

The Agentcities Network Architecture

Steven Willmott

EPFL
steven.willmott@epfl.ch

Jonathan Dale

Fujitsu
jonathan.dale@fla.fujitsu.com

Jerome Picault

Motorola
jerome.picault@motorola.com

Matteo Somacher

University of Parma
somacher@ce.unipr.it

Stefan Poslad

Queen Mary College of London
stefan.poslad@elec.qmul.ac.uk

Juan Jim Tan

Queen Mary College of London
Juanjim.tan@elec.qmul.ac.uk

Ion Constantinescu

EPFL
ion.constantinescu@epfl.ch

David Bonnefoy

Motorola
david.bonnefoy@motorola.com

ABSTRACT

Agentcities is a worldwide initiative designed to help realize the commercial and research potential of agent-based applications by constructing an open distributed network of platforms hosting diverse agents and services. The ultimate aim of the Agentcities initiative is to enable the dynamic, intelligent and autonomous composition of services to achieve user and business goals, thereby creating compound services to address changing needs. In this paper, we present the progress and current status of the Agentcities Network, six months after the launch of the project. The architecture of the Network, consisting of agents, services and platforms, is described. Finally, the plans and challenges for enhancing the Agentcities Network in the next phase of development are also discussed.

1. INTRODUCTION

The Agentcities Network [1] (hereafter referred to just as the Network) represents the first attempt to build an open, global and standards-based agent environment for research and future commerce on the Internet. The Network effectively came into being in October, 2001 with the simultaneous launch of agent platform software environments in 14 world cities. The Network has been growing and evolving ever since and now comprises 41 agent platforms in 21 countries, as well as a set of basic Network-support services.

The objective of the Network is to bring together technologies from both agent and AI research¹ with industry-led technology initiatives² to create a global, open, dynamic environment that enables:

- Rich, flexible communication between software entities deployed within it, and,

¹ Such as agent communication languages [13][14][6], conversation protocols [15][16], ontologies [19], coordination [17], negotiation [18] and open systems theory.

² Such as Web Services [7], JXTA [8], XML [9] and RDF [10].

- Software entities to trade automatically with each other in a dynamic and flexible way without the constant intervention of humans.

The initial basis for development and deployment of this environment is the FIPA agent standard [2] that provides specifications for interoperability between agent-based systems which represent distinct communicating and trading entities.

This paper provides an overview of the current Network, in terms of architecture and technologies, as well as a discussion of future challenges. We focus only on the Network infrastructure and architecture and not on the business and application interactions which are intended to take place over the Network.

2. AGENTCITIES NETWORK

The first generation Network consists of a set of independent agent platform environments deployed by many individual organizations tied together by adherence to a set of common communication mechanisms and a set of Network-support services tying them together into a coherent network.

2.1 Technology Basis

The current Network is based on a selection of standards developed by FIPA that are built on other, well-known standards in the following areas:

- **Message exchange and routing:** including message transport, message envelopes message transport protocols.
- **Agent communication:** including message structure, s-expression syntax, communicative acts and interaction protocols.
- **Agent management:** including agent naming services, service discovery and management services.

The full list of standards used in the Network architecture can be found in the current Agentcities Network Architecture recommendation [3].

2.2 Network Architecture

The Network consists of three main elements:

- **Agent platforms:** Software environments that support agents running in the Network with access to Network-support services³. Each instance of an agent platform represents a single node in the Network.
- **Agents:** A computational process that implements the autonomous, communicating functionality of an application. In particular, it may provide services to others and access services provided by others.
- **Services:** A series of one or more actions carried out by a service provider (an agent or something else) on behalf of a service consumer (an agent or something else).

Individual agents, services and agent platforms are able to interact with one another due to their use of standardized transportation protocols, such as HTTP and IIOP, to ontology frameworks, such as DAML+OIL [11], and agent communication languages, such as FIPA-ACL [4]. The Network is bound together through a set of Network-support services that enable entities in the Network to find each other, which are described in the next sections. Each of these directories has both human and agent interfaces.

