A Hierarchical Collective Agents Network for Real-time Sensor Fusion and Decision Support

Plamen V. Petrov[1], Qiuming Zhu[2], Jeffrey D. Hicks[1], and Alexander D. Stoyen[1, 2]
[1.] 21st Century Systems, Inc., Omaha, Nebraska, USA 68154
[2.] Department of Computer Science, University of Nebraska at Omaha, Omaha, Nebraska, USA 68182

Abstract

This research addresses a problem of how to make effective use of real-time information acquired from multiple sensor and heterogeneous data resources, and reasoning on the gathered information for situation assessment and impact assessment (SA/IA), thus to provide reliable decision support for time-critical operations. A hierarchical collective agents network (HCAN) is employed as a solution to this problem. The agents network supports multi-sensor registration, real-time sensor/platform cueing, level-2 and level-3 information fusion, and has an arm toward the level-4 fusion objectives. An agent component assembly and decision-support-system-development environment, the 21 Century systems' AEDGE™ software package, is used for the design and implementation of a HCAN-DSS system.

The ability to integrate and correlate a vast amounts of disparate information from multiple sensor and heterogeneous data resources with varying degrees of uncertainty in real-time is an impediment issue for mission-critical decision support systems (DSS). For example, in crucial military operations command officers need real-time information and intelligences from various sensors/data resources in a theater of reconnaissance and surveillance to build a whole picture of the battle-space. It is critical for the commanders to know and to understand the relationships among the information collected. Questions are asked: what are the physical and functional constituencies among the objects in a given geographic sector? Are there sequential or temporal dependencies of the objects and what will trigger them? What are the possible consequences of the action and re-actions? Decision making based on these situation assessment and impact assessment (SA/IA) are particularly important for identifying and prioritizing "gaps" between the operation planning and the real-time interactions.

To support making effective SA/IA, a data fusion and decision support system is required to use a set of coherent patterns derived from the available data sets and infer the implications (e.g., causal relations) toward the real world situations. The attribute coherence that is critical to the formation of the meaningful knowledge patterns is often obscure in the data sets obtained from heterogeneous resources. The data collections are often incomplete, imprecise, and inconsistent due to various natural constraints and human faults. Decision makers naturally desire to access large quantities of information expressed in diverse forms. However, as new sensor technology and various information sources have combined to create quantity and diversity, it has become increasingly difficult to provide decision makers with the right information at the right time and in the right quantity and format.

Real-time computerized decision support systems are constructed by integrating a number of diverse components from a variety of software modules. Software developers have come to a numerous ways of querying the local and centralized data resources to access and distill the large and diverse information for the purpose of providing effective decision support. Meanwhile DSS are becoming more and more complex in terms of knowing which data resources to connect, how to keep track of the data dynamics, and assess the reliability of information from each resources. These tying links make the use of intelligent agent architecture necessary and desirable for allowing real-time responsibility and adaptive control of the DSS.

Many popular agent systems of today deploy agents in a uniform level of operation. The agents respond to the same calls and cooperate at the same time toward the same goal of operation. The architecture endues some difficulties in agent communications and task control. When applied in complex real-time DSS with intensive human and system interactions, the cooperative nature makes the system less robust because the disability of one agent would affect the successive operations of the entire agent assembly. The collective nature of the agents in a HCAN paradigm overcomes some of these difficulties, for example, relieving the burden of data-exchanges between fellow agents by limiting agent communication to vertical layers of the assembly only. The hierarchical architecture simplifies the functional design of the agent interactions and enhances the security and efficiency of the process. The HCAN architecture also strikes a balance between the centralized control and distributed computation by allowing distributive agent operations within layers of the hierarchy and enforcing centralized control between the layers of the hierarchy, thus creating a federated agents integration structure. Basically, the HCAN has the functionalities of.

1). A flexible software architecture for accommodating system augmentation and evolutions;
2). A powerful representation schema for accommodating heterogeneous forms of information;
3). A diverse interface for various input resources, output formats, and human interactions;

4). An ability of reasoning on incomplete and inconsistent information, and extracting useful knowledge from the data of heterogeneous resources;
5). An ability of incorporating real-time dynamics of information resources into system at time of operation, and promptly adjusting the reasoning mechanisms;
6). An ability of summarizing and refining knowledge extracted, and distinguishing mission and time critical knowledge from insignificant and redundant ones;
7). A capability of supplying meaningful and accurate explanations, both qualitatively and quantitatively, of the automated system actions; and
8). A capability of providing adequate control and scrutiny of the system operations w.r.t. environment constrains.

