SCALABLE COMPUTING: PRACTICE AND EXPERIENCE
Volume 7, Number 1, pp. 1–10. http://www.scpe.org
ISSN 1895-1767

AGENTS GO TRAVELING

DONAL O'KANE , G. M. P. O'HARE*, DAVID MARSH , SONG SHEN , AND RICHARD TYNAN

Abstract.
This paper is concerned with infrastructural support for nomadic agents. Agent migration provides a wide range of advantages and benefits to system designers, however issues relating to security and integrity mobile agents has mitigated against the harvesting of their true potential. Within this paper we introduce the Agent Travel Metaphor (ATM) which offers a comprehensive metaphor fostering integrating of control and security for mobile agents. We describe the metaphor together with its incorporation within the Agent Factory multi-agent system.

1. Introduction. In recent years much attention has been focused on the area of multi-agent systems and mobile agents. The term agent has numerous connotations and can represent various software entities.

In this paper the term agent implies an entity comprising of but not limited to the following properties: autonomy, social ability, responsiveness, pro-activeness, adaptability, mobility, veracity, rationality, and human cognition modeling techniques such as belief desire and intention. These define a strong notion of agency, [18]. A mobile agent refers to the capacity of an agent to electronically navigate a network in which it exists, [17].

Many arguments have been proffered as to the benefits and disadvantages of mobile agents, [5], and on the use of agent-oriented programming as a design paradigm, [11].

Numerous pitfalls have been identified, such as potential security issues involved in agent migration, [10, 5, 14, 4], interoperability and language translation, [13, 8, 2] and dynamic creation and management of an agent's itinerary, [15].

Various solutions to these drawbacks have been offered by which to address the difficulties associated with agent migration. If solutions could be found to these impediments, then the potential benefits to be harvested from secure agent mobility are immense. As yet, no system or approach has garnered universal support. The itinerant agent framework, [4] is an example of a framework and design that, allows agent platforms to offer a selection of services targeted at roaming agents, with no specific origin or home. [16] introduce the concept of airports for internet agents. These airports provide a framework for ad-hoc and unstable internet agents to access resources and maintain communicate channels. However these are targeted at specify types of mobile agents, not at the general population.

To address this lack of unity this paper introduces the Agent Travel Metaphor, (ATM), which describes and provides a natural method to deploy and implement a vast range of tools and services designed to offer mobile agents various services.

The metaphor draws from the experience of human travel and utilizes techniques from the natural world to provide services such as security, adaptation, core code protection, cooperation and interoperation.

This paper also proposes that the use of the ATM may provide a service backbone for mobile agents to exist, operate and lengthen their lifespan, by providing a range of services and adaptation mechanisms.

2. Mobile Agents—The Way Forward.

2.1. Related Research. The ability of an agent to migrate is defined as electronic transfer of an agent from one point in a computer network to another, [17]. Many multi-agent systems support agent mobility and provide services to enable their agents to migrate efficiently.

A diversity of enabling technologies have been adopted in order to underpin agent mobility. These include mobility security measures, agent language translation, middle agents, load balancing mechanisms, ad-hoc network tools and agent code protection.

Mobility Security: A variety of security issues arise when enabling agents with mobility. The main issues are protection of platforms from malicious agents, protection of agents from malicious platforms, protection of agents from malicious agents and protection of agents and platforms against other entities [10]. Many systems have been developed to tackle these four problems, including design techniques that limit an agents ability to affect the agent platform and scanning systems that ensure migrating agent are not carrying harmful code such as viruses and systems that utilize passport and visa documents and digital certificates, [9, 4], to enable

*Adaptive Information Cluster (AIC), Department of Computer Science, University College Dublin (NUI), Belfield, Dublin 4, Ireland. Gregory.OHare@ucd.ie

identification and the origin of migrating agents, and thus provide agent platforms with a method to measure how trustworthy a migrating agent is.

Agent language translation: If an agent is migrating between heterogeneous multi-agent systems then it is possible for an agent to migrate to an agent platform that utilizes a different agent communication language, (ACL), than the migrating agent. Agent language translation allows the migrating agent to communicate and cooperate with foreign agent platforms as well as foreign agents. Services and agents that facilitate interoperation between heterogenous multi-agent systems exist and use meta languages, [2], direct translation, [4], or ontologies,[8], to provide agents with a means of language translation.

