# KOALA: An Object-Agent Design System

#### Kym Jason Pohl Collaborative Agent Design (CAD) Research Center Cal Poly, San Luis Obispo, California, USA

### Abstract:

This paper describes KOALA, a design system with primary focus on the predesign stage of architectural design. KOALA combines object-agent technology with high level representation to form a partnership between the human designer and the computer-based design environment. A diverse taxonomy of agents including Domain agents. Space agents, and Monitor agents work in a collaborative fashion to provide the human designer with expert evaluation and assistance in developing a design solution. Spaces, in their roles as agents attempt to formulate various design solutions based on individual perspectives through negotiation with other agents. The result is a highly interactive design environment rich in agent-based assistance and exploration.

# **Keywords:**

distributed computing, collaboration, decision-support, agents, object-agents, AI, deconfliction, pre-design

# Introduction:

Despite offering increased functionality and performance, the new generation of CAD applications still suffers from inadequate representation. CAD systems typically only understand a low level geometric description (i.e., points, lines, polygons, primitive geometric solids) of the evolving design. Such geometric characteristics provide only a partial description of the designer's intent (Kalay 1989, Myers et al. 1993, Pohl 1995). Absent are notions of climate, structure, sound control, access, and other abstract, domain specific information. In addition, descriptions at this level have the potential to infer a considerable number of equally abstract relationships. Most CAD systems have no method of representing these kinds of characteristics of an evolving design.

KOALA addresses this dilemma through the incorporation of various agent technologies together with high level representation of both the design objects and the relationships which link them together. The notion of an agent takes on many definitions throughout the field of artificial intelligence (AI). However, for the purpose of this discussion, an agent will be defined as a collection of constraints and functionality having the following characteristics (Woolridge and Jennings 1995): Agents are largely autonomous in nature in that they operate without the direct intervention of other parties (i.e., humans). By virtue of this autonomy, agents employ at least some degree of control over their actions and internal state. Agents communicate information through inter-client interaction. This social ability may even expand to interaction with the human user. Agents also exhibit a reactive nature. Whether dealing with the physical world, a collection of other agents, or

a human user, agents perceive their environment and respond to changes in a timely fashion. In the case of what will be termed "object agents", this activity may be proactive in nature exhibiting goal-oriented behavior. In other words, an agent may in fact take the initiative with respect to satisfying its goals and objectives (Durfee 1988, Durfee and Montgomery 1990, Larsi et al. 1990, Lesser and Corkhill 1983).

Still in its infancy, agent technology is being successfully integrated into the design environment creating a highly effective human-computer design partnership. The Intelligent Computer-Aided Design System (ICADS) developed by the CAD Research Center (CADRC) at California Polytechnic University is an example of such a relationship (Pohl et al. 1989, Pohl et al. 1991, Pohl et al.1992). ICADS consists of nine expert system agents centered around an "off-the-shelf" 2-D Computer-Aided Drawing (CAD) system. Each agent represents an expert in a particular domain. These domains consist of typical areas of concern in architectural design such as structural system selection, thermal, lighting, and cost control. Based on prototypical information, the Domain agents constantly evaluate the evolving design flagging potential violations and proposing various suggestions of how to correct them.

However, despite such agent-based assistance the designer is still forced to communicate the solution via a geometric CAD environment. Essentially driven by prototypical knowledge, this description is then translated into more abstract, non- geometric representations. Highly presumptive in nature, by its very essence this approach is prone to misinterpretation and loss of meaning. With this in mind, a more efficient environment is one which is essentially void of any translation between the designer's view and that which is represented by the application. In such an environment design elements are created and manipulated in the same environment as they are evaluated without any concern of translation.

KOALA attempts to create a holistic design-assistance environment capable of providing online expert evaluation and assistance throughout the predesign activity. Similar to the ICADS model, assistance is provided by collections of expert agents and is based on prototypical knowledge. However, the design environment enhances such functionality by extending the notion of an agent and presenting a somewhat more sophisticated collaborative model. Following is a theoretical description of such an environment including agent psychology and collaborative nature. Specifically, the following sections describe five agent types; namely: the Designer agent, Domain agents; Space agents; Monitor agents; and, Message agents.

