The Underlying Design Principles Of the ICDM Development Toolkit

By

Kym Pohl
Collaborative Agent Design Research Center
Cal Poly State University
kpohl@cadrc.calpoly.edu
http://www.cadrc.calpoly.edu

For the past sixteen years the Collaborative Agent Design Research Center (CADRC) at the California Polytechnic State University (Cal Poly), San Luis Obispo has pursued the design and development of agent-based decision-support systems. Throughout this journey the CADRC has relied on a suite of development tools that greatly assist in the creation and management of such systems. This suite of tools is known as the Integrated Cooperative Decision-Making (ICDM) software development toolkit (Figure 1).

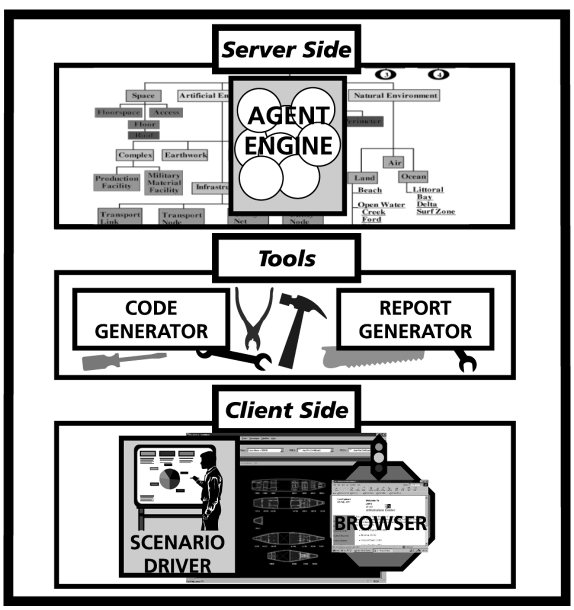

Figure 1 – ICDM Development Toolkit

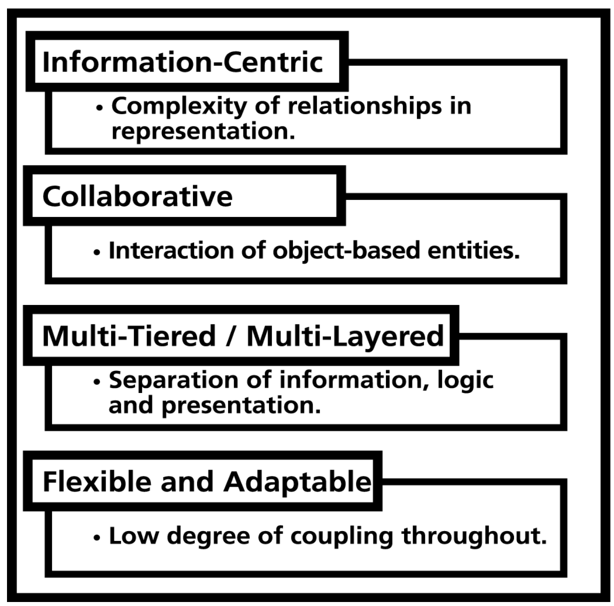

Figure 2 – ICDM Design Principle

Not only does ICDM function as an accelerator (i.e., rapid development) and stabilizer (i.e., built-in robustness and fault tolerance) in the development of decision-support systems, but it also provides a concrete vehicle for representing the key concepts and philosophies that the CADRC has found to be useful for the success of these types of systems (Pohl et. al., 2000; Pohl, 1997). This paper focuses on the key design principles on which ICDM is founded; namely, collaboration-intensive, context-based representation, flexibility and adaptability, and multi-tiered, multi-layered architecture (Figure 2).

Collaboration-Intensive

Certainly in the real world, collaboration among decision-makers and experts is a critical ingredient in making educated and effective decisions. This is especially true when operating across an extensive and varied set of domains. Through years of research in collaborative design the CADRC has found that this same quality extends to the realm of agent-based decision-support systems. Conceptually, the systems developed by the CADRC consist of dynamic collections of collaborators (both human and software-based) each playing a role in the collective analysis of a problem or situation and the consequential decision making assistance required in formulating an accurate assessment and /or solution.

