V.A. (2002b); Mathematical Foundations of Representing Meanings of Texts for the Development of Linguistic Informational Technologies. Part 2. A System of the Rules for Building Semantic Representations of Phrases and Complicated Discourses; Informational Technologies (in Russian), No. 11 (pp. 34-45).

Fomichov, V.A (2002c); Theory of K-calculuses as a Powerful and Flexible Mathematical Framework for Building Ontologies and Designing Natural Language Processing Systems; In: Troels Andreasen, Amihai Motro, Henning Christiansen, Henrik Legind Larsen (Eds.), 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: Berlin, Heidelberg, New York, Barcelona, Hong Kong, London, Milan, Paris, Tokyo. (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. (2005); 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)

Hasselbring, W. and H. Weigand (2001); Languages for Electronic Business Communication: State of the Art. Industrial Management and Data Systems, Vol. 101 (pp. 217-226)

Kamp, H. and U. Reyle (1996); A Calculus for First Order Discourse Representation Structures. Journal for Logic, Language and Information (JOLLI), Vol. 5 (pp. 297-348)

Kimbrough, S.O. and S.A. Moore (1997); On Automated Message Processing in E-Commerce and Work Support Systems: Speech Act Theory and Expressive Felicity; ACM Transactions on Information Theory, Vol. 15 (pp. 321-367)

Schubert, L.K. and C.H. Hwang, (2000); Episodic Logic Meets Little Red Riding Hood: A Comprehensive, Natural Representation for Language Understanding. In: Iwanska, L., Shapiro, S.C (eds.): Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language. MIT/AAAI Press, Menlo Park, CA, and Cambridge, MA (pp.111-174)

Sowa, J.F. (1999); Conceptual Graphs: Draft Proposed American National Standard. In: Tepfenhart, W., Cyre, W. (eds.): Conceptual Structures: Standards and Practices. Lecture Notes in AI, Vol.1640. Springer-Verlag, Berlin Heidelberg New York (pp.1-65)

Xu L. and M.A. Jeusfeld (2003);: A Concept for Monitoring of Electronic Contracts; Tilburg University, The Netherlands (URL http://infolab.uvt.nl/pub/itrs010.pdf). 19 p.