# **Designer Agent:**

The agent taxonomy begins with identifying the human designer (i.e., the user) as the most intelligent agent in the computer-based design environment. Capable of a wide variety of cognitive skills ranging from in depth analysis to highly abstract conceptualization, the human designer is essentially the driving force behind the progressing solution. Unique to this agent is the notion of intent. Intent refers to the goals and objectives of the designer. KOALA represents such notions with the addition of a Designer agent. It is the responsibility of this agent to not only acquire the designer's

intent, but to also maintain its reflection in the decisions being made by the agents in the system. Intent may be explicitly expressed in the form of design criteria, such as performance requirements, or implicitly hidden in decisions that are influenced by vaguely defined perceptions and subtle nuances. In the design activity, the notion of intent is essentially embedded in the strategy employed by the designer. Unfortunately, this implicit notion is not readily identifiable by observers or for that matter, at times by the human designer as the initiator (Schon 1988). Existing as such an intangible entity, this notion continues to be extremely difficult to capture in an electronic environment. Much of this difficulty relates back to the inadequate representation dilemma described earlier. However, the ability of the computer to understand a designer's intent is by far the most crucial element in obtaining a truly comprehensive human-computer partnership. Once achieved, the computer would be able to essentially anticipate the designers interests, desires, and actions. For example, if the computer had an understanding of the designer's intention of imposing a particular architectural style onto the design, the computer application could assist in its implementation by offering numerous design alternatives adhering to that particular style. Further, in an agent-based design system, the actual logic applied by the assisting agents could be adapted to reflect the desired architectural style. Despite its potential, such functionality continues to elude design applications. With current technology, intent must essentially be derived through observing the designer's actions. While this approach is often prone to substantial misinterpretation, it does succeed in providing at least some potential of more effectively assisting the designer in formulating a design solution.

# **Domain Agents:**

The second category within KOALA's agent taxonomy is the Domain Agent. Serviceoriented, Domain agents provide expertise within specific domains of knowledge. Each agent provides expert evaluation and consultation based on its particular domain. Such analysis is largely driven by prototypical knowledge. In other words, an educated comparison can be formed between the various attributes and characteristics comprising the current solution and those commonly associated with design elements of a similar nature in a related environment. The exact set and depth of domains represented depends on the context in which the application is to be employed. For example, recall that the agents in the ICADS design assistance application provide expertise in domains such as lighting, thermal, structure, and sound. Whereas, agents within a transportation planning decision support application involved in the stowage of cargo onto a ship may represent domains such as hazardous materials, trim and stability, and accessibility. Despite the substantially different collection of domains represented, both sets of agents work in the same manner to provide expert assistance.

# **Space Agents:**

The next division within the taxonomy is the Space Agent. Space agents are a specific instantiation of a more general type of agent that can represent the interests of a high level object which plays a significant role in the decision making process of the application environment. In the building design application, spaces or rooms play an important role in the development of floor plans. The architect manipulates spaces as

complex data objects with strong relationships to each other and equally important relationships to data entities that are related but substantially different in nature (e.g., occupant activities, privacy, security, etc.). The ability of the human designer to reason about the relationships among complex data objects is an essential part of the decision making process that underlies the design activity. Computer-assisted design systems, such as ICADS, utilize some form of semantic modeling approach (Myers et al. 1993) to define a common vocabulary that serves as an internal high level representation of real world objects, such as spaces, walls, and opening. This approach provides a workable basis for Domain agents to monitor the evolving design solution and communicate with each other and the designer through some type of coordination facility. The success of this approach must rely heavily on predefined knowledge that is embedded in the agents, and user interactions (i.e., the intervention of the user to maintain and prioritize relationships as a reflection of his/her design intent).

