CIRP Annals Vol.47/1, pp.107-110, 1998

# An Agent Supported Approach to Collaborative Design

Yan Jin, Stephen C-Y. Lu (2)

The IMPACT Laboratory, University of Southern California, Los Angeles, USA

**Abstract**

Contemporary design problems are inherently complex and involve many highly coupled sub-tasks that require multiple designers to work together collaboratively. The designers' objectives are often in conflict and activities inconsistent. How to provide effective technology to support collaborative design has been a challenge for the research community. This paper presents an agent-supported framework, called ASCAD, for collaborative design. In ASCAD, a collaborative design team is viewed as a collection of *design cells* and each design cell is composed of a designer, a software agent, and a number of computer tools. By monitoring designer's design activities, the agent in ASCAD can help its designer coordinate with other designers through identifying needs for coordination, establishing links between designers, and providing suggestions for coordination decision-making. A brief description and discussion of an initial demo application of ASCAD is included to demonstrate the effectiveness of the ASCAD framework and to illustrate future research directions.

**Keywords**: Design, Coordination, System

## 1. Introduction

The development of products, such as automobiles and buildings, is a large scale and complex process. It often takes a long time to complete and requires a large number of engineers from different disciplines to work on interdependent product components and design phases (1)(2). The engineers' objectives are sometimes in conflict and activities inconsistent. For example, in automotive body design, styling designers try to make smooth and fashionable styles of cars and may not pay attention to the strength of the overall body. Inner panel designers try to meet the strength requirement to support the outer panel but may have problems because their inner panel design may not allow enough space to place engine and other equipment under the hood. To achieve the overall quality of product and efficiency of process, engineers responsible for styling, inner panel, and equipment designs need to do collaborative engineering, i.e., to exchange information and make decisions collaboratively and coherently based on their design contexts.

Collaborative engineering involves multiple and parallel activities. To make it successful, achieving effective and efficient coordination among engineers' design activities is critical. Coordination is the key to maintain the consistency of concurrent decision-making and the integrity of information in product development lifecycle.

There are three basic issues involved in providing computer support for collaborative engineering design. The first issue is task decomposition and representation. *Task decomposition* is concerned with identifying the sub-tasks that can be divided in the way that minimizes interactions among the sub-tasks. *Task representation* is related to defining a design task and its sub-tasks in the form that can be easily handled by designers and computers to identify interactions among the sub-tasks and cross the lifecycle of product development.

The second issue is the need for a *communication infrastructure* to facilitate communications among designers. Since in most cases complete task decomposition, i.e., no interactions exist among sub-tasks, is impossible, a sophisticated communication infrastructure is needed to facilitate flow of information among designers. Progress has been made recently to improve the system level communication by increasing the interoperability of computer systems (3) (4) (5).

Given task decomposition, representation and communication infrastructures, the third issue—*coordination support*—still remains. *Coordination* is generally considered as the activity to resolve dependencies among sub-tasks. From a designer's point of view, coordination means to take into consideration decisions made by others in making local decisions as well as to provide information for other designers when needed. Active *coordination support* should help designers identify needs for coordination, create contents of communication to address the needs, select relevant parties to send the communication and resolve conflicts among the coordinating parties if any.

Our research on ASCAD, agent-supported collaborative design, addresses the issue of coordination support. We have conducted our research on collaborative design in automotive design domain and specifically focused on the automotive inner panel design problem. In the rest of this paper, we first present the ASCAD framework and then briefly discuss our initial ASCAD prototype and its application to collaborative inner panel design problems.

## 2. The ASCAD Framework

### 2.1 Design Cell

Figure 1 provides a partial view of collaborative design in a computer-aided design environment. Each engineer is working with a computer system. While design results and other formalized information are stored in computers, less formalized and highly ambiguous issues related to the design (e.g., design rationale) are usually in engineers' mind. During collaborative design, an engineer can coordinate and communicate with other designers

through two channels: human level communication and system level communication, as shown in Figure 1.

