

# Agents Go Traveling

Donal O’Kane, David Marsh, Song Shen, Richard Tynan and G.M.P. O’Hare\*

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

## Abstract

This paper is concerned with infrastructural support for nomadic agents. Agent migration offers much potential however issues relating to the security and integrity of their temporary resting nodes 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. We describe the metaphor together with its incorporation within the Agent Factory multi-agent system.

## 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, (Wooldridge & Jennings 1995). (Wooldridge & Jennings 1995). A mobile agent refers to the capacity of an agent to electronically navigate a network in which it exists, (White 1997).

Many arguments have been proffered as to the benefits and disadvantages of mobile agents, (Chess, Harrison, & Kershenbaum 1994), and on the use of agent-oriented programming as a design paradigm, (Jennings 1999).

Numerous pitfalls have been identified, such as potential security issues involved in agent migration, (Jansen 2000), (Chess, Harrison, & Kershenbaum 1994), (Rothermel, Hohl, & Radouniklis 1997), (Chess *et al.* 1997), interoperability and language translation, (Poslad & Charlton 2001), (FIPA 2002), (Bordini & Campbell 1995) and dynamic creation

and management of an agent’s itinerary, (Sato 2001).

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, (Chess *et al.* 1997) 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. (Tozicka 2003) 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.

## Mobile Agents - The Way Forward

### Related Research

The ability of an agent to migrate is defined as electronic movements about a network, (White 1997). Many multi-agent systems support agent mobility and provide services to enable their agents to migrate more efficiently.

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

Copyright © 2005, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

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 measures:** Systems have been developed that utilize passport and visa documents and digital certificates, (Guan, Wang, & Ong 2003), (Chess *et al.* 1997), to enable identification and the origin of migrating agents.

**Agent language translation:** There are services and agents that facilitate interoperation between heterogenous multi-agent systems, using meta languages, (Bordini & Campbell 1995), direct translation, (Chess *et al.* 1997), or through the use of ontologies, (FIPA 2002).

**Middle agents:** These agents provide a variety of static services to mobile agents such as naming and access to shared resources, (Decker, Sycara, & Williamson 1997), (Chess *et al.* 1997).

**Load balancing mechanisms:** Tools and services that provide load balancing for their network have been developed, (Cao, Wang, & Das 2004).

**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, (Tozicka 2003).

**Agent code protection:** Protection for the core of an agents mental state against viral code and unauthorized access has been developed, (Karnik & Tripathi 1998). Agent platform administrators can assume benevolence of known agents once their critical code is protected.

## 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.

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. Figure 1 shows a general set of process that occur when undertaking travel.

Three classes of 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.

```

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

```

Figure 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 ATM is designed to facilitate configuration and policing of mobile agents policies and services in a heterogenous environment.

The algorithm described in Figure 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 affiliations with other platforms and multi-agent networks.

## 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.

**Passport** an official certificate proving an agents identity, issued by a trusted source.

**Ticket** a certificate that the ticket holder has paid for and is entitled to a specific service.

**Ticket Issuer** a trusted agent or service that can be contracted to provide travel tickets.

**Visa** a certificate proving that an agent is permitted to migrate to a destination, issued by the destination.

**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.

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

**Port Authorities** a trusted agent or service charged with the task of operating the port. 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.

**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.

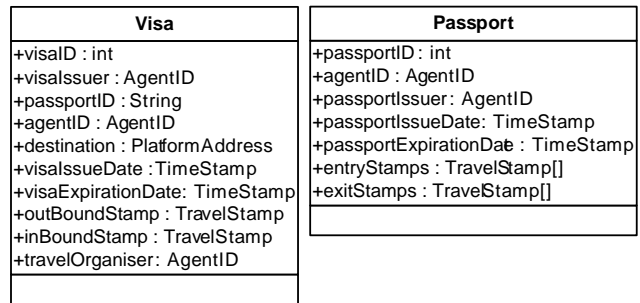


Figure 2: The Visa and Passport Data Structures.

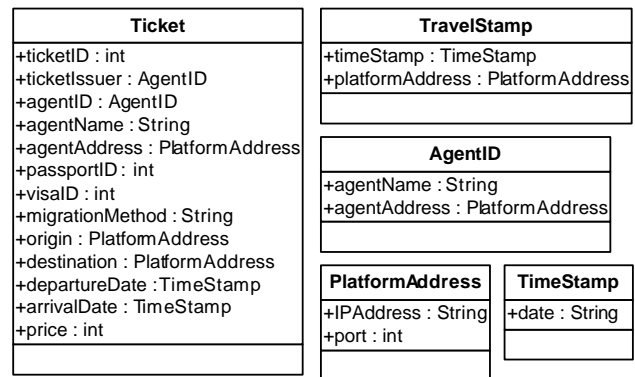


Figure 3: The Ticket, TravelStamp, AgentID, PlatformAddress and TimeStamp Data Structures.