A different approach is to treat the objects that play a major role in the problem environment (e.g., building design), not as passive data entities, but as active agents. Such object-agents can utilize communication capabilities to dynamically create and maintain relationships to other object-agents. Potentially, this would appear to be a significantly more promising approach. Such an environment allows a complex problem system to be decomposed into sub-problems without diluting or losing relationships. To the contrary, relationships are greatly strengthened through the dynamic nature of communication in a collaborative environment. Space agents then are object-agents that have knowledge of their own nature (i.e., essentially the same descriptions that are contained in a 'space' data-object) and the ability to interact with other agents through their communication capabilities. They can act on their knowledge, gain additional information, and request services from other agents.

Prototypical knowledge relative to a space is utilized by a Space Agent to form a set of interests and desires. With each addition of a space into the evolving design, a Space Agent is created and associated with that space. The sole purpose of a Space Agent is to represent the interests of its space. These interests are largely based on prototypical knowledge describing a space of the particular type in a similar environment. Consequentially, each Space Agent views the world (e.g., solution space) from its own, potentially biased perspective. Such biases are an important ingredient of an autonomous environment. As in human group collaborations they reflect the variety of viewpoints that can apply in a given context, and must therefore not be suppressed in the computerassisted environment. Extensively analyzed, argued, and negotiated, these viewpoints lead to a more comprehensive understanding of the problem and presumably a higher quality solution. This approach constitutes a significant deviation in representation as compared with the ICADS model. The design components (i.e., spaces, doors, windows, etc.) in ICADS exist only as informational descriptions. A space in the ICADS model does not have desire, motivation, and most importantly, functionality. With this in mind, collaboration within ICADS is primarily driven by a static set of Domain agents. However, in the KOALA design environment, collaboration is driven by the design objects themselves, namely spaces. In fact, the granularity of this representation could be even further refined to include Wall agents, Window agents and Door agents (Pohl et al.

1994). Each agent would view the progressing solution from its own perspective attempting to satisfy its specific set of interests.

### **Monitor Agents:**

In this highly collaborative environment there is a need for facilitators to detect conflicts and moderate arguments among object-agents. This role is assumed by Monitor agents. In the ICADS model non-convergence (i.e., the inability of the agents to come to a consensus) could be controlled through various techniques, such as user interaction and the assignment of priorities (*add reference*). In the KOALA system this problem is much more serious, not only because of the relatively large number of object-agents (i.e., Space agents) but also because of the different viewpoints that these agents represent.

For this reason the concept of Monitor agents has been introduced in the KOALA system. The purpose of a Monitor agent is to identify possible conflicts and assist in their resolution through the application of moderating techniques that have been successful in human collaborations. Before identifying these techniques and discussing their application by Monitor agents, it is necessary to consider agent collaboration behavior in more detail.

When a modification to the progressing design occurs, each affected Space Agent will formulate a supporting set of design decisions based on individual constraints and interests. These decisions may include a new building material or structural system, and are presented to the other agents with the intent of achieving global acceptance. As a result, each agent gains exposure to various alternative solutions. However, due to their autonomous nature, Space agents tend to lobby only for outcomes that best satisfy their particular interests. In other words, if left to their own devices, Space agents are reluctant to accept anything that offers a less than perfect outcome from their perspective. Inevitably, this stubbornness may lead to deliberations reaching a stalemate. In such an event, a Monitor agent enters the collaboration as a third party facilitator. The goal of this agent is to bring about agreement through assisting the agents in maintaining clarity and focus with respect to the relevant issue(s). To assist in this task, Monitor agents have several strategies to apply among the deliberating agents. Three such strategies are discussed below; namely the 'persuasive' strategy, the 'imposive' strategy, and the 'user-directed' strategy.

#### The *persuasive* Strategy

As the name implies, this conflict resolution strategy attempts to use persuasion as a means of achieving global consensus. In essence, the Monitor agent attempts to persuade one or more agents to reevaluate previously unacceptable solutions based on a more flexible heuristic. If this reevaluation leads to global consensus, the common design decision is adopted by the agents and reflected throughout the system. If the agents find suitability with multiple solution, the Monitor agent selects the solution that requires the least amount of loss for the agents. In any case, even if a common solution is found, it can be assumed that at least one of the agents is not completely satisfied with the outcome. In such an event, dissatisfied agents have the ability to express their discontent through the posting of formal protests. To perform such an action, a Space agent

formulates a report describing its dissatisfaction and turns the border of its space 'red'. Once notified by the change in a space's border color, the user may obtain a detailed description of the protest by selecting the particular space. After reviewing the agent's argument, the user may choose to modify the design in the agent's favor. However, the user may also choose not to concern himself with the agent's discontent and continue with the design. Under these circumstances the agent continues to protest the decision but proceeds with the evolving design.

