Gero JS and Brazier FMT (eds) (2002) *Agents in Design 2002*
Key Centre of Design Computing and Cognition, University of Sydney, pp. 39-54

USER MODEL OF A PERSONAL ASSISTANT IN COLLABORATIVE DESIGN ENVIRONMENTS

SHUMIN WU, HAMADA GHENNIWA
Department of Electrical and Computer Engineering
University of Western Ontario Canada

and

WEIMING SHEN
Integrated Manufacturing Technologies Institute
National Research Council Canada Canada

Abstract. During the past decade, intelligent agents have been applied in attempts to a number of collaborative engineering design systems. Intelligent interface agents or personal assistants play a very important role in such systems. A fundamental issue in designing a personal assistant is how to build adequate models of the user and the domain. This paper proposes a cognitive user model of a personal assistant in collaborative design environments, which comprises a user interest model, a user behavior model, an inference component and a collaboration component. These models and components can help agents effectively achieve the user's goals. The proposed user model is being implemented in a coordinated, intelligent, rational agent (CIR-Agent) model on a FIPA-compliant platform. A case study in a multidisciplinary design optimization (MDO) software environment is also presented.

1. Introduction

During the past decade, intelligent agents have been applied in attempts to a number of collaborative engineering design systems (Shen et al. 2001). Intelligent interface agents or personal assistants play a very important role in such systems. The notion of personal assistants originates from the intersection of Artificial Intelligence (AI) and Human-Computer Interaction (HCI) areas and its development involves technologies such as Internet/Web technologies, multi-media, and real-time technologies. The agent technology that flourishes in recent years gives a renaissance to the personal assistants. A personal assistant can be designed as an autonomous intelligent interface agent (FIPA 2001) that assists the user with daily computer-based tasks (Lashkari et al. 1994). Since one of the primary objectives of a collaborative design environment (CDE) is to facilitate the communication and cooperation among human designers, effective personal assistants in such environments should be capable of collaborating with other agents (including human agents) in the same environment to achieve the user's goals and satisfy the user's needs. A fundamental issue in designing a personal assistant is the cognitive user modeling. The ideas of modeling the user and the domain are not new. There has been a significant amount of research on different user modeling approaches over the past 20 years, but user modeling for personal assistants faces new-sprung challenges through constructing and evolving adequate user models in terms of their representations and inference approaches. In collaborative design environments, user modeling is even more difficult due to the geographically distributed nature of these environments and the multidisciplinary characteristics of collaboration.

This paper proposes a user model for a personal assistant in collaborative design environments, which comprises a user interest model, a user behavior model, an inference component and a collaboration component. The goal of this research work is to find an effective and efficient way of user model representations, inference approaches, and collaboration mechanisms to assist the users with their collaborative tasks.

The rest of the paper is organized as follows: Section 2 provides a brief literature review; Section 3 describes the proposed user model; Section 4 presents a personal assistant designed for CIR-Agent; Section 5 is concerning the implementation issues followed by discussions on the user modeling issues in Section 6; and Section 7 concludes the paper.

2. Literature Review

User modeling has become increasingly important in designing personal assistants. Researchers have examined special issues in designing user models for a wide variety of intelligent interface agent applications and used different approaches to resolve the problems generally in terms of (1) dynamism of the user's interests and preferences, and (2) uncertainty of the user's behaviors and goals. A utility theory-based user model for an active user interface was proposed by Brown et al. (1998a) to handle the dynamism of the user's interests and preferences in information retrieval and knowledge discovery tasks. The user model is described in terms of a user profile to store the static knowledge about the user, a Bayesian network model to capture the uncertainty of the user's actions and goals, and a utility model to capture the user's utility to achieve a goal (Brown et al. 1998a). Explicit requirements and metrics are introduced to enhance the accuracy of the user model (Brown et al. 1998b). The work by Horvitz et al. (1998) mainly deals with the uncertainty in the reasoning about the user's goals. The approach involves building and assessing Bayesian models and obtaining a stream of events based on the user's behavior, for which Lumière, an event language, has been developed to transform events into observational variables in Bayesian models, and persistent profiles to capture changes in a user's expertise. The Personal Assistant in Microsoft Office applications was derived from the Lumière project. The work by Fleming and Cohen (1999) investigated several effective user agent interaction mechanisms including: always consulting rules defined by the user first; using Clarification Factor to determine whether to bother the user to make decisions; and allowing the user to view/update agent’s user models. The objective was to allow the user to update the agent’s user model and to restrict the agent’s interaction with the user so that the user won’t be bothered unduly.

