Service-oriented Network Selection

Patrick Mandic
Rechenzentrum Universitaet Stuttgart
Stuttgart, Germany
Email: patrick@mandic.com

***Abstract*—The trend of merging telecommunication infrastructures with traditional IT infrastructures is becoming increasingly more common. The driver behind this development is partly the strong need for enhanced services from the telecom operator's side aiming for value added services provision to the customer. The effect of this new trend is that existing value chains in the Telco infrastructure will disappear and the merge between the Telco infrastructure and the IT infrastructure can bring added value to the Telco's network, which is needed in order to increase the willingness to pay for the services. Thus, Telecom operators are very keen on participating in the overall service delivery chain being placed on top of their transport infrastructure. This paper proposes a method, to help to propel the previously mentioned trend by enabling the user to query for aggregated composed services directly from the network edge. A network selection is therefore achived based on the services provided by the access network and their capabilities and costs, parallel to other traditional pure network-related parameters such as signal strength and network QoS.**

I. Introduction

Not so long ago telecom operators and ISP providers were practically the only dominant players of the telecommunication market. In the past decade, however, even after the Internet bubble crashed, other players have established themselves in more than a conformable position obtaining great added value from services provided over the telecommunication paths provided by the formers [1]. With the unavoidable sinking prices of the telecommunication industry in a market on the verge of saturation [2], the future does not look particularly rosy for the telecom operators. This problem has been spotted and solutions have been given in certain areas like in [5], restricted to services delivered over a SIP platform.

With the advent of pervasive computing and the collision between cell-phone networks and the current Internet, the new Internet will also have to contemplate, assimilate and make use of other factors that were superfluous or non existant in the old static Internet such as user-context handling, terminal capabilities, mobility, unavailability, bandwidth management, multihoming and many others. In order to improve, personalize and optimize the delivery of contents. These techniques have been studied in several projects such as Akogrimo [3] and Daidalos [4] among others.

It is important to notice that the first step a mobile user needs to take when using a service is to select which network to use. Networks of different types are becoming nowadays more and more ubiquitous and devices tend to provide more and more interfaces to different access technologies. It is not unordinary to realize that in certain locations more than five networks are available and even more of them in busy areas. Currently the only factor that influences the user selection of a network is the price. The operators have difficulties differentiation their network from other networks. Most operators offer binding contracts with extremely complicated price plans to achieve differentiation - a sign that the market has reached saturation. A possible way out if this decline is to introduce methods for service oriented network selection. A major driver for the selection of a concrete network in front of the other would then be the need for a certain service. Currently when a user wants to use a service over the network he purchases the network access and afterwards, she might pay extra for the service to use, e.g. a pay-per-view movie. Just to realize that the service costs more over one network than the other, does not offer any kind of real-time QoS and the download speed is so slow that 24h hours will not be enough to let the customer enjoy the movie. This signifies the loss of this customer for the provider. This work proposes an architecture to provide network differentiation by more than just pricing by providing a vertical service composition architecture and focusing on the service-based selection of the network.

This paper is structured in the description of the value chain of the previously described scenario. A description of the anatomy of service discovery is explained in Section III, followed by an analysis of the service usage bootstrap. Section IV describes the service oriented network selection. Finally some conclusions are extracted.

II. Providing Value Added Services in future commercial network infrastructures

Division of labour is a concept which has been proven to allow more efficient productive processes in various branches of our society. The telecommunication market can exploit this classic approach. Whereas during the Internet bubble at the beginning of this century, the Telco operators strategy was to widen the portfolio and extend their data transport services with high value content. On the other hand one of the bigger risks for the operators are the IT organizations that play a central role while the data transport is being relegated to a second plane and great investments such as the UMTS licenses can not be justified. In-house service provisioning offered by Telcos do not seem to be able to compete with specialized service IT organizations and therefore should opt for the division of labour playing the role of integrators and offering interfaces for network-related aspects to the services on top of it. In order to use this interfaces, service providers need to have some kind of agreement with the Telco, that might be created dynamically. The role of integrator and service composer can basically be taken over by any stakeholder, however, is is clear that Telco's owning a large customer database and maintain a trust relationship to the customers are in a very comfortable position to take over that role.