Middle agents: These are agents that provide a variety of static services to mobile agents. They are generally accepted to be system level agents or trusted agents that provide services such as naming and access to shard resources, [7, 4]. Systems such as the the itinerant agents framework, [4], and the SOMA system, [1], use a variety of middleware software to ensure security, interoperability and communication services to mobile agents.

Load balancing mechanisms: Tools and services that provide load balancing for their network have been developed, [3].

Ad-hoc network tools: The ad-hoc nature of ever changing networks presents another massive set of problems for a mobile agent environment. For example, node disappearance and sudden unannounced reappearance causes a serious failure and recovery discovery problem. Nodes that fail are no longer reachable, agents executing on such nodes are also no longer reachable. Any dependant agents or services must either have the ability to handle this failure or be notified of its occurrence. Upon recovery or re-establishment of a node connection, the other nodes or entities must be made aware of the recovery, this is a non-trivial problem, [16].

Agent code protection: Protection for the core of an agents mental state against viral code and unauthorized access has been developed, [12]. Agent platform administrators can assume benevolence of known agents once their critical code is protected.

2.2. The Agent Travel Metaphor. The Agent Travel Metaphor, (ATM) adopted and introduced within this paper, borrows heavily from human travel. It consolidates and expands previous work by other researchers that have adopted components and segments of the overall travel scenario. Three classes of human travel can be identified as useful for mobile agents, *International* travel, *National* travel and *Metropolitan* travel. In the human environment these classes of travel all possess diverse procedures that determine issues such as how travel is initiated, how secure the travel is and a plethora of further services provided for the travelers.

Metropolitan: An example of metropolitan travel would be traveling on a bus through a city. For a traveler that lives in that city this form of travel will result in destinations that are both similar to the origin location and familiar to the traveler. The origin and destination locations are almost identical, travel can be initiated at any time with little or no preparation, security measures at at a minimum, the traveler has no requirement to prove their identity and must only purchase a ticket for a bus at the time of travel, and travelers already possess most of what they need to carry out their tasks at the destination, (language translation, behavior learning etc).

National: An example of national travel would be traveling on a train from one city to another, within the same country. This method of traveling is much stricter than metropolitan travel, there are more stringent security measures, the travelers must usually purchase a ticket in advance and must present this ticket upon request. The schedule is also much more regulated, train time tables are usually much more strictly adhered to than a metropolitan bus timetable, also a traveler may be required to pack items necessary for their tasks as they may not be available at the destination.

International: An example of international travel would be traveling on an airplane from one country to another. This method of travel has the highest level of security requirements, passport, visa, ticket and baggage checks at both origin and destination, a extremely limited timetable, only a few flights per day to particular destinations, and a limited choice of direct destinations, requiring several stops to get to a particular destination. Along with these is the possibility for a traveler to arrive at a destination with a different language and different behavior as standard, the traveler must be taught or discover how to conform to these new requirements. Packing baggage is most important for international travelers as it becomes more difficult for them to procure the necessary items at foreign destination that will enable them to perform their tasks.

An example of the steps taken by a traveler undertaking an international class of travel involves the following steps:

- Decide on destination(s).
- Contact a travel agent and negotiate a ticket price.
- Travel agent issues a ticket.
- Traveler contacts destination for visa requirements.
- Contact vaccination clinic.
- Query and obtain necessary vaccinations for destination(s).
- Pack necessary baggage.
- Upon arrival at the origin port contact port authorities.
- Ticket and passport are verified and baggage is scanned/checked in.
- Port initiates migration.
- Upon arrival at destination contact destination port authorities.
- Ticket and visa are verified and baggage is scanned and collected or left in secure box.
- Once through security measures contact language translator and/or behavior teacher as needed.
- Normal operation resumes.

```
Begin
    Select Destination                          // (a)
    Get Destination Description                 // (b)
    if( ! same language ){
        Contract Translator                     // (c)
    }
    if( ! same behavior ){
        Contract Teacher                        // (d)
    }
    Determine Packing Requirements              // (e)
    if( packing necessary){
        Pack Baggage                            // (f)
    }
    Obtain Necessary Travel Documents           // (g)
    Submit Travel document to Port Authority    // (h)
    Migrate once Authorized                     // (i)
End
```

FIG. 2.1. *The agent travel algorithm.*

In the metaphor, agents play the role of travelers, an evolution from the human travel analogy, as agents are typified by their use of human cognition techniques in their decision making process.

