Author(s): Melomey, Divina., Mouratidis, Haralambos., Imafidon, Chris.
Article Title: An Evaluation of Current Approaches for Modelling Mobility of Agents
Year of publication: 2007
Citation: Melomey, D., Mouratidis, H., Imafidon, C. (2007) "An Evaluation of Current Approaches for Modelling Mobility of Agents", In Dodds, S.J., Hosny, W., Jahankhani, H., Mouratidis, H., Palmer-Brown, D., Perryman, R., Wijeyesekera, D.C. (eds.) Advances in Computing and Technology, ICGES Press, London 2007 pp 71-78
Link to published version:
http://www.uel.ac.uk/act/proceedings/documents/ACT07.pdf
ISBN: 0-9550008-3-1
Publisher statement: (not stated)

AN EVALUATION OF CURRENT APPROACHES FOR MODELLING MOBILITY OF AGENTS

Divina Melomey, Haralambos Mouratidis, Chris Imafidon
Innovative Informatics Research Group, School of Computing and Technology
{divina, haris , chris12}@uel.ac.uk

Abstract: The development of agent-based systems requires methodologies and modelling languages that are based on agent related concepts. Towards this direction, research has proposed a large number of Agent Oriented Software Engineering (AOSE) approaches to modelling mobility of agents. This paper will evaluate the current approaches and methodologies with respect to modelling mobile agent systems and it will propose a number of concepts required to adequately model agent mobility.

1. Introduction

An agent is a computer program that demonstrates characteristics such as social ability, reactivity, pro-activeness, and autonomy (Wooldridge and Jennings 1995).Mobile agents are special types of agents that possess all the characteristics of an agent but they also demonstrate the ability to move or migrate from one node of a network to another. Mobile agents (Milojicic et al., 1999) (Jansen and Karygianni,1999) have received considerable attention from industry and research community, since their special characteristics help to address network issues such as network overload, network latency, and protocol encapsulation, just to name a few .

Due to the popularity of the agent technology, mainly in the research environment, there has been an influx of software engineering methodologies for the development of multi-agent systems (i.e. systems that consist of more than one agent). Current approaches model static agents and little or no attention has been given to the modelling of mobile agents.

Nevertheless, for mobile agent systems to become widely acceptable there is a need for a methodology to be developed which addresses various issues related to the mobility of agents. For instance, methodologies should assist developers to determine at the onset which agents should remain stationary and which needs to migrate on the network and hence how these could be modelled.

This paper provides an overview of current approaches and modelling languages for modelling multi-agent systems, and their limitations with respect to mobile agent systems modelling. It proposes a set of concepts and a modelling language necessary for modelling mobile agent systems. The layout of the paper is as follows; section 1 provides an introduction to agent technology while section 2 presents the state of the art and limitations (with respect to agent mobility) of existing approaches and modelling languages. Section 3 presents the concepts to model mobility of mobile agents while section 5 concludes the paper and also presents future works.

2. State of the art and Limitations of existing approaches and modelling languages:

A number of approaches and modelling languages have evolved since the emergence of agent technology. Notably among approaches for modelling agent systems that have emerged are Gaia (Wooldridge et al., 2003), MESSAGE (Caire et el., 2000), TROPOS (Bresciani et al., 2003) and Multi

Agent Systems Engineering (MASE) (Self & DeLoach, 2003), Prometheus(Padgham and Winkoff, 2002). Some of the approaches mentioned above mostly concentrated on design issues such as modelling static mobility. Few attempts were also made at modelling the dynamics of mobility of the agents. There are inadequate concepts to specifically model mobility of mobile agents.

Chhetri et al. (2006) developed ontology that describe concepts, and the relationships that exist between them to model mobility issues. The core concepts defined does not include a continuous link that depicts mobility among different components or interactive agents, which presumes the survival of mobile agents. Their ontology did not implicitly define what agent and mobile agent are but presume an agent becomes a mobile agent when it is assigned a role and also see mobility as attribute.

This therefore implies that a designer cannot reason about mobility during the requirement phase of systems development. Their ontology did not specify security to mobile agent.

2.1 Overview of Current Approaches

There are other approaches to model mobility of agent. These approaches do not form a complete methodology on their own, but they stem from the component, elements and diagrams of the Unified Modelling Language (UML).UML provides unification and formalization for methods of numerous approaches to the object oriented software systems lifecycle while Agent UML provide same functionality but for agent oriented systems.