In a dynamic environment where each service is offered on a market, integrators can compose different services into a super service that might or might not look as an atomic service, to the end-user, but is able to query its needs for the integrator to be able to integrate the different services and provide the users' wishes.

In any case, the generic scenario underlines the need for a very comprehensive concept for service description, service discovery and cross-layer service composition that contemplates the necessity of a user discovering these services on the move from the very edge of the network thus being able to compare different alternatives dynamically.

III. Service Discovery Overview

The need for the introduction of semantics in services has become a clear target in the world of Web Services (WS), backed-up by ontologies based on e.g. OWL-S [16] or WSML [10] in order to provide discovery agents with the ability of reasoning and automatic compose services. This goal has been achieved in "controlled" environments to accomplish very specific tasks. Parallel to this, traditional network discovery methods such as SLP or mDNS are characterized by their static semantics. Recently rather light efforts are being put into giving them more versatility by using some of the methods in which semantic WSs are based. The environment of application is however still limited to traditional network services. The application of services in a Next Generation Network scenario brings along new challenges due to strong cross-layer interaction such as real-time requirements, roaming across domains, mobility and so forth.

A. Service Discovery building blocks

Service discovery mechanisms are typically composed of three basic building blocks which are: the service requestor, the service provider and the directory (or repository) where information about the published service information is stored. This directory does not always exist in all mechanisms, e.g. UPnP and usually mDNS do not make use of it, these systems are useful in concrete for local networks but are not manageable in infrastructure mode, where the use of a directory is practically mandatory.

The high level choreography of a service discovery system is as follows. Service providers typically publish the information of the services they provide or, more rarely, some other entity does it on their behalf (which will not be consider in this work). The directory assimilates and processes the information (or refuses it). On the next step the service requestor queries for a certain service on the directory and the description of the matching services are conveyed. Afterwards, the service requestor chooses one or several of them and sometimes triggers a negotiation with it before finally using the most suitable one. In this architecture we can differentiate on one side the way messages are sent - in particular if they follow a query-response, a broadcast or a subscription notification paradigm - and the service query and matching logic on top of it in the other. The latter comprises the following fields:

- **Service Description:** The way services are described is one of the fundamental parts of the system, and generally imposes restrictions onto other parts of the architecture built around it. The description semantics can be simple and efficient (but then again limited) such as in the case of SLP where just attribute-value pairs are used, or complicated and powerful but yet harder to implement and apply as the new semantic WS proposals are.
  The service description published by the service provider in the directory can be of different natures: it might be complete in which case the service requestor can automatically contact the service provider after evaluating the different results; it might only contain certain parts of the description and the rest such as grounding[1] is obtained afterwards; or the last possibility is that the service is described with very high level semantics and after one candidate has been chosen a more concrete and possibly particular service description is provided or negotiated. The attributes a service description describes are of different natures. Examples of information contained in a description are: The grounding or description of how to use the service, the owner or service provider, the requirements that are necessary to use the service (e.g. software, or protocol), description about what the service does, its cost, etc...
- **Query matching**: The core function of a service discovery system is the matching between a specification of the service searched for and the description of the available services. The specification of the service searched for can have different forms: the simplest one is a simple keyword that is matched against all service descriptions, this usually returns a large number of results with low accuracy. Another method is to use a filter, e.g. using Xpath or similar, this already provides means for search based structured information and e.g. a distinction between functional and non-functional parts of the service description can be made. A more complex but powerful method is to provide ontology-structured descriptions of abstract services that form a knowledge base that can be queried. The resulting services can then be instantiated at a later point in time.
- **Filtering:** Results need to be filtered according to the privacy preferences imposed by the service provider and the general policies that might apply. This implies the restriction to or exposition of different information to different users. This feature is rarely supported by any standard service discovery mechanism, and when it does

[1] Grounding is understood as the functional description of how a service needs to be contacted

it, it is in a very simplistic way.

- **Ranking of results:** Once a query has been performed the matching results are delivered. These results can be divided in different degrees of matching, depending if they fully match the requestors query or they only approximate. These degrees range from exact matching, plug-in, subsumtion, intersection or fail [6]. In addition, inside these groups the results can be further sorted taking into account parameters, such as the user profile, user context, popularity... or implicit semantics such as patterns or relative frequencies of terms. This can be done by either the directory, the service requestor or both. However it is recommended that the directory produces a ranking and limits the amount of results to the deemed necessary, especially in the area of wireless or low bandwidth networks.
  In order to use a service, the basic steps are typically discovery of the service, SLA negotiation and finally the use of the service.

