

# Service-Oriented Distributed Communities in Residential Environments

Pierre Parrend, Yvan Royon and Noha Ibrahim

INRIA Ares team, CITI Laboratory,

21, Avenue J. Capelle, INSA Lyon, France.

Phone: +33 (0) 04 72 43 71 29

E-mail: {firstname.lastname}@insa-lyon.fr

**Abstract**—Since the inception of the World Wide Web, two major shifts have occurred that have deeply impacted the way people use the Internet. The first shift is the development of virtual communities, which enable people to share data or communicate together according to common work targets, or common interests. The second shift is the dramatic increase of on-line services, ranging from online book stores to intricate Virtual Organizations. Nonetheless, little effort has been done to bring these two worlds together, mainly due to insufficient technological support. Recent evolutions in Internet broadband access, personal execution platforms and semantic for communities let foresee that the existing gap between services and communities could soon be bridged. We propose an architecture for service sharing in distributed communities, specifically targeted at residential environments.

## I. INTRODUCTION

### A. Evolutions in Internet Access

In the area of Internet broadband access, we see today in both research and development a focus on provisioning services to the user. Yesterday, the focus was on bringing high speed connectivity to the home. Today, it has switched to providing network services to connected homes, *e.g.* voice and video over IP. The next coming evolution is breaking the access providers' monopoly on service delivery. The central element of service provisioning is the Home Gateway, which is defined as the “high tech device ensuring continuity between the home network(s) and the in home connected devices and the external world represented by a wide area network (WAN)” [1]. Several works ([2], [3]) aim at allowing multiple parties to concurrently provide services to the home by taking advantage of this Home Gateway.

Business models attached to ongoing approaches are all about allowing companies to sell services to the end-user in a transparent and user-friendly way. However, one knows that end-users want to have control when dealing with hi-tech tools or electronic devices. The degree of control is indeed an important factor for a successful popular equipment: power users always want to play with electronic goods, open them, tweak their performance and functionalities. This sometimes even brings a second life to devices that support unexpected usages which the designers did not foresee.

### B. Characterization of Communities

We believe that an advanced user-targeted service-oriented system must give back control and freedom to the end-user. We therefore propose to define Communities that take advantage of Home Gateways to enable users to share not only data, but also services and code. Promising criteria for resource sharing is to organize them according to the centers of interest of the users. Principles and structure of these Communities are defined, as well as their behavior and the processes that support service use. Users choose which resources to share on their own service environment. They can then create, join and participate in communities driven by centers of interest. The hardware element which connects the user to the community is the home gateway, or any mobile device, provided that it can run a tweakable service environment.

In this paper, the word “community” refers to a group of people gathered by common centers of interest and sharing resources related to these centers of interest. Such resources are data or applications that can be downloadable, remote or distributed, and may be used jointly for instance through software composition. The community and its resources are entirely distributed.

Communities are focused on social aspects much more than on technological ones, which does not prevent them from being strongly dependent on the tools being used. Three main types of communities exist currently: data publication, data sharing, and distributed applications communities. A more complete analysis on communities can be found in [4].

Data publication communities are made of data distribution (mainly peer-to-peer) networks and communication environments. Data publication involves a mostly anonymous process of data provisioning and search engines. It thus makes possible to find a software package, learning-oriented resources, a music theme or a film according to the needs and wishes of the user. Communication-oriented environments propose centralized tools (chat, fora, blogs, newsgroups) or distributed ones (Instant Messaging). They enable people to keep connecting together. Data Communities provides generic tools that can be used without regard to the actual activity or centers of interest of members. A great majority of them is open, *i.e.* no restriction exists on insertion of new members.

Data and code sharing communities comprises Development

<sup>1</sup>This work is partially funded by MUSE IST FP6 Project n°026442 and AMIGO IST FP6 Project n°004186.



and Collaborative Work, where resources are meant to be modified by several persons. Development communities are centralized around a code repository. Community support is often weak, and provided by third party tools. Collaborative Work communities target at a common achievement through shared edition tools.

