CONSTRUCTING A CONFERENCE MANAGEMENT SYSTEM WITH AGENTS*

Chia-Lin Hsu, Je-Way Huang, Ren-Ji Liao, Feng-Jian Wang, Wen-Nung Tsai

Department of Computer Science and Information Engineering,
National Chiao Tung University, Hsin-Chu, Taiwan, Republic of China
E-mail: {glshu,jwhwang,rjliou,fjwang,tsaiwn}@csie.nctu.edu.tw

ABSTRACT

Software agents have become very popular in the last few years. They have been successfully used in distributed application, information retrieval and automation of repetitive behavior. This paper presents the design and implementation of an agent based conference management system whose primary goal is to support automated paper collection, review dispatching and result collection, acceptance notice, …etc.

1. Introduction

Organizing an academic conference includes a set of complex activities for many people who need to cooperate on different levels for achieving coherent results. These activities include conference planning, call for papers, paper submissions, paper reviews, paper acceptance, conference registration, fee charge and so on. Thus an automatic and efficient supporting system for the paper submission and review process may be required. Recently, several software systems [1][2] have been constructed to automate the collection and counting of paper and review results. However, these systems do not handle other activities and the heterogeneous system environment problems.

Software agents are personalized, continuously running, and autonomous. They are suitable for the applications that require communications between components, sensing or monitoring of the environment, or autonomous operations [3][5][9][14]. The software agents are software entities that perform a set of operations on behalf of a user or a program to achieve the user's goal [8]. The software agent can be used to automate the tasks of the electronic submission process, as well as paper and review comment management in the conference supporting system. An engineering approach for agent based systems development is required and the development principle is needed while the number of agents increases [6].

The conference system is an Agent based Conference Management System (ACMS), which is a multi-agent system based on the DSSL agent system [4]. The design process generates three models. Firstly, the agent model identifies the agent types in ACMS where the agent instances might appear at run time. Secondly, the service model identifies the main services and objects that will be used by each agent type. Finally, the agent-interaction model describes the interaction between related agent type and the associated component diagram. This paper will present a conference system based on ACMS. The notations used are UML [10].

2. The Agent Based Conference Management System (ACMS)

2.1 Basic Requirements

ACMS is an automated, flexible agent-based electronic submission system, which is expected to simplify the work of accepting paper submissions, dispatching papers and collecting the results from reviewers, reporting review results to the authors, and dispatching the information or rules from the conference program committee.

There are at least four roles in the ACMS: author, reviewer, program-committee member and conference agency. The author can submit a paper and query the conference information. The reviewer is in charge of paper review and submits the review comments to conference agency. The program committee is responsible for setting up the reviewer list, holds the review meeting, and decides whether a paper is accepted. The conference agency is responsible for paper and review comment collection and conferment information such as paper acceptance dispatching. Besides, every body can give a request to ask the general information regarding the conference. Figure 1 illustrates the conceptual model of the proposed ACMS system.

2.2 Expected Profits of ACMS

ACMS focuses on the automation of paper submission and review report handling process through Internet. During the conference preparation process, the program committee may change the policy of paper submission, so that the flexibility of dispatching policy is an important issue. The whole process involves many people in different places, and thus ACMS is naturally a distributed application. On the other hand, how to overcome the hardware and software heterogeneity is also an important issue. ACMS is also considered the extensibility and reusability of the whole system. An agent-based approach

* This work was supported in part by 資策會 under contract (89)資系約字第 0321 號.