Coordination in collaborative design takes time and requires knowledge as well. If a designer has an assistant to help him identify needs of coordination, manage computer tools and interaction with other designers, and perform delegated routine tasks, then the designer would have more time to concentrate on creative design activities, and trying more alternatives and more analyses. This assistant-supported idea led us to adding intelligent agents into the picture of Figure 1 and introducing the concept of *design cell* as a combination of a designer, a computer software agent associated with the designer, and a number of computer tools used by the designer, as shown in Figure 2. A collaborative engineering environment usually consists of multiple interacting design cells.

Designers play the primary role in a design cell. They perform creative tasks in design such as problem analysis, conceptualization, decision-making, materialization and evaluation. These activities are the core of design that entails deep thinking and sparkling imagination.

In ASCAD, a tool is defined as a computer process that can be used by the user or agent to solve certain design or communication/coordination tasks. Examples of tools include CAD modelers and FEM analysis programs. For a tool to be part of a design cell, it must be able to interoperate with its agent. In ASCAD, the interoperation between tools and their agent is implemented through defining an interface on the tool side. The interface is called *tool actions*. A tool action is defined by a triplet (<Action>, <Info-Needed>, <Info-Produced>). <Action> specifies the name of action or process the tool can perform. <Info-Needed> specifies what information is required to perform the specified action, and <info-Produced> denotes the information or product that will be produced by the tool through the specified action.

### 2.2 Agent as a coordinator

The agent in a design cell performs following coordination activities:

*Identify coordination needs:* An agent identifies need for coordination either by receiving an explicit request from its designer or tools, or by noticing exceptions in monitored design and coordination actions. When a tool needs some information which it does not have access, it will issue a RFI (request for information) to its agent. The designer may also issue a RFI or RFA (request for action) to the agent to get certain task done. Furthermore, the agent is monitoring its designer as well as tools. When the designer makes a design, for example, the agent will check if there is a need to propagate the design change. In ASCAD, an agent evaluates the impact of design change and new information by consulting *a constraint and dependency manager* tool. If any constraint is violated due to the design change of the new information, the agent will go to the next step to start a coordination session.

*Find with whom to coordinate:* After a need for coordination is identified, the agent then searches for who should be the other party or parties of the coordination. If the coordination is RFA, then the agent will find an action performer from its knowledge about registered tools and agents. If the coordination is general FYI (for your information) then the other parties will be determined

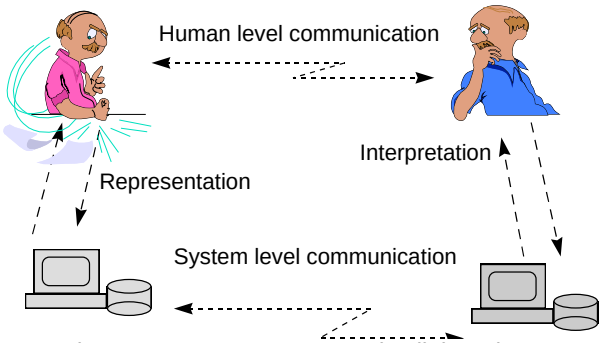


Figure 1: Computer supported collaborative design environment

based on the interests of the other agents. If the coordination is design change propagation, then the range of propagation will be determined by consulting the constraint and dependency manager and based on registered agent interests.

*Establish coordination links:* After the other party or parties of coordination are determined, the agent then establishes a direct link between the coordination parties. Establishing a coordination link is not only providing a communication channel but also to provide a context of communication. For example, in a process of three parties negotiating about a design change, it will be very helpful to visualize the same part of a product model at three different locations.

*Monitoring coordination process:* Once the coordination link is established, the agent starts a new "coordination record page" and record the process of coordination as much as it can. The record will be used later for resolving conflicts when they occur.

### 2.3 Agent as a personal assistant

Besides providing direct coordination support, an agent can also assist its designer in design and other peripheral tasks to save designers' time and avoid mistakes that are easy to make for human designers.

