Agents and Their Cities

Patricia Charlton[1] and **David Bonnefoy**[2] and **Nicolas Lhuillier**[3]

Abstract. The interactive Web seems a natural place for the use of agent technology, to bring access to e-services with solutions to dynamic, open and changing environments. However, agent technology is still quite low key in this expanding area of Internet access and e-business space. Part of the reason for this has been the lack of a large-scale test-bed. Some fundamental changes in the development of agent systems have helped to move the research into the area of directly addressing high-level interoperability: emergence of a standard, availability of many agent systems for interoperability testing, better understanding of agent infrastructure, etc. In this paper we look at the developments of agent technology and the challenges, which will enable agent technology to play a key role in the Internet space. This paper describes the design of high-level interoperable agent services through the concept of Agentcities, its motivations, drivers and its implementation. Finally, the key challenges to be addressed to make this initiative a success are evaluated.

1 MOTIVATIONS

The Internet enables to connect computers throughout the world and more specifically, the web has become a ubiquitous medium, which uses are expanding rapidly. But the web has been designed as a system for navigating documents through hyperlinks. This has several important consequences:

- It is meant to be accessed by humans, not pieces of software, making efficient search engines difficult to implement;
- The content was originally intended to only consist of pages and files;
- The web has no natural notion of service.

A lot of effort has been put in the recent years for overcoming these problems: some additional meta-data can be added to web pages, HTML code is now often dynamically constructed from databases, and web sites are indeed often seen as services. But this is a very primitive notion of service. These services are still meant to be accessed by humans only. This makes it very difficult for a service to use other ones. And even if it were possible, the result would be very static: services have no way to describe themselves, so human intervention is necessary to compose services into more complex ones. No automated service co-operation is possible.

This situation has many drawbacks. A well-known one is that it is more and more difficult to find the service that exactly fits one's need. Moreover would this right service be found, there is no coherent way of accessing and using it, which would make automation possible. In addition, the Internet tries now to move to handheld devices. One of the main issues there is that the current form of the web is not suited for devices, which are constrained in terms of display, processing power and connectivity.

The solution to these problems does not lie in compressing or reshaping the content of the Web, but rather in revolutionizing the way Internet services are accessed, bringing more intelligence and leading to an electronic lifestyle. The result of a service should seamlessly derive from users' wishes rather than from user' skills. The ultimate step is to break the limitations of existing Web services, to enable them to combine themselves in order to solve new problems that one single service cannot solve. This is actually starting to appear with initiatives such as Microsoft .NET [5] and Sun ONE [7]: "Sun Jumps on the Web Services Bandwagon With 'Sun One'", announcement, Giga Information group. But these systems are based on low-level software components, which lack the ability of truly autonomous behaviour. Intelligent agents represent another potential technology to face these challenges, since they enable to hide much of the complexity of accessing Web services, while bringing additional value by customizing and composing the services. Agents are by definition particularly adapted to the characteristics of the Internet environment, which is:

- Open: any party must be able to join, providing it conforms to the standard;
- Dynamic: parties must be able to join and leave at any time;
- Distributed: the management of the architecture is not centralized.

Taking these key characteristics of agents described above in this paper we define in the section 2 the Agentcities concept and its comparison with the classic Web approach. In section 3 we describe the current activities and developments of Agentcities implementations and deployment, reviewing the links with standards. Section four provides a summary of key conceptual challenges for the deployment of Agentcities framework for achieving high-level semantic aggregation of services. The conclusion looks at the broader potential impact of the Agentcities concept and its deployment.