Accordingly, agent platforms play the role of cities, also a natural evolution from the travel analogy as an agent platform is the location in which an agent exists and interacts with its surrounding environment. The potential for agent platforms to create federations with groups of agent platforms and the analogous relationships between neighboring cities and countries, e.g. countries that have travel agreements and do not require visitors to apply for a visa to enter. This reinforces the previous argument in favor of the metaphor.

The variety of processes that we go through when we undertake a journey is unique for every journey and yet distinct patterns can be extracted, for example generalizing the security requirements, documents needed for travel and inquiring at the destination if language translation or behavior teaching is required leads to a generalized travel process as seen in Figure 2.1. The ATM is designed to facilitate configuration and policing of mobile agents policies and services in a heterogenous environment. The algorithm described in Figure 2.1 proposes a foundation for a framework, providing agent platforms, and their agents, with modular, secure, agile and adaptive agent mobility services. The provision of these services, allows for a flexible and easily configured environment, giving platforms the capability to create and control affiliations with other platforms and multi-agent networks.

2.3. Important Actors and Data Structures within the Agent Travel Metaphor. In order to implement an initial realization of the agent travel metaphor and its accompanying framework for incorporation within Agent Factory, it is first necessary to define some of the principle actors and data structures necessary to support the metaphor.

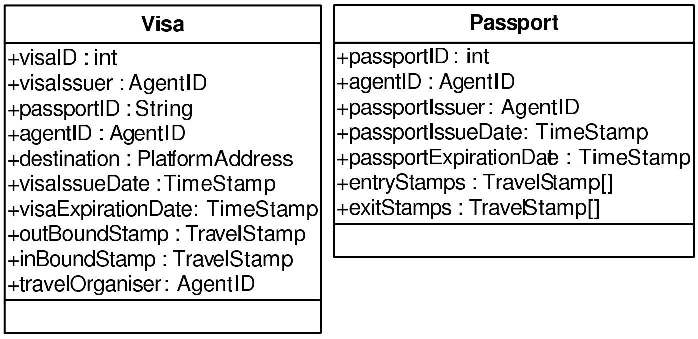

Fig. 2.2. *The Visa and Passport Data Structures.*

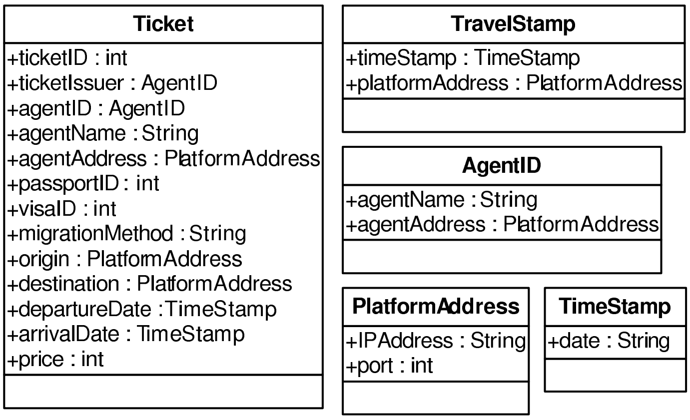

Fig. 2.3. *The Ticket, TravelStamp, AgentID, PlatformAddress and TimeStamp Data Structures.*

Passport Passports are an official certificate issued by a trusted source providing the identity of an agent and providing information on, an agent's origin, the creator of the passport and a history of an agent's travels.

Passport Issuer A passport issuer is a middle Agent contracted to create passport data structures for agents. Passport Issuers retain a copy of all created passports and provide a verification service for any agent wishing to ensure a particular passport is genuine.

Ticket Tickets are certificates that prove that a ticket holder has procured permission for a scheduled migration event.

Travel Organizer A travel organizer middle agent that is contacted by a traveling agent wishing to obtain permission to travel to a particular destination. The travel organizer contacts destinations and procures visas and creates a ticket for the traveling agent. Travel organizers also provide a verification service for any agent wishing to ensure a particular ticket is genuine.

Visa Visas are temporary, once off, certificates that are provided by destination platforms, providing an agent with entry permission to a destination platform.

Port a conceptual location at which all migration to and from an agent platform is coordinated.

Port Authorities Port Authorities are trusted agents charged with the task of operating the port for each agent platform. Each agent platform contains both a port and a port authority agent. The responsibilities of this agent include coordinating agent migration, upholding the local security policies and validating tickets, visas and passports.