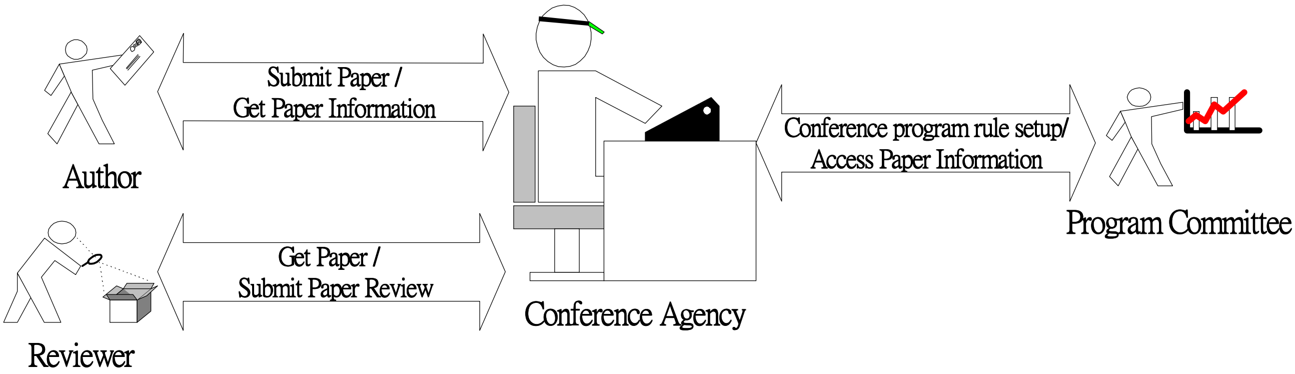


Figure 1 ACMS Conceptual Model

adopted in our system is a good idea to fulfill most of the requirements. The description of the agent based system construction methodology is in [11].

The expected benefits using mobile agents in ACMS include:

- Accommodate hardware and software heterogeneity:

  Since the electronic submission process is naturally a distributed application. The interaction between systems of different environments is complicated. To build the ACMS based on a cross-platform mobile agent execution environment might accommodate hardware and software heterogeneity.

- Encapsulate protocols:

  Each host in a distributed system, to interpret input data and output data, owns the code of the protocol designed well. As protocols evolve to accommodate new requirements for upgrade, efficiency or security, it is cumbersome to upgrade protocol code on each host properly. Mobile agents may carry the upgraded code to reduce the overhead on each host.

- Reduce the network traffic load:

  Traditional distributed systems often rely on communication for multiple interactions to accomplish a task. These systems may contain heavy load of communication traffic. A mobile agent may pack the messages of a conversation and carry it to the destination host where interactions occur to reduce the network traffic.

- Decisions according to user preferences

  Traditional systems do not support preferences very well, and thus a user must transfer the preferences related parameters to a server for decision. The user needs to modify the server site's system to have a new alternation decision, and it is very unlikely to accommodate various needs of multiple users simultaneously. In agent-based systems, what is done is to modify the decision logic in an agent interested. The agent will go to the right place to do the right thing according to user's preferences. Agents with highly intelligence to make decisions are called intelligent agents.

- Scalability:

  Another benefit of ACMS based on mobile agent system is the scalability of the agents. Since the agents are inherently modular, it is much easier to