2 THE AGENTCITIES CONCEPT

The Agentcities concept [8] aims at creating a new user experience on using the Internet, by defining a new approach in providing and accessing Web services. This intelligent and coherent way means users no longer need to cope with service search and usage, but seamlessly get the right result at the right time. The idea behind Agentcities is to create a worldwide network of agent-based services, where access to services is:

- Automated: users themselves are represented by personal agents, which interpret their desires and co-operate with agent services in

[1] Motorola Labs Paris, Espace Technologique Saint-Aubin, 91193 Gif sur Yvette, France, email: Patricia.Charlton@motorola.com
[2] Motorola Labs Paris, Espace Technologique Saint-Aubin, 91193 Gif sur Yvette, France, email: David.Bonnefoy@motorola.com
[3] Motorola Labs Paris, Espace Technologique Saint-Aubin, 91193 Gif sur Yvette, France, email: Nicolas.Lhuillier@motorola.com

order to best fulfill them. These agents are also able to take proactive decision they consider relevant to their user;

- Dynamic: users and services can join and leave at any time, and also the dependencies between them is not static so the failure of one party has no impact on the others; and
- Customized: service access is based on negotiation and cooperation. User agents have some knowledge about their user preferences and context, so that they are able to specialise the service in terms of functionality and output;
- Ubiquitous: access to the services is based on agent-to-agent interaction, so depends less on the characteristics of the users' devices.

In this scheme a Web service does not simply deliver information, but can be defined as any atomic entity that provides on demand an added-value output. Deployed services are expected to be of interest both to businesses (B2B approach) and to end-users (B2C approach).

The ultimate advantage of this architecture is to enable services to collaborate directly with each other. At some higher level, services can co-operate with other services, or make them co-operate, to achieve higher goals, thereby creating added-value compound services. This dynamic, intelligent and autonomous composition of services is essential to address singular or complex tasks that cannot be fulfilled by single and existing services.

Agent services are hosted on agent platforms, which form a network that is open, in the sense that heterogeneous platforms and services, coming from various implementations, can dynamically join. This openness obviously means that strong interoperability issues have to be solved to enable the vision. The interoperability problem among agent platforms has been studied in [15]. The Figure 1 (from [15]) shows how the interoperability problem can be analysed in different layers. Agentcities has to deal with all these layers to be successful. It is necessary to conform to a standard for the platforms to successfully inter-operate. FIPA [2] is one possibility, as using such a standard allows to solve the inter-operability problems for the bottom two layers. Having platforms inter-operate is only one part of the problem. Agentcities is an environment where services themselves can interact, so it is important to look at the Agentcities network not only as a network of platforms, but also as a network of co-operating services.

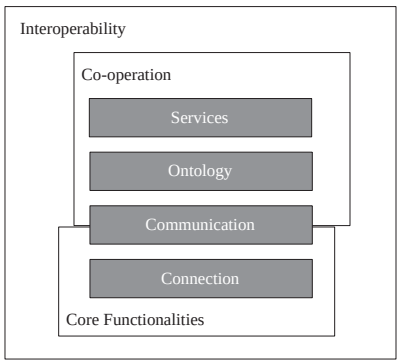

Figure 1. Interoperability Layers

The heart of Agentcities is the network of agent platforms. Services are hosted on agent platforms, each of them being linked to a city. What the paradigm of the city brings to the model is:

- Localisation: some services, like entertainment, are more relevant in a geographical context. This location-based information, which is currently missing from current Web is achieved by the concept of cities. Connecting to a city, users can automatically filter the services, which are next to them. However, this does not prevent them to access services on distant platforms. Some services, which are not linked to a geographical context, like banking, can benefit from other grouping factors like sharing common topic. Hence the concept of a city does not always require to be mapped to geographical entity; such cities can be defined as virtual.
- Scalability: Regrouping the services into cities brings some scalability and hierarchy in the network. Defining main-cities, which are responsible for representing and monitoring other cities referring to them, may be required for management and certification concerns so that openness does not become anarchy. There are core features discussed in this section which illustrate the advances the Agentcities concept brings to e-service access. In order to illustrate further these key features of the Agentcities Concept Table 1 summarises a comparison between the classic concept of the Web and that of the Agentcities.