The last kind of community is built by resources sharing distributed applications, namely data or calculus grids. In calculus grids especially, a resource-consuming process is executed on several distant machines, so as to take advantage of unused CPU resources.

In the world of communities, a clear gap exist between data sharing, code manipulation and distant execution. Service-Oriented Communities based on residential gateways can help bridging it by providing both the technological support and the suitable environment.

### C. Particularities of Service-Oriented Distributed Communities

Service-Oriented Distributed Communities are center-of-interest based communities supporting scalability through an underlying peer-to-peer network. They enable service sharing between mutually unknown persons by supporting service execution. The P2P layer provides also support for classical data and applicative components publication.

The remainder of this paper is organized as follows. Part II presents the pervasive architecture of the system, along with use cases and involved actors. Part III details the behavior of a community during the various phases of its life. Part IV presents the main security challenges brought in by proposed community architecture. Part V is an overview of related works, and part VI concludes the paper.

## II. SERVICE-ORIENTED COMMUNITIES

Service-Oriented Communities are a new kind of community we propose to build so as to take advantage of novel infrastructures. We first need to define the use cases of such communities, as well as their overall architecture, actors involved, and existing interactions.

### A. Overall View

Service-Oriented communities are communities where users share not only data and application packages, as in existing mainstream communities such as peer-to-peer, but services that are remotely accessible. All these resources share a common theme, the center of interest. Many real world communities can benefit from this approach: research communities, collaborative work users, virtual enterprises, developers, as well as open communities of interest that wish to share data, applications and services related to a specific theme.

Service-Oriented communities hosted on Home Gateways are mainly built from users that have broadband access to the Internet. This enables stable connections and better availability of data and services while the user is online. They are ad hoc communities, in the sense that they are made of the users that are present at a given time, with no possible

control over connection or disconnection of any user. As we already highlighted, someone who wants to be part of such a community needs a suitable service environment. This service environment can be located on a gateway (modem). This enables coupling with advanced network services (data streaming, *e.g.* video on demand, and so on, as in the Muse-IST project [2]). Therefore, the user can connect to the Service-Oriented community either directly through the gateway, or through a mobile equipment or laptop, if routing facilities are available. This make mobile access to the Service-Oriented community an easy extension to achieve, and opens the way towards connection of ambient home devices.

Figure 1 shows this architecture.

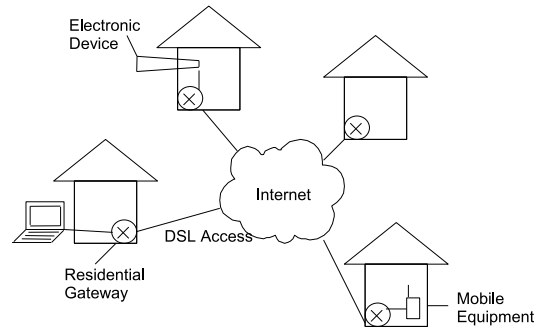


Fig. 1. The Global Architecture of a Community of Services

Building communities in residential gateway environments makes it possible to exploit and adapt the actor model for Residential Gateways in the context of communities. This also provides better understanding of interactions inside and between the execution environments. Involved actors are:

- The *network access provider* provides IP connectivity to the home;
- The *home gateway provider* sells the equipment that connects the home network to the internet, and that hosts the execution environment for services;
- The *user*, who typically uses services on the service gateway;
- The *service provider* provides services to the user.

In Service-Oriented communities, the user also plays several management and service-providing roles:

- The *service environment provider* owns the service environment (or service gateway);
- The *service environment manager* supervises and configures the service gateway;
- The *service manager* supervises and configures a particular service.

The community acts as a service provider, which means each user plays this role for other members of the community.