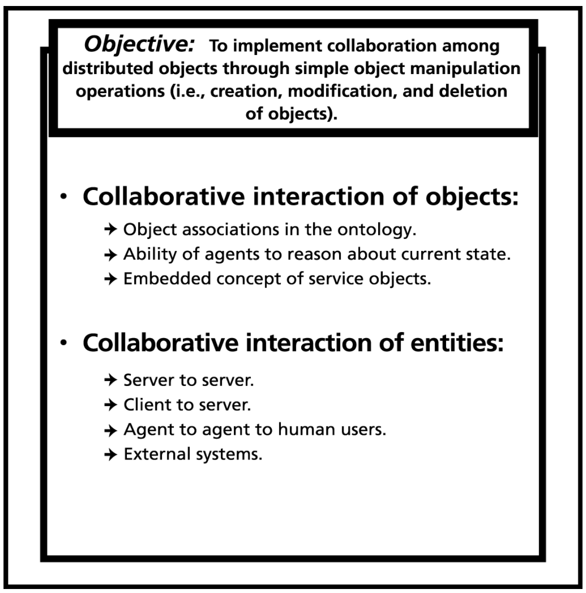

Figure 3 – Design Principle: *Collaboration*

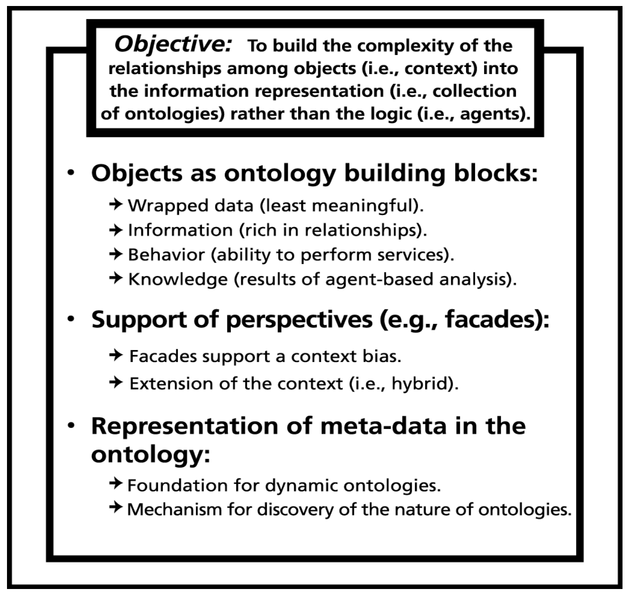

Figure 4 – Design Principle: *Information-`Centric (Context)*

Whether human or software-based, collaboration within an ICDM-based system occurs in terms of a descriptive ontology (Chandrasekaran et. al., 1999). Until recently these ontologies were limited to describing information and knowledge that represents various aspects of the domain(s) over which the system is to operate. For example, in the domain of architectural design the applicable ontology would describe such notions as spaces, walls, accessibility, appropriate lighting, and so on. Although effective and certainly a fundamental element of an information-centric system a considerable portion of the system still remains in a form not necessarily supportive of highly collaborative environments. Further, these non-ontology based components require separate and dedicated interfaces along with specialized management. A number of the services that collaborators within an ICDM-based system interact with (e.g., time, query, subscription, execution, reasoning, etc.) were still presented as client-side adjunct-based interfaces requiring additional management to support collaboration. For example, if two clients wish to share or discuss the same subscription profile, a separate mechanism for identifying and referencing the collection of interests is required. In this case, the interface would be the client-side Application Programming Interface (API) maintained by the subscription service itself.