*Routine design support:* Design activities of a design cell usually involve multiple computer tools working together. Practically, a designer has to manage the execution of the tools during design. When the procedure become a routine, it will be appropriate to have the agent to mange this kind of "local collaborative design" activities. During local collaborative engineering, the agent of the design cell coordinates activities of the tools either by following pre-defined procedures, or based on the coordination

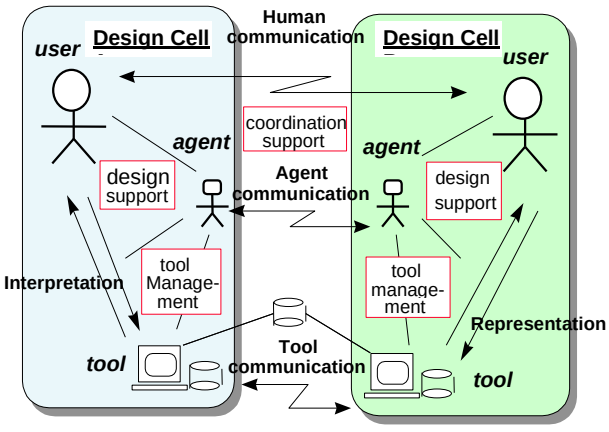


Figure 2: Design Cell: designer, agent, tools and their interactions

steps described above. Our experience has shown that most local collaborative design tasks follows certain work flows.

*To-Do-list management:* By monitoring users and interaction with other users, an agent creates a commitment list for the user. The commitment list may also contain the items told by the designer.

## 3. The I-Agent Architecture

In the ASCAD framework, agents play a central role for coordination. The design and implementation of the agent model is critical for our research. The agent model must satisfy several requirements. First, it must be able to possess knowledge required for coordination support. To facilitate coordination, an agent needs to maintain its information at both design object level and tool operation level. It must be able to apply the knowledge to make coordination decisions. Second, the model should be general enough so that it can be used to support a wide range of interactions over product development lift-cycle. Third, it should be easy for users to customize agents to adapt to their own collaborative engineering environments. Last, the agent should be able to work autonomously so that the design cell can keep working even in the absence of the user.

To build an agent that can satisfy the above requirements, we developed a generic agent structure called I-Agent (6). Figure 3 illustrates the conceptual architecture of the I-Agent.

### 3.1 Agent Configuration

As shown in Figure 3, an I-Agent has two inputs and to outputs. The vertical input-output represents the communication capability of agents and the horizontal one depicts the capability of "perceiving" or "monitoring" and "action" capabilities. These input and out relationships are managed through a set of internal actions: *information filtering*, *belief update, commitment decision*, and *committed problem solving*. The agent internal actions are further governed by agents' cognitive attributes.

Having cognitive attributes is the main feature of I-Agent and distinguishes it from regular programs. An I-Agent's cognitive attributes fall into two categories: *character* and *mental state*, as shown in Figure 3. While *character* describes relatively static characteristics of an agent, the *mental state* of an agent represents the agent's cognitive model of the real world. Following is a brief description of the cognitive attributes:

*Values:* The values of an agent specify the criteria used by the agent to select its goals.

*Interest:* An agent's interest defines problem domains in which the agent is interested.

*Capability*: The capability of an agent specifies the potential for the agent to *directly* perform some action.

*Expertise*: Domain dependent knowledge (e.g., rule-base) that is required for an agent to perform functional tasks.

*Strategy*: Strategy specifies the general plan for coordination behavior that constrains responses to the incoming message and balances the agent's local interests with global (team) interests.

*Goals***:** Generally, a goal is a proposition that the agent tries to make true and the proposition can be anything.

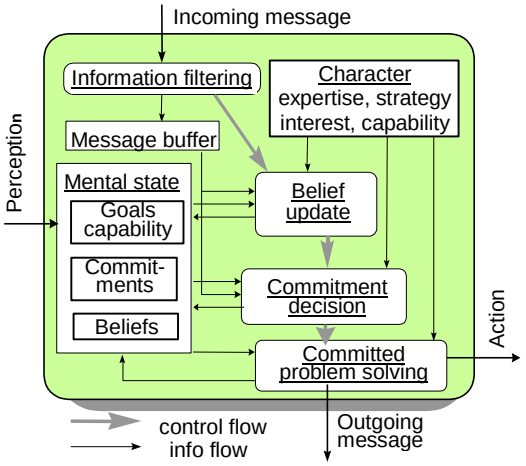


Figure 3: I-Agent Architecture

*Social-role:* Social-roles are expectations for, or evaluative standards employed in, assessing the behavior of occupants of specific social positions.

