

DARPA-JFACC Symposium on Advances in Enterprise Control, San Diego, California, Nov.15-16, 1999. RESU68F

IMMACCS: A Military Decision-Support System

Jens G. Pohl
Anthony A. Wood
Kym Jason Pohl
Arthur J. Chapman

*CAD Research Center
California Polytechnic State University
San Luis Obispo, California*

jpohl@calpoly.edu; awood@cdmtech.com; kpohl@cadrc.calpoly.edu; achapman@calpoly.edu

Abstract.

The Integrated Marine Multi-Agent Command and Control System (IMMACCS) is a multi-agent, distributed system, designed to provide a 'common tactical picture' with integrated and meaningful decision-support facilities to authorized operators at any access node. IMMACCS has been implemented as a three-tier architecture that distinguishes between information, logic and presentation. It utilizes an object-serving communication facility with subscription and multi-casting capabilities that is based on the Common Object Request Broker Architecture (CORBA). With an emphasis on application, IMMACCS was designed and implemented in concert with its military users as an integral component of experiments conceived by the Marine Corps Warfighting Laboratory to test emerging concepts in military command and control. It was field tested as the command and control system of record during the Urban Warrior Advanced Warfighting Experiment conducted by the Marine Corps Warfighting Laboratory in Monterey and Oakland, California, March 12 to 18, 1999.

Principal Design Notions and System Components.

IMMACCS is an adaptive, distributed, open architecture system that is intended to assist military commanders under battle-like conditions when dynamic information changes, complex relationships, and time pressures tend to stress the cognitive capabilities of decision makers and their staff. IMMACCS incorporates four design notions that are fundamental to its decision-assistance capabilities.

Notion 1: Whereas legacy systems typically process data, IMMACCS processes information. The key to the assistance capabilities of IMMACCS is that the system has some 'understanding' of the information that it is processing. In IMMACCS every entity in the screen display of the battlefield (e.g., road, building, truck, tank, enemy unit, civilian group, etc.) as well as intangible entities such as weather, attack, defense, and so on, are represented as individual objects with characteristics and relationships to each other. Therefore, the military commander and staff officer interacts with a computer display that consists of hundreds of real world entities (objects) that all have some 'understanding' of each other's nature, interests and objectives, and a great deal of 'understanding' of their own characteristics and capabilities.

Notion 2: IMMACCS is a collection of powerful collaborative tools, not a library of predefined solutions. This approach is intended to overcome the deficiencies of legacy systems in which built-in solutions to predetermined problems often differ significantly from the complex operational situations encountered in the real world. IMMACCS is a collaborative decision-support system in which the operators interact with computer-based agents (i.e., decision tools) to solve problems that cannot be precisely nor easily predetermined.

Notion 3: IMMACCS incorporates agents that are able to reason about the characteristics and the relationships of the many real world entities (i.e., objects) that are recognized within its representational schema (i.e., ontology).

During its first field test (held in California, March 1999 [1]) IMMACCS included agents capable of providing assistance in decision areas involving: weapon selection and deconfliction; Rules of Engagement; potential fratricide situations; enemy engagements and decision points; and, logistical re-supply requirements. In addition, IMMACCS supports mentor agents that can be dynamically created to represent the interests of warfighters and warfighting machines. Mentor agents are intended to extend the capabilities of Marines at all levels by warning friendly units of hostile intrusions into their territory.

Notion 4: IMMACCS integrates planning, execution and training within one common command and control user environment. The computer-based agents and the IMMACCS users continuously collaborate as they interact with each other in rapidly changing battlefield situations. In this respect IMMACCS reflects the complexity of the real world where problem solutions must be continuously reviewed as conditions change, and it becomes increasingly difficult and inconvenient to separate planning, re-planning, execution, and training functions into artificially discrete activities supported by different applications.

