

8-15-1997

An Agent-Based Architecture Of An Adaptive Decision Support System

Ta-Tao Chuang

Texas Tech University, ogtao@ttacs.ttu.edu

Surya B. Yadav

Texas Tech University, odsby@ttacs.ttu.edu

Follow this and additional works at: <http://aisel.aisnet.org/amcis1997>

Recommended Citation

Chuang, Ta-Tao and Yadav, Surya B., "An Agent-Based Architecture Of An Adaptive Decision Support System" (1997). *AMCIS 1997 Proceedings*. 41.

<http://aisel.aisnet.org/amcis1997/41>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 1997 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

An Agent-Based Architecture Of An Adaptive Decision Support System

[Ta-Tao Chuang](mailto:ogtao@ttacs.ttu.edu) (email: ogtao@ttacs.ttu.edu)

[Surya B. Yadav](mailto:odsby@ttacs.ttu.edu), PhD (email: odsby@ttacs.ttu.edu)

ISQS, Texas Tech University

Lubbock, TX 79409 - 2101

Introduction

This article describes an agent-based architecture for an adaptive decision support system (ADSS) shown in Figure 1. The conceptual model shown in Figure 1 has been presented in an earlier article (Chuang & Yadav, 1996). Please refer to the cited article for details of the features and operations of the conceptual model.

The conceptual model is characterized by a variety of knowledge bases, highly modular structure, and extensive coordination and message-passing activities among the model components. These characteristics lend themselves to an agent-based architecture for implementing the ADSS.

We will address three issues in this article:

- What are the essential agents for embodying the conceptual model of ADSS?
- What is an appropriate type and structure of each agent in the architecture?
- What is an appropriate organization for making the agents work together and exhibit an adaptive behavior?

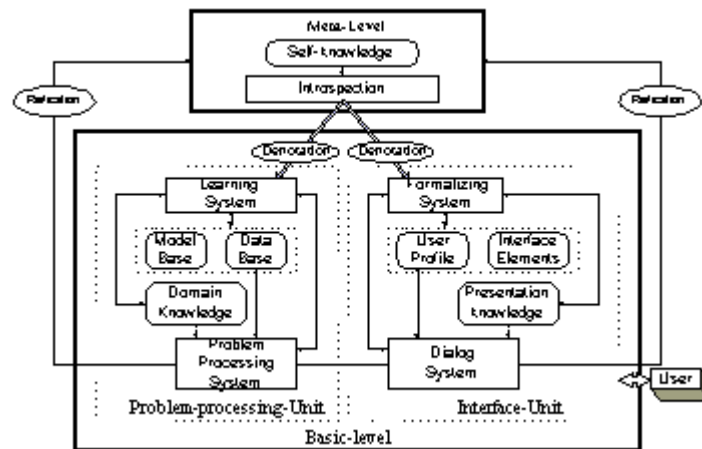


Figure 1. A Conceptual Model of Adaptive Decision Support Systems. (Source: adapted from Chuang and Yadav, 1996)

We propose a federated agent-based architecture as in Figure 2 for implementing the conceptual model of ADSS. The major feature of a federated architecture (Genesereth & Ketchpel, 1994) is that the whole system is divided into several agencies (Minsky, 1986), each of which consists of certain agents. Agents within an agency surrender part of their autonomy and capabilities to a facilitator, which is responsible for communicating with other facilitators in other agencies. Each agent within an agency is responsible for a particular task. Hence, agents have different structures and capabilities.

According to Russell & Norvig (1995), there are four different types of agents: simple reflex agents, agents that keep the track of the world, goal-based agents, utility-based agents. Similar to production systems, simple reflex agents associate actions with conditions. Agents that keep the track of the world can take more appropriate actions by capturing information about how the environment evolves and about what effects their actions have on the environment. Goal-based agents possess information about their goals in

order to assure they will take correct action and reach the goals. To assure that the performance of a goal-based agent is acceptable, a performance measure (a utility function) can be incorporated into it and make it rational. Based on this classification, the rest of the article will discuss the function and type of the agents in the architecture.