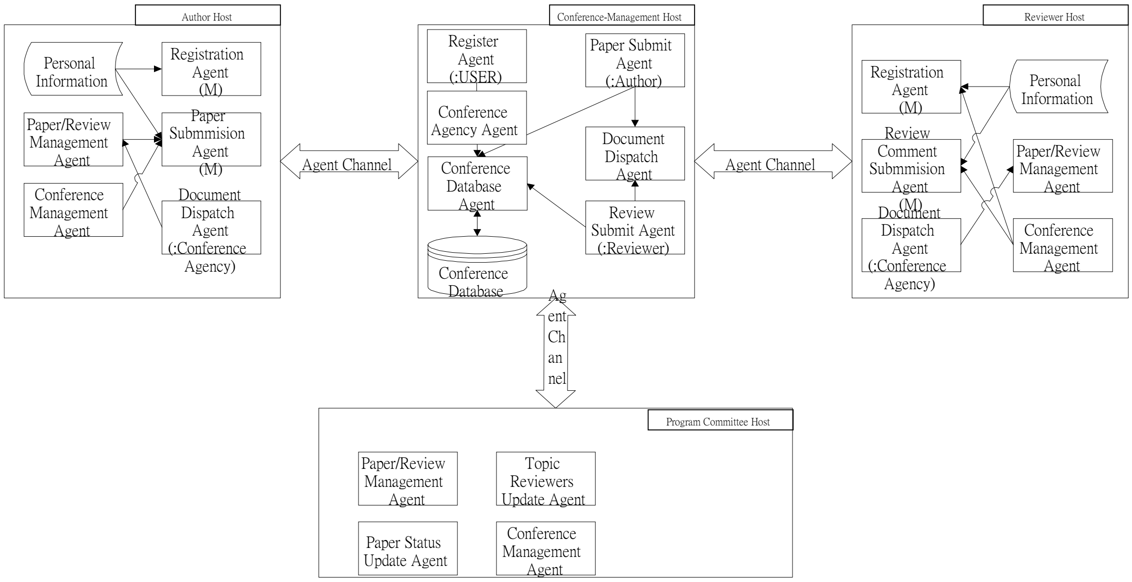

Author Host
Personal Information
Registration Agent (M)
Paper/Review Management Agent
Paper Submmision Agent (M)
Conference Management Agent
Document Dispatch Agent (:Conference Agency)
Agent Channel
Conference-Management Host
Register Agent (:USER)
Paper Submit Agent (:Author)
Conference Agency Agent
Document Dispatch Agent
Conference Database Agent
Review Submit Agent (:Reviewer)
Conference Database
Agent Channel
Reviewer Host
Registration Agent (M)
Personal Information
Review Comment Submmision Agent (M)
Paper/Review Management Agent
Document Dispatch Agent (:Conference Agency)
Conference Management Agent
Agent Channel
Program Committee Host
Paper/Review Management Agent
Topic Reviewers Update Agent
Paper Status Update Agent
Conference Management Agent

Figure 2 The elaborated role (agent) types

add new agents than to add new capabilities to ACMS. System's capabilities and parameters are likely to change over time, so that the modification through agents can be done in the code of client site only.

3. System Specification of ACMS

The main goals of ACMS are collecting the user registration, paper submission, review of comment submission, revision of topics and reviewers, submitting call-for-paper, paper receive and acceptance, and changing the paper status and registration information. As mentioned in section 2, there are four typical role types in an ACMS: *author*, *conference agency*, *reviewer*, and *program committee*. Each of them has distinct goals and permissions.

Intuitively, the interactions inside an ACMS are those between the conference agency and the authors, between the conference and the reviewers, and between the conference agency and the program committee. Based on them, one can elaborate the role model recursively. The process might not be done until a detailed role model, in which the type of a role can be transformed into an agent type, is presented. Our elaborated role (agent) interaction diagram is illustrated in figure 2.

4. Detailed specifications for ACMS

The abstract models can be transformed into three models that can be implemented, agent model, service model, and interaction model. The agent model identifies the classes of agents that will make up the system. The service model identifies the main services that will be associated with each agent. Finally, the agent interaction model describes the communications for each agent and each component diagram represents the deployment of the agents for each site.

4.1 The Interaction of Role Types in ACMS

This section discusses the interactions between the conference agency and the author, between the conference agency and the reviewer, and between the conference agency and the program committee.

- The Author and the Conference Agency

  If the author performs ConferenceRegistration() action with the conference agency, he/she transfers his/her personal information to the agency. When the conference agency accepting 'ConferenceRegistration' message, the agency processes the registration of the incoming author. The agency generates a personal identify number for the author and appends his/her personal information to conference database by performing UserRegistrationCollection() action. If the author performs the PaperSubmission() action with the conference agency, he/she transfers his/her paper to the agency. When the conference agency accepting 'PaperSubmission' message, the agency processes the incoming paper registration. The agency generates a paper identify number for the paper of the author and appends hi/her paper to conference database by performing PapaerCollection() action. If the conference agency performs DispatchReview() action with the author, the agency dispatches the review comment to the author. When the author accepting 'AppendReview' message, he/she processes the incoming review comment and appends the paper to his/her local database. If the conference agency performs InformAcceptance() action with the author, the agency sends the paper acceptance to the author. When the author accepting 'UpdatePaperAcceptance', he/she updates the incoming paper acceptance notification to his/her local database. The interaction diagram between the author and the conference agency is illustrated in Figure 3.

