

DiSUS: Mobile Ad Hoc Network Unstructured Services

Paul Fergus, Anirach Mingkhwan, Madjid Merabti, and Martin Hanneghan

Networked Appliances Laboratory
School of Computing and Mathematical Sciences
Liverpool John Moores University,
Byrom Street, Liverpool L3 3AF, UK.
{cmpfferg, cmsaming, M.Merabti, M.B.Hanneghan}@livjm.ac.uk

Abstract. Mobile devices are situated in an environment that consists of infrastructure and ad hoc networks. Unlike infrastructure networks, ad hoc networks are highly dynamic, wireless and do not have any centralised administration. These two environments accommodate two types of services: one type is registered and discovered using toolsets found within structured networks, e.g., JINI and UDDI, and the other type of service is situated within a network that has no structure, as in P2P. Our work highlights a key solution for providing unstructured services in a mobile ad hoc environment. This paper shows how mobile ad hoc systems can construct and semantically discover unstructured services in their environment. We describe the Distributed Semantic Unstructured Services (DiSUS) framework; the protocols used and describe the functionality of the prototype system we have developed.

1 Introduction

Visualise a high street shopping area, which is a simple outdoor environment. The street is full of shops, restaurants, street vendors and other people. We pop in and out from one store to another (structured services), buy a quick snack from a street vendor – here today gone tomorrow (unstructured services) – and greet people we know (neighbour nodes). All of these activities happen within our focal view. Mobile devices within real-world environments have to work the same way as our shopping area analogy. This provides users of mobile devices, with the ability to interact and use services in the same way we interact with shops and people within our real environment.

Information Space [8] is the concept of integrating information and services from the environment the user has access to. By considering the Mobile User as the centre of the surrounding information and services we find that, in reality, the environment the user moves into provides both structured and unstructured services. The ability to select and use both of these types of services to offer the maximum flexibility for the user is of paramount importance.

The need for an integrated Information Space will require the unification of wired and wireless networks and their services. In particular, the challenge is to bring together services within mobile ad hoc networks and infrastructure networks like the Internet [9]. Service nodes provide services throughout the Information Space by using Gateways that interconnect infrastructure networks and mobile ad hoc networks together.

We describe services within the Information Space as structured and unstructured and define these as follows:

- *Structured Services* use third party services to register and advertise services the peer provides, e.g., Directory Services, Proxy Services and Naming Services. These kinds of services typically have complex structures, such as network connectivity, database access and multimedia functions.
- *Unstructured Services* provide services independent of any third party intervention. This concept is based on a simple service definition, such as a kiosk that provides quick information, a TV remote control that simply changes the channel or a file-sharing application that exchanges digital content.

There are an increasing number of structured services available to users over the Internet and mobile ad hoc networks, yet unstructured services remain far behind. Internet-based structured services like JINI [12] and UDDI are already well defined; however they are incapable of providing services within dynamically changing network environments. This limitation can be overcome by situating services within the Information Space, using decentralised P2P concepts [14].

In addition to using a distributed ad hoc network, our Distributed Semantic Unstructured Services (DiSUS) framework contributes additional knowledge to this area by enabling peers to discover services dynamically based on semantic mappings between queries and distributed services.

The DiSUS framework enables every peer within the network to be a stranger. An individual with a notebook does not have to register with any centralised registry – we envisage that the individual will connect the device to the P2P network irrespective of where they are in an ad hoc fashion.

In the following section, we discuss the requirements for unstructured services in mobile ad hoc networks; in section 3 we provide an overview of the DiSUS framework and its components followed by an explanation of how the prototype works; section 4 documents our conclusions and our proposed future work.

2 System Requirements

Before we can distribute and discover unstructured services, it is paramount that our framework addresses a number of key requirements. The very nature of P2P networks means that peers will come and go over time, therefore we cannot guarantee the availability of unstructured services at any given moment. The challenge is to enable unstructured services to exist in such a non-deterministic environment and allow mobile ad hoc peers within the P2P network to effectively expose and discover unstructured services. We describe these requirements as follows:

- **Interoperability:** The framework must support all types of services, and not be restricted to specific services like Gnutella [4].
- **Decentralisation:** Services have to be completely decentralised; every peer that joins the P2P network must be capable of reaching any other peer without the help of a centralised third party.
- **Unstructured services in Information Space:** Services must exist in infrastructure and mobile ad hoc networks.
- **Dynamic environments:** Services have to be able to work in dynamically changing environments [16].