## Agent Factory and the Agent Travel Metaphor

### Agent Factory Mobility Support

Agent Factory, (Collier *et al.* 2003), 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.

### 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 1. These operations give rise to the creation of three middle agents and three key data structures.

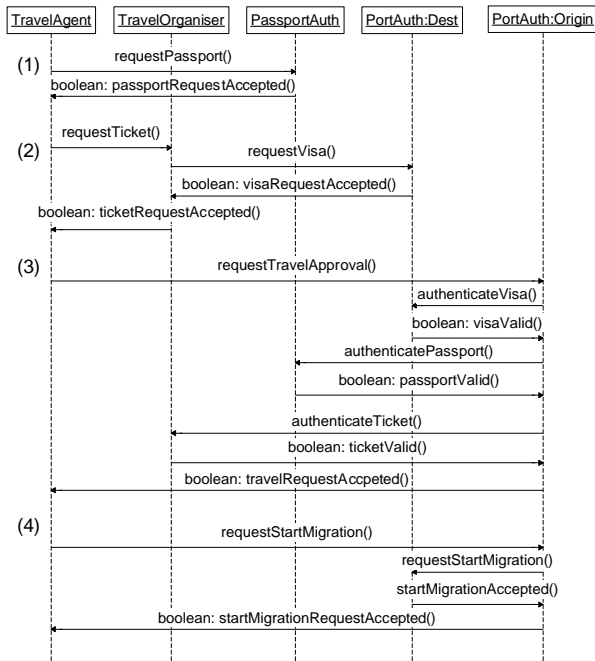


Figure 4: UML interaction diagram show the 5 implemented agents from the agent travel metaphor and the sequence of messages that occur when an agent migrates using the passport, visa and ticket system of authentication.

The middle agents, TravelOrganiser, PassportAuthority and PortAuthority are responsible for issuing the data structures, Tickets, Passports and Visas, Figure 3 and Figure 2 respectively. Along with this responsibility for issuing travel documents to traveling agents, Figure 4 sections (1) and (2), the middle agents also must provide a verification service should another inspecting agent wish to make certain that a presented travel certificate is valid, Figure 4 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 4 section (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

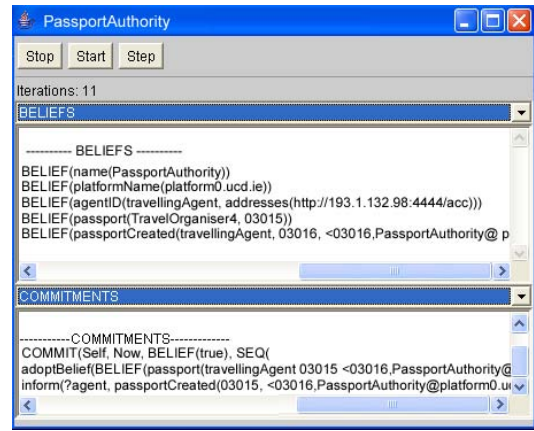


Figure 5: A PortAuthority agent receiving a migration request from a migrating agent, TravellingAgent, the PortAuthority agent handles this request by firstly verifying the documents legitimacy with the issuers.

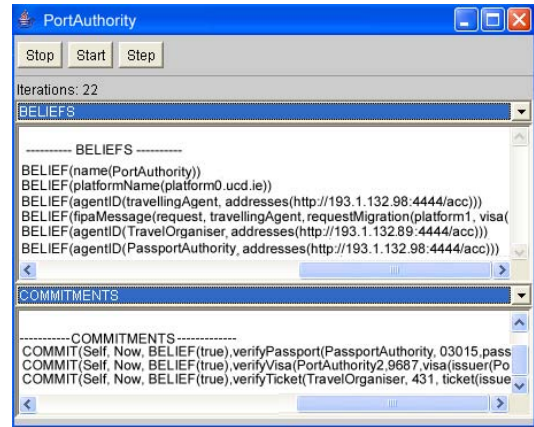


Figure 6: A PortAuthority agent receiving a migration request from a migrating agent, TravellingAgent, the PortAuthority agent handles this request by firstly verifying the documents legitimacy with the issuers.

their security policies can be defined by a local developer or administrator.