Considering the user’s interests vary over time and the agent only has partial knowledge about the user at one time, a user model of a personal assistant should consistently evolve as a result of the accumulating experience with the user. Early studies in machine learning provide the opportunity to enable a user model to learn from a user, i.e., the user plays the role of the learning environment (Langley 1999). In fact, some ideas that have been adopted in user modeling today are originally motivated by machine learning, e.g., the separation of the user’s short-term and long-term interests comes from the multi-strategy machine learning approach. One example is a hybrid user model for news story classification (Billsus and Pazzani 1999) that consists of a short-term and a long-term user interest models. Therefore it is possible to choose from different algorithms that suit each model, e.g., they model user’s short-term interests with the nearest neighbor algorithm to track the user’s multiple and novel interests, while model long-term interests with a Naïve Bayesian Classifier to learn the probabilities and to classify the news stories labeled with their probabilities of belonging to interesting classes. The system can thus produce individualized news stories and adapt to the user’s preferences according to the user’s feedback.

All of above-mentioned approaches have addressed more the modeling of the user's interests and behaviors, but less about collaboration strategies for groups of users to carry out the goal-directed collaborative tasks. Exploring a user model that suffices the user’s needs in a collaborative environment is still an ongoing research activity. To offer the user with desired assistances at the right time in a felicitous manner has an ever-growing sophistication in collaborative environments, because novel applications with diverse interfaces have been largely employed these days, but users who have their own design experience on different sets of software are reluctant to switch to unaccustomed interfaces. This problem known as “organizational boundary” (Monell and Piland 1999) is significant especially in multidisciplinary design environments that require information exchange to be performed media-independently. Additionally, the user will exhibit more complex behaviors in collaboration, which gives rise to challenges of identifying the user's goals with the uncertainty about understanding the user's behaviors.

3. A Cognitive User Model of a Personal Assistant in CDE

The cognitive user model should involve models to capture the user’s interests and behaviors, because they represent two different dimensions of the agent’s knowledge about the user. The user's interests are the topics or subjects that the user is interested in, whereas the user's behavior is the user's way to achieve the goals. Furthermore, the separation of the representation of a user model and approaches such as assessing and evolving a user model enables a flexible way of exerting various inference strategies on the user model. Lastly, in collaborative design environments collaboration strategies can vary from user to user and time to time. Extricating them from other inference strategies and designing a separate model to encapsulate them can provide a more flexible way of modeling the user’s collaborative behaviors. Based on these ideas, our user model is designed to comprise two sub models and two components as described below.

3.1. THE USER MODEL STRUCTURE

A proposed user model for a personal assistant is a model consisting of two sub models correspondingly to capture the user’s interest and the user's behavior, and two components that encapsulate the approaches for reasoning and decision making based on the two sub models (Figure 1).

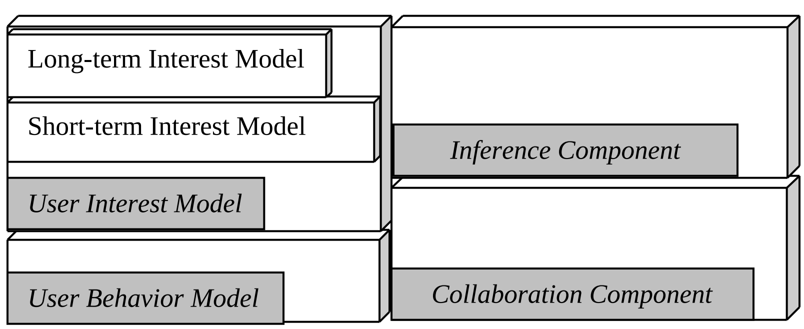

Figure 1. User model architecture

The user interest model is further decomposed into two parts, namely the user short-term interest model and the user long-term interest model. The short-term interest model and long-term interest model exhibit an agent's different levels of sophistication of modeling the user's interests. Empirically, the agent will perform more efficient tasks if it can access a wider range of knowledge about the user. The short-term interest model keeps knowledge of the user’s novel interests in relation to current goals, which handles the dynamism of the user’s interests in timely fashion. The long-term interest model keeps knowledge about the user’s interests over time and takes advantage of the comparable stability of the user’s long-term interests.

The behavior model aims at modeling the user’s actions, and identifying the precedency relations between them. It keeps the observations of the user’s behaviours from a hole on the user interface.