2.2.1 Agent Platforms

Agent platforms that are deployed in the Network are runtime instances of agent software toolkits that support the standards mentioned in Section 2.1. There are currently 12 different implementations (both commercial and non-commercial), and 10 of these are provided under some form of an Open Source License.⁴ Agent platforms in the Network are registered in a simple centralized Agent Platform Directory (APD) that regularly contacts each using a simple “ping/alive” protocol. Agent platforms which respond correctly to the “ping” request are listed alive in the APD. Consequently, the information on which agent platforms are currently alive forms the basis of the agent and service directories.

2.2.2 Agents

Agents in the Network reside on agent platforms and are able to communicate to provide services to one another using the message exchange and communication features provided by each agent platforms. Agents are registered in the Network if they appear in the global Agent Directory (AD) which is dynamically compiled by a global directory agent who regularly queries the list of local ADs on each agent platform which is tracked as alive in the APD.

2.2.3 Services

Services are registered in the Network if they appear in the global Service Directory (SD), which is compiled in a similar manner to the AD.

2.3 Overall Architecture

The overall architecture of the Network is currently very simple and comprises:

- A single domain in which all agent platforms are visible to each other and can communicate directly, that is, there are no message gateways and no mechanisms for explicitly handling firewalls.
- A centralized APD (relying on a single root node) that is overlaid by the global AD and SD, each of which uses a star topology (a single root node dependent on the APD and local ADs and SDs on each Network node).
- A centralized registration mechanism for agent platforms which is dependent on the human managed data provided through the Web site.

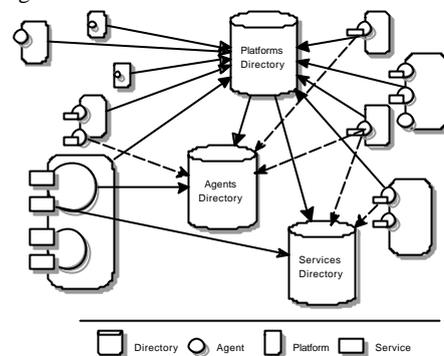


Figure 1: The Agentcities Network is composed of a large number of Agent platforms deployed in a single domain and spanned by three types of directories.

Many improvements are necessary to ensure the sustainability of the Network in the long term, which will be developed as the Network grows. The Network has been running permanently since its launch (166 days) and amongst the original 14 agent platforms, the average platform uptime is 122 days (76%) with some agent platforms logging an impressive 158 days (95%) in uptime. Many of the newly launched platforms have shown near 100% uptime since their launch.

3. CHANGES AND CHALLENGES

Whilst the current Network deployment represents a significant achievement, there are clearly many challenges to come in creating a viable long-term infrastructure. Briefly, these include:

- **Refinement of definitions and models:** For Agentcities to become a ubiquitous network of business interest, the network must allow interaction with existing technologies, but at the same time Agentcities is to make advantage of agent technologies. This will be achieved by adapting the top level definitions and corresponding structures (agents, services and agent platforms) used in the Network to:
 1. Map these onto other network environments, such as Web Services, P2P and ebXML systems, to enable

³ This definition is taken from [4].

⁴ See <http://www.agentcities.org/Resources/Software/> for a list of these platforms

entities from other environments to interact directly with entities in the Agenticities Network.

2. Apply the requirements that are generated by deployed applications and agent theories to the Network.

Finding definitions that are sufficiently generic to satisfy a large number of applications and yet prove useful in practice is likely a difficult challenge.