### B. Versatile Communities

Figure 2 shows the various possible interactions based on Service-Oriented facilities, namely data, components and service support.



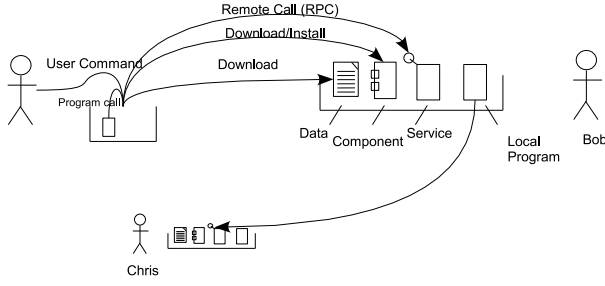


Fig. 2. Possible Interactions between Users in a Service-Oriented distributed Community

Combining data, application and service sharing in a common environment makes such an architecture pretty versatile, compared to existing tools proposing community support. Two types of resources and three different actions can be identified in order to analyse community tools: respectively data and process resources, and publication, integration and modification of resources. We claim that Service-Oriented communities can support all actions over both types of resources either directly or through higher level tools. Further work is required in order to formalize proposed classification. Figure 3 shows characteristics of existing community tools.

|         | Publication          | Integration                  | Modification                    |
|---------|----------------------|------------------------------|---------------------------------|
| Data    | P2P, IM, Chat, Fora  | Collaborative Work           | Collaborative Work, Development |
| Process | Grid remote services | Component based Applications | Service Composition             |

Fig. 3. Characteristics of existing Community Tools

Publication of data is provided by the peer-to-peer layer. Data integration and modification needs additional facilities, that can be provided as applicative components.

Integration of different processes can be achieved through component-based application extensions with available components. It requires that code loading is available [5]. Fine-grained modifications of processes can be achieved through service composition strategies (see [6] or [5]), which goes beyond the scope of this paper.

### III. COMMUNITY BEHAVIOR

A Service-Oriented Community gathers users around a common center of interest, or theme. This center of interest is made up of various resources that the user can look up for, add or remove. The community itself requires a specific life-cycle management process. We define the necessary mechanisms for supporting this behavior.

#### A. The Community

A Community is accessible for new users either through invitation by a friend or acquaintance or through a centralized

Community Repository. This repository is the entry point for Communities. It contains Community description data necessary to allow potential members to search for communities matching their wishes and to join them. These data are a set of keywords defining the community center of interest as well as addresses of several active members that can introduce the new member. So as to comply with fully distributed characteristics of defined communities, central repository must not be involved in the process of member management or resource publication.

A Community is defined by Service meta-data that are stored locally on each member's platform. These meta-data contain the semantic description of the center of interest as well as required information for retrieving resources associated with each specific topic. A topic of interest is a keyword that represents a subset of the center of interest. Topics can be independent from each other - that is to say they build a list of keywords - or formally organized in an Ontology. Because no central point supports community life-cycle, an update mechanism needs to be available for propagating modifications when they occur.

The life cycle of the community contains following steps: creation of the community, insertion of new users in the community, users withdrawal, community destruction.

1) *Community creation*: a Community exists when users (at least one) exist that provide services and data for it. So its creation can be made locally by a single user that defines its own center of interest. For being active, a community needs either to have a second member or to be published on the central Community repository.

2) *User joining a Community*: Communities can be accessed by contacting directly a member of the community (private communities), or by looking up communities that are relevant for a given user in the centralized repository. In the second case, the future member contact the repository, and chooses one (or several) target communities. He then contacts one of its members, in order to be introduced in the community. The latter provides the newcomer with the community meta-data, so as to enable him to have access to available resources.

Figure 4 shows the mechanism of research, discovery and subscription to a community by a given user: request for communities, registration, and request for services or data.

