Ontology-based Agent Mobility Modelling

Mohan B. Chhetri, Rosanne Price, Shonali Krishnaswamy and Seng W. Loke
Faculty of Information Technology, Monash University, Australia
mchhetri@it.swin.edu.au, {Rosanne.Price, Shonali.Krishnaswamy, Seng.Loke}@infotech.monash.edu.au

Abstract

There is increasing recognition of the potential benefits of mobile agent technology for distributed and mobile applications. However, thus far there has been only very limited consideration of the specific requirements for conceptual level modeling of agent mobility to support analysis and design of such applications. Furthermore, the fragmented nature and inconsistent terminology characterizing such research to date complicates the effort to understand and compare the work. An ontology describing the abstract concepts and inter-relationships required to model agent mobility is presented in this paper, intended to serve as the necessary first step to providing conceptual level support for mobile agent applications and as a cogent means of comparing existing work in the area.

1. Introduction

The term *Agent Oriented Software Engineering (AOSE)* refers to the use of a systemic approach to constructing agent applications [19]. With increased recognition of the potential utility of agent technology and adoption of this paradigm for building large-scale real-life systems, the need for methodologies that support the analysis and design phases of agent applications has become as critical as the need for tools and environments that support the implementation phase. This realisation fuelled the growth and development of AOSE methodologies [19]. These methodologies typically focus on the modeling concepts that are core to agent applications in general, such as the *roles* that agents assume and the *tasks/responsibilities* that agents have to fulfil.

In the area of mobile and distributed systems, a class of agents that possess the attribute of mobility is considered particularly useful [5]. Mobile software agents are typically light-weight processes that have the ability to migrate from host to host of their own accord. Mobile agents are largely perceived as being able to improve performance in terms of response time as well as in bandwidth usage in many data-intensive, distributed applications, since they provide a flexible means for migrating processes rather than potentially large amounts of data (e.g. in distributed data mining). Their migratory capabilities are also very useful in mobile and ubiquitous environments that are characterised by resource constraints and volatile network connectivity [4].

Given these advantages of the mobile agent paradigm, several mobile agent implementation environments or toolkits have been developed for both handheld and traditional devices (http://reinsburgstrasse.dyndns.org/mal/preview/preview.html, Accessed September 2004). These toolkits provide the APIs and the servers necessary to implement and deploy mobile agent applications. However, to date, efforts to cater for the engineering of mobile agent applications in early phases of the development cycle, i.e. analysis and design, have been quite limited.

The same rationale that was the driving force behind the development of generic AOSE methodologies provides the motivation for developing methodologies that facilitate the systemic construction of mobile agent applications. That is, with the increasing uptake of this technology, it is essential to provide strategies for formal analysis and design of these applications. However, mobile agent applications have received only minimal attention thus far in research reported in the area of AOSE methodologies [1] [12] [14] [15]. While these research projects take a step in the direction of conceptual modeling of mobile agent applications, they are still constrained by retention of a specific frame of reference or context as follows:

- They are often constrained by specific implementation considerations. That is, they fail to separate the conceptual model from implementation concerns.

- They are typically limited by the methodology that is used to represent the concepts. For instance mGAIA inherits the notational limitations and lack of formality of GAIA [17] and AUML [1] [6] (or any methodology based on the representative formalism AUML) is constrained by UML [8]. Attempting to adhere to the notational framework of existing methodologies and focusing on maintaining consistency at the notational level impedes the ability to develop a conceptual model that is not constrained by these considerations.
- The approach to modeling mobility of agents has thus far remained fragmented – in that different modeling approaches tend to focus on different concepts. Thus, the current state-of-the-art provides users with varying concepts depending on the methodology that they choose.
- The current approaches have a non-orthogonal perspective of mobility. For example, they tend to associate mobility only with agents. However, it is often likely that it is the roles that an agent assumes that determines its mobility and therefore it is important to consider the mobility as an attribute that can be assigned to roles as well as agents. This in turn opens up a whole gamut of associations, such as itineraries that are formulated as a consequence of role mobility and so on.
- Many methodologies also include in a fragmented manner concepts that are not specific to mobility. For example several current mobile agent-modeling methodologies include concepts such as *cloning* and *communication* [3, 9, 11, 13] since they are supported by mobile agent implementation environments/toolkits even though they are not specific to mobility at a conceptual level and pertain to the broader spectrum of agent systems. Thus, the methodologies tend to reflect the toolkits and this in turn results in the loss of certain abstractions that are implicit in the implementation but essential at the conceptual level (e.g. the tasks that an agent performs).