Although certainly possible, supporting such specialized functionality requires the particular services (i.e., the subscription service in this case) to present and manage a specific API to expose or match global references. Although subtle in nature, complexity such as this can easily escalate when considering the high degree of collaboration inherent in multi-agent decision-support systems.

Recently, however, the CADRC has overcome this limitation by taking the notion of *objectified collaboration* to the next level (Figure 3). This approach extends the once solely information-based ontology to include behavioral aspects of the decision-support system. More specifically these constrained behavioral objects constitute the services within the decision-support system (i.e., the services themselves are represented in the collaborative ontology in the same manner as information and knowledge). The only difference is that these distributed and shareable objects offer behavior in addition to information. As a result collaborators are able to interact with these services through the same distributed object operations that they would perform on the information and knowledge objects. Any constraints identified in the behavior are enforced by the standard ontology management facility. The operations that can be performed on these ontology-based objects consist of the basic creation, deletion, and modification functionality. To support the aforementioned example in which two collaborators wish to reference and discuss aspects of the same subscription profile, the two collaborators would treat the profile in question as just another set of multi-faceted, shareable distributed object. In other words, similar to the manner in which rich information models, or ontologies are used as a basis for collaboration this notion is extended to include interaction and collaboration across the services that constitute the system itself. The effect is essentially that interaction with and collaboration across information, and now behavior (e.g., services), is reduced to a basic set of object manipulation capabilities. In this sense, an object is-an-object is-an-object. The only difference is that some distributed, shareable objects offer information and some offer behavior. The client-side portion of the ontology replaces the need for specialized client-side functionality.

Context *(Information-Centric)*

Representation can exist at varying levels of abstraction (Figure 4). The lowest level of representation considered in this paper is *wrapped* data. Wrapped data consists of low-level data, for example a textual e-mail message that is placed inside some sort of an e-mail message object. While it could be argued that the e-mail message is thereby objectified it is clear that the only objectification resides in the shell that contains the data and not the e-mail content. The message is still in a data form offering a limited opportunity for interpretation by software components.

A higher level of representation endeavors to describe aspects of a domain as collections of inter-related, constrained objects. This level of representation is commonly referred to as an information-centric ontology. At this level of representation context can begin to be captured and represented in a manner supportive of software-based reasoning. This level of representation (i.e., context) is by far the most empowering design principle on which ICDM is based. Further, as mentioned in the previous section portions of this context may be extended to exhibit behavior. In addition to services, however, distributed behavioral objects can also be employed as a mechanism for supporting the notion of facades.

Existing as one of the fundamental design patterns employed in object-oriented design (Pohl K. 2001) facades provide a level of derivation attained from the particular representation or ontology on which they are based. In the case of ICDM and the type of ontologies it manages facades offer a method of supporting and managing an alternative *perspective* from that modeled in the ontology from which they are derived (Pohl K. 2001). In other words, ICDM-based facades allow the perspective inherent in a particular model of a domain to be augmented, or in some way altered to support a more appropriate (i.e., to the façade user) representation of the concepts, notion, and entities over which that *user* is operating (Figure 5). Note that *user* in this sense refers to any accessing component. While certainly useful in systems supporting multiple perspectives caution must be employed in preventing abuse by introducing inconsistency and unnecessary duplication.