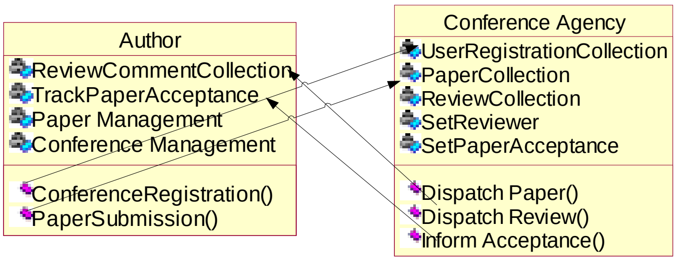


Figure 3 The interaction between author and conference agency

- The Reviewer and the Conference Agency

  If the reviewer performs ConferenceRegistration() action with the conference agency, he/she transfers his/her personal information to the agency. When the agency accepting 'ConferenceRegistration' message, the agency processes the incoming reviewer registration. The agency generates a personal identify number for the reviewer and appends the reviewer's personal information to conference database by performing UserRegistrationCollection() action. If the reviewer performs the ReviewSubmission() action with the conference agency, he/she transfers the review comments to conference agency. When the conference agency accepting 'ReviewSubmission' message, the agency appends the reviewer's review comments to conference database by performing ReviewCollection() action. If the conference agency performs DispatchPaper() action with the reviewer, the agency dispatches the

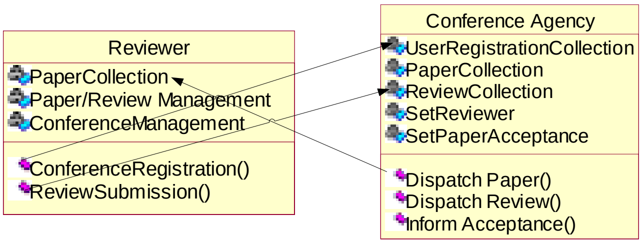


Figure 4 The interaction between reviewer and conference agency

paper to the reviewer. When the reviewer accepting 'AppendPaper' message, he/she appends the paper to reviewer's database by performing ReviewCollection() action. The interaction diagram between the reviewer and the conference agency is illustrated in Figure 4.

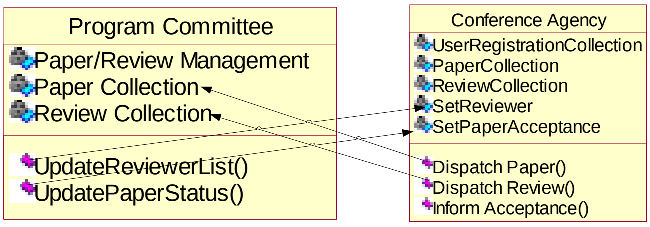


Figure 5 The interaction between program committee and conference agency

- The program committee and conference agency

  If the program committee performs UpdateReviewList() action with the conference agency, the committee sends the reviewer list of topics to the conference agency. When the conference agency accepting 'UpdateReviewList' message then performs the SetReviewer() action to update the reviewer list of topics to the database. If the program committee performs UpdatePaperStatus() action with the conference agency, the committee sends the paper acceptance information to the conference agency. When the conference agency accepting 'UpdatePaperStatus' message then performs the SetPaperAcceptance()action to update the acceptance information to database and performs the InformAcceptance() action to the specific author. The interaction diagram between the program committee and the conference agency is illustrated in Figure 5.

4.2 Agent Types and Services

In the design phase, the agent model can be mapped from the elaborated roles model. In ACMS, each human role interacts with an interface agent directly. The interface agent sends messages to the local host, which then dispatches the messages to specific agents or the other hosts. Besides interface agents, there are at least ten types of agents booted in an ACMS: registration agent, paper submission agent, review comment submission agent, document dispatch agent, paper/review management agent, conference management agent, reviewers update agent, paper status update agent and DB agent. We do not describe this agent in detail. Each of other agent types is described as following:

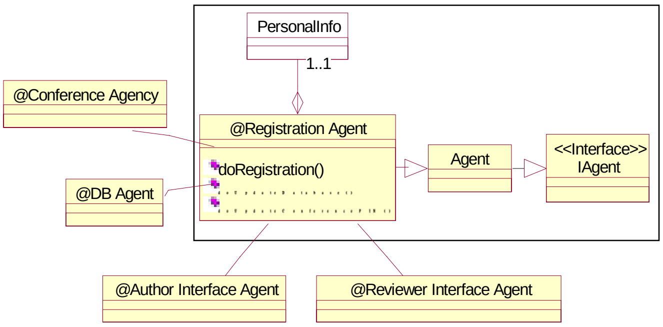
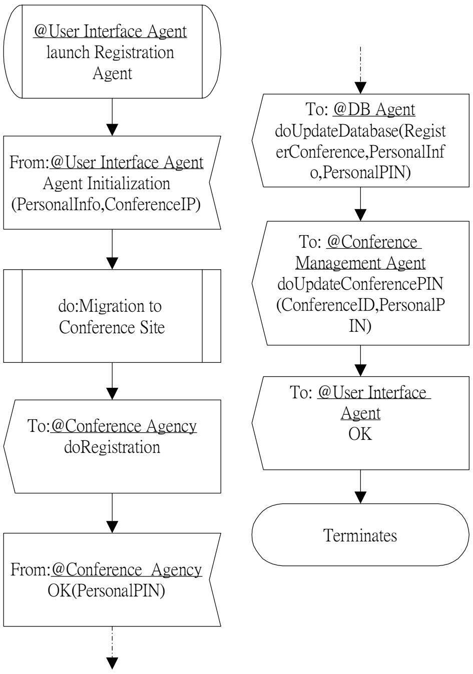