There is clearly a need for a holistic approach both to conceptual modeling of agent mobility—independent of a specific methodology, toolkit, implementation, or application domain—and to evaluating the level of conceptual support offered by current mobile agent methodologies. In fact, our preliminary investigation that yielded mGAIA and aimed to extend the GAIA methodology [15] clearly demonstrated the need for a comprehensive approach that is decoupled from any specific AOSE methodology. This provides the motivation for the current research—the development of an independent and holistic ontology describing agent mobility concepts, where an ontology can be defined in the information technology context as a domain-specific world view whose essentials are articulated either formally or informally in terms of a set of concepts, their definitions, and their inter-relationships [16]. This ontology is based on an extensive exploration and explication of concepts and relationships useful in the context of mobile agents. The development of our ontology follows a top-down approach where we focus first on the concepts that are essential and integral to modeling mobility in agent applications. We then establish the relationships between the identified concepts. We iterated this process to continuously refine our ontology. In representing the concepts and their inter-relationships, at this stage we use a simple Entity-Relationship notation that is well understood and clearly shows the base concepts and their relationships. The significance of this approach in terms of its support for conceptual modeling is as follows:

- It serves as a frame of reference to support conceptual modeling of mobile agent applications, i.e. it clarifies a range of mobility concepts/relationships that need to be considered and provides the context to aid application designers' thinking.
- It has the potential to facilitate documenting design decisions in terms of the ontology. This in turn can result in establishing the thinking behind key questions in AOSE process such as determining the number of agents based on the partitioning/assignment of roles to agents. As discussed in [18], a significant pitfall in the development of agent applications is finding the balance between too many and too few agents. Our approach facilitates documenting the rationale and design decisions that drive the choice of the number of agents in the system. This serves to clarify, evaluate, and potentially amend decisions regarding the number of agents as well as the roles they are assigned.
- Mobile agent application design based on the frame of reference provided by our ontology would make explicit the relationship between roles and agents and their connections to other core mobility concepts such as resources and itineraries.
- It aids in reasoning about application mobility concepts independent of a specific implementation, methodology or language.

We note that in disciplines such as database technology that have reached a more advanced stage of maturity, the advantage of establishing an ontology initially and the need for separation of concerns between conceptual and logical/implementation levels have been well recognised and enshrined in the methodology. This results in distinct roles being assigned to personnel involved in the various stages of development (e.g. analyst vs. programmer). As an emerging and primarily implementation-driven discipline, the mobile agent world is only starting to move in this direction. Thus the process of application development, including current conceptual modeling efforts, is driven by toolkit, methodology or implementation rather than a primary consideration of the conceptual requirements for a range of application domains. The result is the omission of important mobility concepts or the inclusion of extraneous concepts (i.e. not directly related to mobility or implementation-level artefacts) in models intended to support conceptual modeling of agent mobility, as discussed in Section 3. For example, the inclusion of implementation-level concerns such as *resource binding by value* or *by type* in [3] unnecessarily complicates the resulting mobility model.

The paper is organised as follows. Our conceptual model for agent mobility—called AMM (Agent Mobility Model)—is presented in Section 2. In Section 3, we then compare our conceptual model of mobility to those described by existing mobile agent modeling methodologies. Although we discuss research into procedural/dynamic aspects of agent mobility (e.g. activity diagram based approaches such as [2]), our primary focus is on models describing core mobility concepts and their inter-relationships, namely AUML [1] [6] [13], mGAIA [15], SODA [12] and MASE [14]. We provide mappings between our concepts and the mobility concepts, whether equivalent or distinct, described by these methodologies. This comparison contributed to the refinement of our ontology. It clearly establishes the holistic and integrative perspective that our work provides as compared to the fragmented landscape of current mobile agent modeling methodologies. It is the first attempt to provide both (1) a comprehensive ontology for mobility as a sound basis for developing methodologies and tools to support mobile agent applications and (2) a comprehensive and cohesive comparison of current mobile agent modeling methodologies. We also note that our approach of (1) first identifying key concepts for a property of agents (e.g., mobility) in the form of an ontology and then (2) studying the manifestations of these concepts within methodologies (such as those studied here) can be used for other properties of agents. Different ontologies (each corresponding to a property) can provide different "lenses" through which one can view an AOSE methodology, each "lens" focusing on an aspect of the methodology (i.e., the level of conceptual support for that property). Finally, we conclude in Section 4.

2. Key Concepts for Modeling Mobile Agent Systems

This section presents the key concepts for modeling mobile agent systems. We note that the identification of key concepts and their inter-relationships is typically a pre-requisite stage in any conceptual modeling exercise and is the precursor to the formal specification of concepts and their inter-relationships. In fact, these different stages can essentially be considered to be specification of an ontology at increasing levels of formality, with the initial stage expressed informally in natural language [16]. Thus an informal specification of an ontology for modeling mobile applications is presented in this section.

We distinguish the *core* concepts that are specific to modeling mobility of agents from *orthogonal* concepts that are applicable to agent systems in general. Orthogonal concepts are typically addressed in current AOSE methodologies that do not focus on mobility of software agents. These include concepts such as: *Permissions, Responsibilities, Cloning, Communication* and *Coordination, Billing, Internal Reasoning Models* and *Services.* Concepts identified as core for describing agent mobility include: *Roles, Agents, Home, Itineraries, Visits, Tasks, Locations, Migration Constraints,* and *Resources.* These concepts are considered individually and are depicted pictorially with their inter-relationships (solid lines linking entity rectangles with 1:1, 1:N or N:N cardinality) and sub-classes (entity rectangles augmented with a small box on the left with dotted lines to their super-class) in Figure 1 below. Note that (for readers who are not familiar with ER notation), following common ER diagram convention, all relationships shown have bi-directional semantics, but names are shown from the perspective of the left-most (or top-most for those on the same vertical line) entity in a binary relationship.