- **Scalability and robustness:** The current Network-support services are very simple and rely on centralized or star topologies with a single point of failure and no means for distribution of authority. The security and robustness of both Network-support services and individual agent platforms are also rudimentary. Each of these areas poses major challenges to be addressed as the Network grows.
- **Authority and management:** As the Network expands and some agents and services become critical components, there will be challenges in establishing notions of identity, authority, reputation and trust. Presupposing that agents in the future will be able to effectively trade with each other, it is not clear what infrastructural support is required for identity services, for example, how to ground in human legal frameworks and existing Internet infrastructures?
- **Technology heterogeneity:** The technology basis for the Network is based upon diverse technologies and is likely to become more so since there are already efforts to incorporate Web Services interfaces [12]. Such extensions are likely to be a reality of the Network if it is to become generally useful but they will be challenging since:
 1. Different parts of the Network are likely to support only a subset of all technologies used, leading to a patchwork of solutions and potential requirements for gateways or other conversion methods.
 2. Different technical solutions are likely to have different properties and there cases will exist where no complete mapping (or appropriate abstraction) can be found for a particular set of technical solutions. An example is the varying expressive power of languages used to describe services, such as, DAMLS, WSDL or FIPA DF entries.
- **Testing, monitoring and verification:** If the Network infrastructure is to be relied upon, rigorous testing methods and benchmarks must be established and used in regular monitoring of performance (on for example the availability, speed, accuracy of Network-support services).

In many ways, these problems mirror the development of many other network environments, such as P2P networks, GRID computing and, ultimately, the Internet itself. The Network has yet to be seriously tested with substantial applications, but it is hoped that the applications now planned for deployment will help drive the evolution of the Network.

4. AGENTCITIES NETWORK FUTURE

A basic requirement is the openness of the Network, not only in terms of deploying different multi-agent systems, but also in incorporating technologies that deal with dynamic and distributed environments. To help address this, a future Network architecture is being developed that will be based upon an abstract model that is general enough to allow the integration of such technologies.

4.1 Abstract Model Elements

We divide the whole abstract Network architecture elements into three levels. Each level is independent and can be represented by a concrete architecture that implements its elements. These levels define a vertical model where the upper levels are built on the top of the lower levels.

4.1.1 Core Elements

Core elements are necessary things that describe the key, base entities in the abstract Network. Each entity in the Agenticities world can be considered as instance of *actor* or *service*:

- *Actor*
 - **Definition:** An entity which does things in the world and which can act as a service provider, a service consumer or both.
 - **Additional:** It may be realized as an agent, an object or anything else.
 - **Data:** An actor must have at least one description and at least one identity.
- *Service*
 - **Definition:** An activity carried out by service provider on behalf of a service consumer, that is, a service instance.
 - **Additional:** It may be governed by a service level agreement or contract.
 - **Data:** A service must have at least one description.

4.1.2 Structural Elements

Structural elements are needed to manage the interactions among the core elements. A Network implementation can implement a subset of the following:

- *Contracts:* The relationships and interactions among the actors can be regulated by *contracts* (also known as service level agreements). When actors need explicit rules to govern their interactions, they mutually agree to a contract, which can refer to services, such as establishing a quality of service or specifying conditions for service completion.
- *Domains:* Core elements may be members of zero or more domains. A domain is represented as an extensional set and has the following characteristics:
 - Contains zero or more members.
 - Every member in the domain has an identifier.

- Specifies a classification for the contained members.
- Can be managed by one or more actors that provide the service of accessing the domain and act as authorities for the domain.
- Can have policies to manage members, for example, rules for accepting and rejecting membership requests.
- Domains may be members of other domains (which may lead to policy conflicts).
- *Interfaces*: All interaction mechanisms provided by an actor to allow it to communicate with other entities in the Network are defined through public interfaces.
- *Policies*: Actors may have policies that circumscribe their behavior in the Network and state their conditions of action and interaction with regard to particular states of the world. Policies can also be attached to actors that manage domains to provide membership conditions for that domain.
- *Goals*: Actors enter the Network with goals, whether explicitly or implicitly stated. To explicitly define the concept of a goal can help to promote the interactions between the core elements and goals can be defined as the motivating factor for service provision and consumption.

4.1.3 Functional Elements

Functional elements are actual instances of core and structural elements that are required to provide its functionality, such as actor instances, like brokers and mediators, and domain instances, like domain managers. This set of elements should define also some more detailed characteristics about directory and domain management.