IMMACCS is one integrated system and not a confederation of loosely linked sub-systems. Its architecture is based on the Integrated Collaborative Decision Model (ICDM) [2] [3] multi-agent system development framework applied by the CAD Research Center previously in engineering design [4] [5], transportation planning [6] [7] and military command and control [8] [9]. In its field testing state in March 1999, IMMACCS consisted of the following integrated components (Fig.1):

An Object Model (designed and developed by the CAD Research Center) that facilitates the internal representation of information (rather than data). In particular, IMMACCS supports the dynamic formation of associations among objects at both the user and agent levels.

An Agent Engine (designed and developed by the CAD Research Center) that automatically initiates an agent session in support of any desired View of the battlespace.

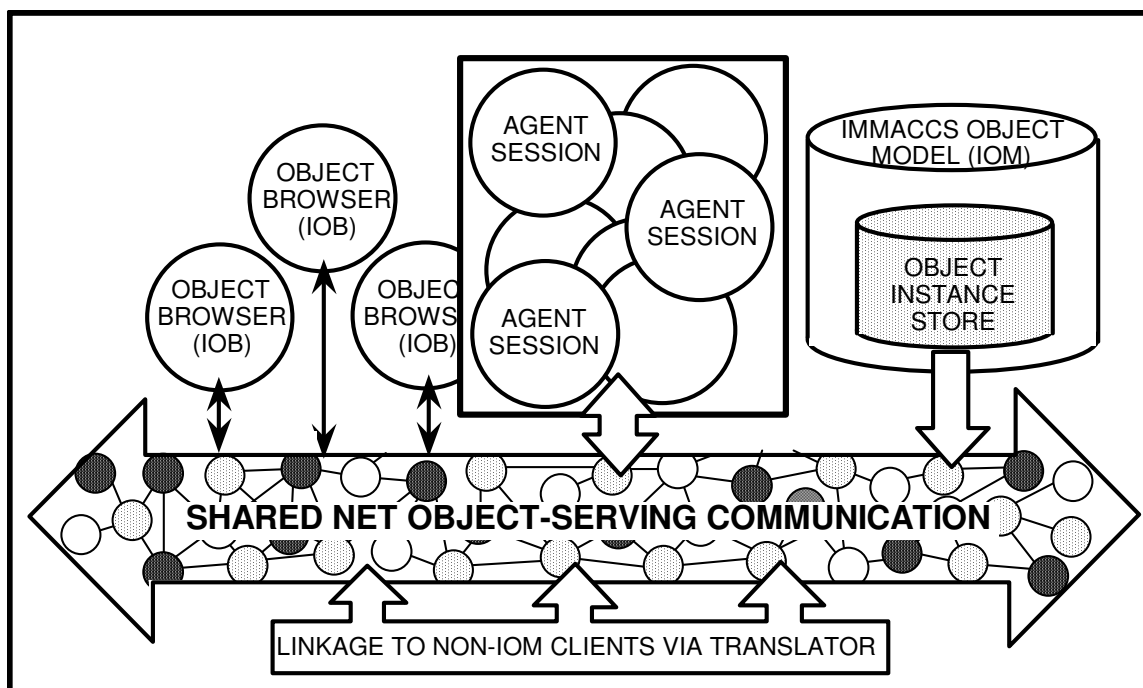


Fig. 1: Schematic representation of the IMMACCS components

A Shared Net communication facility (designed and developed by the Jet Propulsion Laboratory (JPL)) that manages the object-based interactions among the various components on a subscription basis. All IMMACCS components are clients of the Shared Net and indicate their information interests by registering a subscription profile. Whenever, information that is within the subscription of one or more clients (whether military

commander or squad leader) becomes available the Shared Net automatically pushes this information into a cache memory area and sends an alert message to the client. In addition, clients may also query for information for which they have not subscribed. Even individual agent sessions are clients to the Shared Net and can therefore take advantage of these efficient communication capabilities.