Table 1. Analysis of the Web and Agentcities High-level Features

Feature	Web	Agentcities
Applications	Originally documents, tweaked to support more advanced e-commerce services.	Any atomic entity that provides on demand an added-value output.
Openness	New pages can be added, but nothing ensures they will be publicized and accessed from others.	Any service can join provided that it conforms to the standard.
Dynamics	Join and leave is dynamic, but since relations are static it may result in broken links. Content may be dynamically created.	Dynamic creation of teams, where leave of one party does not hurt others.
Scalability	Managed at the network level. Service search and access degrades when number increases.	Abstraction of agent concept is recursive and offers different levels: Platforms in cities, agents on platform, services into agents
Autonomy	Not real autonomy of services but rather independence.	Agents are autonomous entities that are not hurt by failure of others and can draw proactive decisions.
Localisation information	Non-existent	Provided through paradigm of cities
Co-operation/ composition	Basic (ex: portals) and always static (does not support change in format).	High-level co-operation and at language level. Autonomous, dynamic and intelligent composition of services

3 IMPLEMENTING THE CONCEPT

Today, most of the basic technology requirements are met to start implementing the concept: The FIPA agent standard has released a set of specifications in an experimental stage, which are now to be validated on practical examples. Platforms implementing these specifications are available, and an intensive interoperability trial known

as ”FIPA bake-off” [3] recently proved the viability of the specifications. More generally, there is currently a clear need for an open test bed, where interoperability, collaboration and composition issues between agent services can be pragmatically tackled. Many organizations around the world have already shown their interest in the Agentcities concept. As a consequence, a lot of different projects are being set-up to deploy the network and apply it to various domains. The updated list of current projects can be found on the Agentcities web site [8].

In addition to deploying the network, these projects will build compelling applications. Agentcities is open to all kind of applications, the initial focus being on entertainment and other location-based services, which take benefit of the grouping of services onto cities. But other applications can include: healthcare, manufacturing, pervasive and wireless applications, e-learning etc. Some network management services have also to be provided at the platforms level, to allow nodes to get information about the status of the network and to enable new platforms to publicize. These services can be inspired from Internet mechanisms, they must be automated and must not create a central point of failure, making Agentcities a centralized network would result in loosing most of the advantages of agent technology.

In addition to agents, Agentcities will build on a wealth of innovative technologies including Semantic Web technologies, UDDI discovery services [10], Grid Computing [9], service description languages.... Agentcities will also have to take care of integrating with existing environments so that a large community of users can accept it. This is especially true as Agentcities deals in part with electronic commerce, an area where many systems are currently in development. Agentcities will have to take these systems into account, while taking care of not being limited by them.

The great number of initiatives making up Agentcities makes it more difficult to maintain the consistency of the network, so that all the platforms can actually inter-operate and Agentcities is really a single network. In addition, the different initiatives should co-operate so that no effort is lost as solving several times the same problem. This led to the creation of an independent body, the Agentcities Task Force (ACTF), on Oct. 5th 2001, to represent Agentcities in conference and standard organizations and to ensure the liaison between the different Agentcities initiatives and the relevant standards. The ACTF does not intend to constrain the different initiatives, which could retain their freedom, but on the contrary to allow them to better progress towards a common goal.

4 CHALLENGES

The potential impact and future use of Agentcities, as a new engineering paradigm for access to interactive Web like services has three key challenges:

1. The acceptance and common understanding of the concept;
2. Large research organisations and industrial take-up with core concepts being either standardised or standard approaches (de-facto in some cases) being integrated and deployed;
3. Addressing key research concepts that provide high-level service interoperability in an open and dynamic way.

The latter sections have provided the Agentcities concept and the results of the take-up of the current concept. In this section, a set of initial conceptual and design challenges that need to be addressed in order to deploy high-level service interoperability: a) Ontologies, b) Aggregation, and c) Trustworthiness.

4.1 Ontologies

The ad hoc structure of web pages to provide access to services, the maintenance of these services and user access (via high-level search engines, portals etc.) have provided the drive for a new initiative for developing Web-based services - the Semantic Web. ”The Semantic Web will bring structure to the meaningful content of Web pages” [19]. To achieve this goal a Web-based concept of ontologies is used.

The use and need for ontologies takes on a variety of perspectives in the software engineering environment. A concept of ontologies and modelling, in an Agentcities environment, is used at one level to ease automation of:

- Service and domain re-use as an engineered component;
- Service and domain knowledge sharing;
- Service and domain aggregation and specialization.

For Agentcities a concept of service and domain can be applied to many levels of abstraction, e.g. a service can be seen as part of an Agentcities infrastructure, a broker, or an application itself. The concept of domain can also make the same mapping.