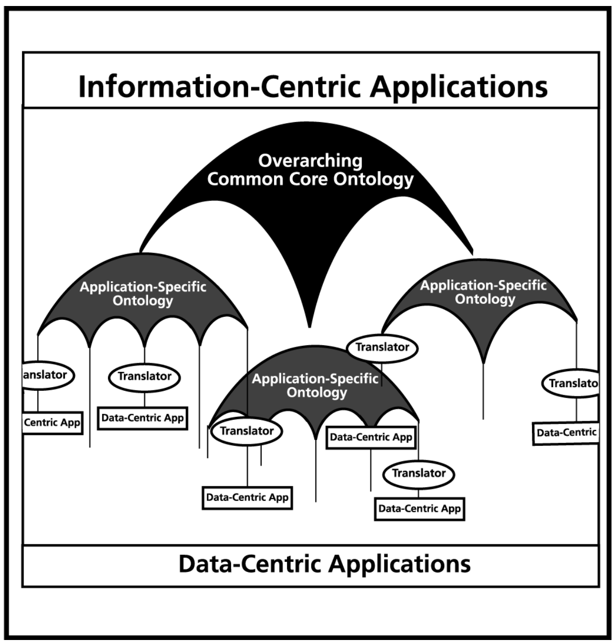

Figure 5 – Ontology Cluster
Supporting Multiple Levels of Abstraction

Figure 6 – Design Principle:
Extensibility and Adaptability

Facades can also be utilized to support real-time calculations. In this sense, the façade derivation would involve a calculation or algorithm perhaps based on one or more attributes of the base object(s). For example, consider an architectural space exhibiting length, width, and height described in English standard units which is to be accessed by a design system that only understands metric units and also requires space volumes. Utilizing ontology-based facades a model could be developed in which, not only the length, width, and height, but also the volume of the space could be calculated and presented to the design system in terms of metric units. Although there are a number of approaches to supporting calculated attributes in the case where an alternative perspective is to be supported, the façade approach permits an extensible (i.e., one perspective extended from another) and encapsulated (i.e., easily maintainable) solution.

Extensibility and Adaptability

One of ICDM's primary goals is to support a high degree of flexibility in respect to the configuration of its components both at the development and execution levels. ICDM supports the addition, replacement, and reuse of software components in the context of agent-based, decision-support systems, and achieves this goal by reducing inter-component coupling to an absolute minimum (Figure 6). There are two key ICDM properties that permit this flexibility. First, all collaboration between clients takes place via, and in terms of the informational ontology (i.e., distributed objects). No direct communication exists between collaborators. The result is a collaborative environment in which client identities are essentially irrelevant in respect to this process. This low degree of coupling permits the reconfiguration (i.e., component addition, removal, or replacement) of collaborating components at any point during an execution session.

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

Apart from the ability to adapt to an evolving definition of a domain, adaptability is also supported in interaction with external systems. This level of adaptability functions in conjunction with the concept of façades mentioned earlier. Replacing the classical approach of building a dedicated and separate translation bridge between collaborating systems, ICDM promotes the incorporation of such translation into the ontology itself. In other words, using ICDM's support for ontology-based facades, translation or derivation of each system's perspective can be encapsulated and managed solely within façade objects. The resulting translation facility exists as a set of behavioral façade objects accessed and manipulated in a manner no different than is applied to other ontology objects. The result is an elegant design where support for translation-based communication between disparate systems is seamlessly incorporated as part of the ontology.

Multi-Tiered, Multi-Layered

The forth design principle to which ICDM adheres addresses the architectural organization of ICDM-based systems. More specifically, this principle identifies distinct separations between areas of functionality at both the conceptual (i.e., tier) level and the more concrete (i.e., layer) level. Conceptually, the architecture of an ICDM-based decision-support system is divided into three distinct tiers namely, information, logic, and presentation. To manage its particular domain each tier contains a number of logical layers that work in sequence (Figure 7).

As the name suggests the information tier houses both the information and knowledge (i.e., ontology) being operated on in addition to all of the mechanisms needed to support management, transport, and access. The information is further delineated into layers. The first of these is the Object Management Layer (OML) described in an earlier section. Below the OML resides the Object Access Layer (OAL) responsible for managing access to the information tier. The OAL exists as a level of abstraction below OML and interfaces directly with the Object Transport Layer (OTL). Based on the CORBA specification (Mowbray and Zahavi 1995) the OTL is responsible for communicating the various requests and subsequent replies for distributed information and behavior issued through the OAL throughout the system. The OTL is the only layer that forms a dependency on an underlying communication protocol. As such, support for alternative communication facilities can be implemented with minimal impact on either the OAL or the OML. This is an excellent example of the benefits of a layered architecture in supporting component reuse and replacement.