A hardware independent Object Browser (designed and developed by the CAD Research Center) that facilitates user interaction within the object-based information context and the collaborative agent assistance capabilities of IMMACCS. Through the Object Browser the user may: set alert conditions (e.g., request warning of enemy advances to within a user-specified radius of the current position of the operator); obtain agent reports and suggestions; request agent explanations; explore the location and capabilities of key resources (e.g., local police and fire stations, hospitals, and government buildings) on the object-based infrastructure display of the battlefield; and, enter information to automatically activate any other client(s) of the Shared Net.

A set of Translators (designed and developed by the SPAWAR Systems Center) that are capable of mapping data received from external applications, such as the Joint Maritime Command Information System (JMCIS), to the object-based representation held within the IMMACCS Object Model.

A hardware independent, lightweight 2-D Viewer user interface (designed and developed by SRI International) that connects the Marine in the battlespace via wireless communication to IMMACCS. Each 2-D Viewer hardware device is provided with a differential GPS (Global Positioning System) facility that transmits automatic position reports to IMMACCS. In this way IMMACCS is able to automatically track the current position of all beacon equipped friendly units, and make this information available to agents as they spontaneously and opportunistically reason about events which might affect these units.

A Geographic Infrastructure Database (designed and developed by the Naval Research Laboratory at Stennis Space Center) that provides objectified battlespace infrastructure from NIMA Vector Product Format (VPF) data.

The Fundamental Requirement of 'Information' Representation.

Although technological advances in computer hardware and communication systems have been truly astounding over the past 20 years, the direct utilization of these advances in the area of decision-support has been less than remarkable. The fact is that we are still using computers largely as *data* processing devices that perform only the most menial and least intelligent data transmission and manipulation tasks. While computers are performing these tasks with great speed and accuracy, and while they are able to provide connectivity among a virtually unlimited number of access points, the higher level and much more rewarding tasks of analyzing, interpreting and abstracting data as 'information' and inferring 'knowledge' is almost entirely left to the human users.

This serious deficiency has become increasingly apparent as technological advances have increased computing power, data storage capacities, and data transmission speeds by orders of magnitude in such a short period of time. Convenient global access to users and data has increased the need for information filtering, so that individuals might take advantage of the opportunities for material and personal profit that this connectivity and processing power present to the user. Needless to say, the capabilities of a computer to assist in the intelligent assessment of information are basically non-existent if the computer processes this information as bitmaps and alphanumeric text strings. Any significantly useful human-computer collaborative partnership carries with it the expectation that information is held within the system environment in a representational form that is, if not equivalent to, at least compatible with human cognition.

The current approach for achieving this objective is to represent information in the computer as objects with behavioral characteristics and relationships to other objects [10]. While this approach is hardly sophisticated it does allow real world objects (e.g., airfield, tunnel, building, weapon, tank) to be represented symbolically so that computer software modules can reason about them.

It is important to note that the relationships among these objects are often far more important than the characteristics that describe the individual behavior of each object. For example, the word 'house' holds little meaning if we strip away the many associations that this word represents in our mind. However, such associations to our knowledge of

construction materials, our experiences in having lived in houses, and our understanding of how our own home is impacted by external factors (such as rain, sunshine, neighbors, mortgage interest rates, and so on) constitute the rich meaning of the object 'house' [11]. Accordingly, any useful representation of information in the computer must be capable of capturing the relationships among the entities (i.e., objects) in the problem system.

While some of these associations are fairly static (e.g., a weapon is a kind of asset and a lethal weapon is a kind of weapon) many of the associations are governed by current conditions and are therefore highly dynamic. For example, as a platoon of soldiers moves through the battlefield it continuously establishes new associations (e.g., to windows in buildings from which snipers could fire on individual members of the platoon), changes existing associations (e.g., higher levels of risk as the platoon nears an active combat zone), and severs previous associations (e.g., as the platoon is forced to abandon its compromised command post).