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.

The administrator can set much looser security policies on

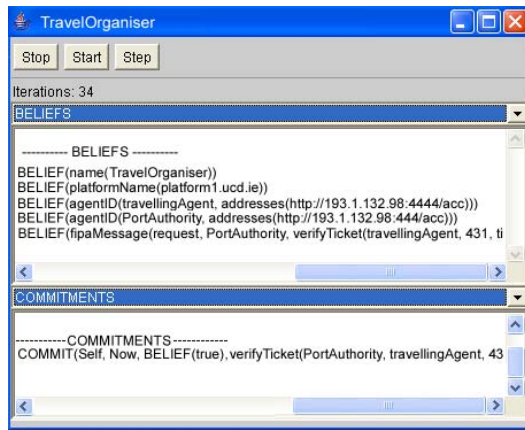


Figure 7: A TravelOrganiser agent receiving a validation request on a Ticket that was issued by this TravelOrganiser. The agent compares the requested ticket against its records.

the 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.

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

## 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 5 and Figure 6 show a PassportAuthority agent and a PortAuthority agent respectively, these agents have been configured to provide a high level of security and can easily be altered to provide a slackened security policy.

**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 deciding to initiate a migration to completion, is substantially increased by 50% to potentially greater than 300%. The percentage increase can be apportioned to the possibility of a large increase in the number of agents that are involved in any migration event. In the examples described above and seen in Figure 4, four agents are involved in the migration process and the number of messages passed increases from 6 without ATM to 19 in the outlined example.

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.

## 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.

## References

- Bordini, R. H., and Campbell, J. A. 1995. Anthropologically-based migration of agents: a new approach to interoperability.
- Cao, J.; Wang, X.; and Das, S. K. 2004. A framework of using cooperating mobile agents to achieve load sharing in distributed web server groups. *Future Gener. Comput. Syst.* 20(4):591–603.
- Chess, D.; Grosz, B.; Harrison, C.; Levine, D.; Parris, C.;

and Tsudik, G. 1997. Itinerant agents for mobile computing. In Huhns, M. N., and Singh, M. P., eds., *Readings in Agents*. San Francisco, CA, USA: Morgan Kaufmann. 267–282.

Chess, D.; Harrison, C.; and Kershenbaum, A. 1994. Mobile Agents: Are They a Good Idea? Technical Report RC 19887 (December 21, 1994 - Declassified March 16, 1995), IBM, Yorktown Heights, New York.

Collier, R. W.; O'Hare, G. M. P.; Lowen, T. D.; and Rooney, C. 2003. 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*, volume 2691 of *Lecture Notes in Computer Science*, 383–393. Springer.

Decker, K.; Sycara, K.; and Williamson, M. 1997. Middle-agents for the internet. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*.

FIPA, F. F. I. P. A. 2002. Fipa abstract architecture specification.

Guan, S.-U.; Wang, T.; and Ong, S.-H. 2003. Migration control for mobile agents based on passport and visa. *Future Gener. Comput. Syst.* 19(2):173–186.

Jansen, W. 2000. Countermeasures for mobile agent security.

Jennings, N. R. 1999. Agent-oriented software engineering. In Garijo, F. J., and Boman, M., eds., *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99)*, volume 1647, 1–7. Springer-Verlag: Heidelberg, Germany.

Karnik, N. M., and Tripathi, A. R. 1998. Design issues in mobile-agent programming systems. *IEEE Concurrency* 6(3):52–61.

Poslad, S., and Charlton, P. 2001. Standardizing agent interoperability: the fipa approach.

Rothermel, K.; Hohl, F.; and Radouniklis, N. 1997. Mobile agent systems: What is missing? In *International Working Conference on Distributed Applications and Interoperable Systems DAISY97*, 111–124. Chapman & Hall.

Satoh, I. 2001. Application-specific routing for mobile agents. In *Proceedings of International Conference on Software Engineering, Artificial Intelligence, Networking & Parallel Distributed Computing*. ACIS.

Tozicka, J. 2003. 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*, volume 2691 of *Lecture Notes in Computer Science*, 373–382. Springer.

White, J. E. 1997. Mobile agents. In Bradshaw, J., ed., *Software Agents*, 437–472. AAAI/MIT Press, Menlo Park, CA.

Wooldridge, M., and Jennings, N. R. 1995. Intelligent

agents: Theory and practice. *The Knowledge Engineering Review* 10(2):115–152.