A Scalability Study of the MUSDAC Platform for Service Discovery in B3G Networks

Pierre-Guillaume Raverdy, Sebastien Armand, Valérie Issarny
INRIA-Rocquencourt, Domaine de Voluceau, 78153 Le Chesnay, France,
firstname.lastname@inria.fr

Abstract— **Beyond 3G (B3G) environments bring new challenges for mobile service platforms, as they strive to support new interaction models between mobile users and to leverage the services deployed in the environment. For users to discover and access the networked services, an innovative solution is required that overcomes both the heterogeneity of the runtime environments (and therefore protocols) of the clients and services, and the limited interconnectivity between the different networks available at a location. We proposed the MUSDAC middleware platform that interoperates with existing service discovery protocols and manages the dissemination of service information in dynamic multi-network compositions. In this paper, we present our assessment of the scalability of the MUSDAC platform, focusing on the dissemination of discovery requests. We in particular analyze the impact of the network dynamics, of the maximum hop number and of the network interconnectivity level.**

*Index Terms***— Interoperability, heterogeneous networks, scalability, service discovery,**

I. INTRODUCTION

The availability of consumer-oriented mobile devices powerful enough to host services, and the deployment of heterogeneous networks based on wireless networking technologies have enabled the emergence of service-rich computational (or B3G) environments aimed at supporting users in their daily life [13]. Such B3G environments bring new challenges for mobile service platforms, as they strive to support new interaction models between mobile users and leverage the services deployed in the environment. Discovering the networked services available in such environments is a crucial first step for mobile service platforms.

Many Service Discovery (SD) protocols [14] have already been proposed (e.g., SSDP, SLP, WS-Discovery and UDDI) for different networking environments (e.g., home networks, Internet) and have usually been integrated in middleware platforms (e.g., UPnP, Web Services platforms). Service discovery has however to be rethink for B3G networks due to their intrinsic heterogeneity:

- **Protocol heterogeneity**: the use of various middleware platforms (e.g., UPnP, Jini, Web services) results in the concurrent use of discovery and access protocols that do not directly interoperate with each other, further limiting the number of accessible services.
- **Network heterogeneity**: the use of various wireless technologies (e.g., cellular networks, WiFi, or Bluetooth) and network management models (e.g., ad hoc or infrastructure-based) results in many independent networks being available to users at a location. As users can only be connected to a limited number of networks at the same time (often a single one), many services may not be reachable (i.e., firewalls, private IP addresses, limited support of IP multicast…).

Several projects have investigated interoperability solutions for SDP [1][4][8][10], and primarily focused on home networks/LANs and multicast-based SD protocols such as SLP or UPnP. Two approaches [5] have been considered for such interoperability: (i) an explicit approach where clients directly access the interoperability layer that translate the client's request into the specific format of the supported SD protocols; (ii) an implicit approach where network messages are automatically translated between the different discovery domains. Clients and services are thus unaware of the translation process. While the transparent approach facilitates interoperability with legacy clients, the explicit approach enables the extension of existing SDPs with advanced features such as context awareness [6].

All these projects however consider interoperability within the same IP network. On the other side, several projects have investigated peering (gateways) and application-level routing for multi-network service discovery. They however investigated single-protocol solutions such as extension of LAN-oriented protocols (e.g., mSLP for SLP), federation of service registries (e.g., UDDIv3), or discovery protocols targeted at P2P networks [2][3][9]. In [7], a hybrid solution is presented that integrates UPnP and SDP for local discovery and INS/Twine for wide-area discovery. This project however and focus on a user-centric approach to B3G networks (i.e., pre-defined set of networks/devices) and assumes that global IP routing is supported in the environment.

We have developed the *MUlti-protocol Service Discovery and ACcess* (MUSDAC) middleware platform [11] to enable clients to discover and interact with services deployed in B3G environments. In MUSDAC, we model B3G environments as a dynamic composition of heterogeneous IP networks belonging to different administrative domains. We assume that global IP routing may not be available (i.e., a client in a network may not be able to directly communicate with a service in another network). Figure 1 illustrates such environment with users in a shopping mall being able to connect to various wireless networks deployed there (a MANET, a hotspot and a Bluetooth PAN) and being able to connect to their intranets or home networks through Internet or cellular networks. Services in these various networks are advertised using different SD protocols.