Air Side a restricted area of an agent platform. Once An agent commences migration it is restricted from normal operation until it arrives landside at its destination.

Land Side the term used for the normal space for agent operation.

Baggage a collection of code or data, external to an agent's mental state, that the agent makes use of in order to fulfil its goals.

Secure Box a secure and private storage location attached to a port. An agent carrying baggage may deem portions unnecessary for the current location. These unnecessary portions can be stored ready to be retrieved once the agent requires them or leaves the current location.

Vaccinations a security and protection measure that allows agents to defend themselves from infections before migrating to a potentially dangerous/malicious location as well as allowing agent platforms to guard themselves from unknown migrating agents.

Language Translator an agent that can be contracted by a mobile agent to bestow the ability to converse with other agents that use different communication languages.

Behavior Teacher an agent that can be contracted to give an agent the ability to adapt to local operating behaviors. Some platforms within the network may require agents to register with its white/yellow pages services for example, while other locations may not.

3. Agent Factory and the Agent Travel Metaphor.

3.1. Agent Factory Mobility Support. Agent Factory, [6], is a multi-agent systems developed using the strong notion of agency. Agent Factory provides support and infrastructures that allow for rapid prototyping of agents. It imbues its agents with mobility via HTTP socket connections, transferring agent mental state and serialized java code. Federations of agent management services, (AMS), and directory facilitators, (DF), provide white and yellow pages services that supply agent and service naming.

3.2. Enabling Agent Factory with the ATM. In order to evaluate the usefulness of the agent travel metaphor, we identified and extracted a subset of this architecture to be initially implemented. This subset consists of the operations (a), (g), (h) and (i) defined in Figure 2.1. These operations give rise to the creation of three middle agents and three key data structures.

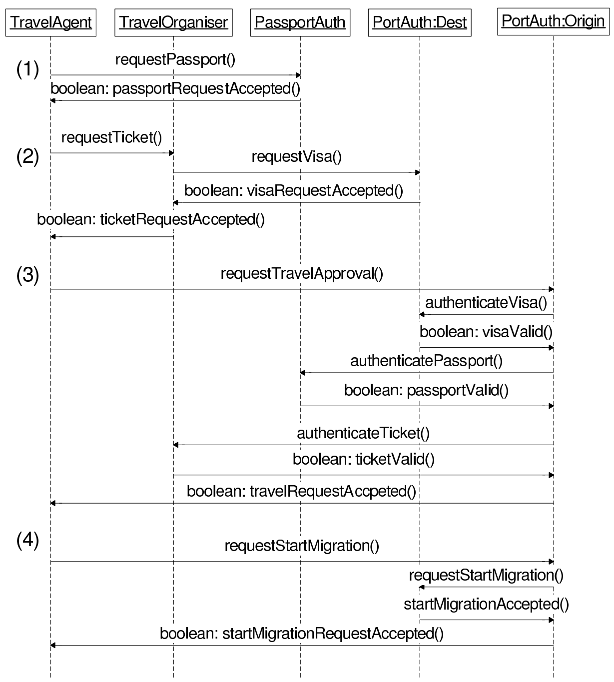

Fig. 3.1. *UML interaction diagram show the 5 implemented agents from the agent travel metaphor and the sequence of messages that occur when an agent migrates using the passport, visa and ticket system of authentication.*

The middle agents, TravelOrganiser, PassportAuthority and PortAuthority are responsible for issuing the three key data structures, Tickets, Passports and Visas, Figure 2.3 and Figure 2.2. These agents also have the responsibility for issuing appropriate travel documents to traveling agents upon request and proper authorization as in Figure 3.1 sections (1) and (2). The TravelOrganiser, PassportAuthority and PortAuthority also must provide a verification service allowing other middle agents that are inspecting travel documents to request that the creator of travel certificates verify that said certificates are valid, Figure 3.1 section (3). In conjunction with providing a verification service for Visa documents, PortAuthority agents are responsible for initiating authentication of incoming and outgoing agents' travel documents, Figure 3.1 section (4).

3.3. Modular Deployment of the Agent Travel Metaphor. The use of middle agents to imbue an agent platform with the ATM takes advantage of the flexibility inherent in intelligent agents, giving rise to the modular nature of the metaphor.

The middle agents, for example the PortAuthority agent, can use locally written software to perform their tasks. This means that the *exact* protocols that the agents use to enforce their security policies can be defined by a local developer or administrator.