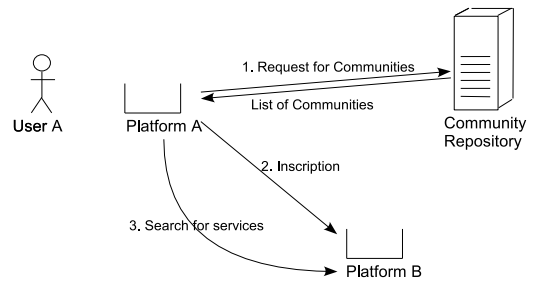


Fig. 4. The Mechanism of Community Research and Discovery



3) *User leaving a Community*: in existing communities, a user can leave by simply stopping to provide and to look for resources. It is thus impossible to determine whether withdrawal is permanent or temporary. We propose a suitable mechanism in part ‘user disconnection’.

4) *Destruction of a Community*: a Community is destroyed when no more Service meta-data exists that defines it. As far as all members own a copy of this meta-data, this occurs when no more member is active. A mechanism for managing community publication on the Community repository still needs to be defined. Two main options can exist: either a community administrator is responsible for keeping community data up to date, or a heart-beat probe should be done by the community repository so as to ensure that advertised Communities are available.

## B. Use Scenarios

While a member of a given community, a user can be active or not, that is to say connected to other members or not. A Service-Oriented Community must thus support user connection and disconnection, searching, finding, adding and removing resources. In case Service meta-data are organized according to an Ontology, this latter may be extended.

1) *User connection*: a user connects itself by advertising resources he makes available. This is done by adding all resources he owns to reference lists of the community.

2) *User looking for resources*: it can be done by sending a request to other members of the community for resources matching a given topic of interest in the community. The specification of the request format can be subject to discussion, according to the degree of precision and of exhaustivity required. If an Ontology is used, it structures the relationship between themes of interest through the community and enables the specification of attributes to these themes: synonyms, translation in several language, and so on. Such attributes facilitate the publication of resources with matching topics, and make the search less dependant to keywords chosen by the publisher, thus providing communities with quasi exhaustive searches over all relevant topics. Strictly specifying the different topics in an ontology moreover allows to realize research over several communities, which may have partially overlapping themes.

3) *User adding resources*: First step of resource publication is to associate meta-data to the resource: type of resource (data, application, service), topics of the community to which it is relevant, optionally the name of the publisher, a version number. Resources are then advertised though flooding or DHT according to chosen strategy.

4) *Removal of a resource*: A resource disappears when all members that have a copy of the resource become inactive or withdrawn from the community. In case that a resource becomes out of date, it can be updated or removed, provided that either the resource reference list is available (DHT strategies), or that a resource revocation list exists (flooding strategy).

5) *Extension of the Ontology*: In the case the community’s topics are described through an ontology, evolution of this

latter can be necessary. The user first updates its local copy of the ontology. Thereafter, these modifications are propagated by flooding the members of the community: each member notifies its neighbours, who do the same, and so on. Such a mechanism implies that at a given time, it is not possible for users to have an exact knowledge of the current state of the ontology. Some resources may then be hidden to some members for some time. The alternative to this would be to make use of a centralized server containing the ontology for each community. This is contrary to our hypotheses. Further work is still required in order to specify such a protocol.

6) *User disconnection*: The occurrence of frequent user disconnection is part of the definition of communities, in which users come and go as they wish. No service or data can be assumed to be always available. Replication of data, handled by the peer-to-peer overlay network, can help decrease the effect of disconnection.

## C. Ontology Management

When a user of a community searches a specific service or data, he chooses specific keywords that are matched with those in the ontology for a quick and efficient search. Ontology thus enables services and knowledge sharing and reuse. It is possible to use languages such as OWL that provide richer expression and semantic reasoning.

An example of an ontology that could be used in a community interested in animals is given figure 5.