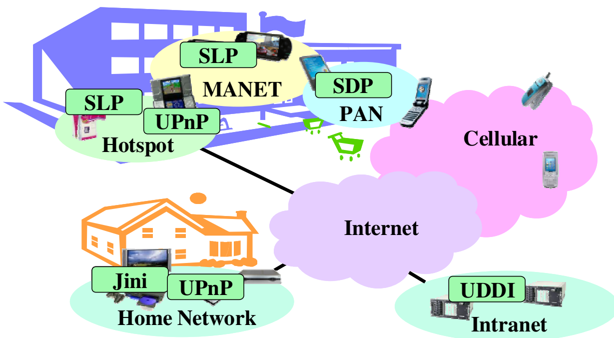

Figure 1: B3G Environment

In Section II, we present the MUSDAC platform and detail how service discovery is performed. We focus in particular on the scalability issues linked to the dissemination of discovery requests. In Section III, we report on the performance evaluation of the first prototype implementation. We then report in Section IV on the simulations we conducted to identify the influence of key parameters (e.g., number of bridges, maximum hop count) on the scalability of the MUSDAC platform. We then conclude in Section V.

II. The MUSDAC Platform

A. Platform overview

MUSDAC is a lightweight middleware platform that facilitates service discovery and access in B3G environments. MUSDAC is an additional layer on top of existing discovery and access protocols, and is provided as a service through existing discovery protocols. Clients explicitly use the APIs and service description format of the platform. This explicit approach allow the extension of existing SD protocols with new functionalities such as context-awareness [12].

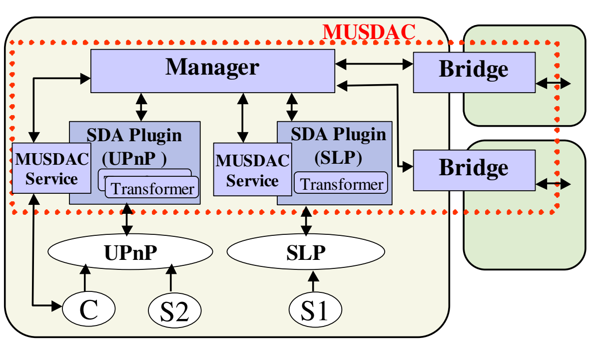

Figure 2: MUSDAC Platform

As shown in Figure 2, the MUSDAC platform includes the following components:

- The **Manager** that processes within a network the discovery and access requests from local and remote clients and manages the activation of hr other components.
- **Service Discovery and Access (SDA) Plugins** and **Transformers** that interact with specific discovery domains to collect service information and perform service access.
- **Bridges** that interconnect the networks accessed through the network interfaces of the device, and manage the dissemination of discovery information as well as the access to distant services.

B. MUSDAC deployment

The MUSDAC platform can be deployed on one or more devices in a network, and one of the MUSDAC instances is dynamically elected as the manager of the network. In the current prototype, we implemented the manager as a centralized component, as the size of a network is limited by nature (i.e., its number of devices and services in a subnet can be managed in a centralized fashion). The manager periodically sends presence beacons so that other MUSDAC instances in the network can detect its absence, as well as duplicates, and recover. The elected manager then activates SDA Plugins and transformers to interact with the local discovery domains. The manager may also dynamically select some of the MUSDAC instances in its network to act as bridges based on several criteria (i.e., connectivity to other networks, expected lifetime, processing power).

The main issue for the manager is indeed to select the appropriate bridges. An optimistic activation may lead to redundant bridges, and unnecessary duplication of discovery messages. On the contrary, a too cautious activation may lead to weak interconnectivity between networks and therefore only partial accessibility to the services offered in the environment.