*Capacity:* The capacity of an agent describes the availability of resources that are required for the agent to perform actions.

*Commitments:* The commitments of an agent specify the constraints on future actions of the agent. Commitments create mutual beliefs in a collective plan among agents, as well as responsibility that holds each agent to his or her part.

*Beliefs:* Beliefs of an agent construct a cognitive model of the real world that is composed of the physical world (e.g., design task) and the social world (e.g., reporting relations between designers).

## 4. The ASCAD Prototype

The ASCAD prototype system has been implemented in Java language. Figure 4 illustrates a diagram of a design cell implementation of ASCAD and Figure 5 shows a screen image of the agent-related windows.

As described above, a design cell is composed of a designer, an agent that assists the designer, and a set of tools. The designer works with the agent and design tools through a set of graphical user interfaces.

The agent of a design cell talks to computer tools and other agents using KQML and KIF. A limited ontology set called ASCAD Ontology was developed to facilitate agent-tool and agent-agent communication. As a computer tool, AutoCAD has been integrated into a design cell through a tool wrapper program, as shown in Figure 4.

While external tools are "plugged-in" through wrappers and KQML-KIF for flexibility, local tools, such CDM and PPM in Figure 4, are linked to agent directly for efficiency. In this sense, local tools are exclusive functional components of agents, and are under full control of agents. From an agent's point of view, the distinction between external and local tools has important implementation implications. First, it clarifies who has control over which tools and makes information management responsibility clear. Second, tight link between agent and internal tools can significantly improve the performance of tools and agents.

## 5. Demo and Discussion

We conducted a demo application for two designers to collaborate with each other on a partial inner penal design

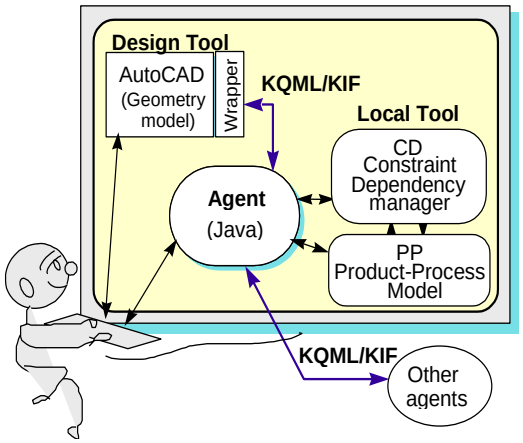


Figure 4: ASCAD Prototype

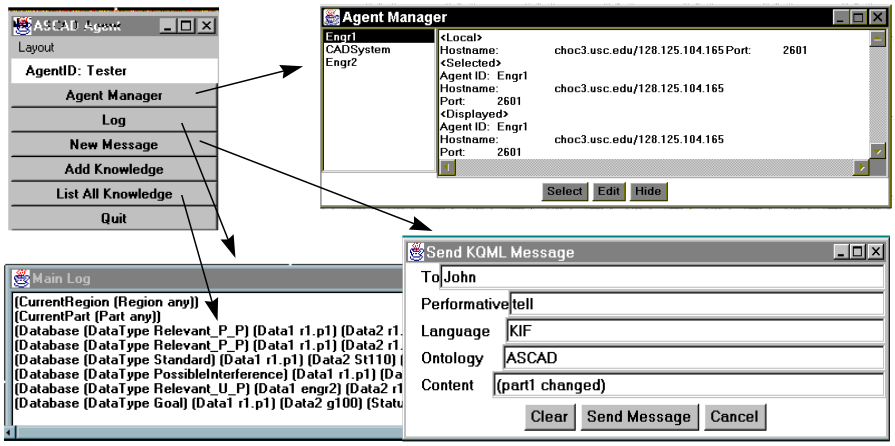


Figure 5: ASCAD Agent and Related Windows

problem. In the following we briefly report the reaction from the initial evaluation and discuss some of our observations.