An Agent-based Architecture of ADSS

1. User interface agency

1. Facilitator: The facilitator in the user interface agency is a communication channel between the agency and two other agencies. The facilitator accepts system-level messages and requests from its agents, and routes them to one of the other two agencies or vice versa (Genesereth & Ketchpel, 1994). The facilitator keeps two levels of information to perform this function: the system level and the agency level. Information about the system level allows the facilitator to communicate with other facilitators, while information about the agency level allows it to communicate with its agents.
 1. System level: information about the self knowledge agency and information about the problem processing agency.
 2. Agency level: information about the dialog agent and the formalizing agents. With the specific information about the two levels, communication rules can be established to route messages between agencies and agents. Thus, the facilitator can be constructed as a reflex agent.
2. Dialog agent: An agent which communicates with users during a decision support session. The major functions of the dialog agent include passing problems and results between users and the problem solving agent, and collecting information about the users. Communications between the dialog agent and other components occur at the agency level. The dialog agent can be constructed as a reflex agent. The dialog agent must possess information about the following agents:
 1. Facilitator: The dialog agent communicates with the problem solving agents via the facilitator during a decision support session.
 2. User profile agent: The dialog agent passes information about the user to the user profile agent, which subsequently updates the user's status or profile.
 3. Element agent: The dialog agent should have access to the interface element base when it is necessary to tailor the interface to the user's preference.
 4. Presentation knowledge agent: The dialog agent communicates with the presentation knowledge agent in order to determine an appropriate presentation mode.
3. Formalizing agents: Four formalizing agents are responsible for interface composition at the component level, collection level, screen layout level, and application level. Formalizing agents can be constructed reflex agents. Each of them must possess information about the following agents:
 1. Dialog agent: Formalizing agents may need to communicate with the user when they do not have enough information about the user to compose presentation knowledge.
 2. Element agent: Formalizing agents may need access interface elements in order to allow the user determine the presentation mode when their knowledge cannot decide which elements to use.
 3. User profile agent: Formalizing agents need to access user profiles via the user profile agent in order to create new presentation knowledge.
 4. Presentation knowledge agent: Newly generated presentation knowledge will be added to the presentation knowledge base.
4. Element agent: An agent which is responsible for retrieving interface elements upon requests from the formalizing agents and/or the dialog agent. The element agent functions like the DBMS in a database system and can be constructed as a reflex agent.

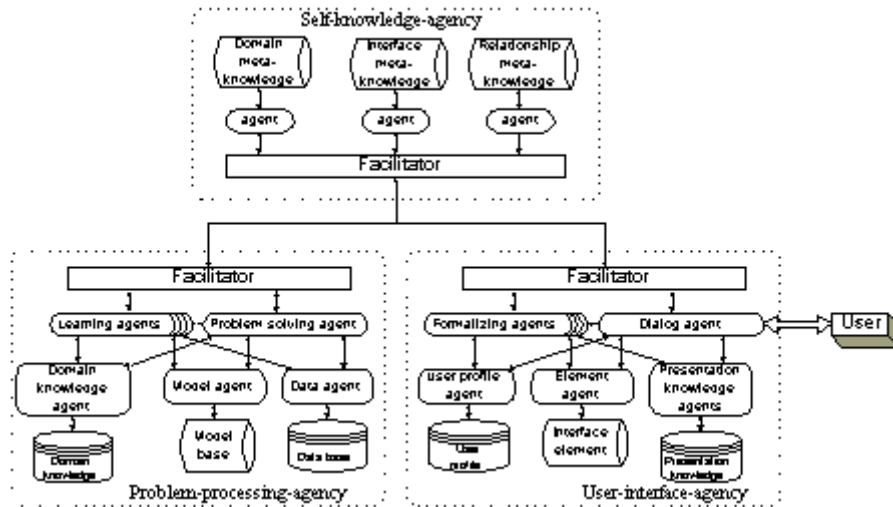


Figure 2. A Federated Agent Architecture of the ADSS Model.