Of these core concepts, itineraries represent the single most important concept for modeling mobility. Fundamental to the orthogonal modeling of mobility is the understanding that mobility of agents can be *task driven, role driven, resource driven* (i.e. based on resource availability), or *functionality driven* (i.e. based on the requirements of a specific task) and the provision of explicit support for this via the direct

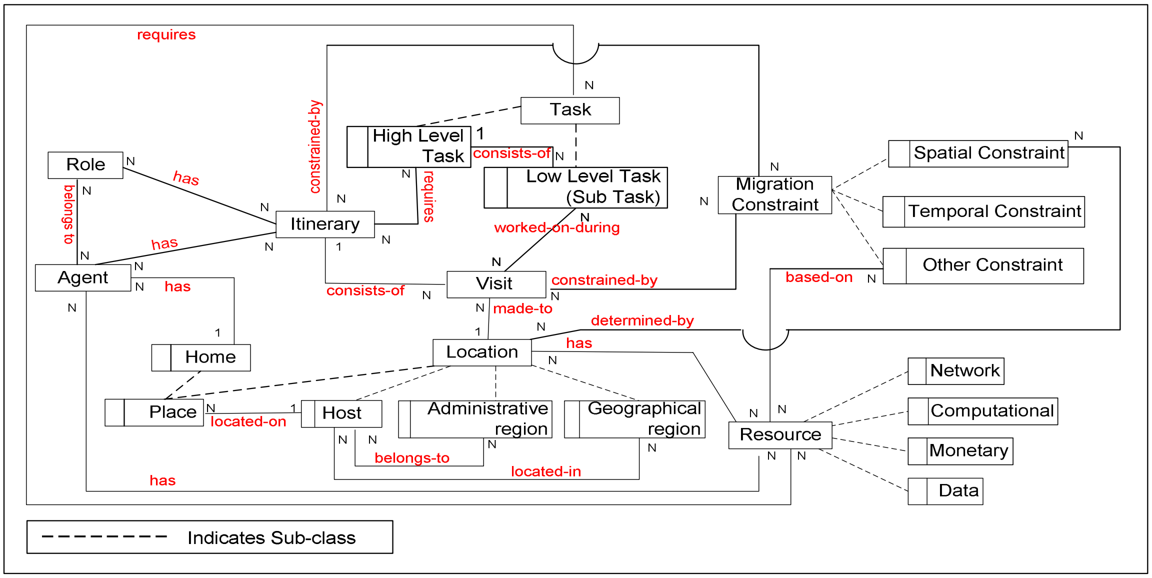

Figure 1. Agent Mobility: Core Concepts and their Inter-relationships

association of itineraries (an hence the attribute of mobility) with a role, agent, or task.

Roles: A role by definition is a position or purpose that any object/entity/agent has in a system, situation, organization, society or relationship. The notion of roles is widely prevalent in AOSE methodologies (e.g., [19] [17] [1]), particularly in the *analysis* phase, where it is essential to identify the requisite purpose of agents that need to be created. Typically, an agent in a system needs to acquire or take at least one role and may fulfil several roles as deemed necessary.

In the context of modeling mobile agent systems, we perceive *roles* as a core concept because a *role* may require the attribute of mobility in order to fulfil its responsibilities, commitments and obligations. Thus, an agent that embodies a role that has the attribute of mobility would necessarily be a *mobile* agent. This view of mobility of roles explicates the addressing of mobility at the analysis phase of developing mobile agent systems.

Agents: In general, an agent can be defined as a proactive, reactive, autonomous and communicative software entity situated in an environment. In the context of agents used for mobile and distributed applications, mobility of agents is obviously a core concept that needs to be modelled. Mobility may be associated with an agent indirectly by virtue of an agent role or task that requires that attribute, as discussed in the *Role* and *Task* sections respectively. However, even in cases where neither roles nor tasks specifically require the attribute of mobility; a software specification may require that agents be mobile in order to cope with unexpected changes in resource availability, e.g. forced migration in the case of host failure. This requires direct support for agent mobility, i.e. by allowing the attribute of mobility to be directly associated with an agent irrespective of that agent's roles or task.

Home: The notion of *home* is explicitly catered for in several mobile agent implementation environments. Home refers to the origin or place of creation of a mobile agent. Any agent – mobile or stationary - does indeed have a "home", but this concept becomes particularly important when an agent possesses the ability to move and visit several *locations.* In our model, *home* is a sub-type of place and is modelled separately as it possesses additional attributes such as forwarding messages to agents and may have distinct constraint specifications (e.g. spatial constraints such as an agents *home* must be within a certain administrative region).

Itineraries: Mobile agents or roles that require mobility implicitly have *itineraries* that determine a migration or travel path. This is the concept most critical to the conceptualization and modeling of mobility in agents. An itinerary typically specifies an ordering of *visits* to various *locations* to accomplish specific *tasks* (i.e. a *low-level* or *sub-task*). An itinerary could also be directly associated with a task or goal (i.e. a *high-level task* requiring the agent to migrate to certain locations in a specific ordered fashion), which can then be further specified in terms of the specific sub-tasks to be performed during individual visits. Furthermore, an itinerary is associated with *migration constraints* that determine the sequence of visits within an itinerary, and why and when a particular itinerary is chosen and undertaken. In order to ensure that the conceptualization of mobility is at a truly abstract level and thus supports the full range of possible mobile agent applications, it is essential that the concept of itineraries be treated as an orthogonal modeling concept. This means that itineraries, and hence mobility, can be associated freely with any other component of agent systems that might require the attribute (i.e. characteristic) of mobility, including agents, roles, and tasks.