C. Service Discovery in MUSDAC

In MUSDAC, services are described using the *MUSDAC Description* format, which is a generic and modular service description format. This format serves as a canvas for the discovery requests issued by clients, and for discovery responses returned by SDA Plugins and managers.

MUSDAC-aware clients first discover the MUSDAC service in their discovery domain (In Figure 2, the UPnP client uses SSDP to discover the UPnP MUSDAC service). Clients interact with the MUSDAC service to perform service discovery in the B3G environment. The discovery request is first processed by the local manager, which forwards it to local SDA Plugins and active bridges:

- Each SDA Plugin either (i) translates the request in the SD protocol-specific format and executes it (pull-based SD protocols), or (ii) matches the request against its cache of service descriptions (push-based SD protocols).
- Bridges forward incoming MUSDAC requests on their other network interfaces. Managers on the

other networks will then process the remote request as a local one.

For each service matching a request, a MUSDAC description of the service is returned to the manager of the client. While the client may collect all results after a given timeout, it can also collect partial results over time as different SD protocols display noticeably different response times (e.g., from 1ms for SLP to up to 6s for UDDI).

D. *Propagation path and dynamic routing*

In MUSDAC, bridges append on the fly propagation information to the discovery requests that they forward. This propagation information consists of the identifier of the bridge, as well as the identifiers of the source and destination networks. The *propagation path* (the list of propagation information added by the bridges) is primarily used to return discovery responses to the client's network following the reverse path. Indeed, one of the primary concerns in MUSDAC was to limit the processing and communication overhead on the bridging devices, as mobile devices are usually resource-constrained (CPU and battery). Using the (static) propagation path embedded in the messages enables a quick and inexpensive processing of requests and responses by bridges (i.e., no routing tables and associated processing/communication overhead). If a bridge in the propagation path is not available anymore when the response is returned (i.e., the device disappeared), the response is simply discarded. However, as discovery responses are returned within a few seconds of the requests, we can safely assume that the same bridges will (usually) be available.

In the current prototype implementation, the propagation path is also used to route access messages between the client and service's networks. A client is thus only able to access a service it discovered as long as all the bridges in the propagation path remain active. While this scheme limits the processing overhead on bridges, it also clearly limits the accessibility to distant services. Indeed, propagation paths are unlikely to stay valid for a long period of time due to the dynamic nature of B3G networks. A solution is for managers to dynamically determine a path to route access messages to distant networks based on the topology information collected from the propagation paths in the discovery requests. This solution has the advantage of still limiting the overhead on bridges. It however requires the dissemination of enough discovery requests (in terms of frequency) so that each manager has complete and up-to-date information on the B3G network topology. One goal of the scalability assessment presented in this paper is to evaluate whether such semi-dynamic routing of access messages is adequate, or if dynamic routing on the bridges should be implemented.

III. PROTOTYPE EVALUATION

We implemented a first prototype of the MUSDAC middleware platform in Java and released it as Open Source Software (http://www-rocq.inria.fr/arles/download/ubisec/). This prototype supports several standard SD protocols (e.g., UPnP, SLP, UDDI) as well as SD protocols for ad hoc networks and MANETs. The current implementation only supports multi-network access for SOAP services, with local SOAP endpoints being managed by the MUSDAC service on behalf of distant services. SOAP messages sent by a local client to this endpoint are encapsulated in a MUSDAC access message and are routed by the bridges to the destination network. The relevant SDA plugin then extracts the SOAP message and sends it to the service endpoint on behalf of the client.

In [11], we evaluated the MUSDAC platform overhead when performing service discovery and access in multi-network configurations. To summarize the results, the main platform overhead is related to the cost of accessing the MUSDAC Service (from a few 10's of ms for its XML implementation to a few 100's ms for its SOAP implementation), while the overhead associated with the processing and dissemination of discovery requests or access messages through bridges is limited. The propagation overhead is around 30 ms per bridge for discovery messages and 20 ms per bridge for access messages.