Figure 6 Class diagram of registration



@User Interface Agent
launch Registration Agent
From:@User Interface Agent
Agent Initialization
(PersonalInfo,ConferenceIP)
do:Migration to Conference Site
To:@Conference Agency
doRegistration
From:@Conference Agency
OK(PersonalPIN)
To: @DB Agent
doUpdateDatabase(RegisterConference,PersonalInfo,PersonalPIN)
To: @Conference Management Agent
doUpdateConferencePIN
(ConferenceID,PersonalPIN)
To: @User Interface Agent
OK
Terminates

Figure 7 The service model of registration agent

- Registration Agent (RA)

  A RA is responsible for conference agent registration, by taking user's personal information to a conference server. When the RA arrives at the conference agency site, it sends a 'doRegistration' message to the conference agency and waits for reply. When accepting such a message, the conference agency first checks whether the request is allowed to be accepted. If yes, the conference agency will reply an 'OK' message with personal identity number for the user to the RA which then sends the identity number back to user site's conference management agent. Finally, the RA terminates in the conference site automatically. If the registration process causes an error, the RA will send 'sorry' with caused message back to the original site of the agent and terminates. The class diagram and service model of the RA are illustrated in figure 6 and figure 7.

Using the same elaboration process, we can elaborate the following agents:

- Paper Submission Agent (PSA)

  A PSA is launched by an author using the user interface agent (UIA). It packs the paper submitted and migrates to the conference site that author selected. When the agent arrives at the conference site, it sends a 'doSubmitPaper' message to the conference agency and waits for reply. When accepting such a message, the conference agency first checks whether the paper is allowed to be accepted. If yes, the agency sends an "OK" message with 'PaperPIN' (a paper ID) to the PSA, which then contacts with the database agent to update the status of related paper and sends 'PaperPIN' back to paper/review management agent (PRMA) of original author site. If no, the agency replies an 'error' message to the PSA which will send 'sorry' with caused message back to the original site. No matter yes or no the PSA terminates after it replies the message to the author's interface agent.

- Review Comment Submission Agent (RCSA)

  A RCSA is launched by a reviewer through the user interface agent. It packs the review comments and migrates to the conference site. When the agent arrives at the conference site, the agent sends a 'doSubmitReview' message to the conference agency and appends the review comment to the database by contacting with database agent. In case has error, the RCSA will send 'sorry' with caused message back to the original site and terminates.

- Document Dispatch Agent (DDA)

  The DDA is responsible for deliver of papers and review comments. When a PSA migrates to the conference site and sends the 'doSubmitPaper' message to the conference agency. The conference agency determines dispatching to which reviewers and launches the DDA initialized with the destination IP address for each reviewer. Then the DDA migrates to the reviewer's site and contact with PRMA for appending the paper to local database. When a RCSA migrates to the conference site and sends the 'doSubmitReview' message to the conference agency. The conference agency launches two DDAs, one is initialized with program committee's IP address and the other is initialized with author's IP address. Then the agents migrate to their destination sites and contact with PRMA agent for appending the review comment to local database. If the document dispatching process causes an error, the DDA sends 'sorry' with caused message back to the original site and terminates.

- Paper/Review Management Agent (PRMA)

  PRMA manages the paper and review comments for users. When a paper is accepted or rejected, the agent will inform the interface agent to display the acceptance information about that paper. When one paper or review comment is delivered the agent will inform the UIA. Besides, this agent can manage the paper review comments about specific paper for the user. If the paper/review management process causes an error, the PRMA will send 'sorry' with caused message back to the interface agent.