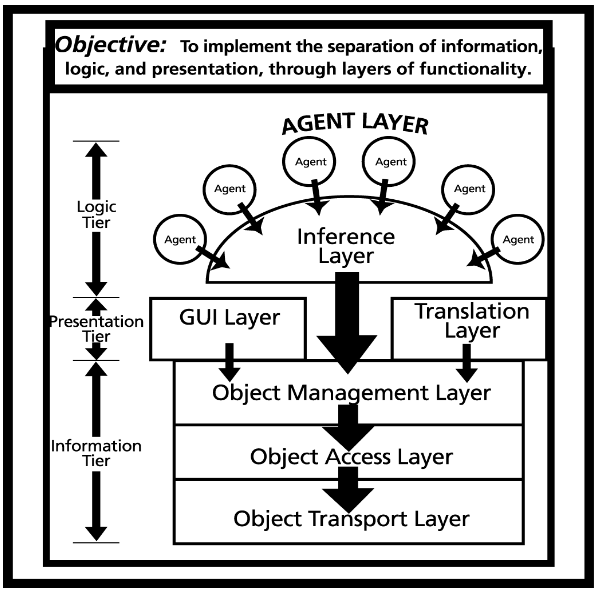

Figure 7 – Design Principle:
Multi-Tiered, Multi-Layered Architecture

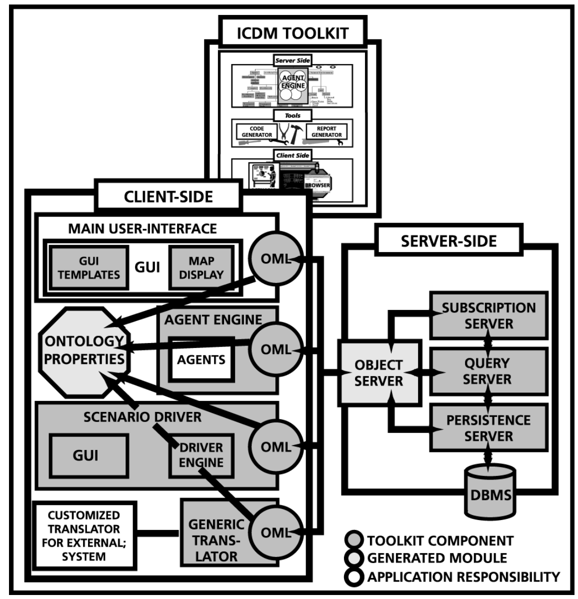

Figure 8 – ICDM-Based System
Component Types

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

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

Conclusion

As a toolkit for the development of agent-based, decision-support systems ICDM supports three types of components each working in conjunction with the others to form a complete decision-support system, namely: the toolkit facilities; the automatically generated modules; and, the application-specific code that must be created manually (Figure 8). The first type of component is automatically generated from the ontology. Among the generated artifacts is a property file that contains detailed characteristics of each object described in the ontology. These properties are used to configure the second type of component. Configuration of the second type of component takes place during runtime and essentially conforms this category of components to the specific system in which they are operating. The third type of component is system-specific and the responsibility of the particular project. This set of components primarily includes the agent rules and user-interface. Together, these three types of components are integrated at the project level to formulate a specific agent-based, decision-support system.

Adhering to the design principles of collaboration, context-based representation, extensibility and adaptability, and a multi-tiered, multi-layered architecture, ICDM can be effectively utilized in the rapid development of agent-based decision-support systems spanning a variety of complex domains. ICDM has been successfully employed by the CADRC in the development of decision-support systems ranging from architectural design (e.g., ICADS and KOALA, (Pohl et. al. 1991; Pohl K. 1996)) to tactical command and control (e.g., IMMACCS and FALCON, (Pohl et. al. 1999)). Profiting from being founded on a framework embodying the principles described in this paper each such decision-support system exhibits the key qualities (e.g., collaborative, high level representation, and tools as opposed to predetermined solutions) that the CADRC has found to be vital to the effectiveness of agent-based, decision-support systems.