1. User profile agent(s): A user profile agent manages information about the user. When more than one user is using the system, common characteristics of the users are kept in the user profile base and an agent is responsible for managing information about each user. User profile agents can be constructed as reflex agents since they function like a DBMS.
 2. Presentation knowledge agent: An agent which is responsible for retrieving presentation knowledge. This agent should be constructed as a goal-based or utility-based agent because it needs to inference interface knowledge for an appropriate interface.
1. Problem processing agency
 1. Facilitator: The function of the facilitator in this agency is similar to that of the facilitator in the user interface agency. This facilitator can be constructed as a reflex agent. The information that the facilitator possesses can be classified into two levels:
 1. System level: information about the self knowledge agency and about the user interface agency. The facilitator reifies its counterpart in the self knowledge agency or communicates with facilitator in the user interface agency.
 2. Agency level: information about the problem solving agent and about individual learning agents. The facilitator communicates with learning agents when the system is engaged in learning mode.
 2. Problem solving agent: An agent which performs the function of decision support. The problem solving agent should be constructed as a goal-based agent so that it can correctly reach the desired state. Communications between the problem solving agent and other agents within the system mainly occur at the agency level. The problem solving agent needs information about the following agents:
 1. The facilitator: The problem solving agent communicates with two other agencies via the facilitator during a decision support session or when it is necessary to generate new problem domain knowledge.
 2. The domain knowledge agent: The problem solving agent needs to access problem domain knowledge via the domain knowledge agent.
 3. The model agent: The model agent retrieves various models for the problem solving agent or for learning agents to perform scenarios analysis or to generate new problem domain knowledge.
 4. The data agent: The problem solving agent receives data or historical cases from the data agent.
 3. Learning agents: Six learning agents are responsible for different learning process at the object level, association level, exemplar level, prototype level, concept combination

level, and problem solving level (Rohatgi, 1994). The six learning agents can be constructed as reflex agents since they are implemented as learning algorithms. The six learning agents need information about the domain knowledge agent, model agent, and data agent to generate new knowledge.

4. Problem domain agent: The problem domain knowledge agent should be implemented as a goal-based or utility-based agent because it needs to retrieve the most relevant knowledge to the current problem.
 5. Model agent: The model agent functions like a DBMS. It can be constructed as a reflex agent.
 6. Data agent: The data agent functions like a DBMS. Thus, it can be constructed as a reflex agent.
2. Self knowledge agency: In addition to a facilitator, there are, at least, three agents for three kinds of meta-knowledge.
1. Facilitator: Like the facilitator at the other two agencies, the facilitator can be constructed as a reflex agent.
 2. Problem domain meta-knowledge agent: This agent should be constructed as a goal-based or utility-based agent in order to retrieve the most relevant meta-knowledge to trigger an appropriate learning agent.
 3. User interface meta-knowledge agent: Similarly, this agent should be constructed as a goal-based or utility-based agent in order to trigger an appropriate formalizing agent.
 4. Relationship meta-knowledge agent: This agent is responsible for managing the knowledge about the relationship between the domain meta-knowledge and the user interface meta-knowledge. The relationship meta-knowledge exists in the form of heuristics. The relationship meta-knowledge agent should be constructed as a goal-based or utility-based agent.

Conclusion and Future Research

We have described briefly a federated agent-based architecture for implementing an integrated conceptual model of ADSS. We are currently developing a prototype system based upon the proposed architecture. A detailed design of the prototype has been completed. We are implementing the prototype using an ES shell. The major objective of the prototype system is to demonstrate the feasibility of the conceptual model. In addition, it opens up several research opportunities: using the prototype system as a vehicle to re-examine the relationship between task complexity and presentation modes, and the impact of presentation modes on the quality of decisions.

References

Chuang, Ta-Tao & Yadav, Surya B. (1996). A Conceptual Model of an Adaptive Decision Support System (ADSS). Proceedings of the Second Americas Conference on Information Systems (p. 464-466).

(The rest of references are available upon request.)