- Conference Management Agent (CMA)

  The CMA manages the information about the topics and the due date information of a specific conference that users concern. For example, if the due date of paper review submission is in several days, it will inform the interface agent to display due date information for the reviewer. Besides, this agent can manage multiple conference that using ACMS for the user. If the CMA process causes an error, the conference management agent will send 'sorry' with caused message back to the UIA

- Reviewers Update Agent (RUA)

  Program committee can modify the reviewer list for specific topic or set up the min/max papers of specific topic by launching RUA. If the reviewers updating process causes an error, the RUA will send 'sorry' with caused message back to the program committee agent and terminates.

- Paper Status Update Agent (PSUA)

  Program committee can determine whether a paper is accepted or not by this agent. Then the agent migrates to the conference server and updates the information to the conference database. If the paper status process causes an error, the PSUA will send 'sorry' with caused message back to the program committee UIA and terminates

- Conference Agency Agent (CA)

  Because of the needs to resolve the participants' IP addresses, the registration agent, paper submission agent, review submission agent, and reviewers update agent make interaction with the conference agency to achieve their goals.

- Accepted Paper Registration Agent (APRA)

  If a paper is accepted, the conference agency informs the author the conference standard for an electronic version of their final paper. When the author sends the final paper to the conference, he/she instantiates an APRA which packs paper number, author information, payment (visa or master) and the paper to the conference agency for final paper registration. The APRA, launched by the author interface agent, carries the information to the conference site. When an APRA arrives at the conference site, it sends a 'doSubmitFinalPaper' message with the carried information to conference agency agent and waits for reply. The conference agency checks whether the paper and its format are acceptable (It also checks the payment). If yes, the agency sends an 'OK' message with 'AttendeePIN' (a user ID), and acceptance result to the accepted paper registration agent, which then contacts with the database agent to update the database. If no, the agency replies an 'error' message to the accepted paper registration agent which will send 'sorry' with related comments back to the original site. The accepted paper submission agent terminates after it replies the message to the author's interface agent.

- Conference Registration Agent (CRA)