In any case, if the employment of the Persuasive strategy fails to produce a acceptable solution, the Monitor agent employs the second of its conflict resolution strategies. This strategy is known as the Imposive strategy.

#### The Imposive Strategy

Employing a more forceful approach than its predecessor, Imposive conflict resolution again attempts to bring about a global consensus through compromise. Utilizing this strategy, the Monitor agent searches for a solution to essentially impose onto the deliberating agents. However, this solution is by no means arbitrary. Rather, the solution is not only a product of the initial agent deliberation, but it may in fact already be held favorably by a number of the agents. In determining which solution to select, the mediating Monitor agent searches for a majority opinion. That is, the Monitor Agent asks the question, "Is there a solution which is deemed acceptable by a majority of agents?". For example, suppose that three out of ten agents find acceptability with a certain solution 'A'. Further, suppose that of the remaining seven agents no more than two agree on any one solution. Therefore, solution 'A' would attain a majority status. In this case, the Monitor Agent would impose solution 'A' on all ten agents. Again, any agent displeased with the decision would be free to express its dissatisfaction via a formal protest. While being somewhat dictatorial in nature, Imposive conflict resolution does attempt to provide a solution that is desirable to the agents as a whole. However, if a majority opinion cannot be identified among the deliberating agent, the Monitor agent employs a third strategy in an effort to produce an acceptable design solution. This final strategy is known as the User-Directed strategy.

#### User-Directed Strategy

Considering that both Persuasive and Imposive conflict resolution strategies were unsuccessful in bringing about agent consensus, a more drastic approach is now employed by the Monitor agent. As the name implies, User-Directed conflict resolution involves the human designer as the definitive mediator. The Monitor agent initiates a dialog with the user presenting the particular dilemma at hand. In addition, the Monitor Agent provides the user with a description of the various solutions as presented by the deliberating agents. Based on this information, it is the task of the designer to decide on the most appropriate solution. However, the user is by no means confined to the solutions proposed by the agents. At any time during the design activity, the designer is free to explore any number of alternative solutions through direct collaboration with the agents. The user may select a subset of Domain agents and Space agents with which to engage in an exploratory discussion of alternatives. By allows the selection of a subset of agents, the user is able to omit particular concerns, such as sound control, cost control, or even the interests of another space, from influencing the discussion. However, the user may also choose not to become involved in resolving the conflict at the particular time thus postponing its resolution until a later date. In this case the design continues despite the outstanding conflict With the collaborative model employed within the KOALA environment, such conflicts may actually end up resolving themselves through future deliberations irrespective of whether the user becomes involved or not.

# Message Agents:

Although not within the scope of the current design, mention should be made of another possible type of agent. Up to this point, the agents described in KOALA have ranged from passive, service-oriented Domain agents to highly interactive, goal-oriented, Space agents capable of initiating action. Whether in a passive or active form, each of these agents displays some degree of functionality. A vital component of this functionality is the communication. With respect to agent collaboration, communication either triggers an action or assists in its implementation. In both cases, communication exists as passive data messages passed from one agent to another. Such messages are consequentially lacking of any functionality. Unable to provide any form of self-management, such responsibility falls on the shoulders of other parties. While the physical management of these messages is the responsibility of the communication facility, logical management is typically orchestrated by the sender.

This is where the concept of a Message agent may offer superior performance and flexibility. Suppose that messages passed between agents were, in fact, agents themselves. Autonomous in nature, these Message agents would embody the same ability to take impromptu action when faced with unforeseen circumstances as their Space agent counterparts. Accordingly, the Message agents would provide a high degree of self-management at both the physical and logical levels.