- Platform Independence: To support a variety of services and devices, a service has to be lightweight and capable of communicating across different platforms for example by using W3C standards [15].
- Ubiquity: The deployment of services must include a variety of mobile devices ranging from workstations to mobile phones.
- Intelligence: Service location must support discovery using semantic languages and toolsets that have inferential capabilities [6, 7].
- On-demand Services: Services will not run continuously and will be invoked as and when they are required [13].

3 Distributed Semantic Unstructured Services (DiSUS) Framework

In this section we describe our Distributed Semantic Unstructured Services (DiSUS) Framework, which implements several subsystems to distribute and semantically discover services within a P2P network. It provides a description of subsystem interaction and overall framework capabilities.

3.1 Framework Design

The requirements described in Section 2 are addressed within the DiSUS framework using three core components. These three components implement a P2P network, a knowledge base with inferential capabilities that allow us to semantically represent and query distributed peer services and a Web container capable of dynamically hosting Web Services as illustrated in Fig. 1. The integration of these components comprises the DiSUS framework and provides the foundation for our research.

The framework consists of two types of peer service that can choose to implement any, all or none of these components dynamically. These peers in turn are described as a Simple-Peer and a Specialised-Peer.

The DiSUS framework describes three peer services, which communicate with each other using standardised XML messaging wrapped in FIPA-ACL [11] compliant objects, as illustrated in Fig. 1. The two peers labelled Specialised-Peer-A and Specialised-Peer-B are peer services that have the ability to host Web Services, semantically match queries with local services on the peer machine and propagate queries within the P2P network. The peer labelled Simple-Peer-C is a simple peer and has limited capabilities. This type of peer service does not have the capability to host Web Services or respond to queries propagated from neighbouring peers. This peer service propagates its own queries and invokes discovered Web Services.

In this framework Web Services are hosted locally on peers and are described within the peer's knowledge base best describing, conceptually, what the Web Service means. This process enables the peer to capture the capabilities the service offers. Information about the Web Service itself is not entered into the knowledge base automatically, but rather a knowledge engineer relies on his/her understanding of the services capabilities and manually adds this information to the knowledge base. This is a temporary measure, which enables us to test our solution and we are currently

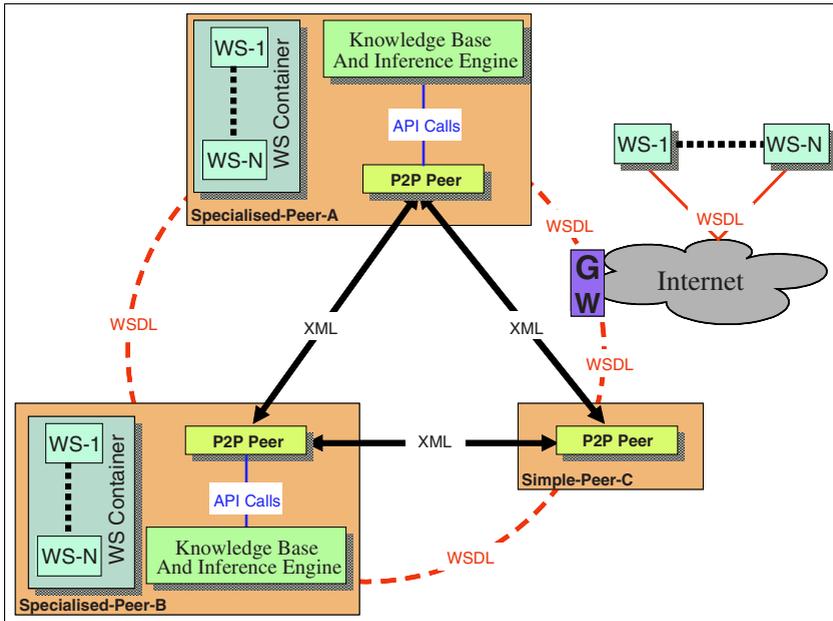


Fig. 1 Distributed Semantic Unstructured Services (DiSUS) Framework

working on algorithms to automate this process [2]. The challenge is to automatically create these representations and enable mapping processes between information contained in the knowledge base and the service itself.