Visits: As discussed in the context of itineraries, a visit refers to one stop within an itinerary. A visit involves a specific *location* and one or more *sub-tasks* that need to be performed at that location. Furthermore, the migration from one location to another may be dictated by visit-level *migration constraints* (i.e. specified at the level of an individual visit) as well as by those migration constraints specified for the itinerary as a whole.

Tasks: Tasks are activities that an agent is required to complete and are modelled in AOSE methodologies. We model tasks as high-level tasks or low-level tasks (subtasks). A high-level task normally requires migration and describes a high-level goal or job. A subtask normally does not require mobility and describes an activity to be performed in one visit.

Location: The concept of *location* is integral to mobile agent systems. Most mobile agent modeling methodologies such as mGAIA and AUML include constructs/notations for modeling location. However, they limit their scope to the *place* (also known as *agency* or *context)* when modeling location. A *place* is the run-time environment that allows hosting of mobile agents and supports their interaction with external computer systems. We take a holistic view of *location* wherein it is essential to model *geographic/administrative* domains and *hosts* (also known as *nodes* or *devices*) in addition to *places.* The modeling of location in these terms is to facilitate the greatest degree of flexibility and expressive power when specifying location, whether in conjunction with a visit, resource, or a spatial migration constraint. Modellers can thus vary the precision (e.g. specific *place* versus *host* versus *geographical region*) and complexity (e.g. a combined specification of a specific mobile host during the periods it is located in a specific administrative and/or geographical region) of location specifications. Note that this model of location explicitly accounts for the possibility of movement between places on a given host, as well as the typically considered migration between hosts. It further allows for the possibility of hosts and their associated places themselves having mobility, e.g. mobile devices such as personal digital assistants (PDAs) or places running on PDAs. This is important since, increasingly, mobile agent applications are being viewed as being particularly suitable for pervasive environments, which involve mobile and resource constrained devices.

As an example of the use of location in mobile agent application specification, consider the specification of spatial migration constraints for a visit or itinerary based on differential access rights/permissions. Another example would be the indirect specification of a variety of spatial constraints for a given agent, role, or high-level task via the use of an associated itinerary with an accompanying spatial constraint. By incorporating the additional notions of *geographic/administrative regions* and *place* in the modeling of location, we provide a more general modeling paradigm for mobile agents.

Migration Constraints: In specifying mobility through visits and itineraries, a concept that is often ignored is the *constraints* that determine the migration and their inter-relationships with other agent concepts. We note that these constraints can be associated with an individual *visit* as well as with *itineraries* as a whole. Migration constraints specify priorities that determine *when* an agent migrates, *where* an agent migrates to, and *why* an agent migrates. There are three types of constraints (separate or in combination) that can drive the migration:

1. *Spatial* – These constraints specifies where an agent should migrate to (or where not).
2. *Temporal* – These constraints specifies when an agent migrates (including time bounds).
3. *Other* – These constraints include all other conditions that are associated with migration and can include:
   - *Agent Population* - If the number of agents at a particular location exceeds a specified number, the agent under consideration should migrate to another location where it would face less competition for *resources.*
   - *Resource Unavailability* – If a particular *resource* is not available or accessible, the agent under consideration should migrate.
   - *Task Requirement* – A *task* requires the agent to be in a particular *location,* which necessitates the agent under consideration to migrate.
   - *Forced Migration* – An agent is forced to leave a specific location unexpectedly, e.g. in the case of host failure.

Resources: This concept is associated with *migration constraints, tasks, agents* and *locations.* A task may require specific resources (e.g. the availability of a network connection in order to book a movie) and a migration constraint may depend on the availability of certain resources. Conversely, an agent or location may be associated with (i.e. have) specific resources. A *resource* includes:

1. *Computational resources*
2. *Network resources*
3. *Data resources*
4. *Monetary resources (An agent might be given what is akin to monetary units in order to pay for computational resources they use.)*

3. Comparison of various methodologies

In this section, we review the mobile agent concepts from specific methodologies in terms of our proposed ontology and identify their limitations with respect to modeling mobility and mobility related concepts. In order to facilitate comparison of our proposal with existing agent modeling methodologies we use a consistent representation approach, the Entity-Relationship model, to illustrate the mobility concepts in the different methodologies. Furthermore we adopt the following conventions in the ER diagrams:

- we use the AMM naming convention (with the original author's terminology in parentheses) for concepts analogous to those in AMM,
- we indicate specific semantic differences between AMM and other authors' analogous concepts using asterisks with an accompanying explanation in the text,
- we represent any additional concepts in the methodologies that are not included in the AMM ontology with using the original authors' terminology,
- we include only those modeling concepts directly related to mobility (so concepts such as permissions, implementation level error handling, and agent communication would not be included as they are not considered core for conceptual modeling of mobility), and
- we represent any relationship cardinality not clearly specified in a given methodology using a question-mark ('?').