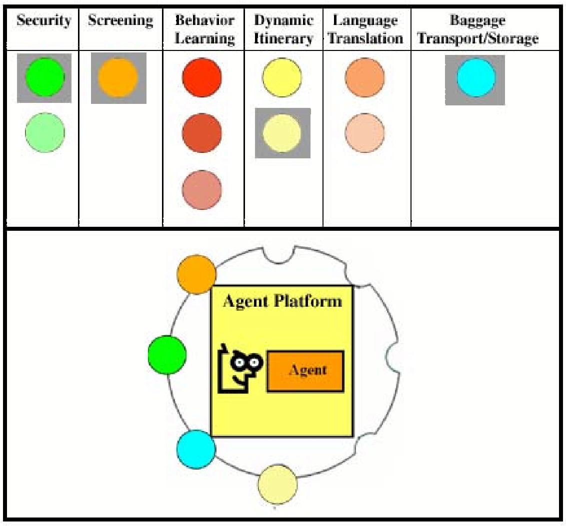

Fig. 3.2. *An agent platform configured with security, screening, dynamic itinerary, and baggage trasport/storage services.*

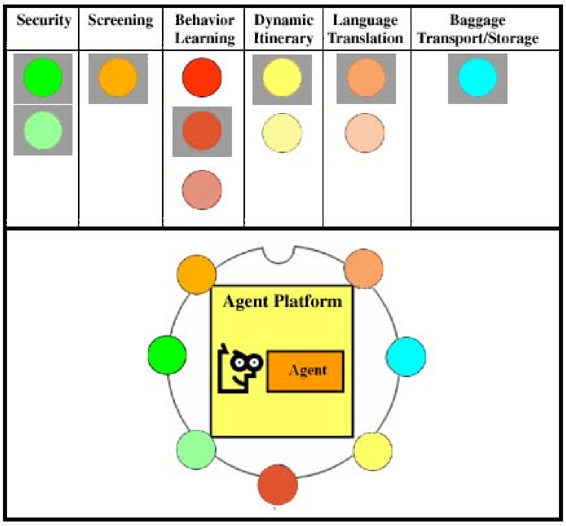

Fig. 3.3. *An agent platform re-configured with several extra services, behavior learning and language translation.*

Platform administrators can use the modular structure of the ATM to setup policies for services such as security, language translation etc, Figure 3.2. Administrators can also dynamically modify the services that exist on a particular platform as well as edit and augment existing services on a platform, Figure 3.3.

Consider the following scenario: An administrator controls a particular set of five agent platforms. The administrator knows that agent language and behavior are identical across all of the platforms and that only one of the platforms has access to critical assets that need to be protected.

The administrator instructs the critical platform's PortAuthority agent to demand that incoming migrating agents present Ticket, Passport and Visa documents. Furthermore it must scrutinize presented travel documents aggressively, verifying them with their issuers along with performing a query to federated PortAuthority agents to ensure that the migrating agent or its origin has not been blacklisted or quarantined for misbehavior.

An agent, PassportTraveller, decides that it wishes to migrate to the critical platform in order to access the secure data there. The agent is informed of the requirements set by the administrator, a passport, visa and ticket, by the TravelOrganiser agent. The agent then either contacts the appropriate agents in order to satisfy these requirements if it has knowledge of the behavior that is necessary to do so or, if the agent does not currently have the necessary knowledge to satisfy the migration requirements the agent can contact a behavior teacher to learn how to satisfy the requirements so that it may migrate.

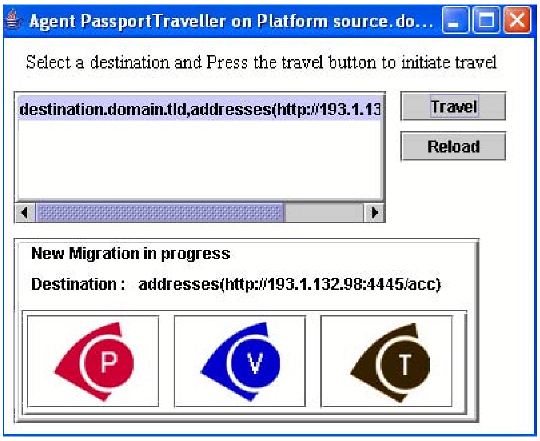

Fig. 3.4. *A screen shot showing an agent, PassportTraveller, migrating to destination* `http://193.1.132.98:4445`

Figure 3.4 shows an agent, PassportTraveller, that is migrating to the critical platform. The agent, as required, must satisfy three requirements, presenting a passport, visa and ticket in order to successful migrate. Figure 3.4 shows the agent has successfully obtained the first two travel documents as the red passport and blue visa icons are no longer grey, and is waiting upon the ticket document so that it can proceed with its migration.