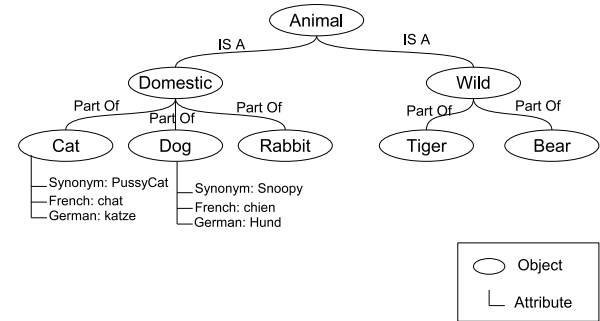


Fig. 5. A part of an Ontology Example

A member of this community would like to search for services related to cats. He queries the system by providing the keyword cat in german, KATZE. His system matches this keyword with the related ontology and finds out that Katze is a translation of cat. This ontology will let two users searching for the same information but using different keywords like in different languages share their resources transparently.

## IV. TRUST IN COMMUNITIES

A systematic analysis of security problems implied by Service-Oriented communities is out of scope of this work. However, it is necessary to highlight the main problems that can occur in such an environment.



### A. Protecting the Community

Security questions appear to be largely dependent on the kind of community considered. Open communities, *i.e.* communities where everyone can join and leave without control, are the most exposed ones. Actually, it is not possible to assume anything about the behavior of members in communities where it is possible to join and leave, and where anonymity is guaranteed. On the opposite, closed communities often reflect real world organizations. Traceability can then be the only necessary feature of such environments, as far as liability of members is provided. Communities often take an in-between position: members, even though not personally known, are identified through a given identifier, and associated to a personal trust level. This trust level evolves according to the user's activity in the community and the duration of its presence. [7] provide a detailed introduction to trust<sup>2</sup>. Service-Oriented communities are more likely to be of the third category, that is to say trust based.

Two elements are to be carefully protected: the community itself, and the resources it contains. The community exists as far as users can join it, and as members can propose and find resources. First, integrity of the community needs to be guaranteed at the community repository server. Second, community should keep being available to members as far as other members exist to provide resources. The problem becomes delicate for handling community evolution: the description of the community should be expandable, but should not be possible to prune. Resources also need to be protected, in particular code and applications. Lack of security in this domain would enable a malicious person - who could even be member of the community - to modify code, and introduce malicious routines. Malicious code can imply backdoors to members' computers, propagating viruses, and so on. A fined grained approach can be achieved by defining capabilities specific to each user, which limits the access right to resources and services [8].

A wide-spread strategy for protecting code is to guarantee its integrity and origin. This can be achieved through code signing, for instance through PGP facilities. Code signing allows to identify the emitter of the application, but does not answer to the question: can I trust this emitter ?

### B. Trust between Members

Members of a community can be more or less reliable persons. It is thus necessary for each member to evaluate whether a given member can be trusted, in particular when he releases code or applications. This can be achieved by monitoring and analysing actions of the members of the community. Users that interact often with others, providing useful and safe content, will have a high degree of trust. Members who have joined recently, or who are not very active, will have a low degree of trust. Members who release malicious content will be categorized as distrustful (see [9],

[10]). Such a mechanism can be also used so as to enable parental control.

Two problems appear in a trust mechanism: the insertion of new users, and the propagation of trust information [11]. If new members have a too low degree of trust, they will have difficulties in providing content, and in becoming a full member of the community. A solution can be to provide high level of trust at the beginning, with high accountability for any suspect action. Moreover, a single user will only have information about members he is loading resources from. A solution is to provide a reputation service, which enable every user to know to what extent the others trust the members of the community. Such a mechanism also provides better reactivity to malicious members, which can be isolated before having harmed every benevolent user.

### C. Safe Execution of Code

As far as open communities can not force users to be previously known by the system and thus be already trusted, it is necessary to protect the execution environment from potential malicious actions. Service platforms provide isolation through sandboxing, and through fined-grained execution permissions for untrusted process, especially preventing access to the local file system, or to sensitive system operations.