An approach such as Gaia was not built on UML. Agent Modeling Language(AML) is also another modelling language specified as an extension to UML 2.0(Cevenka et al.,2005) (Cevenka et al.,2005b) (Cevenka and Trencansky,2004). Some approaches such as Gaia did not use UML at all.

2.1 Gaia Methodology

The Gaia methodology (Juan et al., 2002) focuses on analysis and design of agent based system. It provides analyst tools to develop a system from the systems requirement to detailed design which allows for direct implementation of the system (Wooldridge et al., 2000). Gaia models a complex system using agent concepts. Gaia defines responsibility when it assigns roles to agents.

However, Gaia lacks concepts and graphical notations to support modelling and reasoning about mobility of agents' vis-à-vis their social interaction with each other in a multi agent environment.

2.2 TROPOS

TROPOS as a requirements-driven methodology was developed to support analysis and design activities (Bresciani et al., 2004) (Castro et al., 2002). TROPOS covers the early and late requirement phases, as well as the architectural design and implementation phases. Its greater strength lies only in identifying early requirements for the system in spite of the fact that it has a broader coverage of the entire software development process.

However, TROPOS has not been developed with mobile agents in mind and therefore it fails to provide the necessary processes and concepts to model mobility of agents (Bresciani et al., 2004).

2.3 MaSE Methodology

Multi-agent Systems Engineering (MaSE) is a methodology (DeLoach & Self, 2001), (DeLoach, 2004),(DeLoach, 2006). From all the available AOSE methodologies, it is only MaSE that managed to model some aspects of agents' mobility using UML. In particular, MaSE makes provision of tools which enables developers/designers to specify where and which location an agent can migrate to, which task and

communication processes should be retained and which should not (Self & DeLoach, 2003). However, they only focused on the output models of the analysis phase of the systems lifecycle, and they also fail to identify why mobility is needed by some agents, and the association with the system requirements.

2.4 AUML Extensions

As mentioned above, apart from the methodologies for the development of agent systems, there have been few efforts to develop modelling languages and definition of some concepts that can be employed for the modelling of agent and mobile agent systems. In particular, Poggi et al. (2004) extended AUML deployment and activity diagrams with concepts and notations such as home, mobility path, destination, visitor, dotted lines to represent messages and dash lines with arrows pointing towards platforms that a mobile agent might be visiting. These concepts and notations have been introduced to extend the deployment diagrams (Poggi et al. 2004). All these concepts and notations introduced are geared towards modelling the static movement of the mobile agent, without paying particular attention to the dynamic mobility of an agent. Another important issue for mobile agent systems is security. However, the proposed concepts and notations fail to allow developers to consider security issues that might be present on their mobile agent systems.
Furthermore the issue of time was also not addressed by the proposal put forward by (Poggi et al., 2004). For instance, it is not possible to model when a mobile agent decides to move from one node to another.

Regarding activity diagrams, Poggi et al. (2004) introduced concepts such as return path, bounced failure and notations to indicate two statements with two arguments. These concepts and notations are intended to capture the dynamics of the agents i.e. concurrency, sequence and iterations of the movement of the mobile agent. This extension only captured the sequence of activities and knowledge provided by the designer so that a mobile agent can make an informed choice.
However, this was not fully realized since there was no continuous established link for which the mobile agent can make independent decision on its movement i.e. to and from it previous platform. There was also no indication whether a mobile agent has the necessary permissions to visit certain platforms. In addition, there was no mention or indication whether the agent has any kind of itinerary or not, and the kind of activities it does on its way to accomplish a task or a goal.
Similar to the work by Poggi et al (2004), Baumeister et al (2003) presented new stereotypes such as mobile, mobile location, at location, clone and move to model mobility in mobile systems which is an extension to activity diagram. New concepts such as mobile objects, locations and actions to moving mobile objects were introduced by the authors. Location that is contained in another is called nested location was also considered. Two notional variants were also introduced. These are location and responsibility centred. These provide answers such as who is performing an action and where the action is being performed. Swimlanes were introduced to represent objects showing who is performing a particular action as well as mobility of an object with respect to topology of location
However, the concept of nested location was not properly defined and illustrated. The idea of mobile location lacked clarity. Even though the extension to the activity diagram was to model mobility in mobile systems, concepts introduced has no direct bearing to neither agents nor mobile agents. All references were made to objects.
(Kosiuczenko, 2003) introduced the stereotype class *move* in sequence diagram to model mobile objects. It further introduced stereotypes for cloning objects which are *create* and *copy*. Mobile objects in

this extension can change its location when it performs a jump action. Concepts on nested topology were also presented by the authors.
Changes do take place during the life line of a mobile objects and hence the ability to trace mobile objects that perform the jump action. The lifeline therefore contains all the jumps right from the first place the mobile objects appeared. According to the author, the lifelines contain all jump arrows of the mobile object and its host and ends where the lifeline of the mobile object ends or terminates.
This extension, however, focused on objects and not agents. There is also no formal semantics for modelling the sequence diagram, hence lack of tool support to aid the analyst to perform a thorough analysis of systems.