Abstract concepts such as privacy, security and power, are less amenable to this approach since their meaning and role in our day-to-day activities is less easily defined. For example, the characteristics of 'privacy' are neither static nor can they be accurately described in relational terms. They depend on a wide range of factors that relate to both environmental and personal circumstances and dispositions. These factors can be only partially accounted for through embedded knowledge and rules, and therefore become largely the purview of the human members of the collaborative human-computer partnership.

Nevertheless, even with these shortcomings this form of representation of real world objects can provide the basis of usable problem solving support and decision making assistance. Improvements are possible with the addition of knowledge bases and user interaction. In the latter case the user becomes as much a helper to the system as the system serves as an assistant to the user. However, this occurs in quite different ways. The system uses its computing and logical reasoning capabilities to monitor, analyze and evaluate the actions, requests and interests of the user in an opportunistic manner. The user, on the other hand, helps the system to understand the nature of the objects and relationships that it is processing in a more deliberate manner [12].

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.

In IMMACCS a comprehensive object model representing the characteristics and relationships of the entities expected in the urban battlespace environment, forms the basis of all agent capabilities [9]. Utilizing a partially automated process and a standard graphical methodology (i.e., Unified Modeling Language) it became possible to directly produce final application code. While past efforts by others [13] [14] [15] contributed valuable information relating to the identification and description of object classes for battlefield simulations, these references place little importance on the associations between objects. Accordingly, the development of the IMMACCS object model was required to focus heavily on defining and representing the relationships among battlefield entities.

The IMMACCS System Architecture.

Based on the ICDM framework the IMMACCS model is based on a three-tier architecture that makes clear distinctions between information, logic, and presentation. These tiers are represented by the three major IMMACCS system components; namely: the Shared Net (information tier), the Agent Engine (logic tier), and the IMMACCS Object Browser (IOB) and 2-D Viewer (presentation tier) (Fig.2). Included in the information tier are two additional components. The first of these is the Marine Corps System IMMACCS Translator (MCSIT) providing bi-directional information translation between IMMACCS and external systems (e.g., JMCIS, LAWS, TSCM, etc.). The second system is the Geographic Infrastructure Database (GIDB) responsible for providing geographic infrastructure information (e.g., buildings, roads, etc.) to the other IMMACCS components. Each of these tiers functions in an integrated fashion to form a comprehensive agent-based decision-support execution framework. This framework allows multiple human decision makers to solve complex problems in a collaborative fashion obtaining decision-support assistance from a collection of heterogeneous on-line agents.

The Shared Net Information Server: Conceptually, the Shared Net (SHN) represents a library of objectified information which clients utilize to both obtain and contribute knowledge. The only difference is that clients can obtain this information in, not only a pull fashion, but can also have the SHN push them information on a subscription basis. Physically, the SHN exists as an enhanced distributed object server based on the Common Object Request Broker Architecture (CORBA) specification.