As we see, tools for protecting a service environment in the context of community exist. However, future works will imply a deep analysis of risks and suitable protections, so as to ensure that the proposed system is really safe, and not just providing a set of juxtaposed security features.

## V. RELATED WORKS

### A. Context

Our work aims at supporting service and data publication and providing for virtual communities, taking advantage of the important bandwidth available through wide band Internet accesses, and considering security issues as a key problem for such an architecture's viability. As far as we know, no other work seems to have been done about this very problem. However, several fields of research share problems with this configuration, and propose some suitable solutions: Semantic Web, pervasive grids and ubiquitous computing, as well as Service Gateways for supporting access to and execution of services.

Muse-IST (Multi-Service Access Everywhere) [2] is a European Union funded project which aims at providing multi-services wide band Internet Accesses to the end users. The goal is to enhance quality of triple play solutions, and to enable a Service Gateway to interact with home equipment for providing new services, such as remote maintenance, data streaming from multiple providers, and so on. We propose to enhance specified Home Gateways with Service-Oriented Communities.

Amigo-IST (Ambient Intelligence for the networked home environment) [12] is also a European Union funded project. The Amigo project will provide solutions for the major problems that are encountered in the use of home networking today.

<sup>2</sup>You can also refer to IST Working Groupe iTrust for more information, <http://www.itrust.uoc.gr/>



The project aims to improve the usability of a home network by developing open, standardized, interoperable middleware and improve the attractiveness by developing interoperable intelligent user services. Amigo provides technical solutions for integration of services in Service-Oriented Communities.

### B. Support for Communities

This work aims at proposing mechanisms for supporting communities more than to analyse the concept of community itself, which relates mainly to a sociological approach. Several fields deal with providing tools that can be integrated with great benefits for communities, although they do not directly deal with them: Semantic Web and Web Services, as well as execution and connection environments.

Semantic Web opens a wide range of perspective for advanced data retrieval facilities and new applications in particular through Web Services [13]. It enables automated data analysis, semantic interoperability through ontologies. Security is typically provided through digital signatures and trusted rating servers, for preventing users to be cheated by low quality or malicious services. An extensive presentation of semantic web services is provided by [14]. Semantic web is still a young research topic. Several teams are working at building systems compliant with this technology. [15] proposes a semantic web of services for weak networks, that is to say networks with space, time, semantic disconnections.

The service environment we propose can take advantage of the OSGi platform [16], especially for application component loading and installing. Moreover, required extension such as peer-to-peer support [17] and virtual gateway support [3] for isolation of services are available. Another but similar approach is the one of [18], who proposes an ad hoc interaction framework that plays the role of the overlay, and is similar to OSGi relative to provided services. Though this work is targeted at pervasive systems, it does not give more insight about this aspect. The work of [19] is the most similar to ours, in that it proposes a pervasive grid system over wide band Internet access (over Optical Burst Switching - OBS). This work is very prospective, as far it tries to see how services could look like in 20 years. Our proposition, based on OSGi and DSL connection, is a more realistic short-term one.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an approach where users take control over their own residential gateway and can choose to cooperate with others users sharing a common interest. Service-Oriented communities can support data and process sharing. With suitable tools, they can also not only publish these resources, but also integrate them and even modify them at runtime.

Such an approach can therefore be considered as a powerful support for all kind of communities, from peer-to-peer and instant messaging to remote service execution or Collaborative Work. Introduction of semantic for classifying resources of the community enables advanced search facilities, as well as integration of several communities.

Built communities presented in part II and III are ubiquitous, as far as they enable every person connected to the Internet to have access to them. They can also support mobile users, by taking advantage of wireless Access Point existing in most residential gateway, and even ambient devices. This approach can therefore be said to support pervasive services.

Two major extensions of this work are to be foreseen: first, a detailed analysis of the possibilities brought in by the use of semantic in communities, and secondly, a systematic security study in order to be able to assert the level of security provided by Service-Oriented communities, and thus to determine in which real world environments - leisure, virtual organizations - they can be reasonably introduced without exposing sensitive data or services.