For example, if a Space agent desired a lighting analysis to be performed, the Space agent would proceed by instantiating a Message agent to represent the request. The sole purpose of this agent would be to manage the administration of the particular request. To perform such a task, the Message agent would first attempt to locate an agent that could provide the desired service. According to the agent taxonomy described in this paper, the responding agent would most likely be a service-oriented Domain agent. Once located, the Message agent would instruct the Domain agent of the nature of the request. Further, if the Domain agent is capable of supporting multiple approaches to providing the service, this instruction may even include an indication of the particular analysis method to be employed. In addition, if an unforeseen difficulty arises, the Message agent would be responsible for initiating an appropriate course of action.

Consistent with the concepts of agent autonomy and functional abstraction, there would be no need for the initial Space agent to have any knowledge of how its request was actually being carried out. In any case, the Message agent would essentially be responsible for overseeing the acquisition of the desired analysis and for conveying the results back to the initial Space agent. By refining agent granularity down to the message level, the communication itself is empowered with the ability to take action.

### **Conclusion and Future Enhancements:**

Unique among other design assistance applications, such as ICADS (Pohl et al.1992), the KOALA system combines service-oriented Domain agents, self-motivating Space agents, and facilitating Monitor agents into an abstract agent world. Each of these agents plays a specific role in providing an assistance intensive design environment. The result is a highly collaborative, dynamic model where agents deliberate among themselves based on individual, potentially biased perspectives. Interaction with the human designer is encouraged by the addition of such notification and exploration facilities as agent protest reports and direct human designer/agent collaboration. These facilities provide the designer with a powerful exploration environment where users are free to engage in private conversations with any combination of Domain, Space, and Monitor agents. The resulting system provides an environment where the human designer forms a partnership with computer-based agents in achieving common design goals and objectives.

In view of the highly collaborative, information intensive nature of the KOALA decision support system, the potential for incorporating several additional capabilities arises. While not implemented within the current design, mention should be made of two such additions. These additions take the form of an explanation facility providing insight into agent logic, and an alternative mode of design where agents are essentially given the freedom to be the driving force behind the evolving design. The following sections provide brief discussions of each of these enhancements.

# **Explanation Facility:**

The purpose of an explanation facility is to provide the designer insight into the inferencing logic of the design decisions originating within the system. Such insight allows the designer to verify that his or her intent is indeed being adequately represented within the design system. If this representation is in question, the explanation facility also aids the user in identifying which parameters need to be modified.

Allowing users to communicate in free format English sentences, explanation facilities generally support a number of different question types including *HOW?*, *WHAT?*, and *WHY?* (Klein 1994, Cawsey 1992, Myers et al. 1993). *WHAT?* questions are usually the least difficult to answer simply requiring an explanation of a certain piece of knowledge. *HOW?* questions can be answered by providing a trace regarding the process undergone by the agents in arriving at a decision. *WHY?* questions are more difficult requiring an explanation of a desired goal or objective. Explanation may attain substantial sophistication utilizing a combination of computer graphics in conjunction with English sentences to form a kind of decision network. This type of presentation may be useful in illustrating a particular series of considerations leading to a resulting design decision. Regardless of the level of sophistication, however, providing a facility where agent logic can be clarified and indeed questioned greatly assists in strengthening the partnership between the human designer and the computer application environment.

### **Assisted-Design Mode:**

Up to this point, the design activity described in this paper has been primarily userdirected. The design evolves over time through incremental changes fashioned by the designer. As modifications occur, agents attempt to accommodate such changes through supporting design decisions. While the agents are allowed considerable freedom in making such decisions, their ability to manipulate the design is not without restriction. For reasons of consistency and manageability, spaces are not permitted to resize or relocate themselves without first consulting the user. Although the reasons for the proposed agent action may be perfectly valid, the ability of the agent to unilaterally execute the action may be disconcerting to the designer and disruptive to the design activity. However, permitting such agent freedom in perhaps a different mode of design may add considerable creativity to the design activity.