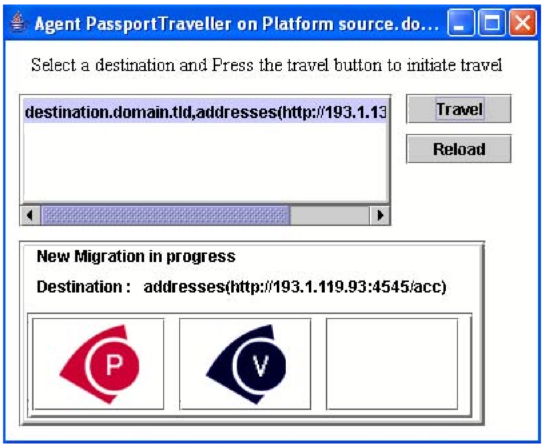

Fig. 3.5. *A screen shot showing an agent, PassportTraveller, migrating to destination* `http://193.1.119.93:4545`

The administrator can set much looser security policies on the other platforms, as security threats are not as potentially catastrophic and damaging to these platforms. The administrator instructs these PortAuthorities to require a Passport and Ticket from incoming migrating agents and to assume agent benevolence, i. e. to accept all presented documents as valid without verifying the documents with their issuer.

Figure 3.5 shows the agent PassportTraveller migrating to another platform with fewer security policies in place. Here the agent is only required to satisfy two requirements, presenting a passport, visa in order to successfully migrate. A ticket is unnecessary as the cost of the migration and the frequency and scheduling of migration between these agent platforms is of minimal importance. This agent has previously satisfied the passport requirements and is awaiting delivery of a visa document before it can proceed with its migration.

The above scenario describes the manner in which agent platforms can be configured in different manners using the ATM and the concepts of *National*, *International* travel and *Metropolitan* travel outlined in the Agent Travel Metaphor section above.

4. Evaluation and Results. In order to evaluate the consequences of enabling Agent Factory with the ATM framework we must consider several issues.

Security: The ATM framework puts into place a configurable set of security measures that allow administrators to set security policies in the manner that they see fit. Figure 4.1 and Figure 4.2 show a PassportAuthority agent and a PortAuthority agent respectively, these agents have been configured to provide a high level of security. In this example the PortAuthority agent is requiring that the traveling agent presents three travel documents, a passport, a visa and a ticket. Along with requiring traveling agents to present these documents the PortAuthority agent also verifies all of the documents' authenticity. These agents can easily be reconfigured to provide a slackened security policy for example if the PortAuthority only required traveling agents to present a single travel document, a passport, and if the PortAuthority agent did not verify the documents authenticity.

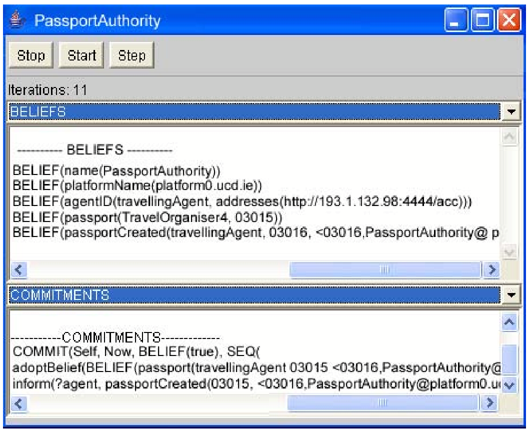

Fig. 4.1. *A PortAuthority agent receiving a migration request from a migrating agent, TravellingAgent, the PortAuthority agent handles this request by firstly verifying the documents legitimacy with the issuers.*

Dynamic itinerary: The TravelOrganiser agent provides mobile agents with a method to choose their migration destination without any previous knowledge the location of this platform. This allows agents to create a random migration pattern incorporating new additions to the agent platform network.

Scheduled migration: When an agent purchases a Ticket, the destination agent platform is informed that an agent wishes to migrate by the Ticket issuer. The origin and destination PortAuthority agents can prepare for the migration event and utilize the time beforehand to modifying the migration schedules on the network.