The methodologies listed below were chosen for this comparison because they included mobility either in their analysis or design phase. We note that there are several approaches to modeling procedural/dynamic aspects of mobile agent systems such as the semantics of a move operation [2, 9, 10, 11] and are typically based on activity diagrams [2]. The majority of this work includes only limited and primitive treatment of mobility as derived from specific toolkits or methodology constructs (e.g. *move, region, agent*) and will not be discussed further here. A more extensive consideration of mobility concerns is found [3], which considers the dynamic/procedural view (as indicated by the illustrations in the paper [3]) and is closer to the logical design/implementation phase is presented. We see this work as complementary to ours in that this stage represents the logical next phase in engineering mobile agent applications. However, we believe that it is essential to take a step back in this process and start with the core mobility concepts and inter-relationships in order to avoid several pitfalls. Common pitfalls include omission of important mobility concepts or inclusion of extraneous concepts unnecessarily complicating the model—such as implementation artefacts or concepts that are not specific to mobility (e.g. communication and cloning). For instance, compared to our proposed ontology, the conceptual model presented in [3] does not include the concept of *task* or *visit,* has a restricted definition of *location (missing home), itineraries (missing relationship to roles),* and *resources (missing computational and monetary resources),* and includes implementation artefacts such as *resource binding by value or by type* and *compile versus run-time specification of movements.* The latter is actually a reference to the implementation of *reactive versus proactive* mobility—which is treated at the conceptual level in our ontology as *forced migration constraints.*

As discussed in Section 1, we henceforth concentrate on comparing those methodologies describing core mobility concepts and their inter-relationships. Amongst the methodologies compared, Agent UML and mGAIA have done more work in modeling agent mobility than SODA and MASE.

- Agent UML [1] [13]
- mGAIA (Mobile GAIA) [15]
- SODA (Societies in Open and Distributed Agent Spaces) [12]
- MASE (Multiagent Systems Engineering Methodology) [14]

3.1. Agent UML

Agent UML, an extension of the object-oriented modeling language, the Unified Modeling Language (UML), has identified mobility as one of the principle modeling areas [1]. However, Agent UML does not consider mobility as a separate and orthogonal concept as it is considered only as an attribute of an individual agent. This is also true of migration constraints, which are considered only with respect to an agent. Furthermore, Agent UML considers movement of agents between hosts only and does not provide support for other types of movements, e.g. movement between two places within the same host.

Using the notational conventions described in Section 3, figure 2 shows the ER representation of Agent UML. One of the main limitations of Agent UML is that the notion of a *visit* is not fully developed or discussed. Agent UML does refer to elements such as the *origin* (host at which the agent begins execution which is indicated by using the stereotype *home*) of the movement, the *destination* (the host at which they stop execution after finishing their task), and the *mobility path* (the intermediate path between the origin and the destination, which could be a series of locations). We feel that there is no significant semantic difference between *origin, destination* and *intermediate visits,* beyond that previously described by *home.* Therefore, having three different concepts does not add information and should not be included as it results in an unnecessary increase in the complexity of the model. This argument holds true for the mGAIA methodology as well [15], wherein the concepts of *origin, destination* and *atomic paths* (or *visits)* are modelled. Furthermore in Agent UML, there is no distinction between the concept of *home* (where an agent is created) and *origin* (where an agent commences an *itinerary).* This approach seems to imply that an agent starts executing every *itinerary* at the place that it was created. This is not necessarily always true and is therefore an incorrect assumption. The cardinality between *itinerary* and *origin/destination* are not specified.

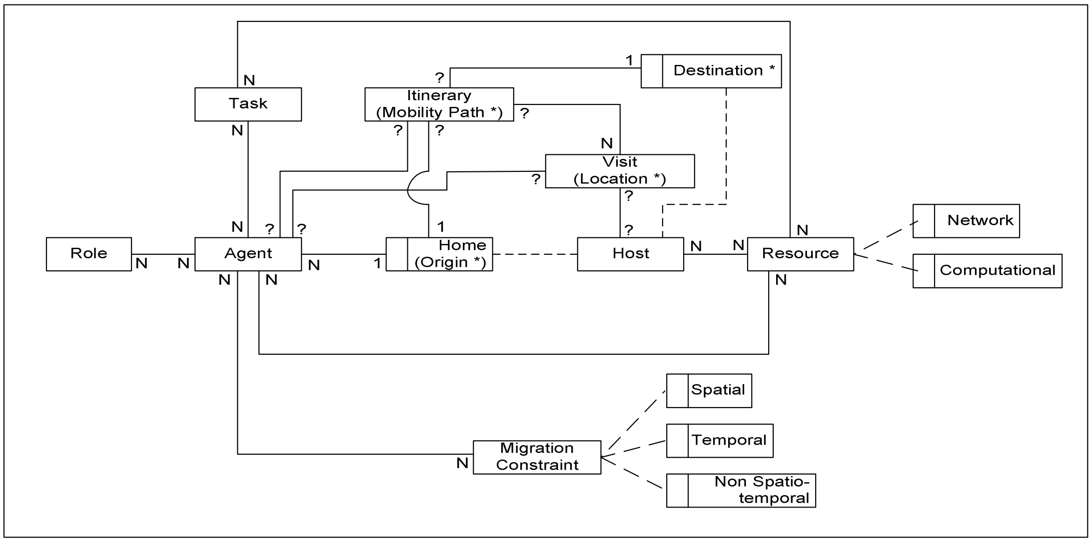

Figure 3: ER Representation of AUML

Agent UML uses the term *visitor* to indicate that the mobile agent is at a node, which is not its origin, or *home* [6]. However, they do not include any means of specifying what the visitor does during a visit (i.e. sub-tasks) or constraints governing specific visits (or itineraries).