2.5 Agent Modeling Language (AML)

AML is specified as an extension to UML 2.0 is a semi visual modelling language. It is used to specify, model and document systems that incorporate concepts and features of multi agent systems theories and existing abstract models such as TROPOS, Gaia, MESSAGE,UML, PASSI, Prometheus and MaSE (Trencansky and Cervenka, 2004b). In modelling the deployment of Multi Agent systems (MAS), AML attempted to provide support for mobility by identifying the following main elements: the agent execution environment, the hosting property, dependencies i.e. the move and clone, and lastly actions of move and clone (Trencansky and Cervenka, 2004b). However, there was no supporting model or construct to model the mobility of the agent. No mention was made of mobile agents and how their movement can be captured. Clearly, AML focus is not on mobile agent but rather on multi agent systems.

3. Building an Ontology for Modelling Agent Mobility

As mentioned and proved above, there is no single approach to guide the designer to reason about mobility from conception of an idea to its completion. An approach for modelling mobility issues of agent-based systems should have a set of modelling tool, a highly expressive modelling language and well documented semantics to assist software engineers to reason and model agent mobility issues as well as incorporating security where necessary.
Below we present a list of concepts (along with their definition) that we have found are necessary to be included in a complete ontology for modelling mobile agent systems.

3.1 Mobility Concepts

To overcome some of the limitations identified in the earlier section, this paper therefore presents a new and enhanced set of concepts to model the mobility of agents. Due to lack of space we present only brief definitions of concepts. These concepts are software agent, stationary agent, mobile agent, platform, home platform, host platform, , summit, mobility link, weak mobility, strong mobility, itinerary, task, goal, zone, permissions, sleep mode and knowledge base.

Software Agent
As mentioned above, software agent can be either stationary or mobile. It is important therefore to allow developers to model both types of agents. An agent comprises of code and state information needed to carry out some kind of computation. We differentiate a software agent to stationary agent and mobile agent.

Stationary Agent is an agent that is stationary. In other words, an agent which executes in the place it started. Stationary agent does not move.

Mobile Agent
This is an agent capable of moving among different platforms.

Platform
For an agent (and therefore mobile agent) to run, a platform is required; in other words an agent platform provides the computational environment in which an agent operates. For the purpose of modelling mobile agents, our work models a platform as networks of computers or independent nodes, irrespective of size. A platform offers resource services to other agents that enter it. For modelling mobility, two types of platforms are required.

Home Platform
This is the location where an agent originates.

Host Platform
Any platform a mobile agent migrates to apart from its home platform.

Summit
Summit allows two or more agents and/or mobile agent to meets in the same computer. Here, a mobile agent decides to migrate to meet with another stationary agent on a server platform for a service.

Mobility Link
A Mobility Link establishes a link or a session between or among agents. A link can be established only if the agents can identify each other. A mobility link can be terminated by either agent at both ends of the established mobility link. Whiles a link is established, an agent must not move to another place or location on the platform; should this happen, the mobility link will be implicitly terminated. Therefore in this context mobility link will be used to synchronise agents that want to meet for a summit. Mobility link allows a connection to be made regardless of the distance. It also enables a mobile agent to obtain a service remotely and the return to its home platform. A user's agent for example should be able to obtain flight information and book a flight for the user. On its return to the home platform, the user's mobile agent should be able explain to the user, the type of ticket booked, be it first class or economy.

Weak Mobility
This involves a situation where an agent gathers or stores no information on previous host visited. This is suitable to collect on line data to perform simple control and configuration tasks from several networks elements. It also leads to the reduction of network load.
Weak mobility copy only code. Program execution starts from initial state e.g. java applets

Strong Mobility
This preserves accumulated information upon migration. In addition it is able to process data from network elements. It is also able to preserves its state and form during previous visits.
Strong mobility copies code and execution. It resumes execution where it stopped but doesn't necessarily have same resources on current platforms.
Migration process ceases at originating site.

Itinerary
Itinerary represents the mobility plan of the mobile agents' movement.

Task
A task is any action or series of actions an agent or mobile agent can perform as part of its itinerary and its goals.