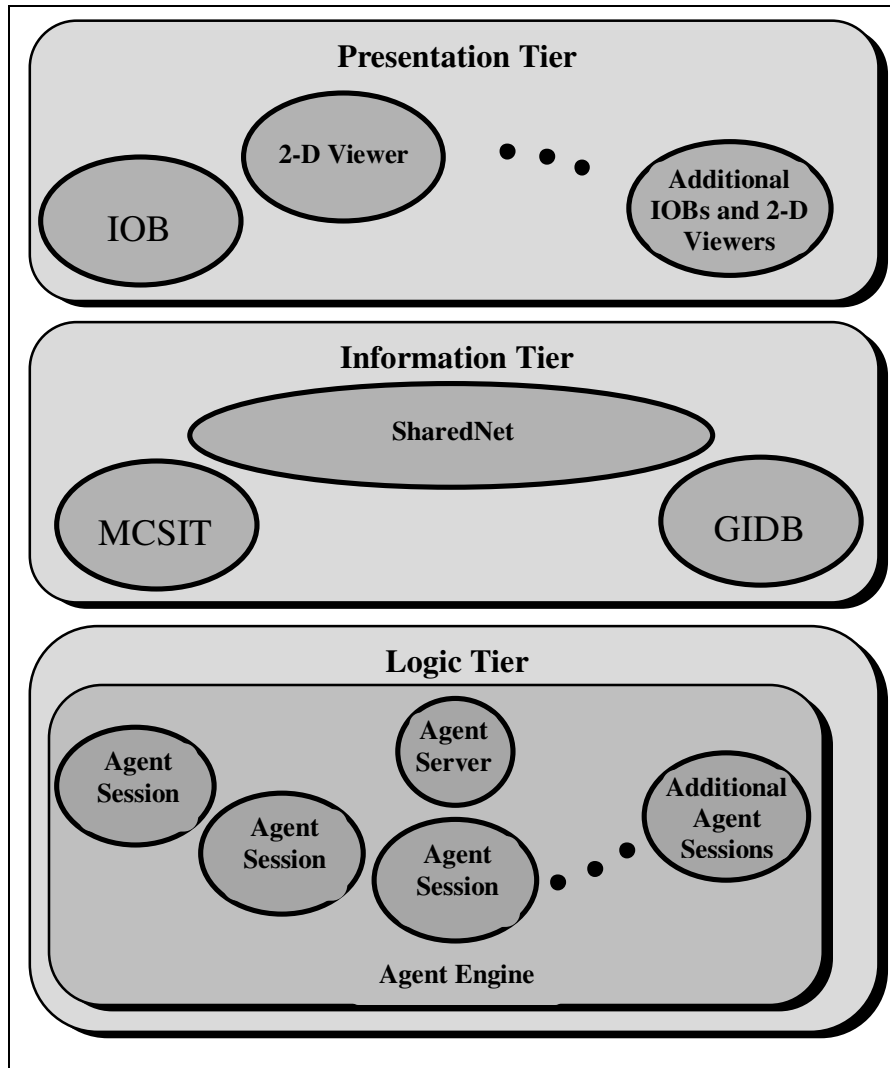


Fig. 2: Three-Tier Architecture of IMMACCS

As the basis of the SHN, distributed object servers are designed to service client requests for information. The knowledge of exactly where the information resides and how it can be retrieved is completely encapsulated inside the object server. This means that clients need not be concerned with which client has what information and in what form that information exists. This feature becomes instrumental in providing an environment where collaborative application components inter-operate in a de-coupled fashion.

Regardless of the native representation of the information, distributed object servers can be used to present information to clients in the form of objects. However, this does not discount the need for information to be modeled as high-level objects in their native form portraying behavior and conveying relationships. While on the surface this representational morphing capability of object servers seems promising, in practice this feature proves to be quite misleading. If the information is not represented at a high level upon its conception, such objectification amounts to little more than wrapping data in communicable object shells. These shells fail to convey any additional insight into the meaning or implication of the information than was present to begin with in its original form. Although in the

future there may be potential for successful research efforts in this area, at present, unless information is originally modeled as objects, knowledge-oriented applications prove to gain little from the distributed object server feature.

However, applications such as IMMACCS that do model information as high-level objects stand to gain considerably from employing SHN-type distributed object servers. Distributed object servers preserve purely objectified representations of information as it moves throughout the system. This is due to the fact that the internal mechanisms of distributed object servers process information as objects themselves.

The IMMACCS model takes full advantage of these object-oriented facilities by integrating an Object-Oriented DBMS (OODBMS) for maintaining persistence. The OODBMS is the facility that the SHN uses to store the application's objects. Employing an OODBMS to store the information objects has two significant advantages.

First, an OODBMS retains the object-oriented representational nature of the information as it exists in its persistent form. Whenever there is representational degradation there is potential for loss of informational content and meaning. By utilizing both transport and storage facilities that are capable of processing and manipulating information as objects, there is virtually no degradation of representation as information flows throughout the system.

