MODELLING MOBILE AGENT APPLICATIONS BY EXTENDED UML ACTIVITY DIAGRAM

Miao Kang, Kenji Taguchi
Department of Computing,Univerisity of Bradford,Richmond Road, Bradford, West Yorkshire, UK

Keywords: Mobile agent, UML, UML Activity diagram, UML Metamodel

Abstract: Mobile agent technology has gained increasing importance in recent years. However, little work has been done in defining notations/languages to capture and model mobile agent applications. This paper presents extensions of UML activity diagrams for modelling mobile agent applications, which capture specific features of mobile agents such as mobility, cloning and communications.

1 INTRODUCTION

A mobile agent is a computing entity that can move around different hosts on the network while carrying its state and procedures. Mobile agent technology can be deployed for distributed systems such as e-commerce, distributed information retrieval and some other services such as personal assistance and telecommunication network services.

Industry has made a lot of effort to implement mobile agent programming languages. However most of the works only address implementation issues and do not address design issues on the development of mobile agent applications.

In this paper, we present an effective new approach to model specific features of mobile agents by extending UML. UML can also be extended or adapted to a specific method, organization, or user by the built-in extension mechanisms. Its wide acceptance in industry is one of our main reasons to adopt UML as our base language.

UML is also widely accepted in Agent Oriented Software Engineering (AOSE). Unfortunately UML cannot be used in a straightforward way for modelling agent-based systems, since it lacks basic notion of agenthood. Hence it needs to be extended by incorporating new structural elements which could enhance the base language's expressive power. Agent UML (AUML) (Bauer, et. al., 2001) is one of those approaches which synthesize a growing concern for agent-based software methodologies with increasing acceptance of UML for object-oriented software development.

The notion of locality is a key concept in mobile agent languages and systems (Taguchi and Dong, 2002), which poses constraints on communications between agents and how they can move around locations. For instance, in the agent-place model, which is the basic computational model of Telescript (White, 1996), communications between agents are only allowed at the same location, which means that an agent must move to a certain location where the agent can communicate with other agent.

In this paper, we extend activity diagrams in UML to model mobile agent applications. Activity diagrams are mainly used for modelling business process and are also good at modelling algorithmic behaviour of a computation. The major extension is to give a new structural meaning to swimlanes to model behaviours of mobile agents. This visual representation makes it easy to capture mobility of agents. Since moving from one location to another is simply represented by a transition between two swimlanes. A metamodel is a model of models. It aims at modelling, visualizing the system, specifying the structure and the behaviour of the system and helping in building the system, documenting the decisions made while building the system (Booch, 1999). We use the UML metamodel to model the syntax of our extension in this paper.

In order to demonstrate applicability of our notation, we designed an electronic auction system in our proposed notation and implemented it in Java Agent DEvelopment (JADE) programming language (Bellifemine, et. al., 1999).

This paper is organised as follows: the next section presents our new approach by extending UML activity diagrams to model mobile agent

Kang M. and Taguchi K. (2004).
MODELLING MOBILE AGENT APPLICATIONS BY EXTENDED UML ACTIVITY DIAGRAM.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 519-522
DOI: 10.5220/0002635505190522

applications. The syntax of the extended activity section 3. We then compare related works and finally conclude our work and explain our future plan of our work.

2 OUR APPROACH

We will use UML Activity diagrams to model dynamic behaviour of mobile agent applications.

UML has extension mechanisms such as tagged values, stereotypes and constraints. Tagged values are mainly used to show extra information. *Stereotype* allows us to extend the UML elements that already exist by decorating them with extra semantics. It is presented within a pair of matched guillemots.

Specific features of mobile agents, which we model, are mobility, cloning, messaging between agents, and the concept of nested places.

We introduce a new stereotype << Host >> + *parameter* for swimlane, which represents the location. The parameter is the unique name (address) of the location. It should be noted that this is just a conventional extension of UML. Moving between hosts can be now captured by a transition between swimlanes. A transition is represented by an activity "go to host" (see (1) in the figure 1).

Now we introduce a new notation, *Subswimlane* that is not realised by the built-in extension mechanisms in UML. A subswimlane is also shown as a swimlane with an icon in the higher right corner representing a nested swimlane (see (2) in the figure 1). An example of subswimlane is given in (5) in figure1 in which Shopping Centre accommodates Directory Service, Ticket Shop and Flower Shop. diagram is defined by using UML metamodel in Mobile agents may travel among these shops gathering information or making trade.

We use *signal sending* and *signal receipt* in UML Activity diagram to enable agents to exchange messages. A new rule is added, which is that whenever an agent would like to send a message to another agent, the parameters(message description) inside these icons must be exactly the same between signal sending and expecting signal receipt. For example, a user defined "Query" type signal send, must have the same type in the expecting signal receipt (see (3) in the figure 1).

Cloning is another important behaviour of mobile agents, which includes several subactivities. This activity allows an agent to clone itself first, then to send the cloned agent to another host. This behaviour could be repeated in order to send the required number of clones. We use subactivity notation to include subactivities and use *Dynamic Concurrency State* to show iterations without having to construct a loop (see (4) in the figure 1).