As mentioned above, the approaches applied to the two sub models to do appropriate inferences can be decoupled from representations into the inference component. The collaboration component contains a set of collaboration strategies that will help the personal assistant agent to execute the collaborative tasks in CDE. These two components support the agent’s reasoning process so that the agent can offer appropriate assistances to the user.

3.2 REPRESENTATION OF THE USER INTEREST MODEL

A formal method is proposed to represent the mental state of an agent, which is described in terms of the representation of each element and their relations in user modeling. In order to build the user’s interest model, the agent has to conceptualize the user’s interests in terms of the elements, C_i, of the domain given by $D = \{C_1, C_{2,\ldots,} C_n\}$ and the relationships between them, R, for any given possible world W. Formally, this conceptualization process can be depicted as a structure S, where

$$S = < D, R \mid W > .$$

A language denoted as L is employed to describe the user interest model. This language has a vocabulary V that represents S, and an interpretation I to map V into the elements of S, such that

$$I : V \rightarrow D \bigcup R .$$

For example, suppose a user interest is “agent” in the context of computer science, and therefore, the conceptualization of the user interest can be represented using English language as:

D =<*Agent, Artificial Intelligence*>, R =<*is_in*>, W =<*Computer Science*>.

In a different context, such as traveling, although “agent” has the same representation in syntactic, the conceptualization of the user interest of “agent” will be represented as:

D =< *Travel Agent, Agent*>, R =<*is_a*>, W =<*Travel*>.

To apply the above representation for modeling the user’s interests we extend the structure representation to capture the user’s preferences as follows:

$$UM_I = <(D \times w_D), (R \times w_R)>,$$

where UM_I is the user’s interest model, w_D, w_R implies the user’s preferences for the concepts and for the relationships, respectively. Therefore, different users might yield different user models within the same context (world).

In implementation, there are several types of languages that can be used to describe the user interest model, such as first-order predicate logic, semantic networks, and conceptual graphs (CG). Among them, we find the *semantic network* (Quillian 1968) is a powerful tool to represent the semantic knowledge about the user’s interest model. A semantic network is a declarative graphic notation in which the nodes represent the concepts and the arcs between the nodes represent the relations among the concepts. It is semantically rich enough to support effective conceptualization in our problem, as one node can lead to all the other related nodes spreading out through the network and the arcs between the nodes can describe all the relations in this domain between the concepts such as inheritance, generalization and even more complex relations. Semantic networks also have some limits at representing the relations with qualifiers, such as “some of” and “all of”. However, since in our problem the user’s interests will focus on the concepts within one specific world, which means the qualifiers of the concepts can be always paraphrased as “all of”, the semantic network is still adequate. The semantic networks provide a way to represent the structural knowledge in integrated and interrelated concepts. Those concepts can be associated with a list of properties, which describe the attributes of the concepts. One of the attributes for the concept node can be used to represent the *weight* indicating the user’s preference for those interests.

Using KL-ONE (Brachman 1979) like semantic networks, the user model can be implemented by representing the relationships as special nodes (or roles) associated with names and weights. The example mentioned above is shown in Figure 2.

In Figure 2, the double-line arrows denote the *inheritance* arcs such as arcs from *Travel Agent* to *Agent* and from *Personal Assistant Agent* to *Agent*. The single-line arrows with a circle represent different types of relations, with which the *Travel Agent* can be linked to *Price* as the travel agent will ask for price, and its weight shows the user's preference of the price, i.e., <*ask_for*<*Travel Agent, Price*>, w_R>.