The second advantage relates to the manner in which SHN clients request information. Whether mining for information or posting a standing subscription, clients formulate their information requests in terms of objects. More specifically, these subscriptions and queries are formed in terms of object attributes and object relationships, and can range from simple existence criteria to more complex relationships incorporating both logical and relational operators. For example, one such query may request all friendly tracks possessing munitions with an Effective Casualty Radius (ECR) of 500 meters. In this example, the client is essentially pulling information out of the SHN. The operands of the query are each specified in terms of the application's object model.

Another method employed to obtain information from the SHN deals with the notion of subscription. Clients can dynamically register standing subscriptions with the SHN which are again described in terms of the application's object model. For example, a client may request to be notified whenever an enemy track moves to within 300 meters of the client's current location. Once registered, this condition is continuously monitored by the SHN. When satisfied, the SHN essentially pushes the query results to whichever client has indicated an interest (i.e., registered an appropriate subscription). The alternative to this subscription mechanism would be to have interested clients perform the same query on an iterative basis until such a condition occurs. Under these conditions each unsatisfied query may potentially decrease resources (i.e., computing cycles) available to other application components and would essentially prove to be wasteful. If a client takes a more conservative approach where the repeated query is made on a less frequent basis, the client risks being out of date with the current state of affairs until the next iteration is performed. With this in mind, the incorporation of an ability to push information to interested clients becomes essential in providing IMMACCS with an efficient, up-to-date information environment.

The Agent Engine: The Agent Engine represents the logic-tier of the underlying three-tier architecture of IMMACCS. Existing as a client to the SHN the Agent Engine is capable of both obtaining and injecting information into the SHN. Architecturally, the Agent Engine consists of an agent server capable of serving collections of agents (Fig.2). These collections, or Agent Sessions, exist as self-contained, self-managing agent communities capable of interacting with the SHN to both acquire and inject information. As a SHN client possessing and registering interests in events and information, agent activity is triggered by changes in the battlespace. Regardless of whether agents are interacting with the SHN or each other, interaction takes place in terms of objects. This again illustrates the degree to which an object representation is preserved as information is processed throughout IMMACCS.

The Client User Interface: Representing the third and final tier of the three-tier architecture employed by IMMACCS the Client User Interface (CUI) exists as a culmination of instances of the 2D-Viewer and the IMMACCS Object Browser. Collectively, these CUIs provide human users with a means of viewing and manipulating the information and analysis provided by the other two tiers of the IMMACCS decision-support system. Understanding the importance of data presentation, the CUIs present the user with this information and analysis in a robust and graphical manner.

As clients of the SHN, CUI users have the ability to interact with each other in a collaborative fashion. That is, by virtue of either injecting or obtaining information from the SHN, CUI users working on the same view have the potential of exchanging strategic or other kinds of information in a collaborative manner. This type of information exchange occurs regardless of whether the relevant view represents the 'common tactical picture' or exists as a localized strategy explored by a subset of users. All information and analysis remains localized within a particular view unless explicitly copied into another view through user interaction. In this manner, no informational or analytical collisions occur between conceptual views without the potential for user-based supervision and subsequent reconciliation.

Implemented Agent Capabilities.

The following agent capabilities are currently supported by IMMACCS and were field-tested during the Urban Warrior Advanced Warfighting Experiment, held in Central California during March 1999:

Sentinel Agents: These mentor agents are dynamic and can be created by users and tasked to monitor and alert on simple conditions. For example, Sentinel Agents can be used to generate alerts if an enemy unit or a hostile civilian group enters within the area of a given radius around a location in the battlefield. This agent capability can also be useful for monitoring potential targets of indirect fires, or may be employed by each friendly unit in the battlefield to monitor its immediate surroundings. Once a condition is no longer valid, the agent associated with it can be removed by the operator.

Fires Agent: This service agent capability is static and responds to 'Call for Fire' messages. The agent(s) will select the best weapon that is available, deliverable, and acceptable. During the Urban Warrior AWE weaponing portion of this capability addressed range, time of flight, target type, urgency, circular error of probability (CEP), effective casualty radius (ECR), availability, and rules of engagement (ROE). The deconfliction portion of this capability addressed trajectory of munitions relative (i.e., within time and space) to the position of other friendly assets (i.e., people, equipment and other munitions), civilian tracks, and infrastructure objects.