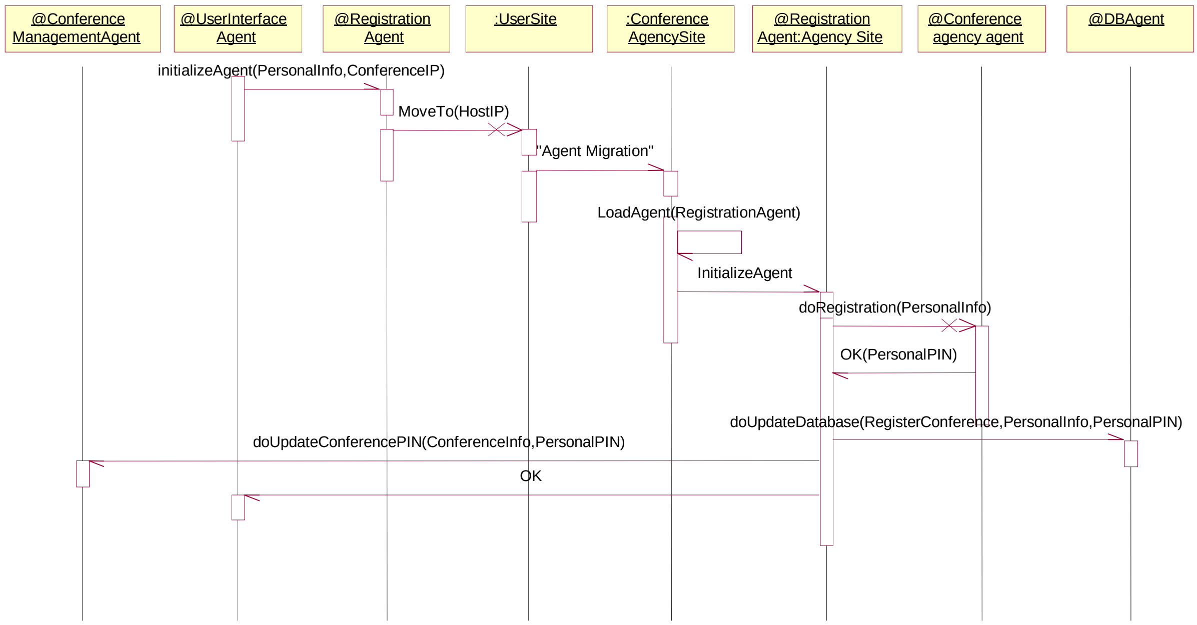

@Conference ManagementAgent
@UserInterface Agent
@Registration Agent
:UserSite
:Conference AgencySite
@Registration Agent:Agency Site
@Conference agency agent
@DBAgent
initializeAgent(PersonalInfo,ConferenceIP)
MoveTo(HostIP)
"Agent Migration"
LoadAgent(RegistrationAgent)
InitializeAgent
doRegistration(PersonalInfo)
OK(PersonalPIN)
doUpdateDatabase(RegisterConference,PersonalInfo,PersonalPIN)
doUpdateConferencePIN(ConferenceInfo,PersonalPIN)
OK

Figure 8 The user registration process

A CRA is responsible for those who want to participate a conference but not the pay authors. When such a user wants to register for a conference, he/she instantiates a CRA, which is then launched by the UIA. The CRA carries the information to the conference site for registration. When the agent arrives at the destination site, it sends a ‘doConferenceRegistration’ message including the information to the conference agency and waits for reply. The conference agency then checks whether the registration is acceptable (It also checks the payment). If yes, the agency sends an ‘OK“ message with ‘AttendeePIN’ (a user ID) to the CRA, which then contacts with the database agent to update the database and sends ‘AttendeePIN’ back to user interface agent of original user site. If no, the agency replies an ‘error’ message with related information to the CRA which will send ‘sorry’ with caused message back to the original site. A CRA terminates after it replies the message to the user’s interface agent.

- Conference Schedule Agent (CSA)

CSA is responsible for informing the events and warning message of a conference. Program committee set up the deadline for the events of call for paper, paper submission, paper review, PCB meeting, final paper in camera ready format and early conference registration through program committee interface agent. CSA monitors the deadline of coming events and check whether the date comes in a week. For each deadline, the agent sends several messages to program committee UIA to show the information at the specified dates correspondingly.

- Budget Management Agent (BMA)

BMA is responsible for conference budget information management. The income includes conference registration fees, advertisement, corporate sponsorships, grants (government or fund), etc. The costs include program committee expense, promotion advertisement, proceeding printings, hotel bedrooms, food (break and meals), and conference space. BMA counts the total income, cost and the difference when program committee requests through UIA.

4.3 Agent Interaction model

This section describes how the agent interaction model can be used to achieve the four main goals of ACMS: user registration, submission of paper, submission of review comments, update topic reviewer and paper status update.

- User Registration

The agent interaction diagram is illustrated in figure 8.

1. In the beginning, the User Interface Agent launches a Registration-Agent, and sends a “initializeAgent” message with “PersonalInfo” and “ConferenceIP” to Registration Agent.

2. The Registration-Agent sends a “MoveTo” message with “HostIP”, and then the Registration-Agent migrates from user site to conference agency site.

3. When the Registration-Agent arrives at conference agency and is initialized, the Registration-Agent sends a “doRegistration” message with “PersonalInfo” to the conference agency.

4. When accepting a “doRegistration” message, the conference agency generates a “PersonalPIN” number and sends an “OK” message with “PersonalPIN” to the Registration Agent.

5. When accepting an "OK" message, the Registration-Agent sends a "doUpdateDatabase" message with "RegisterConference", "PersonalInfo" and "PersonalPIN" to DB Agent. Then the DB Agent updates the user registration database.
6. The Registration Agent sends a "doUpdateConferencePIN" message with "ConferenceInfo" and "PersonalPIN" back to Conference Management Agent.
7. Finally, the Registration Agent sends an "OK" message back to user interface agent.

Using the same process, we can identify remaining interaction models of paper submission process, review comment submission process, topic reviewers update process, and paper status update process.

4.4 Agent Deployment

This section describes the agent deployment. There are four types of agent sites in the environment: author site, reviewer site, program committee site and conference agency site.