The Web Services themselves are developed and hosted within distributed Web Service Containers. Invocation of Web Services is achieved in conformance with the WSDL 1.0 specification [15] – this ensures that any compliant Web Service can be invoked by a peer irrespective of where the service is located, providing it has been conceptually described within a peer’s knowledge base.

Knowledge base servers reside on each Specialised-Peer, which ensures that the knowledge within the peer network is not centralised. This requirement allows us to move away from centralised registries such as JINI and UDDI and enables Web Service descriptions to be distributed within knowledge bases throughout Specialised-Peers connected to the P2P network. This ensures that there is no central point of failure and that service descriptions are distributed throughout the network – if a peer becomes unavailable you only lose the services on that peer.

We believe this framework to be very flexible and it provides distinct advantages over other areas of research such as the Service Location Protocol (SLP) [5] and Avancha, S. et al. [1]. Our framework has the ability to extend its reach outside an organisational LAN and access services contained within a peer network and semantically match queries with distributed services. Furthermore the DiSUS framework is protocol agnostic and we envisage that we will be capable of enabling interoperability between different protocols such as Bluetooth, WiFi and HomeRF. Our understanding is that [5] and [1] are incapable of achieving this level of functionality.

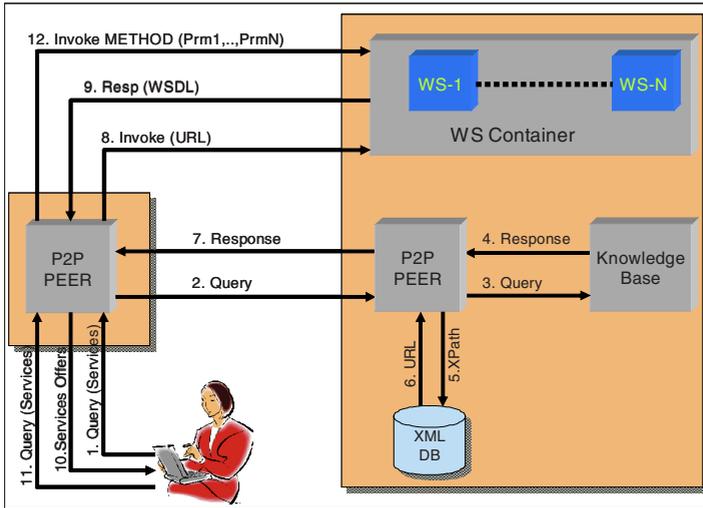


Fig. 2. The DiSUS Framework in operation

3.2 Algorithm and Protocol

This section describes the systematic interaction between subsystems in the DiSUS framework. It shows how services are semantically discovered, selected and invoked.

Fig. 2 describes the interaction between the user and two peer services. This process begins when the user selects a service category from the user interface. After selection a query message is retrieved from an XML configuration file and wrapped in a XML message. This message is propagated to all peers on the network. When a Specialised-Peer receives a message it extracts the queries from the XML message and passes it to its knowledge base to determine semantically if it has a service corresponding to the received query.

It does not matter what name we give to services because services are not retrieved by keyword but rather on their capabilities. Services that have the same capabilities, but are named differently should be conceptually equivalent, therefore they will be returned irrespective of what they are called.

If the peer has a service, the abstract name of the service is retrieved from the knowledge base and used to extract the Web Service’s relative URL from an XML file as illustrated in Fig. 2. We programmatically retrieve the IP address dynamically from the peer machine at runtime and construct the URL by combing the IP address and Port number with the relative URL extracted from the XML file.

Before a Web Service is invoked, the peer service retrieves the WSDL file and programmatically determines what functions the service offers. Typically, the functions might be displayed to the user who then selects the required function (method) they want to use, or the program might intelligently invoke the most suitable service. This is outside the scope of this research. Before the service method is invoked the user is prompted to enter the required parameters. The Web Service processes this method invocation and a query response is created and wrapped within an XML message. This message is returned to the peer that initiated the query and presented to the user.