Rules of Engagement (ROE) Agent: This service agent capability is static and monitors for violations in rules of engagement (ROE), in a simplified manner (e.g., entry of tracks into an area of the battlefield that is designated as off-limits). It will also augment the weapons selection and deconfliction capabilities by alerting on available and deliverable weapons that violate the current ROE.

Engagements Agent: This service agent capability is static and monitors incidents of friendly units subjected to enemy fire. Detection of this condition will produce an alert with associated position information. It will also track and alert on sniper fire, terrorist activity, and rioting, highlighting these occurrences on the map.

Logistics Agent: This service agent capability is static and monitors the general readiness of friendly forces. The levels of certain logistics items, such as fuel and water, is monitored with alerts being generated as levels fall below preset thresholds. Upon alert creation the location of potential re-supply points is highlighted on the screen display of the battlefield.

Hazard Agent: This service agent capability is static and monitors the battlespace for indications of nuclear, biological and chemical (NBC) weapons. Upon receipt of NBC indicators, a warning alert is issued indicating the presence and highlighting the suspected position. The friendly units closest to this location are also automatically identified for reconnaissance (RECON) planning purposes.

Intelligence Agent: This service agent capability is static and allows users to monitor enemy sensors. If the sensor is passive the Intelligence Agent sends an alert to propose the initiation of a 'Call for Fire'. However, if the sensor is active the Intelligence Agent will automatically generate a 'Call for Fire'.

Decision Point Agents: This mentor agent capability is a specialization of the Intelligence agent and allows users to request notification as soon as certain user-specified conditions occur within a Named Area of Interest (NAI). For example, a particular road junction may serve as a 'decision point' and alternative courses of action may be attached to the movement direction of the enemy force as it leaves the road junction.

Blue-On-Blue Agent: This service agent capability is static and monitors threats posed by the proposed actions of one friendly unit to another friendly unit. For example, the Blue-On-Blue Agent will generate alerts if either a friendly unit is directly targeted by the 'Call for Fire' of another friendly unit, or if the likely result of a 'Call for Fire' could indirectly endanger the unit.

Conclusion.

IMMACCS is the forerunner of a new wave of decision-support applications that will become widely available to the military during the first decade in the 21st Century. All of these applications will embody at least three fundamental system design concepts. First and foremost, they will incorporate a high level representation of the application domain (i.e., the problem or decision making situation) in terms of objects and relationships among objects. There will be an increasing trend to model the complexities of the problem situation in the ontology (i.e., object model) rather than the agents. In fact, it is likely that the agents in such applications will become relatively simple code modules, with narrowly focused capabilities, but numerous in number.

Second, the ability of these decision-support applications to process 'information' rather than 'data' will allow them to function in a more collaborative role with the human user. Automation will become very much a secondary consideration. The primary concern will be for computer-based agents to assist the human user and each other in collaborative planning, execution and training tasks. In such a collaborative environment emphasis will be placed on the ability of agents to explain their decision process and conclusions, rather than on the automation of the solution generation sequence.

Third, decision-support software will increasingly assume the character of a set of tools. In legacy applications processing 'data' there is no alternative to the incorporation in the software of predefined and hard-coded solutions. This forces the problem in the real world, which is likely to deviate in one or more respects from the anticipated problem, to be distorted to fit the ready made solution that has been built into the software. Software systems like IMMACCS incorporate no solutions, but rather a set of powerful tools. Agents, with their ability to autonomously navigate within the object model representing the problem situation, freely communicate, and dynamically create relationships, constitute the most sophisticated members of this toolkit.

References.