There are many sources of uncertainty at different levels of the decision support. For example, even if a situation-assessor is aware of the presence of certain objects in the operation space, such as the type of contact, intention, reaction rational, etc., the exact dynamics of the object is still uncertain to the decision maker. While the knowledge about the object dynamics is critical in constructing an optimal strategy of action, various statistical methods and knowledge discovery techniques are applied in the reasoning module. The level of uncertainty forces the reasoning agents to operate with different decision strategies.

The 21st Century Systems, Inc. has developed the AEDGE™ as an open DII-COE and CORBA compliant agent-based environment that enables the development of components-based agent systems. The system is implemented in Java™, with Java Database Connectivity™ for DB access, Java Swing, AWT, and Java3D for visual interfaces, Java Media Framework and Java Speech API for audio/speech interface. AEDGE™ defines Agents, Entities, Avatars and their interactions with each other and with external sources of information. This standardized architecture allows additional components, such as service-specific DSS tools, to be efficiently built upon the core functionality. Common interfaces and data structures can be exported to interested parties who wish to extend the architecture with new components, agents, servers, or clients. When the core AEDGE™ components are bundled with customer-specific components, a clean separation of those components, through APIs, is provided.

The AEDGE™ is based on an extensible multi-component DSS architecture (EMDA, also referred to as the AEDGE™ Architecture). The architectures describe the data objects, interfaces, communication mechanisms, component interactions, and integration mechanisms for the AEDGE™ and its extensions. In the AEDGE™ architecture, components communicate among each other via the Service Provider/Service Requester Protocol (SPSR). Service providers are components that implement an algorithm or need to share their data (data sources). Service requesters are the components that need a function performed for them by some other component or need to import data from another component. Both service requesters and service providers implement remote interfaces, which enables such components to communicate over a TCP/IP network. The remote interface implementation is currently based on Java RMI (remote method invocation), though the Architecture is not dependent on this implementation. AEDGE™ provides multiple levels of customization. The subject-matter users are able to build scenarios and scripts or to automatically generate them using the AEDGE™-based Scenario Editor. Rules and triggers for agent behaviors can be created and modified by the advanced user. AEDGE™ also provides APIs for custom extensions of agents, data bridges, and the entity framework. The practical user will enjoy AEDGE™'s versatile data connectivity and its near-real-time execution and monitoring of DSS functions. As a built-in bonus, AEDGE™ provides connections to a number of simulators and data formats, including HLA, DIS, DTED, DBDB2, XML, as well as support for multiple modes of distribution (CORBA, RMI, TCP/IP).

As an example of the HCAN design using AEDGE™ for data fusion and DSS applications, the Advanced Battlestation with Decision Support System (ABS/DSS) which was developed as an operational agent-based C2 team decision support platform for command and control centers aboard aircraft carriers.. The ABS/DSS is based on AEDGE™'s implementation of HCAN, and provides consolidated situational awareness through real-time, interactive, agent based decision support coupled with a linked 2D/3D battlespace visualization. Additionally, the ABS/DSS supports shipboard distributed training, train-as-you-fight, with a built-in scenario construction and emulation of friendly and hostile entities. Whether the watchstander is in live-feed mode or in training mode the operation of the agent-based decision support system and the 2D/3D visualization is identical. The ABS/DSS was designed and prototyped using cognitive and functional analysis of team member roles, responsibilities, and decision-making process with the help of Subject Matter Experts (SMEs), to optimize applicability of results to operational settings. Systematic descriptions of watchstation roles, responsibilities, requirements, interdependencies, tactics, strategies, and task demands were collected from SMEs, cognitive task analyses and focal-group interviews. These data were examined to identify decision events, which were generic to performance, regardless of mission, and likely to bottleneck under high tempo situations. Based on the analytic agent behaviors, interactions and optimization processes were defined to match the operational tasks and environment. When a watchstander decides to "log in" to a training session in a particular role, he/she may choose to view recommendations generated by the agent for that role or not. The agent recommendations are logged constantly in the system. This enables human decision making to be directly compared to agent-based decision making. The capability facilitates skill acquisition and decision making for events that are both typical and time consuming.