Agent UML tends to group agents belonging to the same agent platform together. Hence it models the *Agent Platform* as a separate Component indicating which agents are housed on the platform itself [6]. We believe that this is an implementation artefact and should not be included in the conceptual modeling phase.

Agent UML does acknowledge the need for an agent environment (i.e. *locations* in our terminology) but defines it quite broadly, including not only physical entities in the agent's environment but also the principles and processes under which agents exist and communicate. Thus the term *environment* in Agent UML encompasses communication concepts (termed *communication environment* in Agent UML) and such concerns as determinism, diversity, and volatility (part of what Agent UML terms the *physical environment,* along with physical entities) [7]. These concepts are not specifically mobility related, and thus not considered core concepts for modeling agent mobility in our ontology. Furthermore, this conceptualization of *physical environment* does not correspond with our intuitive/common-sense understanding of the term (perhaps a more appropriate term to use in this context would be *metaphysical environment*). On the other hand, Agent UML does not support explicit modeling of *places, geographic regions,* or *administrative regions.* Thus *home* is treated as a type of host rather than as a place on a host, only migration between hosts is considered, and only resources of hosts can be modelled.

3.2. mGAIA (Mobile GAIA)

mGAIA, an extension of the GAIA methodology, does not handle mobility in the analysis phase but instead it models it in the design phase [15]. As with Agent UML, the main drawback of this approach is that mobility is considered only in conjunction with a specific agent rather than as an orthogonal concept. A further similarity to Agent UML is that mGAIA does not fully develop the concept of visit. Figure 3 below shows the ER representation of mGAIA. It can be seen (using the notational conventions described in Section 3) in figure 3 that several core concepts identified in AMM including *home*, *location*, *resources,* and *migration constraints* are not modelled in mGAIA.

The *travel path* of mGAIA is analogous to *itinerary* in AMM. It consists of several *atomic movements* which are analogous to *visits* in AMM. Each itinerary consists of the *origin* (where the mobile agent starts the movement to accomplish the task assigned), *final destination* (where the mobile agent will reside after completion of the assigned tasks) and *paths* (which are composed of a list of *atomic movements*). However, there is no notion of an agent *home*, of *migration constraints,* of *resources*, or of a separately defined concept of *location,* all core concepts in AMM.

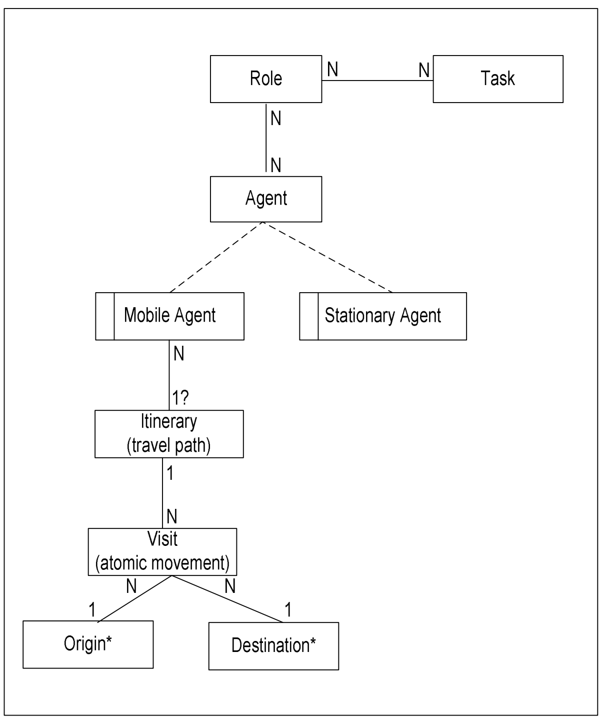

Figure 3: ER Representation of mGAIA Methodology

3.2. MASE (Multiagent Systems Engineering Methodology)

In an attempt to bridge the gap between existing AOSE methodologies and mobile agent systems and to support the modeling of mobility independent of implementation level concerns, the authors of MASE have incorporated mobility into MASE in [14]. However, the authors do acknowledge that they do not explicitly discuss the issues of *when, where* and *why* of agent movement [14]. There is no explicit consideration of *itineraries, home*, or *resources* as illustrated in figure 4 using the notational conventions described in Section 3.