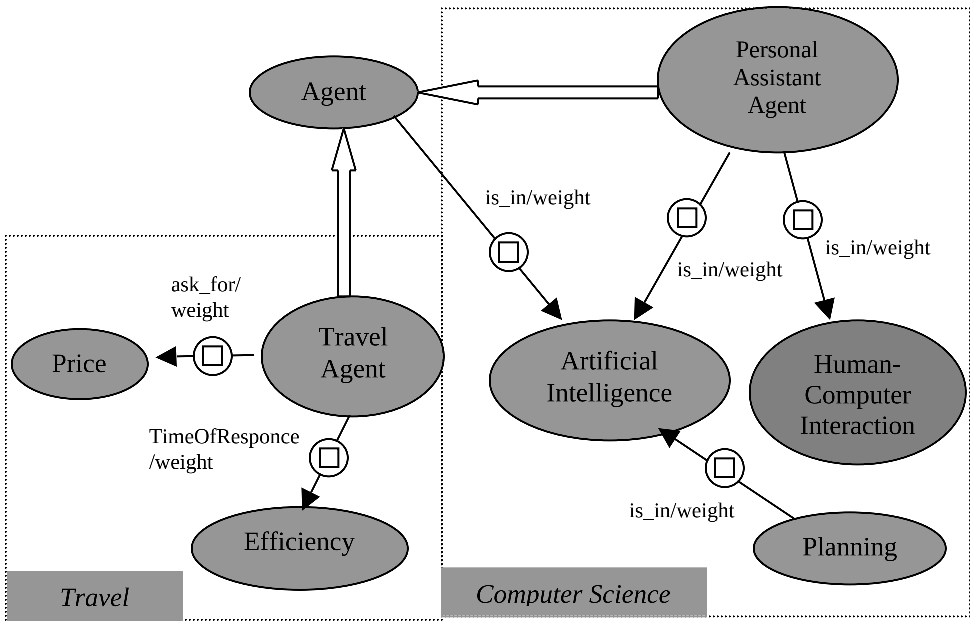

Figure 2. Semantic network

3.3 REPRESENTATION OF THE USER BEHAVIOR MODEL

A user behavior model will keep the following information: (1) observed user's actions; (2) contexts such as specific applications and workplaces; (3) user's predefined rules; (4) the history of the assistances provided by the agent; (5) the cases of tasks and their solutions. This paper covers only the first two aspects.

To assist the user to achieve his/her goals, the agent will observe the user's actions when he/she performs tasks. The user's actions in a specific context can be classified into various types, such as operations on tools, menus, or graphical components in a word processing application. Each type has primitive actions, such as selecting the "view" button, opening a URL, and placing a query, etc., which cannot be divided further. Akin to the user's interest model, we use semantic networks to represent the user's behavior model, in which the primitive actions are represented as nodes. The arcs between the nodes represent the precedence order among the nodes. The weight assigned to an arc in the network represent the likelihood that the user will choose the precedent action given the parent node (action). Formally, the behavior model can be represented as:

$$UM_B = < (a, a', p(a / a' \mid a, a' \quad A)) > ,$$

where, A denotes the set of primitive actions that the user might perform to achieve its goals, and $p \in [0,1]$ to represent the likelihood between two nodes. Figure 3 shows an example on how these primitive actions are organized in the network.

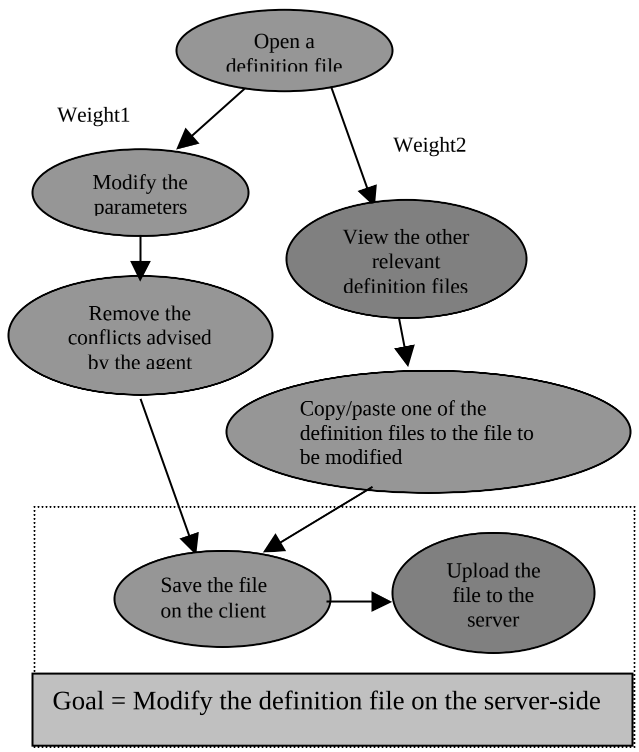

Figure 3. The user behavior model

The *Naïve Bayesian Network* is adopted to implement the representation of the user behavior model. One of the important reasons we use Bayesian network is that it is capable of dealing with incomplete data sets, which enables a personal assistant to formulate plans even when it merely possesses partial knowledge of the state of the world.

Assume that the user's goal is to modify a definition file and upload it to a server, and that it must be consistent with other definition files in the project maintained by a group of users. The user will either modify the parameters in the old file or directly copy from another definition file. The tree-structured network (Figure 3) displays the two possible paths that the user would take to achieve the goal. Each node will maintain a probability table that will be used to calculate the likelihood of sequences by the modeling approaches.