The approach is to create a common structure and representation languages that will enable common concepts to be matched. Currently a descriptive approach has been used by the Semantic Web initiative. How concepts are shared and matched is left to the application developer. Although the future vision of Web-based services fits in well with the Agentcities concept there are currently some deficiencies, which need to be addressed. The deficiencies in the current Web model can be seen in considering two common characteristics of Agentcities: that of openness and autonomy.

The Agentcities vision is to support a higher degree of automation of the service through utilizing the service model rather than a whole a priori knowledge engineering process that is presupposed by the web. In the Agentcities concept the definition of the service model and its constraints is part of the ontology model. However, it is also clear that although there will be a clear core set of representation languages and structures for utilizing an ontology there will not be one and that dealing with diversity of potential models will be key to high-level semantic interoperability even if the agent or service can not work directly with a particular model. Hence some specific key challenges can be summarized as:

- Upper ontologies which may specifically cover policies, conventions, social and cultural contracts that address accountability between agents;
- Service ontological modelling requirements and agent use;
- Imposed requirements of an ontology representation language by domains specific needs;
- Ontology requirements on agent conversations and communication languages.

These requirements follow closely the requirements that the current FIPA ontology TC proposes as part of its work plan. In line with this the Agentcities implementations also work closely with other standards proposed by W3C etc.

4.2 Aggregation

The dynamic aggregation of services via an agent infrastructure, which has been introduced in the previous section on ontologies, requires that an agent infrastructure support concepts of co-ordination.

In most multi-agent systems coordination refers to an external requirement, defining a process in which agents engage in order to ensure a community of individual agents acts in a coherent manner. Coherence means that agents' actions work well together to solve problems, and that they do not conflict with one another. Coherence refers to, and acts as a measure of, how well a system of agents behaves as a unit [13].

There are a number of approaches, which have been devised to achieve coordination in agent systems. Coordination techniques may be classified in four broad categories [14]: organisational structuring, contracting, multi-agent planning and negotiation.

Without good coordination mechanisms, many of the benefits of the multi-agent paradigm simply disappear. Jennings identifies four major components, which must be present in any comprehensive coordination technique:

- There must be structures, which enable the agents to interact in predictable ways;
- There must be flexibility so that agents can operate in dynamic environments and can cope with their inherently partial and imprecise viewpoint of the community;
- There must be social structures, which describe how agents should behave towards one another when engaged in the coordination process;
- Agents must have sufficient knowledge and reasoning capabilities to exploit both available structure (individual and social) and flexibility.

Agent architectures usually separate out the coordination aspects into three areas of the architecture: content language, communication protocol, and the internal agent computation. There is a fourth aspect that may also be applied - the ontology, which defined the context of the content (often referred to as the domain model).

Essentially to create dynamic service aggregation requires that the concept of social agency be supported as proposed by Singh [18]. The social agency he proposed was to consider higher-level communication protocols. As a minimum Agentcities concept this agency needs to define a possible team or cluster of agents explicitly that are involved in an overall service delivery/access. Currently, in many systems most of the social agency is implicit, that is the agents are assumed to cooperate as the explicit computational behaviour of a MAS is defined by the set of services it supports. It is deemed successful if the behaviour exhibited provides the desired functional support of these services. Hence the social agency has a dependency on the application or service it is supporting.

To incorporate these requirements into a system which supports openness is the concept of policies [12]. The policies model's the social concepts of services in how to define explicitly a set of service constraints and a service commitment. Hence to achieve high-level service interoperability means having a service architecture mapping to agents, where an agent system supports an extended service models, communication protocols and some minimum concept of policies for defining constraints.

For Agentcities there will be a number of challenges to solve in integrating the many standards of services and still supporting a degree of openness. For some partial solution FIPA is currently developing standards in the area of service definition, policies and commitments. Current work that provides some partial semantic solution is the ALFEBIITE project [1].

Some core challenges to be addressed taking into account these developments for Agentcities are:

- Design of rich service model which supports some concept of constraints of use;
- Standard high-level protocols for negotiating e.g. about a particular service per se not just about getting the service;
- Set of standard service policies that provide a high-level set of constraints and explicit commitments to the general concept of service interoperability.