4.2 Concrete Realization

The abstract elements described previously can be mapped into the current Agentcities Network in many ways. Considering the core elements, the reification⁵ of the abstract term *actor* is made by FIPA compliant agents, whereas the abstract term *service* is made by FIPA-Service agents. The Network provides domains of agents with the following properties:

- Unique agent names through FIPA agent identifiers.
- Rules for accepting, rejecting and removing agents. Currently, these rules are handled in an ad-hoc fashion by each platform.
- Service federation and propagation of data through DFs.

Service domains are built by the DF `service-description` and the property of DF federation, with multiple DFs working together to create a domain. However, the current implementation of the DF [5] has some drawbacks and will have to be extended:

- The DF is indexed by agent and not by service.

⁵ Reification is the term used to describe the process of moving from an abstract concept (design) to a concrete realization (implementation).

- The data model does not support structured service descriptions, such as descriptions in DAMLS.
- Most of the attributes are optional.
- DF federation does not propagate service data, but service queries which means that service data cannot be cached and searches can be slow, but the results are up-to-date.

5. CONCLUSION

Whilst clearly a work in progress, the Network already represents the largest agent environment ever deployed and is growing steadily. We believe that the Network provides:

- A worldwide deployment environment for testing agent-based applications and services.
- A vehicle for integrating agent technology with technologies such as Web Services, P2P networking, GRID computing and the Semantic Web.

Finally, the development of the Network has provided a significant test for a subset of the existing FIPA specifications and generated important feedback to FIPA itself. Whilst many challenges lie ahead, it is hoped that with involvement from such a large number of organizations, the Network architecture can be improved significantly as usage continues to grow.

5.1 Acknowledgements

The research described in this paper is partly supported by the EC project Agentcities.RTD (IST-2000-28385). The opinions expressed in this paper are those of the authors and are not necessarily those of the EU Agentcities.RTD partners. We would like to sincerely thank the many people who have participated in establishing the network.

6. REFERENCES

- [1] *The Agentcities Network*. <http://www.agentcities.net/>
- [2] *The Foundation of Intelligent Physical Agents (FIPA)* <http://www.fipa.org/>
- [3] *Agentcities Network Architecture Recommendation*, ACTF-REC-00001, 2002. <http://www.agentcities.org/rec/00001/>
- [4] *FIPA Abstract Architecture*, FIPA00001, 2001. <http://www.fipa.org/specs/fipa00001/>
- [5] *FIPA Agent Management Specification*, FIPA00023, 2001. <http://www.fipa.org/specs/fipa00023/>
- [6] *FIPA ACL Message Structure Specification*, FIPA00061, 2001. <http://www.fipa.org/specs/fipa00061/>
- [7] *W3C Web Services Activity*. <http://www.w3.org/2002/ws/>
- [8] *The JXTA Project*. <http://www.jxta.org/>
- [9] *W3C Extensible Mark-up Language (XML) Activity*. <http://www.w3c.org/XML/>
- [10] *W3C Resource Description Framework (RDF) Activity*. <http://www.w3c.org/RDF/>

- [11] *DAML+OIL*. <http://www.daml.org/language/>
- [12] *Agentcities Task Force Web Services Working Group*, ACTF, 2002.
- [13] J. Searle. *Speech Acts*. Cambridge Univ. Press, 1969.
- [14] Finin, T. et al. *Specification of the KQML Agent Communication Language*. The DARPA Knowledge Sharing Initiative, External Interfaces Working Group. 1992
- [15] B. Burmeister, A. Haddadi, and K. Sundermeyer. *Generic configurable cooperation protocols for multi-agent systems*. From Reaction to Cognition –5th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'93).
- [16] *FIPA Interaction Protocol Library Specification*, FIPA00025, 2001. <http://www.fipa.org/specs/fipa00025/>
- [17] M. Wooldridge and N. R. Jennings. *Towards a Theory of Cooperative Problem Solving*. In Proceedings Workshop Modelling Autonomous Agents in a Multi Agent World (MAAMAW'94). 1994.
- [18] B. Laasri, H. Laasri, and V. Lesser. *A generic model for intelligent negotiating agents*. International Journal on Intelligent Cooperative Information Systems, 1992
- [19] D. Fensel. *Ontologies: A silver bullet for Knowledge Management and E-Commerce*. Springer 2001