3.4 THE INFERENCE COMPONENT

The inference component together with the user interest model and the user behavior model can provide three levels of assistances: (1) advising the user what are the correct actions he/she should take to achieve the goal especially when the agent finds he/she does not act on the right track; (2) proactively offering possibly desired assistances based on prediction of the user's next step; (3) maintaining the user interest and behavior models by adjusting the likehoods of arcs. Let's take a further look at how these three levels of assistances are achieved.

First, the inference component will track the user's actions and compare them with sequences of actions in the network. Once it detects the user does not act on the right track, e.g., that the user consistently plays cyclic actions suggests the user does not know how to proceed, the agent will help the user out from the confusion by advising the user to do proper actions.

Second, as decoupling the inference approaches from the user interest and behavior models allows us to use flexible and adequate strategies to accurately predict the user's intention based on the structural knowledge the agent has established in the user model, we can apply various belief network approaches to assess and update those models. For example, we can apply the Bayesian inference and utility based decision approaches for the user's behavior model. In the behavior model, each arc between two nodes is associated with one probability (weight) represented as $P_r(\theta / A, K)$, where θ is the random variable that denotes a new observation of the user's action, A denotes the set of other observed random variables, i.e., the origin nodes of the arcs that come to θ (the actions taken before the next step), and K is the domain knowledge. In order to find the branch with the maximum utility in the Bayesian network that represents the agent's belief of the user's behavioral preference, we compute the likelihood for each path when the user's action sequence reaches at a certain node. Initially, we will assign the probabilities in a conditional probability table (CPT) for each node and then revise it as more information (or observation) is gathered. Because Bayesian inference is based on the posterior probability, we can assign the initial value of the priori probability based on our belief, then use Baye's theorem to calculate the posterior probability as follows:

$$P_r(\theta / A, K) = P_r(A / \theta, K)\, P_r(\theta / K) / P_r(A, K)$$

where,

$P_r(A, K) = \sum P_r(A/\theta, K) P_r(\theta / K)$

and the $P_{r(A/\theta,K)}$ is the prior probability and $P_{r(\theta/K)}$ is the likelihood of θ using the values in the CPT in each node.

Third, the inference model takes the responsibility to maintain the user model. It will update the likelihood of each arc according to the user's feedback. Bayesian updating methods based on binomial or normal models can be used to modify the agent prior assumptions on the probability distribution among several arcs that derive from the same node.

3.5 THE COLLABORATION COMPONENT

The collaboration component encapsulates diverse collaboration strategies that can be applied to the user's interest and behavior models to assist collaborative users with their collaborative tasks. In a collaborative environment, users can obtain the desired documents, design ideas and suggestions without knowing the information resources. The personal assistant is responsible of locating the appropriate information and providing the timely adequate advise to each user.

Currently, we are investigating the issue of identifying users that share the same interest. Utilizing the user interest model, this type of collaborative task can be carried out through a collaborative reasoning about collaborative users' interest models. For example, two user interest models UM_{I1}^{D1} and UM_{I2}^{D2} represented as $D_1 = \{C_1, C_2, ..., C_n\}$ and $D = \{C_{1'}, C_{2'}, ..., C_{n'}\}$, suggest a potential collaboration between the two users (User 1 and User 2), if $D_1 \cap D_2 \neq \phi$. In this case, the personal assistant of User 1 can ask the personal assistant of User 2 for sharing information of interest on behalf of User 1. Additionally, the execution of the collaboration requires coordination and interaction devices (Ghenniwa 1996) between the agents.

4. The CIR-Agent Model for a Personal Assistant

Our personal assistant is built as an interface agent, particularly based on the coordinated, intelligent, rational agent (CIR-Agent) model (Ghenniwa and Kamel 2000). The CIR-Agent model is a generic agent model for cooperative distributed systems that can be applied in designing interface agents in collaborative design environments. The model consists of the following elements, Figure 4:

- *knowledge* – that contains the information about the user, domain knowledge and the agent environment. In a personal assistant agent, the user model contains the aforementioned sub models and components.