References

Chandrasekaran, B., Josephson, John R. and Benjamins, V. Richard. What are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems, Vol. 14 No. 1, January/February 1999.

Chapman A.; ‘Collaborative Support Systems for Facility Management’; in Pohl J. (ed.) Advances in Collaborative Decision-Support Systems for Design, Planning, and Execution, focus symposium: International Conference on Systems Research, Informatics and Cybernetics, Baden-Baden, Germany, August 17-21, 1998 (pp.71-80).

Fridman-Noy, Natalya, Hafner, Carole D. The State of the Art in Ontology Design: A Survey and Comparative Review. AI Magazine, Vol. 18, No. 3, Fall 1997.

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

Myers L. and J. Pohl; 'The ICADS Expert Design Advisor: An Aid to Reflective Thinking'; Knowledge-Based Systems, 5(1), March 1992 (pp.41-54).

Myers L., J. Pohl and A. Chapman; 'The ICADS Expert Design Advisor: Concepts and Directions'; First International Conference on Artificial Intelligence in Design; Edinburgh, Scotland, UK, June 25 to 27, 1991.

Pohl J.; ‘Collaborative Decision-Support and the Human-Machine Relationship’; Office of Naval Research (ONR) Workshop hosted by the CAD Research Center, Cal Poly, San Luis Obispo, CA, April 20-22, 1999.

Pohl J.; 'Human-Computer Partnership in Decision-Support Systems: Some Design Guidelines'; in Pohl J. (ed.) Advances in Collaborative Design and Decision-Support Systems, focus symposium: International Conference on Systems Research, Informatics and Cybernetics, Baden-Baden, Germany, August 18-22, 1997 (pp.71-82).

Pohl J., A. Chapman and K. Pohl; ‘Computer-Aided Design Systems for the 21st Century: Some Design Guidelines’; 5th International Conference on Design and Decision-Support Systems for Architecture and Urban Planning, Nijkerk, The Netherlands, August 22-25, 2000.

Pohl J. and L. Myers; 'ICADS: A Cooperative Decision Making Model with CLIPS Experts'; 2nd Annual CLIPS Conference, Johnson Space Center, Houston, Texas, Sep.23-25, 1991.

Pohl J., L. Myers, A. Chapman, J. Snyder, H. Chauvet, J. Cotton, C. Johnson and D. Johnson; 'ICADS Working Model Version 2 and Future Directions'; Technical Report, CADRU-05-91, CAD Research Center, Design Institute, College of Architecture and Environmental Design, Cal Poly, San Luis Obispo, CA, 1991.

Pohl J., L. Myers, J. Cotton, A. Chapman, J. Snyder, H. Chauvet, K. Pohl and J. La Porta; 'A Computer-Based Design Environment: Implemented and Planned Extensions of the ICADS Model'; Technical Report, CADRU-06-92, CAD Research Center, Design and Construction Institute, College of Architecture and Environmental Design, Cal Poly, San Luis Obispo, CA, 1992.

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

Pohl K.; 'KOALA: An Object-Agent Design System' in Pohl J.(ed.) Advances in Cooperative Environmental Design Systems, focus symposium: International Conference on Systems Research, Informatics and Cybernetics, Baden-Baden, Germany, August 14-18, 1996 (pp.81-92).

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

Uschold, Mike and Gruninger, Michael. Ontologies: Principles, Methods and Applications. Knowledge Engineering Review, Vol. 11, No. 2, June 1996.

Uschold, Mike and King, Martin. Towards a Methodology for Building Ontologies. Workshop on ‘Basic Ontological Issues in Knowledge Sharing’ held in conjunction with IJCAI-95.