Migration time: The actual time taken to electronically migrate an agent increases from 5% to 7%, as a result of the agent keeping a record of its travel documents. Total time taken, from the initial decision to initiate a migration until the resumption of operation at the destination is substantially increased by 50% to potentially greater than 300%. The percentage increase can be apportioned to the potential for a large increase in the number of agents that are involved in any migration event. In the examples described above and seen in Figure 3.1, five agents are involved in the migration process, the traveling agent, the PassportAuthority agent, the TravelOrgainiser agent and the two PortAuthority agents, (based at the origin and destination). The number of messages that are passed between these five agents increases from 6 without any of the security measures from the ATM to 19 in the outlined example for the agent to acquire a passport, visa and ticket and to present these documents to the port authorities, and for the port authorities to verify the documents with

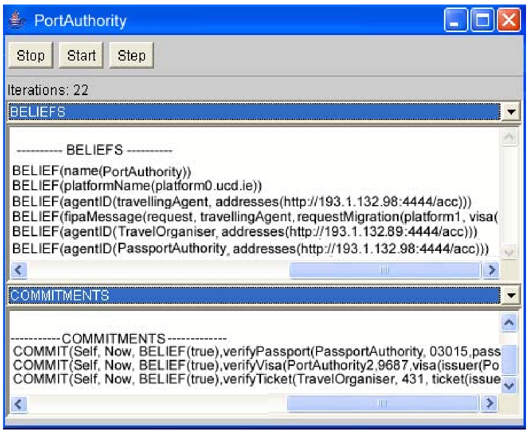

Fig. 4.2. *A PortAuthority agent receiving a migration request from a migrating agent, TravellingAgent, the PortAuthority agent handles this request by firstly verifying the documents legitimacy with the issuers.*

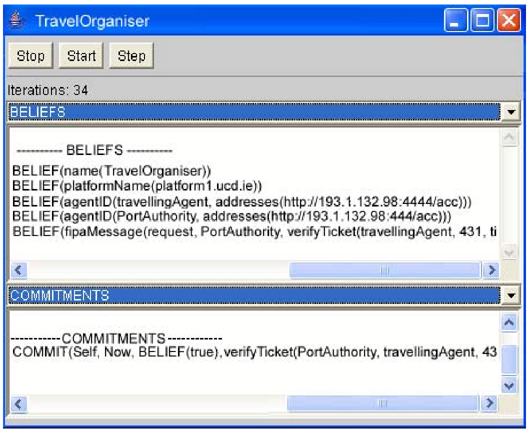

Fig. 4.3. *A TravelOrganiser agent receiving a validation request on a Ticket that was issued by this TravelOrganiser. The agent compares the requested ticket against its records.*

the document creators, Figure 3.1. Each extra security measure that is introduced in this manner, for example a digital certificate, introduces an extra time delay due to the request/present/verify processes.

It is the opinion of the authors that the benefits obtained from imbuing an Agent Factory agent platform with the ATM outweigh the drop in performance and speed. The additional services, such as security, heterogeneity over agent language, behavior, come at a cost. The total time taken for migration to occur and total size of an agent when it is migrated is increased. The modular nature of the ATM however allows for flexibility, for example if speed of migration is a priority, then migration security polices can be set to the lowest levels, increasing performance. If agent platform security is the priority then the resulting increase in the time taken caused by stricter security is an acceptable compromise.

5. Conclusion. This paper has introduced a comprehensive agent migration protocol which it delivers through a set of collaborative intelligent agents. The metaphor has been realized and incorporated within Agent Factory. It represents a consolidation and integration of some previous research that has adopted in part the travel metaphor.

The ATM has been realized in such a way as to support the addition of further modular components and the adoption of configurable local polices, for example baggage allowance, visa issue, security clearance and interoperability between a variety of regimes. This allows agent platform administrators a greater level of management and dynamic control over services provided by the agent platform.

As with the ongoing need for constant vigilance by administrators of computer networks to ever changing threats such as worms, viruses and other malicious attacks, the security threats posed by mobile agents are ever changing. As agents adapt and change so must the agent platforms security countermeasures, the modular service deployment proposed a demonstrated by this paper presents agent platform administrators with the tools necessary to combat and adapt to threats from mobile agents.