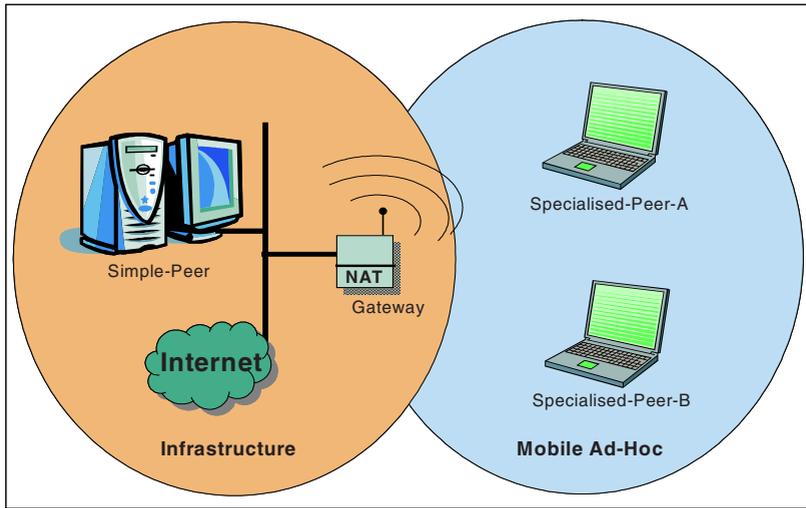


Fig. 3. DiSUS Prototype system configuration

3.3 Prototype

The prototype we have created hosts several Web Services, which are distributed and discovered within three Windows XP peers connected in a JXTA [16] P2P network on a Wireless LAN (802.11b). Two Notebooks were used and one Desktop (see Fig. 3).

Although the machines used within our prototype are fairly high powered we envisage deployment of our prototype to much smaller devices such as PDAs and Mobile Phones, however our current research has not addressed this issue.

All software was constructed using Java based on the 1.4 JRE. We chose three toolsets to construct our framework – JXTA was used to implement the P2P network, OpenCyc [10] was used to implement the knowledge base and Inference Engine, and GLUE was used to implement the Web Service container. In conjunction with the three toolsets described, the FIPA-ACL [11] standard was used to standardise message passing between peer services.

The test environment consists of two Specialized-Peers that host three Web Services each and provide OpenCyc services to semantically process queries received from the peer network. The third peer is a Simple-Peer and provides no services except the minimal functionality required, as described in section 3.

In the test environment, the Specialised-Peer-A hosts the following Web Services; PF Book service; the Restaurant Finder service and the Postcode Checker service. The Specialised-Peer-B hosts the AM Book service; the ABC Restaurant Finder service and the Thesaurus service.

The Web Service descriptions in our prototype are represented in the OpenCyc Knowledge Base using the built-in language provided by OpenCyc, which is CycL [10]. This is a language based on Description Logics and allows information to be represented taxonomically based on classifications obtained from specific domains. As well as a means of representing information, OpenCyc provides a built-in

Inference Engine, which enables peers to make inferences about information in order to determine semantically, what services it hosts, matches queries received from the peer network.

Queries are represented as CycL and have a similar syntax to Common Lisp. These queries are wrapped in FIPA-ACL messages, which are propagated around the peer network. When a peer receives a FIPA-ACL message it extracts the CycL query and sends it to the knowledge base using the OpenCyc API. This API is developed in Java and allows the peer to connect to the OpenCyc server and invoke a number of functions to directly manipulate the knowledge base.

Information retrieved from the peer's knowledge base will be the abstract names of URLs for services that conceptually match the query received. Service URLs are returned to the querying peer and once a peer service has the URL, the Web Service can be invoked. The peer service dynamically discovers the methods supported by the Web Service along with associated parameters by interrogating the WSDL file received when a binding to the Web Service is made. This is achieved by using the APIs provided by the GLUE [13] framework to interact and retrieve all required information from the WSDL file.

4 Conclusion and Future Work

In this paper we have described the DiSUS framework and explained how it can support unstructured services in Mobile Ad hoc Networks. We defined the requirements such a framework needs to address and described the prototype we developed. Within our research into the DiSUS framework our primary goal was to implement a prototype that could ground our ideas and provide a framework on which to build. We believe we have achieved this and have successfully implemented a solution that enables us to semantically discover distributed Web Services in an ad hoc network.