We designed and implemented an electronic English auction system in JADE (Bellifemine, et. al., 1999). JADE is a FIPA-compliant Agent Platform. In JADE, the communication between agents will use the ACLMessage class, in which several message types are provided so that agents could understand what kind of message, e.g., INFORM, REQUEST CONFIRM, they need.

Our system consists of two agents, an auctioneer agent and a bidder mobile agent. The auctioneer agent resides at seller's host and manages an interface to the seller to show the state of the auction and controls the whole process. The bidder mobile agent will circulate round potential bidders while gathering bids by displaying it on GUI to bidders.

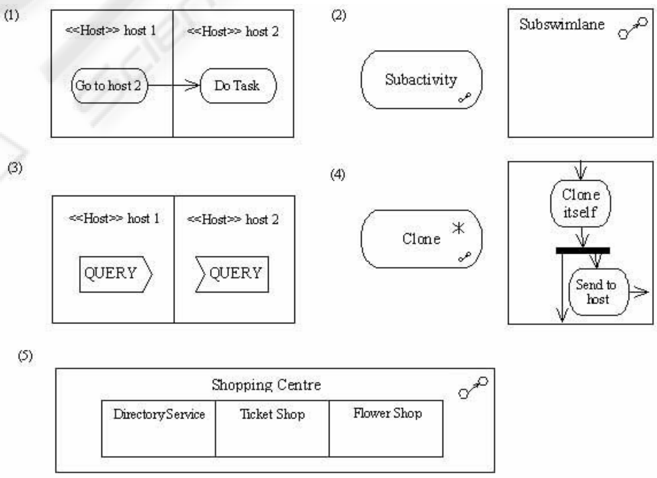

Figure 1: Modelling concepts in activity activity diagram of our new approach

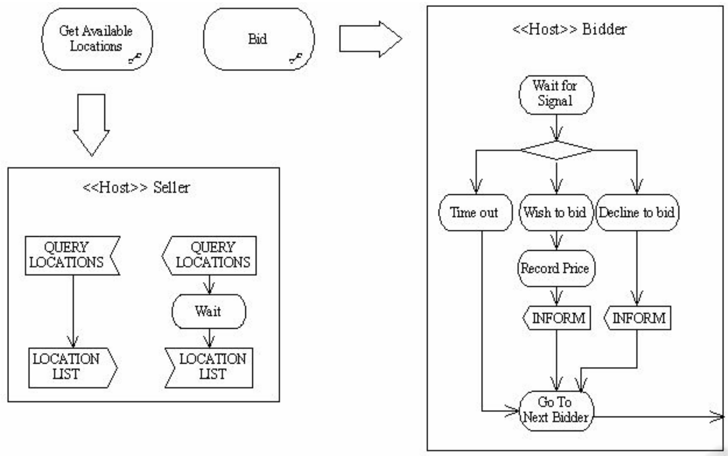

Figure 2: Get Available Locations subactivity and Bid subactivity

There are several repeated activities such as querying for available bidders' locations from management agent (AMS in JADE) in main platform, querying auction information from auctioneer agent and bidding with bidders and so on.

So "Get Available Locations" and "Bid" are modelled as two subactivitites in a swimlane depicted in figure 2.

3 UML METAMODEL FOR OUR NEW MODEL ELEMENTS

In the previous section, we have defined a new stereotype <<Host>> + parameter in order to capture the notion of location. This is defined in the metamodel in (1) in the figure 3. Another element we define is subswimlane, which is syntactically defined in (2) in the figure 3. The swimlane is also syntactically defined as in (3) in the figure 3. We use stereotype and tagged value to present the syntax of moving from host 1 to host 2 of a mobile agent.

We use two tagged values, "Go", which is defined in (4) in the figure 3 and "Clone" (see (6) in the figure 3) to represent the syntax of move and clone behaviours ((5) and (7) in the figure 3) respectively.