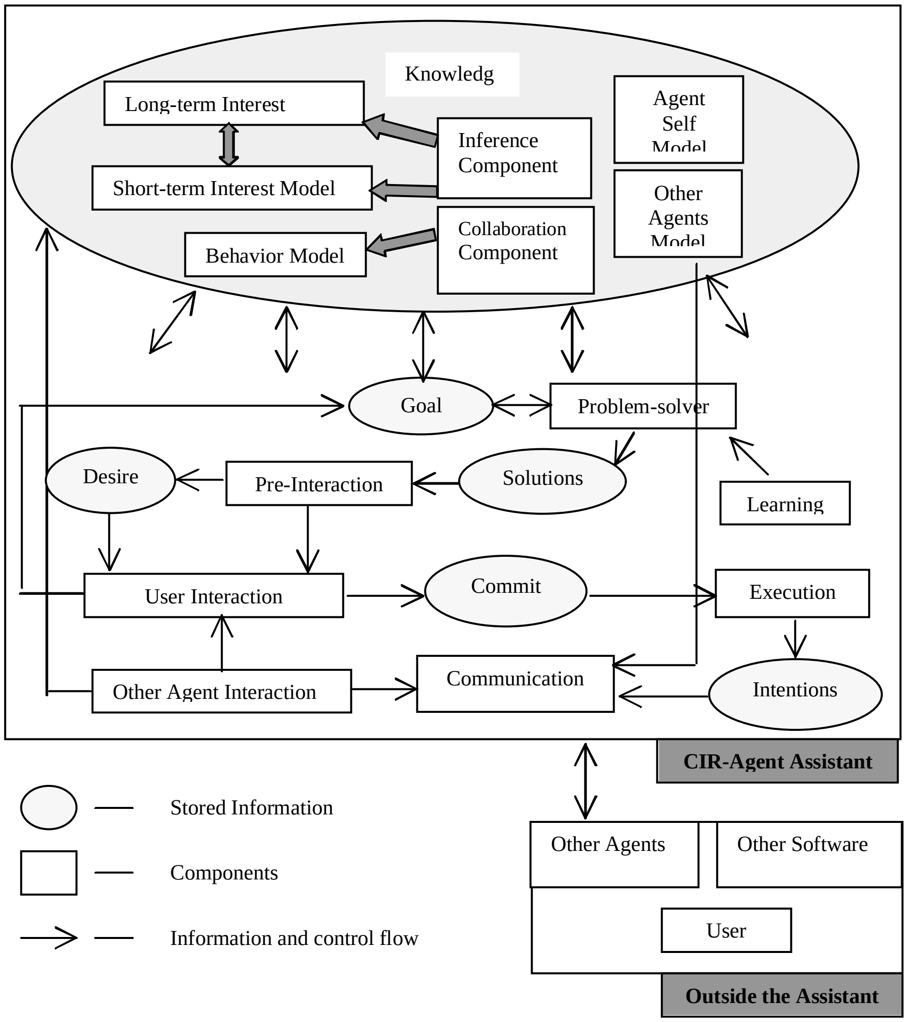

Figure 4. CIR-Agent Architecture

- *problem solver* – that extrapolates and reasons from its knowledge in order to derive an optimized action plan for the user.
- *communication* – that allows the communication between a personal assistant agent and the outside world.
- *interaction devices* – such as:
  - *assignment device* – that allows the personal assistant to interact with the user and other agents to achieve common goals. The Contract Net Protocol (Smith 1980) can be used to select a target agent for a specific task when the current agent is limited by its capability to solve the problem alone.
- *user model maintainer* – that updates the user model according to the system or the user's requirements.

- *user model inference component* – that infers from the user model to predict the next step of the user so that the agent might offer assistances proactively.
- *user model collaboration component* – that encapsulates diverse collaboration strategies.

In the architecture shown in Figure 4, the agent's *knowledge* includes the user model as well as the agent self model and the model of the other agents.

5. Implementation

The CIR-Agent is being implemented in a way that conforms to the FIPA (Foundation for Intelligent Physical Agents) standard (http://www.fipa.org/). FIPA allows the interoperability among the agents and the interaction with other heterogeneous agent systems. As FIPA standardized the three types of interfaces: User-Agent Interface, Agent-Agent Interface, and Agent-Other Software Interface, and the protocols of interaction (FIPA 2001), the FIPA compliant work will enable the multi-modality in interaction, i.e. one single application can run in one or more modalities such as smart-boards and palmtop computers in media-independent fashion.

There are several FIPA-compliant agent platforms developed by different institutes. One of them is JADE (Java Agent Development Framework), a software development framework and distributed application development environment for the implementation of multi-agent systems (http://jade.cselt.it/). Based on this platform, we are developing personal assistants that take advantage of the existing JADE contributions such as the communication (JADE employs RMI technology for communications between JVMs), AMS (Agent Management System), ACC (Agent Communication Channel), and DF (Directory Facilitator). However, the internal architecture of a personal assistant is based on the CIR-Agent model as presented in the previous section.