Mobile agents also greatly benefit from the use of the ATM. With the introduction of a stable and trusted source from which the agents can learn new behavior, mobile agents can now adapt themselves to perform tasks that would not normally be part of their ordinary operation, as seen when the agent PassportTraveller adapted its behavior in order to fulfil the migration requirements given to it by the PortAuthority agent at its destination, namely presenting a passport, visa and ticket object. Also with the provision of security measures on agent platforms it is possible for mobile agent to access information about agent platforms to verify that a particular platform does not have a history of attacks agents mobile agents, thus allowing agents to protect themselves against malicious hosts.

Acknowledgement. The authors would like to gratefully acknowledge the kind support of Science Foundation Ireland under Grant No. 03/IN.3/I361.

REFERENCES

[1] P. Bellavista, A. Corradi, and C. Stefanelli, *Mobile agent middleware for mobile computing*, j-COMPUTER, 34 (2001), pp. 73–81.
[2] R. H. Bordini and J. A. Campbell, *Anthropologically-based migration of agents: a new approach to interoperability*, 1995.
[3] J. Cao, X. Wang, and S. K. Das, *A framework of using cooperating mobile agents to achieve load sharing in distributed web server groups*, Future Gener. Comput. Syst., 20 (2004), pp. 591–603.
[4] D. Chess, B. Grosof, C. Harrison, D. Levine, C. Parris, and G. Tsudik, *Itinerant agents for mobile computing*, in Readings in Agents, M. N. Huhns and M. P. Singh, eds., Morgan Kaufmann, San Francisco, CA, USA, 1997, pp. 267–282.
[5] D. Chess, C. Harrison, and A. Kershenbaum, *Mobile Agents: Are They a Good Idea?*, Tech. Rep. RC 19887 (December 21, 1994 – Declassified March 16, 1995), IBM, Yorktown Heights, New York, 1994.
[6] R. W. Collier, G. M. P. O'Hare, T. D. Lowen, and C. Rooney, *Beyond prototyping in the factory of agents.*, in Multi-Agent Systems and Applications III, 3rd International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2003, Prague, Czech Republic, June 16-18, 2003, Proceedings, vol. 2691 of Lecture Notes in Computer Science, Springer, 2003, pp. 383–393.
[7] K. Decker, K. Sycara, and M. Williamson, *Middle-agents for the internet*, in Proceedings of the 15th International Joint Conference on Artificial Intelligence, Nagoya, Japan, 1997.
[8] F. F. I. P. A. FIPA, *Fipa abstract architecture specification*, June 2002.
[9] S.-U. Guan, T. Wang, and S.-H. Ong, *Migration control for mobile agents based on passport and visa*, Future Gener. Comput. Syst., 19 (2003), pp. 173–186.
[10] W. Jansen, *Countermeasures for mobile agent security*, 2000.
[11] N. R. Jennings, *Agent-oriented software engineering*, in Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99), F. J. Garijo and M. Boman, eds., vol. 1647, Springer-Verlag: Heidelberg, Germany, 30/6–2/7 1999, pp. 1–7.
[12] N. M. Karnik and A. R. Tripathi, *Design issues in mobile-agent programming systems*, IEEE Concurrency, 6 (1998), pp. 52–61.
[13] S. Poslad and P. Charlton, *Standardizing agent interoperability: the fipa approach*, 2001.
[14] K. Rothermel, F. Hohl, and N. Radouniklis, *Mobile agent systems: What is missing?*, in International Working Conference on Distributed Applications and Interoperable Systems DAISY97, Chapman & Hall, 1997, pp. 111–124.
[15] I. Satoh, *Application-specific routing for mobile agents*, in Proceedings of International Conference on Software Engineering, Artificial Intelligence, Networking & Parallel Distributed Computing, ACIS, August 2001.
[16] J. Tozicka, *Airports for agents: An open mas infrastructure for mobile agents.*, in Multi-Agent Systems and Applications III, 3rd International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2003, Prague, Czech Republic, June 16-18, 2003, Proceedings, vol. 2691 of Lecture Notes in Computer Science, Springer, 2003, pp. 373–382.
[17] J. E. White, *Mobile agents.*, in Software Agents, J. Bradshaw, ed., AAAI/MIT Press, Menlo Park, CA, 1997, pp. 437–472.
[18] M. Wooldridge and N. R. Jennings, *Intelligent agents: Theory and practice*, The Knowledge Engineering Review, 10 (1995), pp. 115–152.

Edited by: Shahram Rahimi, Raheel Ahmad
Received: October 28, 2005
Accepted: March 19, 2006