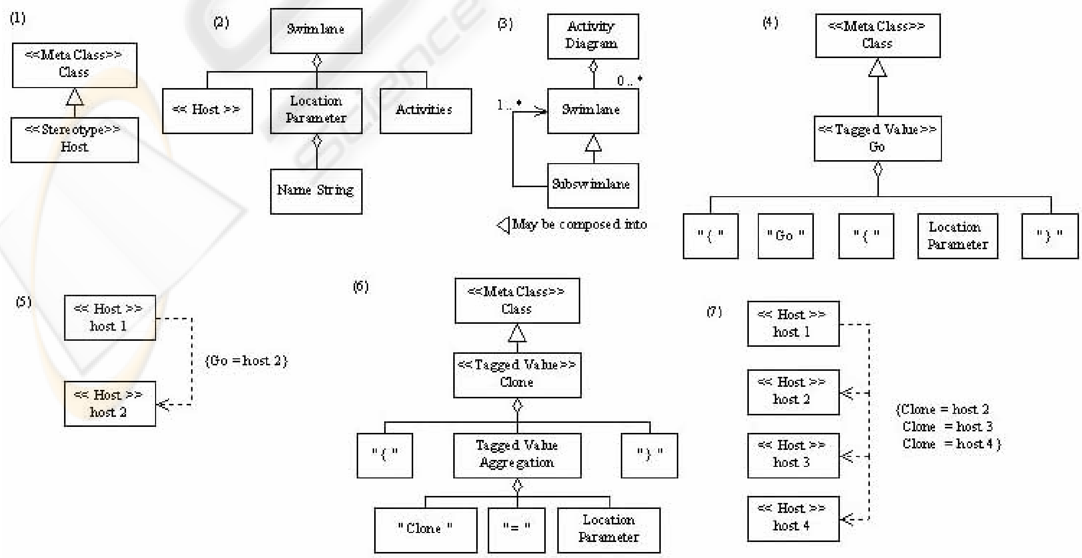

Figure 3: Metamodel for our new approach

4 RELATED WORK

The most relevant work is the FIPA modelling standard in deployment and mobility of multi-agent systems (FIPA Modeling TC, 2003), which uses activity diagrams and deployment diagrams as part of AUML. AUML deployment diagrams statically models movement of agents by incorporating a new association with a stereotype <<moves>> and nods in a network in which agents are modelled as software components. Algorithmic behaviour of an agent is modelled in an ordinary Activity diagram, but movement is captured by using symbols for a node and a decision which an agent will take in order to migrate different nodes in a network.

Lind adopted UML Activity diagrams to model agent interaction protocol without introducing any new modelling elements (Lind, 2002). He used swimlane as a construct which denotes a role of an agent and synchronous communications by giving new stereotypes. However, as mobile agent systems being inherently multi-agent, it is hard to use his notation for more complex communication patterns, which require more channels between different agents.

The most recent approach (Baumeister, et. al., 2002) related to ours is an extension of UML class diagram and activity diagrams to model mobile system. Their notation, which is based on a mobile calculus mobile ambient (Cardelli and Gordon, 1998), assumes both mobile objects and locations could migrate to another location. The crucial difference between ours and theirs is that we use a different mobility model in which we do not allow a location to move to another location while some agents running in parallel.

The biggest advantage of our work is its straightforward visual representation of mobility of agent, which helps to capture dynamic and algorithmic behaviour of mobile agents and to understand how they interact with each other in terms of locality.

5 CONCLUSION AND FURTHER WORK

In this paper, we presented extensions to activity diagram for modelling mobile agent applications. A new stereotype <<Host>> + *parameter* for swimlane, which represents the location is introduced in order to capture mobility of agents. Other specific features of agents such as communications, cloning are defined by existing model elements with a new rule of subactivities. We also proposed a new notation *subswimlane* in order to model nested places.

One of drawbacks is that an activity diagram basically models a *single multi-threaded* agent. Even our proposed notation clearly visualize locality and communications between agents, it is hard to verify all communications and movement are eventually synchronized between agents. The best way to overcome this drawback is to translate activity diagrams into some other formalism such as process algebra, which provide verification procedure of the model. Fortunately our concept of location can be easily mapped into mobile calculi, such as mobile ambient, which provides verification procedure as well as semantic foundations of our notation.

REFERENCES

Taguchi, K. and Dong, J. S., 2002. *An Overview of Mobile Object-Z.* In ICFEM'02, 4th International Conference on Formal Engineering Methods, LNCS2495 144-155.

Booch, G., Rumbaugh, J. and Jacobson, I., 1999. *The Unified Modelling Language User Guide*, Addison-Wesley.

White, J., 1996. *Mobile Agents*. In Software Agents, Bradshaw, J., editor, MIT Press 437-472.

Bellifemine, F., Poggi. A., Rimassa, G., 1999. *JADE – A FIPA-compliant agent framework*, In the Proceedings of PAAM'99, London, April 1999, 97-108

FIPA Modeling TC, 2003. FIPA Modeling Area: Deployment and Mobility.

Lind, J., 2002. *Specifying Agent Interaction Protocols with Standard UML*. In AOSE'01, 2nd International Workshop on Agent-Oriented Software Engineering held at the 5th International Conference on Autonomous Agents, LNCS 2222 136-147.

Baumeister, H., Koch, N., Kosiuczenko, P. And Wirsing, M., 2002. *Extending Activity Diagram to Model Mobile Systems*. In NODe'02, International Conference NetObjectDays, LNCS 2591 278-293.

Cardelli, L. and Gordon, A., 1998. *Mobile Ambients*. In FoSSaCS'98, 1st Conference on Foundations of Software Science and Computation Structure, LNCS 1378, 140-155.

Bauer, B., Muller, J. P., Odell, J., 2001. *Agent UML: A Formalism for Specifying Multiagent Software Systems.* In Proceedings of the First International Workshop on Agent-Oriented Software Engineering (AOSE-2000) held at the 22nd International Conference on Software Engineering, Limerick, Ireland, Springer Verlag.