- Author Site System

  There are five agent types in the author site: author interface agent, registration agent, paper submission agent, paper/review management agent and conference management agent. The author site system is running on the author's host. The system will launch the author interface agent, paper/review management agent and conference management agent as it starts up. The deployment diagram is illustrated in figure 9.

- Reviewer Site System

  There are five agent types in the reviewer site: reviewer interface agent, registration agent, reviewer summit agent, paper/review management agent and conference management agent. The reviewer site system is running on the reviewer's host. The system will launch the reviewer interface agent, paper/review management agent and conference management agent as it starts up.

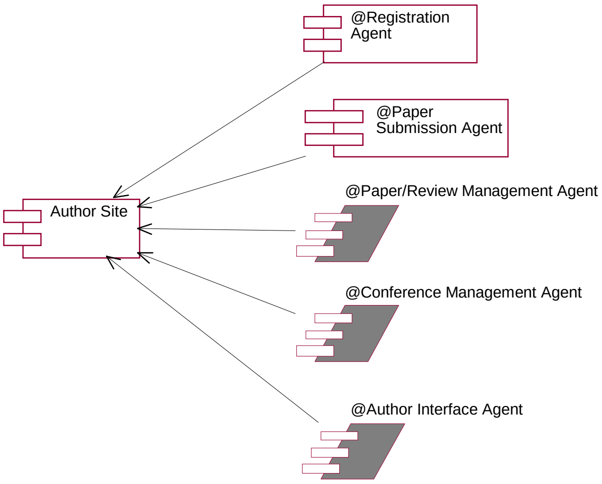


Figure 9 Author site component view

- Conference Agency Site System

  There are three agent types in the conference agency site: DB agent, Conference agency agent and document dispatch agent. The conference agency site is running on the conference agency's host. The system will launch the conference agency agent and DB agent as it starts up.

- Program Committee Site System

  There are four agent types in program committee site: PC interface agent, paper status update agent, reviewers update agent and paper/review management agent. The conference agency site system is running on the program committee's host. The system will launche the PC interface agent and paper/review management agent as it starts up.

5. Conclusion

In this paper, an agent based software system development approach is presented and an example, the ACMS, is used to illustrate the feasibility of this approach. The application is developed upon our DSSL agent platform to achieve cross-platform execution in heterogeneous environment.

Currently, the ACMS supports the authors for electronic submission of review comment, supports the program committee for the collection of papers and review comments, as well as the help of paper acceptance decision. Moreover, ACMS supports authors and reviewers to manage their papers and reviewers' comments and communication on multiple conferences.

Reference

[1] Nicol, D.M. "Conference program management using the Internet." Computer, Vol. 29, pp. 112 – 113, IEEE, March 1996.

[2] Automatica's Editor-in-Chief! http://www.math.utwente.nl/eic/

[3] P. Maes. "Agents that Reduce Work and Information Overload." Communications of the ACM, Vol. 37, No. 7, pp.31-40, 146, ACM Press, July 1994.

[4] C. Z. Liao. "Constructing an Agent's Platform Based on Java", Jun 1999.

[5] D. B. Lange, M. Oshima "Seven Good Reasons for Mobile Agents." Communications of the ACM, Vol. 42, No 3, pp.88-89, ACM Press, March 1999

[6] M. Wooldridge, N. R. Jennings, D. Kinny "A methodology for Agent-Oriented Analysis and Design" , Agents '99: Proceedings of the Third International Conference on Autonomous Agents Seattle, WA, May 1998..

[7] C. A. Iglesias, M. Garijo, J. C. Gonzlez, J. R. Valasco “Analysis and Design of Multiagent Systems using MAS-CommonKADS” , Proceedings of the AAAI'97 Workshop on Agent Theories, Architectures and Languages, Providence, USA, July, 1997. An extended version of this paper will appear in INTELLIGENT AGENTS IV: Agent Theories, Architectures, and Languages , Springer Verlag, 1998.

[8] Maes, P. “Designing Autonomous Agents: Theory and Practice”, Biology to Engineering and Back, London: The MIT press, 1991.

[9] N. R. Jennings, M. Wooldridge. “Agent Technology - Foundations, Applications, and Markets” Springer UNICOM, February 25th, 1998

[10] Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide", 1999

[11] Chia-Lin Hsu, Je-Way Huang, and Ren-Ji Liao, “An agent based software construction method”, 2000