While the MUSDAC overhead for performing service discovery is important when compared to simple network-level SD protocols such as SLP (service discovery performed in less than 1 ms), it is comparable to the UPnP service discovery time, and is small when considering complex protocols such as UDDI (up to 6s when a client sends a request for the first time). As a client searching for services in the B3G environment will wait for all SD protocols to respond, and therefore wait for the slowest one, the MUSDAC overhead is not detrimental.

IV. SCALABILITY ASSESSMENT

Following the initial prototype evaluation, we are now investigating scalability issues of the MUSDAC platform. In this paper we in particular focus on the efficient dissemination of discovery requests: identifying the relation between the number of bridges, the maximum hop count and the network dynamics when trying efficiently to (i) reach the maximum number of networks in the B3G environment and (ii) ensuring the longest accessibility to distant services.

A. *Simulation environment*

Our simulation environment is based on an extended version of JNS[1], which is a partial Java implementation of ns-2. In particular, we extended JNS to support nodes with multiple network interfaces and to allow the dynamic activation and configuration of these network interfaces. This simulation environment enables us to carry out our simulations directly with the code from the prototype implementation of the MUSDAC platform.

For our simulations, we consider an environment with 10 networks (WiFi) with different percentages of MUSDAC-enabled devices (see Table 1), which influence the potential for interconnection. If not noted otherwise, simulations were carried out using 2 high, 3 medium, and 5 small networks. All networks have the same emission rate for discovery requests. We also consider that all networks are in radio range of each other (i.e., a bridge may connect any two or more networks) and that multiple bridges may interconnect the same networks.

[1] http://jns.sourceforge.net/

	MUSDAC nodes	
	Minimum	Maximum
High	8	16
Medium	6	10
Small	3	6

Table 1: Simulation network types

While many models have been proposed for user mobility or generic network traffic in MANETs, we are not aware of any model for B3G networks that would address in particular the dynamic interconnection between heterogeneous networks. Furthermore, we are also not aware of any model or experimental traces focusing on the network traffic pattern associated to service discovery protocols (e,g,, rates of discovery requests in typical home networks or intranets).

In our experiments, we vary the node arrival rate (i.e., network stability), the bridge creation rate (i.e., interconnectivity between networks), and the maximum number of hops used in the dissemination of discovery requests (i.e., network reachability). These parameters influence, among others, the stability of the paths between a client and a service and the overhead sustained by managers and bridges.

Each simulation lasted for 15 minutes, following an initial period of stabilization. Each simulation was repeated 3 times and the median of the 3 runs is presented.

B. *Network reachability*

We first investigated the combined effects of the interconnection level (i.e., number of bridges per network) and the number of hops in the reachability of distant networks (See Figure 3). In our simulation environment, the dominant factor is the number of active bridges per networks over the maximum hop count. With a limited number of bridges (1.37, 1.91 and 2.14 bridges on average per network), a discovery request is only able to reach 3-4 networks at 2 hops, compared to 4-6 networks at 5 hops (out of a maximum of 9). A slightly higher level of interconnectivity (between 2.81 and 3.81 bridges per network) provides a sufficient reachability. To guarantee a complete reachability, a high degree of interconnectivity is however required (5.83 bridges per network).

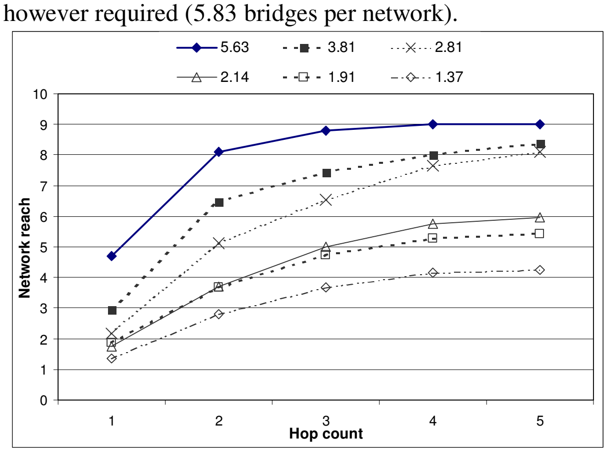

Figure 3: Network reachability

In our simulations, the network reachability is almost stable after 3 hops. It should however be noted that in realistic scenarios, the number of overlapping networks will be limited. The number of hops may then become more important in order to reach nearby networks.