## REFERENCES

- [1] HGI, "Home gateway initiative, vision and whitepaper," 2005.
- [2] MUSE Project, "IST-026442 FP6," <http://www.ist-muse.org/>, 2005.
- [3] Y. Royon, S. Frenot, and F. L. Mouel, "Virtualization of service gateways in multi-provider environment," in *Component-Based Software Engineering*, 2006.
- [4] E. D. Mynatt, A. Adler, M. Ito, and V. L. O'Day, "Design for network communities," in *CHI*, 1997, pp. 210–217. [Online]. Available: [citeseer.ist.psu.edu/mynatt97design.html](http://citeseer.ist.psu.edu/mynatt97design.html)
- [5] N. Ibrahim, F. L. Mouël, and S. Frénot, "Automatic negotiated integration of services in pervasive environments," in *MWS Workshop (Middleware Web Services) co-located with EDOC 2005 Conference*, Sept. 2005, Enschede, The Netherlands.
- [6] S. Majithia, D. W. Walker, and W. A. Gray, "A framework for automated service composition in service-oriented architectures," in *1st European Semantic Web Symposium*, 2004.
- [7] S. Ruohomaa and L. Kutvonen, "Trust management survey," in *Trust Management: Third International Conference, iTrust 2005, Paris, France, May 23–26, 2005*, ser. LNCS, S. S. Peter Herrmann, Valérie Issarny, Ed., no. 3477, 2005, p. 77.
- [8] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke, "A community authorization service for group collaboration," in *Third International Workshop on Policies for Distributed Systems and Networks (POLICY'02)*, 2002.
- [9] S. March, "Formalising trust as a computational concept," Ph.D. dissertation, University of Stirling, 1994.
- [10] M. P. Singh, B. Yu, and M. Venkatraman, "Community-based service location," *Communications of the ACM*, vol. 44, no. 4, pp. 49–54, 2001.
- [11] A. Abdul-Rahman and S. Hailes, "Supporting trust in virtual communities," in *Proceedings Hawaii International Conference on System Sciences 33, Maui, Hawaii*, January 2000.
- [12] N. Georgantas, Ed., *Detailed Design of the Amigo Middleware Core: Service Specification, Interoperable Middleware Core*, ser. Deliverable D3.1b, IST Amigo project, 2005.
- [13] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, May 2001.
- [14] S. McIlraith, T. Son, and H. Zeng, "Semantic web services," *IEEE Intelligent Systems (Special Issue on the Semantic Web)*, March/April 2001. [Online]. Available: [citeseer.ist.psu.edu/mcilraith01semantic.html](http://citeseer.ist.psu.edu/mcilraith01semantic.html)
- [15] R. Krummenacher, T. Strang, and D. Fensel, "Triple spaces for an ubiquitous web of services," in *W3C Workshop on the Ubiquitous Web, Tokyo, Japan*, 2005.
- [16] The OSGi Alliance, "OSGi Service Platform," <http://www.osgi.org/>, October 2005. [Online]. Available: <http://www.osgi.org/>
- [17] S. Frenot and Y. Royon, "Component deployment using a peer-to-peer overlay," in *Working Conference on Component Deployment*, 11 2005.
- [18] G. Coulson, P. Grace, G. Blair, D. Duce, C. Cooper, and M. Sagar, "A middleware approach for pervasive grid environments," in *UK e-Science Programme Workshop on Ubiquitous Computing and e-Research*, 2005.
- [19] E. van Breusegem, M. de Leenheer, J. Cheyns, P. Demeester, D. Simeonidou, M. J. O'Mahoney, R. Nejati, A. Tzanakaki, and I. Tomkos, "An OBS architecture for pervasive grid computing," in *Workshop on Optical Burst Switching*, 2004.