They introduce the notion of a *move* activity in the analysis phase which returns two values: a Boolean value representing the result of the *move activity* (success or failure) and a “reason” (i.e. text) value which gives an explanation as to why the move failed. This move activity is just a process within a state in a concurrent task diagram and is modelled in terms of saving state and transfer, which is an implementation model of a ‘move’ command. Each *move activity* is analogous to the *visit* in AMM. However, the relationship between a task and a visit is not clearly defined in terms of cardinality. Additionally, MASE does not clearly distinguish between the concept of *location* and *host.* Each move activity is associated with a destination but it is unclear whether this destination refers to a *host* or a *place* within a host.

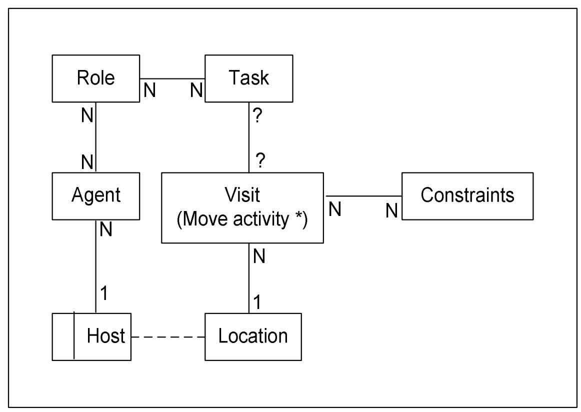

Figure 4: ER Representation of MASE Methodology

3.3. SODA (Societies in Open and Distributed Agent Spaces)

SODA is an extension of the Gaia methodology and considers mobility in the design phase. However, the main focus of SODA is to define a conceptual framework and a methodology for the analysis and design of multi-agent systems and not the mobility of agents. Hence, it does not focus on agent mobility. It explicitly models the agent environment, which can then be mapped onto specific agent infrastructure. It mentions mobility in the design phase in the *agent model*. In the agent model, each agent class which has a set of one or more roles is characterized in terms of its *cardinality*, its *location* (the location of the agent with respect to the topological model of the system) and its *source*. If an agent is static, then its location is specified as *fixed*. If the agent is mobile, then its location is said to be *variable* [12].

We can see from the ER diagram in figure 5 (again using the notational conventions described in Section 3) that in SODA there is no notion of the concept of *itinerary, visit, home* or *migration constraints*, all identified as core concepts in the AMM methodology.

The concept of *location* is not at all developed, minimally referenced as being variable for mobile agents and static otherwise. There is no association between an agent, its location and the tasks that it performs at that location. In fact, mobility itself is not modelled as a separate concept, but instead is seen solely as a characteristic of an agent.

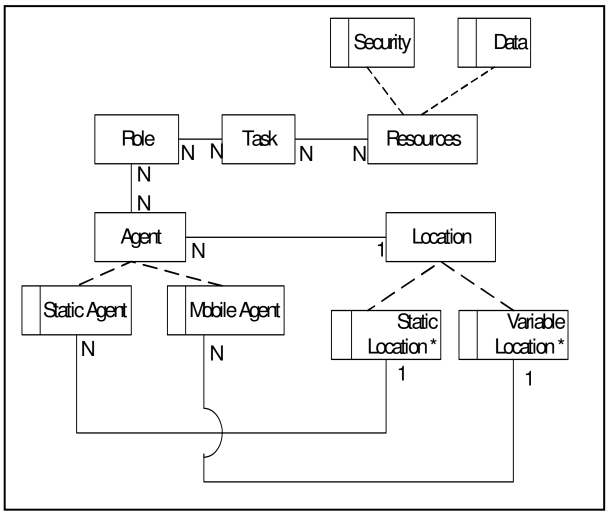

Figure 5: ER Representation of SODA Methodology

4. Conclusion

This paper presents an ontology for mobile agents that is intended to have general applicability and intelligibility independent of a particular AOSE methodology or toolkit and to focus on the essential concepts required to model mobility independent of implementation level concerns. The contributions of our work, differentiating it from other research efforts thus far in conceptual modeling of mobile agents, include:

- clear and explicit modeling of mobility using itineraries, thus an itinerary is identified as the critical concept integral to agent mobility and used to associate mobility semantics with other elements in the agent domain;
- orthogonal modeling of mobility which can be associated as needed with elements of the agent domain such as agents, roles, or tasks through the association of an itinerary. This contrasts with previous approaches associating mobility with a particular agent domain element, usually an agent;
- comprehensive and explicit consideration of mobility concepts not previously considered, including
  - different types of locations and movements (e.g. between places on a host or between administrative versus geographical regions),
  - different types of migration constraints and their relationships with other concepts,
  - different types of resources and their relationships with other concepts;
- clear and explicit specification of inter-relationships between mobility concepts, including those:
  - between itineraries and agents, roles, or tasks (high-level);
  - between an itinerary and its sub-components (i.e. visits);
  - between a visit and its sub-components (i.e. low-level tasks, migration constraints, location);
  - between resources and locations, migration constraints, agents, or tasks;
- provision for variance in mobile agent requirements, including,
  - support for mobility at agent, role or high-level task level using itineraries;
  - support for both general, high-level (for an itinerary) and specific, low-level (for a visit) migration constraints; and
  - support for a wide range of location and movement types not commonly considered.