5.1 A CASE STUDY

A multidisciplinary design optimization (MDO) software environment (Shen and Ghenniwa 2001) is being developed for the design and optimization of blow molded parts such as bottles, containers, toys, etc. The proposed approach includes distributed system integration using intelligent agents and Internet/Web technologies, as well as multiple optimization methods including gradient-based optimization techniques and soft computing techniques. Agents are used to wrap/represent software components (e.g., BlowSim, BlowLoop, Parmesh, BlowOp and BlowView, etc.) as well as interfaces for human designers. Web server agents are implemented to facilitate the integration of various software components from remote locations and separated by firewalls. All these agents form a heterogeneous multi-agent system, Figure 5.

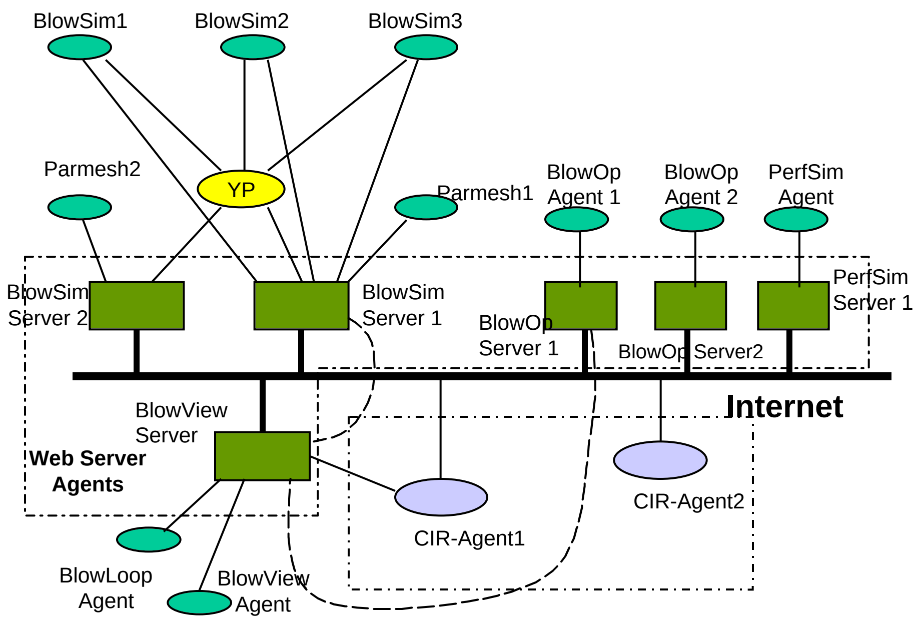

Figure 5. MDO collaborative environment

In this MDO environment, designers initiate design and optimization projects, monitor design and optimization processes through the Web-based graphical user interfaces. Each designer will have his/her own interested projects/files and perform routine tasks such as uploading definition files, sending emails, and viewing the results of blow modeling processes. A group of designers including a project manager and a number of project members will collaborate through the Internet. A user (designer) may play different roles in several different projects. Users can change the parameters in the definition files or through the dialogs to control the blow molding processes provided that they have enough rights and the changes they have made are consistent with other definition files.

Personal assistant agents are employed to provide assistances to designers and to facilitate the collaboration among project members in this software environment. A personal assistant guarantees the appropriateness of the user's modification operations. It also finds out the other agents who share the interest with its user. The user short-term interest model records the user's recent accessed projects/files. At a certain period, the system will store the information with the highest weights in the user long-term interest model. User's actions will be captured and kept in the user behavior model.

The agent will make use of those data to assist the user, e.g., preload the definition files so that the user do not have to wait during the definition files uploading period, or prepare the most often used parameters to let the user avoid constantly changing the default data manually.

6. Discussion

In a collaborative design environment, the user should have the right to view and modify his/her model rather than being imposed to accept the user model. The user's arbitrary actions at the beginning will affect the appropriateness of inference. Considering the user always knows better about his/her own interest, and makes the proper decision whether an action is effective than the computer, the user model can be more accurate with the aid of the user.

Two approaches can be adopted to let the user share the control over the modification of the user model. One approach is to allow the user to interact with the user model through periodically promoted dialogs. This time-oriented method avoids bothering the users when they are concentrating on what they are working on. The other approach is to implement the personal assistant as a "mixed-initiative" user interface (Amant 1997), in a way that once the agent reaches a decision point to modify the user model it will leave the decision to the user. In that case, both the human agents and personal assistant agents can do their best to the system.