4.3 Trustworthiness

The concept of a system's or service's trustworthiness and approaches to a realization varies due to the multifaceted nature of the concept. Attributing some value of trust to a system to be considered trustworthy is often very personal, based upon experience. In fact our very social and cultural approach to evaluating a first meeting of a service can be, for example, strongly influenced by someone's recommendation if we have attributed a high-level of creditability of knowledge to that person about that particular service. Hence, the very success or failure of a service in the physical world could be based on someone's recommendation. Current research has also demonstrated that we bring our social model of the world when we interact with various inanimate objects from the toaster to the computer [17].

The multifaceted nature of creating a high-level concept of trustworthiness requires support for generic concepts of trust, security, and privacy, within Agentcities architecture, defining their roles as:

- Trust: is a social concept for evaluating risk, which is often situated in a cultural environmental and driven by a community's need for cooperation through communication and interactions for the perceived survival of that community;
- Security: is a set of physical realisations, which reduce the risk of harm within the environment. Security can provide fundamental building blocks for supporting concepts of trust.
- Privacy: provides both a conceptual and physical space to the social protection of high-valued items.

While it is quite easy to secure a closed multi-agent system, where access can be controlled and restricted, security is of great importance when dealing with open systems in wild environments, such as the Internet, where the objective is potentially to let any new party dynamically join. Security of multi-agent systems is the first requirement prior to any commercial deployment of agent services. FIPA is currently defining security specifications, which, once more, can be validated and improved on the Agentcities test bed. Some of threats are common to any information system on the Internet: eavesdropping, traffic-analysis, masquerading, denial of service, etc. For these, more or less efficient safeguards have already been found. The real problem comes from new threats, which are linked to the proper characteristics of the agents:

- Privacy: each agent encapsulates some personal information about its users, which it must not publicize to any other agent. Moreover, when required, communication between agent should rely on some level of confidentiality, using for instance encryption mechanisms;
- Trust: in a dynamic environment, parties involved in a cooperation may not have prior knowledge on each other. In order to work efficiently, these parties need to know the level of confidence they can have in the fact that the other party is actually what it claims to be and also in the fact that the other party can actually do what it proposes to do. This requires some kind of standardized authentication or certification mechanism.

For the success of Agentcities as an open electronic service environment the deployment architectures will require to work with classic concepts of security, to enable social concepts of communication to be captured within the model of interaction and to deal with legal aspects of privacy. Some of these concepts are being dealt with, for more information see for privacy [6], trust [11] and security [16].

5 CONCLUSION

The focus of this paper has been the Agentcities concept and its challenges for providing advances in distributed deployment of services and their access across the WWW. In using the web for service access in the future, it is clear that certain demands on the infrastructure will be made and agent technology can provide some of the mechanisms to address these demands, such as service aggregation. However, until now a mapping of agent technology on such a large-scale distribution of heterogeneity has not been defined in any conceptual way. The Agentcities concept bridges this gap by recursively mapping the concept to the web infrastructure, taking into account the heterogeneity of designs through considering high-level interoperability of services. The Agentcities test bed, built from the different initiatives, represents the opportunity to define, implement and validate the key challenges for the concept to succeed. In particular we wanted to bring out the following important features:

- Openness, which can be achieved through providing structured and explicit interfaces of interoperability i.e. a set of standards are essential;
- Interoperability in agent systems is frequently only considered at some communication level. This is not enough for the future of service support, but is an essential starting point;
- The meaning of an action of communicating is at a different level of interoperability more than just sending a message. The content of the message requires agreed and precisely defined ontologies. Ontologies are still defined and abstracted by hand. The automation of this process is still a main issue but aspects are being resolved and are essential for high-level service interoperability.

More general technical and social challenges must also be considered in the future realisations of the Agentcities concept. There is a clear need for tools to support a methodology for this new engineering paradigm both at the concept level of agents and their ontological mapping (see the MESSAGE project which provides a framework for defining the many concepts of a service within an agent framework [4]). There are current advances in UML modelling, which supports the design of agents and their mapping to ontologies. However, there are two key future needs to be addressed: the mapping of agent services to web services and how to deploy agent-enabled services into active web. The latter point is that if services are to be available continuously and interactive we need to be able to add new services to a live heterogeneous service network. Such an approach will require that we have better concepts of security for such deployments.