The construction phase required a great deal of manual work to conceptually create service descriptions and insert them into the knowledge base as well as formulating the queries used to conceptually discover and invoke services. Although we are currently working on algorithms to automate this process [2], it initially enabled us to develop a level of confidence that illustrated that our ideas are viable and uncovered key areas we need a better understanding of. This is very much a work in progress framework, which in this paper forms the base framework for future work.

Within this paper we successfully addressed a number of key issues - firstly we addressed the issue of semantically representing descriptions of Web Services and illustrated how these descriptions can be used to semantically determine if peers contain services that match the query being processed. Secondly we illustrated how unstructured services could be discovered within a dynamic environment devoid of any centralisation or any third party registries. We successfully embedded services within a distributed P2P network and proved that we could retrieve services without have any pre-knowledge of their existence and employ on-demand service invocation.

Within this paper we acknowledge that we need to address a number of issues. Firstly we need to better understand how we can conceptually match natural language queries with knowledge contained in a peer's knowledge base. We are currently addressing this issue and we are working on an algorithm that can determine the semantics of unstructured information using distributed dynamic ontologies. We

believe that our approach will provide an efficient means of addressing semantic interoperability between disparate knowledge structures. Secondly we need to automatically evolve the conceptual capabilities contained within the peer's knowledge base by collaboratively evolving knowledge structures based on a general consensus received from the peer network. We have in part, developed an algorithm to achieve this by using evolutionary programming techniques, however this is still a work in progress [3].

References

- [1] S. Avancha, Joshi, A., Finin, T., Enhancing the Bluetooth Service Discovery Protocol, University of Maryland Baltimore County: Baltimore, MD 21250, Technical Report, TR-CS-01-08, 2001.
- [2] P. Fergus, Mingkhwan, A., Merabti, M., Hannegham, M. Capturing Tacit Knowledge in P2P Networks, to appear in 4th EPSRC Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet2003). 2003. Liverpool, UK
- [3] P. Fergus, Mingkhwan, A., Merabti, M., Hannegham, M., Distributed Emergent Semantics in P2P, John Moores University: Liverpool, Technical Report, TR-CMS-2003-04-002, 2003.
- [4] The Gnutella Protocol Specification v0.4. 2001, Accessed: 23-05-2003, http://www9.livewire.com/developer/gnutella_protocol_0.4.pdf.
- [5] E. Guttman, Service Location Protocol Automatic Discovery of IP Network Services. IEEE Internet Computing, 1999. Vol. 3(No. 4): p. 71–80.
- [6] A. Maedche, Staab, S. Services on the Move - Towards P2P-Enabled Semantic Web Services, Proceedings of the 10th International Conference on Information Technology and Travel & Tourism. 2003. Helsinki: Springer, p. 124–133.
- [7] S. A. McIlraith, Son, T. C., Zeng, H., Semantic Web Services. IEEE Intelligent Systems, 2001. 16(2): p. 46–53.
- [8] A. Mingkhwan, Merabti, M., Askwith, B. Interoperability of Structured and Unstructured Services in Personal Mobility Information Space, European Wireless 2002. 2002. Florence, Italy
- [9] A. Mingkhwan, Merabti, M., Askwith, B., Hannegham, M. Global Wireless Framework, European Personal Mobile Communications Conference (EPMCC'03). 2003. Glasgow, Scotland
- [10] OpenCyc Project. 2002, Cycorp, Inc., Austin, TX, USA, Accessed: 5-02-03, <http://www.opencyc.org>.
- [11] M. Schalk, Liebig, T., Illmann, T., Kargl, F. Combining FIPA ACL With DAML+OIL - A Case Study, Proceedings of the Second International Workshop on Ontologies in Agent Systems. 2002. Bologna, Italy
- [12] JINI Technology. 2002, Accessed: 23-05-2003, <http://www.jini.org/>.
- [13] GLUE. 2003, The Mind Electric, Accessed: 5-02-03, <http://www.theminelectric.com>.
- [14] Framework for Peer-to-Peer Distributed Computing in a Heterogeneous, Decentralized Environment. 2002, Sun Microsystems, Inc., Accessed: 28th November 2002, <http://www.sun.com/software/jxta/mdejxta-paper.pdf>.
- [15] World Wide Web Consortium. 2003, W3C, Accessed: 5-02-03, <http://www.w3c.org>.
- [16] B. J. Wilson, (2002), "JXTA", New Riders Publishing, Indianapolis.