C. *Propagation paths duration*

As detailed in Section II.D, service access messages in the current MUSDAC prototype are routed to the targeted service according to the propagation path returned in the service discovery response (i.e., static routing based on information at the time of the discovery request). The major drawback of this approach is that in a dynamic network environment, clients may only be able to access a service for a short time.

We first evaluated (See Table 2) the average duration of propagation paths (from 1 to 5 hops) depending on the network dynamics (from a renewal rate of 0.6 nodes/minute for slow networks to 3.5 nodes/min and 7.5 nodes/minute for medium and dynamic networks). For the three configurations, the interconnectivity between networks was low.

We then evaluated the duration of a session (i.e., time interval with continuous connectivity between two networks potentially using various paths) for the same configurations. As can be observed in Table 3, the average session duration is significantly superior to the average duration of a propagation path, and is slightly superior to the duration of a 1-hop propagation path. Indeed, as the interconnectivity level between networks is low, it is unlikely to have multiple bridges linking the same 2 networks. We thus evaluated the session duration for the slow environment where the interconnectivity level between networks is increased (from 1.4 to 2.8 bridges per network). As the average number of bridges increases, the average session duration significantly increases (cf. Slow* configuration).

	Average Bridge# per network	Propagation path duration (s)				
		1	2	3	4	5
Slow	1.4	224	168	139	119	93
Standard	1.9	100	53	37	29	20
Dynamic	2.0	61	30	21	16	13

Table 2: Static path duration

It should be noted that the duration of the propagation paths and the sessions represents an upper boundary for service access, as a client may issue a service discovery request at any time.

	Average session duration (s)	Average path duration (s)
Slow	269	139
Standard	112	37
Dynamic	71	21
Slow*	665	227

Table 3: Session duration

D. *Maximum hops*

We then evaluated the influence of the maximum number of hops on the load sustained by bridges and managers. As the number of reachable networks increases slowly after 3 hops (See Figure 3), does the number of messages processed by the managers and bridges also stabilize?

In a standard environment (renewal rate of 3.3 nodes/min and manager request rate of 2.9/min) with a low interconnectivity (2 bridges/network), the number of messages to be processed by managers and bridges (See Table 4) significantly increases with each hop, even after 3 hops, while the session duration remains low. Increasing the maximum hop number from 2 to 4 increases the number of messages by 97% for managers and by 229% for bridges. However, the session duration is only increased by 10% due to the low interconnectivity.

Max hops	Average Bridge# per network	Manager messages (mess/min)	Bridges messages (mess/min)
5	1.9	46.6	62.9
4	2.0	35.3	53.6
3	2.2	19.8	23.7
2	2.1	17.9	15.8

Table 4: Maximum hops and processing overhead

On the contrary, increasing the maximum hop number in a dynamic environment with better interconnectivity (7.0 nodes/min, 4 bridges/network) significantly increases the average session duration (See Table 5), as more alternative paths are available when a bridge disappears while the increase of messages for managers and bridges is negligible (resp. 1% and 13.7%) as most networks are accessed within a short hop count.

	Standard network (3.3 nodes/min)	Dynamic network (7.0 nodes/min)
Session duration	+10%	+100%
Manager messages	+97%	+1%
Bridge messages	+229%	+13.7%

Table 5: Effects of maximum hop variations (2 to 4 hops)

Managers and bridges may therefore dynamically adapt the maximum hop number of discovery requests depending on the topology of the B3G networks to quickly optimize the use of the processing and networking resources, as changing the interconnectivity level (e.g., controlling the configuration of the devices' network interfaces) may not always be possible.