### 5.1 Reactions from on-site demo

To evaluate the ASCAD idea and get feedback for future research, we have done an on-site demonstration to a group design engineers using our initial prototype system. The over all reaction was positive. When asked “do you think this kind of agent-support is useful for your work?”, most people responded “very useful”, and some felt “hard to say”. For the question “when will you need a real ASCAD system?”, many engineers said they will need it within 1 to 3 years. A close look at the survey after the demo reveals that the engineers whose design tasks require intensive interaction or coordination with others highly valued the ASCAD demonstration and they liked the agent-support idea. One common question to the ASCAD approach was how to manage agent knowledge both globally in a team and locally in a design cell.

### 5.2 Personal Agent

The research on agent-based systems thus far has assumed an agent is a tool (or a tool with an agent wrapper) or a combination of a human user and a tool (5) (7). In these systems, tool agents communicate with each other through a facilitator. The agent in ASCAD is an independent process. It is general purpose and is constantly ready to help its designer. It is similar to but different from a facilitator in that it is a personal agent and under full control of its user.

There are several advantages of this personal agent approach. First, linking a tool to an agent is relative simple. It is relative easy to define a tool action set for a tool and the user of a design cell can change both agent and tool to meet any link requirement. Second, independent or routinized local collaborative engineering processes can be delegated to the agent and carried out automatically. Third, by introducing management organization structures, agents of different design cells can be organized either hierarchically or as a flat team. Agents can have different authorities to information access and decision-making.

One disadvantage of the personal agent approach is that in a flat team structure, agents’ knowledge about the team can be highly redundant, e.g., each agent has to know all other agents’ interests. This problem is not obvious for a small team. One way to overcome this problem when the team is getting large is to introduce organization structure among agents.

### 5.3 Domain representation and design context

One of the goals of ASCAD development is to achieve context-based coordination support, i.e., decisions about whether there is a need for coordination, who should be involved and when to coordinate should be based on the design context. In ASCAD, design context has two portions. One is the structured design object (or product) representation and the other is the record of operations and interactions of the designer and tools. To achieve effective context-based support by an agent requires highly formalized domain representation, but this may make it difficult to scale up the system. We plan to get around with this dilemma by paying more attention to operation support and leaving the design details to the designers. Our current research is focused on developing a *decision-based collaborative design process model* to record designers’ communication and decision-making processes and to provide agent-based *preventive*, *informative* and *reactive* coordination support for the designers based on the process model.

## 6. References

(1) Sohlenius, G. (1992) Concurrent Engineering, Keynote paper, CIRP Annals, Vol.41/2:pp489-496.

(2) Krauze, F.-L. T. Kiesewetter, S. Kramer, (1994) Distributed Product Design, CIRP Annals, Vol.43/1, pp.149-152

(3) Cutkosky, M.R., Engelmore, R.S., Fikes, R.E., Genesereth, M.R., Gruber, T.R., and Mark, W.S. (1993) PACT: An Experiment in Integrating Concurrent Engineering Systems, IEEE Computer, Vol. 26, No. 1, pp 28-37.

(4) Genesereth, M. and R.Fikes (1992) Knowledge Interchange Format. Report Logic-92-1, Dept. of Computer Science, Stanford University, CA.

(5) Genesereth, M.R. and S.P. Katchpel, (1994) Software Agents, in Communications of the ACM, Vol.37, No.7, pp.48-53.

(6) Jin, Y. and R.E. Levitt, (1993) "i-AGENTS: Modeling Organizational Problem Solving in Multiagent Teams", in *International Journal of Intelligent Systems in Accounting, Finance and Management*, Vol.2, No.4 pp.247-270.

(7) Khedro,T. and M. Genesereth, (1994) The Federation Architecture for Interoperable Agent-based Concurrent Engineering Systems, Concurrent Engineering, Research and Applications, Vol.2 pp.125-131.