The idea of autonomy and dynamics within an interactive system that supports openness brings a number of potential social consequences to interacting with an Agentcities environment. One clear concern is that of responsibility when an action goes wrong in a high-level service environment. A model of responsibility in society is based on many factors of trust, ethics and conventions, which establish some form of social contract. While electronic contracts are still based around strong human intervention and on point-to-point solutions between two organisations, a notion of responsibility is not often considered. However, once we move into automated service aggregation at any level we need to employ the more powerful concepts of agent technology; that of designs using concepts of society, beliefs, intentions, conversations, conventions, responsibility, high-level knowledge about applications etc. Within these designs researchers and engineers consider models of computational trust, security and privacy in order to see a future in how we can have access to the ever growing and advancing electronic society and be in a position of trusting the environment.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the members of the Agentcities initiatives around the world and the originators of the concept, most notably Steven Willmott and Bernard Burg [20]. Note that the opinions reflected in this paper are those of the authors and are not necessarily those of all the partners developing the Agentcities concept.

References

[1] 'ALFEBIITE project (A Logical Framework for Ethical Behaviour between Infohabitants in the Information Trading Economy)'. http://www.iis.ee.ic.ac.uk/alfebiite.
[2] 'FIPA (foundation for intelligent physical agents)'. http://www.fipa.org.
[3] 'Fipa interoperability bake-off, london, april 2001'. http://sharon.cselt.it/projects/jade/papers/BakeoffReport.html.
[4] 'MESSAGE, Methodology for Engineering Systems of Software AGEnts'. http://www.eurescom.de/~public-webspace/P900-series/P907/index.htm.
[5] 'Microsoft.NET'. http://www.microsoft.com/NET/.
[6] 'PISA'. http://pet-pisa.openspace.nl/pisa_org/pisa/index.html.
[7] 'Sun ONE'. http://www.sun.com/sunone/.
[8] 'The Agentcities web site'. http://www.agentcities.org.
[9] 'The CoABS Grid'. http://coabs.globalinfotek.com/.
[10] 'Universal description, discovery and integration'. http://www.uddi.org.
[11] C. Castelfranchi and R. Falcone, 'Principles of trust for mas: Cognitive anatomy, social importance, and quantification', in *ICMAS'98*, pp. 72–79, (1998).
[12] P. Charlton and R. Cattoni, 'Evaluating the deployment of fipa standards when developing application services', *International Journal of Pattern Recognition and Artificial Intelligence*, **15, No. 3**, 551–577, (2001).
[13] N. R. Jennings, 'On being responsible', in *Decentralized Artificial Intelligence*, pp. 93–102, Amsterdam, (1992). Elsevier/Holland.
[14] M.J. Johnson P. O'Brien M. E. Wiegand N. R. Jennings, P. Faratin, 'Using intelligent agents to manage business processes', in *First International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM96)*, pp. 345–360, London.
[15] D. Bonnefoy P. Charlton and N. Lhuillier, 'Dealing with interoperability for agent-based services', pp. 236–237, (2001).
[16] Stefan Poslad and Monique Calisti, 'Towards improved trust and security in fipa agent platforms', in *Autonomous Agents 2000 Workshop on Deception, fraud, and trust in agent societies*, Barcelona.
[17] B. Reeves and C. Nass, 'The media equation: How people treat computers, televsion, and new media like real people in real places'. Cambridge University Press, (1996).
[18] M. Singh, 'Agent communication languages: Rethinking the principles', pp. 40–47, (1998).
[19] J. Hendler T. Berners-Lee and O. Lassila, 'The semantic web', *Scientific America*, **May**, (2001). http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html.
[20] S. N. Willmott, J. Dale, B. Burg, C. Charlton, and P. O'brien, 'Agentcities: A Worldwide Open Agent Network', *Agentlink News*, (8), 13–15, (November 2001).