- [1] R. Leighton, K.J. Pohl, M. Porczak, A. Davis, L. Vempati, H. Assal, and J. Pohl, "IMMACCS After Action Report", CAD Research Center, Cal Poly, San Luis Obispo, CA 93407, Apr. 1999.
- [2] L. Myers and J. Pohl, "ICDM: Integrated Cooperative Decision Making – In Practice", *Proc. 6th International Conference on Tools with Artificial Intelligence*, New Orleans, Nov.6-9, 1994.
- [3] K.J. Pohl, "ICDM: A Design and Execution Toolkit for Agent-Based, Decision-Support Applications", in *J. Pohl (ed.) Advances in Collaborative Design and Decision-Support Systems*, Focus Symposium, InterSymp-97, Baden-Baden, Germany, Aug.18-22, 1997, pp. 101-110.
- [4] J. Pohl, L. Myers, A. Chapman, J. Snyder, H. Chauvet, J. Johnson, and D. Johnson, *ICADS Working Model Version 2 and Future Directions*, Technical Report (CADRU-05-91), CAD Research Center, Cal Poly, San Luis Obispo, CA 93407, Jan. 1991.
- [5] J. Pohl, J. La Porta, K.J. Pohl, and J. Snyder, *AEDOT Prototype (1.1): An Implementation of the ICADS Model*, Technical Report (CADRU-07-92), CAD Research Center, Cal Poly, San Luis Obispo, CA 93407, Aug. 1992.
- [6] CADRC, "ICODES: Proof-of-Concept System – Final Report", Contract # N47408-93-7347 (Naval Civil Engineering Laboratory), CAD Research Center, Cal Poly, San Luis Obispo, CA 93407, 1994.

Pohl J., A. Wood, K.J. Pohl and A. Chapman; 'IMMACCS: A Military Decision-Support System'; DARPA-JFACC Symposium on Advances in Enterprise Control, San Diego, California, Nov.15-16, 1999.

[7] K. Penmetcha, A. Chapman, and A. Antelman, "CIAT: Collaborative Infrastructure Assessment Tool", in *J. Pohl (ed.) Advances in Collaborative Design and Decision-Support Systems*, Focus Symposium, InterSymp-97, Baden-Baden, Germany, Aug.18-22, 1997, pp. 83-90.

[8] R. Nadendla and A. Davis, *FEAT: Distributed Problem Solving in a Military Mission Planning Environment*, Master Thesis, Computer Science Department, Cal Poly, San Luis Obispo, CA 93407, 1995.

[9] J. Pohl, M. Porczak, K.J. Pohl, R. Leighton, H. Assal, A. Davis, L. Vempati, and A Wood, *IMMACCS: A Multi-Agent Decision-Support System*, Technical Report (CADRU-12-99), CAD Research Center, Cal Poly, San Luis Obispo, CA 93407, Aug. 1999.

[10] L. Myers, J. Pohl, J. Cotton, J. Snyder, K. Pohl, S. Chien, S. Aly, and T. Rodriguez, *Object Representation and the ICADS-Kernel Design*, Technical Report (CADRU-08-93), CAD Research Center, Cal Poly, San Luis Obispo, CA 93407, Jan. 1993.

[11] M. Minsky, "Why People Think Computers Can't", *AI Magazine*, vol.3(4), Fall 1982.

[12] J. Pohl, "The Representation Problem in CAD Systems: Solution Approaches", in *J. Pohl (ed.) Advances in Cooperative Computer-Assisted Environmental Design Systems*, Focus Symposium, InterSymp-95, Baden-Baden, Germany, Aug.16-20, 1995.

[13] C. Conwell, *Joint Warfare Simulation Object Library*, Naval Command, Control and Ocean Surveillance Center, RDT&E Division, Jun. 1995.

[14] DARPA, *Object Model Working Group Command and Control Schema*, Washington DC, Oct. 1996.

[15] GRC, *The Joint Warfare System Object Model*, Office of the Secretary of Defense, Director for Program Analysis and Evaluation, Joint Warfare System Office, Sep. 1996.