With this potential in mind, the addition of an "assisted-design mode" could be provided as an alternative mode of design. In such an environment the progressing solution would be driven by the agents as opposed to the user. Initial user intent could be communicated to the Designer agent through a comprehensive interface. This intent may include user preferences for such design criteria as structural material energy conservation, and overall cost distribution. Once the intent has been communicated, the agents may engage in intensive collaboration in an effort to formulate an accommodating design solution. Attempting to support overall user intent, agents would be given complete freedom in formulating and administering various design decisions. While primarily playing an observational role, the user still reserves the right to modify the design intent at any time during the assisted-design activity. As collaboration proceeds, the user observes the design spaces as they alter both their geometric and non-geometric characteristics. Further, these design modifications are not limited to the initial set of spaces. Spaces may be added or deleted from the progressing solution as the need arises. As the driving force of the evolving design, the agents are free to explore and implement any number of design modifications while maintaining consistency with designer intent and objectives.

The enhancements described above represent only a fraction of the exploratory potential of the object-agent model inherent in KOALA. By no means limited to the field of architectural design, this approach can be applied to a wide variety of application areas including resource deconfliction and facility management. No matter what the application area, the overall objective of the object-agent model described above is to empower the design objects themselves with the ability to act on their own behalf. Together with a highly interactive collaborative model, these dynamic agents work in a cooperative fashion with the human designer to form a human-computer partnership that is rich in online exploration and intelligent agent assistance.

### **References:**

Cawsey, Alison; "Explanation and Interaction", Massachusetts Institute of Technology, 1992

Durfee E. and T. Montgomery; "A Hierarchical Protocol for Coordination of Multi-agent Behavior", Proc. 8th National Conference on Artificial Intelligence, Boston, Massachusetts, 1990 (pp. 86-93)

Durfee E.; "Coordination of Distributed Problem Solvers", Kluwer Academic, Boston, 1988

Kalay, Yehuda E. (ed.); "Principles of Computer-Aided Design: Modeling Objects and Environments", Wiley, 1989 (pp. 402)

Klein, D.A.; "Decision-Analytic Intelligent Systems: Automated Exploration and Knowledge Acquisition", Erlbaum, Hillsdale, New Jersey, 1994

Larsi B., H. Larsi and V Lesser; "Negotiation and its Role in Cooperative Distributed Problem Solving", in Huhns (ed.) Proc. 10th International Workshop on Distributed Artificial Intelligence, Bandera, Texas, 1990

Lesser V.R. and Corkhill D.D.; "The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks", AI Magazine, 4(3), 1983 (pp. 15-33)

Myers L., J. Pohl, J. Cotton, J. Snyder, K.J. Pohl, S. Chien, S. Aly and T. Rodriguez; "Object Representation and the ICADS-Kernel Design", Technical Report, CADRU-08-93, CAD Research Center, Design and Construction Institute, College of Architecture and Environmental Design, Cal Poly, San Luis Obispo, CA, January, 1993

Pohl J.; "The Representation Problem in CAD Systems", in Pohl (ed.) Advances in Computer Based Building Design Systems, 7th International Conference on Systems Research, Informatics and Cybemetics, Baden-Baden, Germany, August 15-21, 1995

Pohl J, J. LaPorta, K. Pohl and J. Snyder; "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, CA, August, 1992

Pohl J., L. Myers and A. Chapman; "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 Ohispo, CA, September, 1994

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 and Construction Institute, College of Architecture and Environmental Design, Cal Poly, San Luis Obispo, CA, 1991

Pohl J., L. Myers, A. Chapman and J. Cotton; "ICADS: Working Model version 1", Technical Report, CADRU-03-89, CAD Research Unit, Design Institute, School of Architecture and Environmental Design, Cal Poly, San Luis Obispo, CA, 1989

Schon, D.A.; "Designing: Rules, Types and Worlds", Design Studies, 9(3), July 1988

Woolridge M. and N.R. Jennings; "Agent Theories, Architectures, and Languages: A Survey", [Proc. ECAI-Workshop on Agent Theories, Architectures, and Languages], Amsterdam, The Netherlands, 1995 (pp. 1-32)

Woolridge M. and N.R. Jennings; "Intelligent Agents: Theory and Practice", (Submitted to) Knowledge Engineering Review -- Revised January 1995