Most significantly, the treatment of mobility as an independent concept orthogonal to other concepts in the agent domain is consistent with the previously stated goal of articulating a new perspective of mobility independent of a particular AOSE methodology, toolkit, agent implementation, or agent application.

On the basis of this ontology, we then provide a comprehensive, understandable, and cohesive comparison of those conceptual-level aspects of agent mobility considered to date by current mobile agent modeling methodologies. Thus the proposed ontology has the potential to serve as a sound foundation for developing methodologies and tools to support mobile agent applications and for evaluating competing methodologies in terms of the level of conceptual support offered. As described in Section 1, it further provides a frame of reference to guide mobile agent application design and design documentation by explicating in a systematic and clear manner the relevant mobility issues to be considered. Future work includes the development of a specification language and associated tools based on this ontology that can be integrated with current AOSE support and used to support higher level conceptual modeling (analysis and design) of mobile agent applications.

5. Acknowledgements

We thank Nitin Arora for his help in formatting this paper.

6. References

1. Agent UML: http://www.auml.org, Accessed September 2004.
2. Baumeister, H., Koch, N., Kosiuczenko, P., and Wirsing, M., Extending Activity Diagrams to Model Mobile Systems, In M. Aksit, M. Mezini, and R. Unland (eds): Objects, Components, Architectures, Services, and Applications for a Networked World. NetObjectDays, NODe 2002, Erfurt, Germany, Oct. 7-10, 2002. Revised Papers, volume 2591 of LNCS, pages 278-293, Springer Verlag, 2003.
3. Belloni, E, A., and Marcos, C, A., MAM-UML: An UML Profile for the Modeling of Mobile-Agent Applications. XXIV International Conference of the Chilean Computer Science Society, Arica, Chile, 8-12 November, pp. 3-13. IEEE CS Press. ISBN 0-7695-2200-9,
4. Brooks, C., and Krishnaswamy, S., Resource Monitoring to Support Mobile Agents in Pervasive Computing, Proceedings of the International Conference on Intelligent Agents, Web Technology and Internet Commerce (IAWTIC 2004), Gold Coast, Australia, July 12-14. 2004.
5. Du, T.C. Li, E.Y. and Chang, A. Mobile Agents in Distributed Network Management. Communications of the ACM, Vol. 6, No. 7, 2003, pp. 127-132.
6. FIPA Modeling Area: Deployment and Mobility (Working Document), 2003, http://www.auml.org/documents/DeploymentMobility.zip, Accessed September 2004.
7. FIPA Modeling Area: Environment (Working Document), 2003, http://www.auml.org/documents/Environment-030412.doc.zip, Accessed September 2004.
8. Henderson-Sellers, B. and F. Barbier. Black and White Diamonds, Proceedings of the International Conference on Unified Modeling Language, 1999, pp. 550-565.
9. Klein, C., Rausch, A., Sihling, M., and Wen, Z., Extensión of the Unified Modeling Language for Mobile Agents, In Siau, K., and Halpin, T., (eds): Unified Modeling Language: Systems Análysis, Design and Development Issues, Chapter 7, 2001, Idea Publishing Group.
10. Mouratidis, H., Odell, J., and Manson, G., Extending the Unified Modeling Language to Model Mobile Agents, Workshop on Agent Oriented Methodologies, OOPSLA 2002, Seattle, USA, November.
11. Muscutariu, S., and Gervais, M., On the Modeling of Mobile Agent-Based Systems, The 3rd International Worskshop on Agents for Telecommunication Applications (MATA'01), LNCS Vol. 2164, pp. 219 – 234, 2001, Springer Verlag.
12. Omicini, A. SODA: Societies and Infrastructures in the Analysis and Design of Agent-based Systems. Proceedings of the 1st International Workshop, AOSE 2000 on Agent-Oriented Software Engineering, Limerick, Ireland, 2001. pp. 185-193.
13. Poggi, A., Rimassa, G., Turci, P., Odell, J., Mouratidis, H., and Manson, G., Modeling Deployment and Mobility Issues in Multiagent Systems Using AUML, Agent Oriented Software Engineering (AOSE 2003), 2003, pp. 69-84.
14. Self, Athie. L. and Scott A. DeLoach. 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.
15. Sutandiyo, W., Chhetri, M.B., Krishnaswamy, S., and Loke, S. W. Experiences with Software Engineering of Mobile Agent Applications, Proceedings of the Australian Software Engineering Conference (ASWEC 2004), Melbourne, Australia, April 13-16, IEEE Press.
16. Uschold, M. and Gruninger, M. Ontologies: Principles, Methods, and Applications, Knowledge Engineering Review, Vol. 11, No. 12, 1996, pp. 93-136.
17. Wooldridge, M., Jennings, N. R. and Kinny, D. The Gaia Methodology for Agent Oriented Analysis and Design in the Journal of Autonomous Agents and Multi-Agent Systems, Vol. 3, 2000, pp. 285-312.
18. Wooldridge, M, J., and Jennings, N, R., Software Engineering with Agents: Pitfalls and Pratfalls, IEEE Internet Computing 3 (3) 20-27, 1999.
19. Weiß, G. Agent Orientation in Software Engineering, Knowledge Engineering Review, Vol. 16, No. 4, 2002, pp. 349-373.