To let the user share control over the user model will sacrifice the automation of construction process, but improve the accuracy of the user model and shorten the long learning curve that the fully automated methods take. Furthermore, the user will feel more comfortable to let the agent delegate his/her tasks and thus build trust between the user and the software personal assistant, which is a very important issue in agent design (Maes 1994).

7. Conclusions

In this paper, we propose a user model composed of a user interest model, a user behavior model, an inference component and a collaboration component. The primary objective of our work is to find a solution to the user modeling for personal assistant agents working in collaborative design environments. The user interest model and the user behavior model are designed to capture the user interests and behaviors in the complex collaborative environments. Based on these two models, we use the inference component and the collaboration component to apply various strategies to perform inferences and facilitate collaboration. In a case study of a multidisciplinary design optimization environment, we investigate the feasibility of exploiting our user model to capture the user interests and behaviors and provide proactive inference to offer appropriate assistances to the user. The personal assistant agent is being implemented on the CIR-agent model while conforming to the FIPA standards.

References

Amant, SR: 1997, Navigation and planning in a mixed-initiative user interface, *in Proceedings of the Fifteenth National Conference on Artificial Intelligence, AAAI Press*, pp. 64-69.

Billsus, D and Pazzani, MJ: 1999, A hybrid user model for news story classification, *Proceedings of the Seventh International Conference User Modeling* (UM'99), Banff, Canada.

Brachman, RJ: 1979, On the epistemological status of semantic networks, *Findler* 3-50.

Brown, SM, Santos, E Jr. and Banks, SB: 1998a, Utility theory-based user models for intelligent interface agents, *in* Lecture Notes in Artificial Intelligence 1418: *Advances in Artificial Intelligence - AI '98*, Springer-Verlag, pp. 378-392.

Brown, SM, Santos, E Jr., Banks, SB and Oxley ME: 1998b, Using explicit requirements and metrics for interface agent user model construction, *Proceedings of the Second International Conference on Autonomous Agents*, Minneapolis, MN, pp. 1-7.

FIPA: 2001, Personal Assistant Specification, Document XC00083B, available at: http://www.fipa.org/specs/fipa00083/XC00083B.html.

Fleming M and Cohen R: 1999, User modeling in the design of interactive interface agents, *in Proceedings of the Seventh International Conference on User Modeling(UM99).*

Ghenniwa, H: 1996, *Coordination in Cooperative Distributed Systems*, PhD Thesis, University of Waterloo.

Ghenniwa, H and Kamel, M: 2000, Interaction devices for coordinating cooperative distributed systems, *Automation and Soft Computing* **6**(2): 173-184.

Horvitz, E, Breese, J, Heckerman, D, Hovel, D and Rommelse, K: 1998, The Lumière project: Bayesian user modeling for inferring the goals and needs of software users, *Proceedings of the Fourteenth Conference on Uncertainty in AI*, pp. 256-265.

Langley, P: 1999, User modeling in adaptive interfaces, *Proceedings of the Seventh International Conference on User Modeling*, pp. 357-370.

Lashkari, Y, Metral, M and Maes, P: 1994, Collaborative interface agents, *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pp. 444-449.

Maes, P: 1994, Agents that reduce work and information overload, *Communications of the ACM* **37**(7): 31-40.

Monell, DW and Piland, WM: 1999, Aerospace systems design in NASA's collaborative engineering environment, *Proceedings of the 50th International Astronautical Congress, Amsterdam*, The Netherlands.

Quillian, M Ross 1968, Semantic memory, i*n* Minsky M. (ed.), *Semantic Information Processing*. MIT Press, Cambridge.

Shen, W and Barthès, JP.: 1996, An experimental multi-agent environment for engineering design, *International Journal of Cooperative Information Systems* **5**(2-3): 131-151.

Shen, W., Norrie, D.H., and Barthès, J.-P.: 2001, M*ulti-Agent Systems for Concurrent Intelligent Design and Manufacturing*, Taylor and Francis, London, UK.

Shen, W and Ghenniwa, HH: 2001, Multidisciplinary design optimization: A framework for technology integration, *Proceedings of the First International Workshop on Multidisciplinary Design Optimization*, London, ON, Canada, pp. 22-28.

Smith, RG: 1980, The contract net protocol: High-level communication and control in a distributed problem solver, *IEEE Transactions on Computers* **C-29**(12): 1104-1113.