Goal
A goal is a specific objective an agent aims to accomplish. This is what motivates it to meet for a summit, hence establishes a mobility link in order to achieve this goal.

Zone
This a collection or a group of platforms operated by the same authority. To this end, a source mobile agent should provide enough proof to the destination zone else access will be denied.
A mechanism therefore will be provided to verify the authority of a mobile agent migrating from zone to zone.
Hence authority will limits what platforms and agents can do at any point in time.

Permissions
Permissions will grant the right to execute an instruction or perform an action i.e. ability to create another agent and to grant them rights to use certain resources and a life to live such as a few hours or days after which it terminates.

Sleep Mode
This affects and monitors changing conditions. This occurs the moment a mobile agent put itself to sleep until such as a time it needs to be active. For example when a trip is book for a later date, on the day of the flight, the mobile agent awake and inform about any delay and/or of the details of the trip.

Knowledge Base
These are rules that will be loaded in to the mobile agent at the start time which will enable the mobile agent to make an informed decision.

4. Conclusion and future work

In this work, we have examined the existing methodologies and approaches used in modelling agent mobility; we have presented critical concepts needed to model mobility. There are still more of these mobility concepts than space will allow us. Our primary aim, in this paper, was to evaluate all current approaches for modelling mobility. Our research indicated lack of a complete approach to model all the issues related to modelling mobile agents. The approaches are also not complete in themselves, in that they lacked proper illustrative examples; all examples used are not complex enough to reveal weaknesses in the approach.
In our future work, these concepts will be modelled and evaluated using an exemplar with a supporting modelling tool as well as a supporting documentation.

5. References

Bauer B., Müller J. P., Odell J. "Agent UML: A Formalism for Specifying Multiagent Interaction,", 22nd International Conference on Software Engineering (ISCE), *Agent-Oriented Software Engineering*, Paolo Ciancarini and Michael Wooldridge eds., Springer-Verlag, Berlin, pp. 91-103, 2001.

Baumeister, Nora Koch, Piotr Kosiuczenko, and Martin Wirsing. “Extending Activity Diagrams to Model Mobile Systems”. In M. Aksit, M. Mezini, and R. Unland, editors, Objects, Components, Architectures, Services, and Applications for a Networked World. International Conference NetObjectDays, NODe 2002, Erfurt, Germany, Oct. 7-10, 2002. Revised Papers, volume 2591 of LNCS, pages 278-293., 2003.

Bergenti, Federico; Gleizes, Marie-Pierre; Zambonelli, Franco (Eds.) Kluwer Academic Publishing (available via Springer), 2004.

Bresciani, P. Giogini, F. Grunchiglia, J. Mylopoulos, and Perini A. “Tropos:An Agent-Oriented Software Development Methodology”. Journal of Autonomous Agents and Multi-Agent Systems. Kluwer Academic Publishers, 2004

Caire, G., Coulier, W., Garijo, F.,Gomez-Sanz, J., Pavon., J. Kearney, P. and Massonet. P.” MESSAGE: A Mehtodology for the Development of Agent-Based Appplications, To appear at Methodologies and Software Engineering for Agent

Systems, edited by Federico Bergenti, Marie-Pierre Gleizes and Franco Zambolleni, to be Published by Kluwer Academic Publishing, 2004

Castro, J., Kolp M., and Mylopoulos. J. "Towards Requirements-Driven Information Systems Engineering: The Tropos Project". In Information Systems, Elsevier, Amsterdam, The Netherlands, 2002.

Cervenka R.and Trenansky I. "Agent Modeling Language". Version 0.9. Technical report, Whitestein Technologies, 2004.

Cervenka R.and Trenansky I., and Calisti M.. "Modeling Social Aspects of Multiagent Systems. The AML Approach". In J.P. Muller and F. Zambonelli, editors, The Fourth International Joint Conference on Autonomous Agents & Multi Agent Systems (AAMAS 05). Workshop 7 : Agent – Oriented Software Engineering (AOSE), pages 85-96, Universiteit Utrecht, The Netherlands, 2005

Cervenka R. and. Trenansky I, and Calisti M., Greenwood D.. "AML: Agent Modeling Language. Toward Industry-Grade Agent-Based Modeling". In J. Odell, P. Gioginin, and J.P. Muller, editors, Agent-Oriented Software Engineering V: 5th International Workshop, AOSE 2004,pages 31-46, Springer-Verlag, Berlin, 2005.