E. *Bridge and manager overhead*

We investigated the influence of the network interconnectivity level on the processing overhead sustained by bridges and managers (i.e., amount of discovery requests to process). In Table 6, the rates of incoming discovery request are provided for 3 configurations and a maximum hop count of 4 (See Figure 4 for average bridge number and reachability for the three configurations). In all simulations, each manager issued on average 8.1 discovery requests per minute.

Interconnectivity (bridge#/netw.)	Manager messages (mess/min)	Bridges messages (mess/min)
Low (1.3)	19.7	14.9
Medium (2.6)	29.2	16.1
High (3.9)	39.9	15.1

Table 6: Manager and bridge message rates (with duplication)

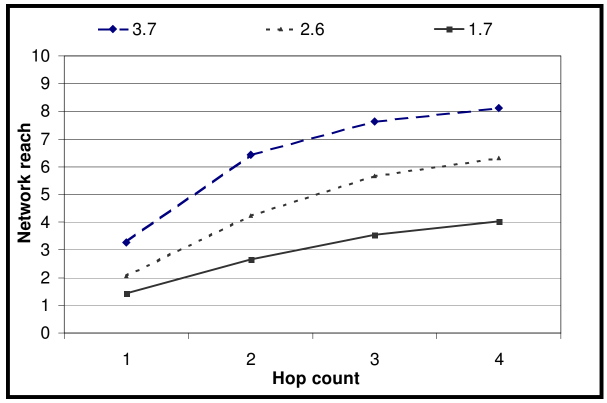

Figure 4: Network reachability

It can be observed that the number of messages processed by each manager greatly increases with the number of bridges. Indeed, in the current prototype implementation, discovery requests are sent to local bridges using multicast, and all bridges forward the request unless the target network is already in the request's propagation path or if the bridge already received the request from an alternative path (i.e., bridges maintain a log of recent request IDs). However, if two bridges interconnect with the same two networks, a discovery request issued on one network will be forwarded twice to the destination network as each bridge processes the request independently and simultaneously. As the number of bridges per network increases, such duplication problem becomes apparent.

Interconnectivity (bridge#/netw.)	Manager messages (mess/min)	Bridges messages (mess/min)
Low (1.7)	19.9	15.2
Medium (2.6)	25.3	13.2
High (3.7)	30.8	11.9

Table 7: Manager and bridge message rates (no duplication)

We thus improved the MUSDAC platform to prevent such duplication without requiring coordination. As bridges are already aware of each other in a network through presence beacons, they can detect such duplicate interconnection. Only one bridge (the one with the lowest bridge ID) will then forward the request. In Table 7, the same simulations were conducted with the new forwarding policy on the bridges, and it can be observed that the number of remote discovery requests processed by the manager is significantly reduced for high-interconnectivity environments.

An important side-effect of this new policy is the implicit global load balancing of the discovery requests across the bridges, and therefore the reduced number of messages that each bridge has to process.

V. Conclusion

We proposed the *MUlti-protocol Service Discovery and ACcess* (MUSDAC) middleware platform to enable mobile users to discover and interact with services provided in the environment. The MUSDAC platform overcomes the network and protocol heterogeneity inherent in B3G networks by providing an interoperability layer on top of existing SD protocols, and by managing the dynamic association between independent networks and the dissemination of discovery information. Our solution is fully transparent to the deployed services, and only requires the client applications to have knowledge of the MUSDAC Service. Through this service, client applications are able to discover and access services that use different SD protocols and/or are located in different networks. An initial prototype of MUSDAC has been implemented and several SDPs have been integrated into the platform.

In this paper, we presented our assessment of the scalability of the MUSDAC platform, focusing on the dissemination of discovery requests. In particular, we analyzed the impact of the network dynamics, of the maximum hop number and of the network interconnectivity level on the network reachability, session time, and processing overhead for the managers and bridges. By assessing the scalability of the MUSDAC platform with JNS (i.e., using the actual code of the prototype implementation), we have been able to detect and quickly correct various duplication problems.

To truly benefit users, the MUSDAC platform must be deployed on a large number of devices, acting as bridges whenever possible. While we can use the maximum hop number to dynamically adjust the overhead of the MUSDAC platform or temporarily activate/deactivate bridges to ensure a given interconnectivity level, the direct management of the network interfaces of mobile devices (e.g., dynamically activating interfaces or changing network association) may lead to instability (e.g., sudden flooding of a bridge/network due to a shorter propagation path). Changes of the network interface configuration may also conflict with other communicating applications on the device.

While the initial evaluation of the scalability of the MUSDAC platform focused on discovery requests, we now intend to investigate bridge activation policies for complex multi-network configurations, as well as policies for selecting the service access path (when multiple are available) and taking into account the networks' bandwidth, latency, processing and cost. A key objective for such an evaluation will be to refine our models for B3G networks and for service discovery traffic pattern.

Acknowledgements

This work described herein is funded by the European Commission under the FP6 IST Project Plastic contract number 026955.

References

[1] J. Allard, V. Chinta, S. Gundala, and G. Richard, "Jini Meets UPnP: An Architecture for Jini/UPnP Interoperability," In Proceedings of the 2003 International Symposium on Applications and the Internet (SAINT 2003).

[2] K. Arabshian and H. Schulzrinne. “Gloserv: Global service discovery architecture”. In First Intl. Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous), August. 2004.

[3] M. Balazinska, H. Balakrishnan, and D. Karger, “INS/Twine: A scalable peer-to-peer architecture for intentional resource discovery,” First International Conference on Pervasive Computing, Zurich, Switzerland, August 2002.

[4] Y.D. Bromberg, V. Issarny, "INDISS: Interoperable Discovery System for Networked Services", In Proc. of ACM/IFIP/USENIX 6th International Middleware Conference (Middleware'05). Grenoble, France, 2005.

[5] Y.-D. Bromberg, V. Issarny, P.-G. Raverdy. Interoperability of Service Discovery Protocols: Transparent versus Explicit Approaches. In Proceedings of the 15th IST Mobile & Wireless Communications Summit. June 2006, Myconos, Greece.

[6] A. K. Dey, G. D. Abowd, D. Salber “A context-based infrastructure for smart environments”, In Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE '99), 1999.

[7] M. Ghader, R. L. Olsen, V. Prasad, M. Jacobsen, et *al* “Service Discovey in Personal Networks; design, implementation and analysis”, In Proceedings of the 15th IST Mobile & Wireless Communications Summit. June 2006, Myconos, Greece.

[8] P. Grace, G. S. Blair and S. Samuel. "A Reflective Framework for Discovery and Interaction in Heterogeneous Mobile Environments". ACM SIGMOBILE Mobile Computing and Communications Review, 9(1), pp 2-14, special section on Discovery and Interaction of Mobile Services, January 2005

[9] D. Gribble, M. Welsh, R. von Behren, E. Brewer, D. Culler, N. Borisov, S. Czerwinski and R. Gummadi. “The ninja architecture for robust internet-scale systems and services” Computer Networks, 35(4):473–497, Mar. 2001.

[10] T. Koponen and T. Virtanen, "A Service Discovery: A Service Broker Approach" In Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04), January 2004

[11] P.-G. Raverdy, V. Issarny, R. Chibout, A. de La Chapelle. A Multi-Protocol Approach to Service Discovery and Access in Pervasive Environments. In Proceedings of MOBIQUITOUS - The 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services. July 2006, San Jose, CA, USA.

[12] P.-G. Raverdy O. Riva, R. Chibout, A. de La Chapelle and V. Issarny, "Efficient Context-aware Service Discovery in Multi-Protocol Pervasive Environments", In Proceedings of IEEE International Conference on Mobile Data Management (MDM), 2006.

[13] M. Weiser, The Computer for the 21st Century. Scientific American, 256, 3(Sept.), Pages 94-104.

[14] F. Zhu, M. Mutka, L. M. Ni, "Service Discovery in Pervasive Computing Environments", Pervasive Computing, IEEE Volume 4, Issue 4, Oct. Dec. 2005 pp: 81–90