Cysneiros L. M., Werneck V. and Yu E. "Evaluating Methodologies: A Requirements Engineering Approach Through the Use of an Exemplar". The Journal of Computer Science and Technology, Vol. 5No.2

DeLoach Scott A.."Multiagnet Systems Engineering of Organization-based Multiagent Systems". 4th International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS'05). May 15-16, 2005, St. Louis, MO. Springer LNCS Vol 3914, Apr 2006, pp 109 - 125.

DeLoach Scott A., Wood Mark F. and Sparkman Clint H., "Multiagent Systems Enginering", The International Journal of Software Engineering and Knowledge Engineering, Volume 11 no. 3, June 2001.

DeLoach Scott A.. "The MaSE Methodology. In Methodologies and Software Engineering for Agent Systems". The Agent-Oriented Software Engineering Handbook Series: Multiagent Systems, Artificial Societies, and Simulated Organizations, Vol. 11. Bergenti,

Ivan Trencansky, Radovan Cervenka: Agent Modeling Language: A Comprehensive Approach to Modeling MAS. Informatica (Slovenia) 29(2) 391-400(2005)

Jansen,W. and Karygianni,T.(1999)"Mobile Agent Security, National Institute of Standards and Technology (NIST) special publication 800-19 , October,1999

Jennings N. R., Sycara K. and Wooldridge M. (1998) "A Roadmap of Agent Research and Development" International Journal of Autonomous Agents and Multi-Agent Systems 1 (1) 7-38.

Jennings N. R, Wooldridge M.."Agent-Oriented Software Engineering (2000)". Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99)

Juan T.,Pearce A., and Sterling L.,. "ROADMAP: Extending the Gaia Methodology for Complex Open Systems". In Proceedings of the first international joint conference on Autonomous agents and multiagent systems (AAMAS2002),

Bolognia, Italy, pages 3--10, 2002.

Kang M, Taguchi K.” Modelling Mobile Agent Applications by Extended UML Activity Diagram”. ICEIS(4) 2004: 519-522

Kosiuczenko P..”Sequence Digrams for Mobility”. Krogstie J. (ed.): Advanced Conceptual Modeling Techniques: ER 2002 Workshops, ECDM, MobIMod, IWCMQ, and eCOMO, Tampere, Finland, October 7-11, 2002, LNCS 2784, Springer, Berlin, 12 pages, 2003.

Milojicic D., Kotz D., Lange D., Petrie C., Rygaard C. “Mobile agent applications” IEEE Concurrency July to September 1999

Szolovits P., Doyle J., Long W. J.,. Kohane I and. Pauker S. G. “Guardian Angel: Patient-Centred Health Information Systems”. TR-604, Massachusetts Institute of Technology, Laboratory for Computer Science, 545 Technology Square, Cambridge, MA, 02139, May 1994

White J.E. “Mobile Agents, in Software Agent”, JM Bradshaw, Editor. MIT Press ... In Software Agents, J. Bradshaw, editor, MIT Press, 1996, pp. 437-472
Wooldridge, M., Jennings, N.R. and Kinny, D. “The Gaia Methodology for agent oriented analysis and design”. Autonomous Agents and Multi-Agent Systems, 3(3), 2000, pp 285-312

Yu, E., Cysneiros L.M. “Agent-Oriented Methodologies- Towards a Challenge Exemplar” in Proc of the 4th International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS 2002) Toronto May 2002.

Padgham L. and Winkoff, M., Prometheus: A Methodology for Developing Intelligent Agents , Proceedings of the Third International Workshop on Agent-Oriented Software Engineering, at AAMAS 2002, July,2002, Bologna, Italy.

Poggi A., Rimassa G., Turci P., Odell J., Mouraidis H., and Manson G. Modelling Deployment and Mobility Issues in Multiagent Systems using AUML, in Agent Oriented Software Engineering IV, P. Giorgini, J. P. Muller, J. Odell (eds.), *Lecture Notes in Computer Science 2935*, Springer-Verlag 2004.

Self A. &. DeLoach Scott A. “Designing and Specifying Mobility within the Multiagent Systems Engineering Methodology”. Special Track on Agents, Interactions, Mobility, and Systems (AIMS) at The 18th ACM Symposium on Applied Computing (SAC 2003). March 9 - 12, 2003, Melbourne, Florida, USA.

Wooldridge, M. and Jennings N. R. (1995), "Agent Theories, Architectures, and Languages: a Survey," in Wooldridge and Jennings Eds., Intelligent Agents, Berlin: Springer-Verlag, 1-22

Zambonelli F.,. Jennings N.R and Wooldridge M. “Developing Multiagent Systems: The Gaia Methodology”. ACM Transactions on Software Engineering and Methodology, 12(3): 317-370, July 2003.