IV. The discovery bootstrap

Considering the scenario proposed in Section II, before being able to use any type of service, the user needs to have network access in order to use the service or the collection of services requested. The necessary steps that need to be take between the moment the user - or the logic in the mobile device - decides the need for a service and just before the execution of the service is defined as the service bootstrap. The service bootstrap is divided in the layer shown in Fig IV. In these layers the following actions take place in the traditional approach.

1) User locates available networks.
2) User selects a network according to the name or the signal strength.
3) The terminal needs to be configured at IP layer as well as with information about local infrastructure services, e.g. authentication authority (if authentication takes place at IP level), DNS, local Service Directory and so forth.
4) A service query is performed, either by using the service directory obtained in step 3 or by using some king of multicast or broadcast depending on the technology used.
5a) Repeat the same with other networks if service was not found or to compare with other services.
5b) Required service found and ready to be used.

This three layered procedure shows the typical approach that is most extensively used currently. Needless to say that this approach is far from being efficient. In order to use a service a user needs to fully configure it's network capabilities and the service directory (higher layer in citefig:NetSelect) before a query can be performed. This is obviously inefficient if one wants to compare the same service in different networks or has to try one network after the other.

The three layers can be flattened in to two layers by giving the user the information that is supposed to be given in the higher layer already in the middle one. Possible solutions include the use, in IPv6, of additional information in the Router Advertisements, which is however very limited and implies a high bandwidth cost due to the constant retransmission. DHCP would reduce the amount of bandwidth but it still would need to send all the networks information everytime when requested. A better solution would be using a srv records mDNS-based system or a tailored multicast method to give the user some information about the network services. While the cost of the connection establishment has been reduced to some degree in this case, the user needs to obtain link access in every probed network.

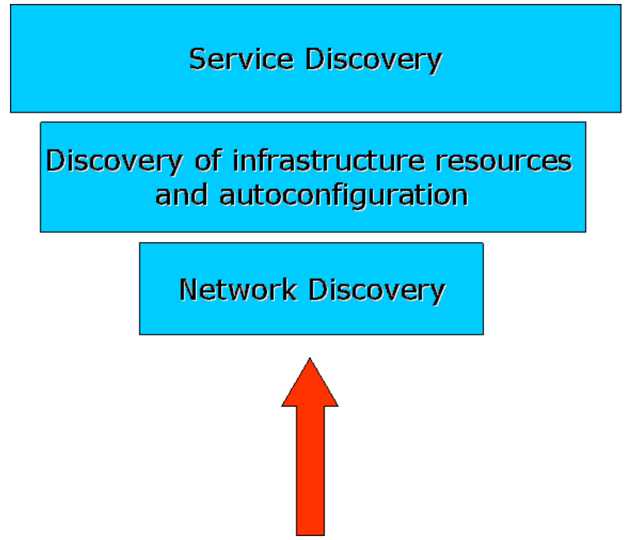

Fig. 1. Discovery bootstrap

One could go further and flatten the original three layers of the service discovery process into just one, by allowing queries already at L2 of the OSI layer. However, this method imposes the problem of having an access technology dependent system, and that is why in the Akogrimo project a simple solution based on mDNS was followed. The 802.21 standard strives to facilitate handovers among different access technologies, which allows an interface at L2 towards the access technology independence that we are seeking.

V. The 802.21 Standard

The 802.21 standard [9] is being developed to permit the long time promised seamless handover between different access technologies. 802.21 works at L2 can collaborates with other protocols at higher layer such as mobile IP enabling so-called fast-handover. This requires a common standard that needs to be implemented accross all 802 media. This protocol helps in the network selection decision process which up to now was practically restricted to signal strength and network name (e.g. in 802.11). To this end several other parameters are independent to improve the quality of the decision taken, such as QoS, costs, user preferences and network operators' policies. Network selection for non-handover purposes is not a direct goal of the standard, but the underlying mechanisms can be reused in other cases such as network selection based on the service to be consumed. That is to say, choosing that provides a certain service or where the network plus the service costs are lower. 802.21 is conceptually composed of three layers: MIH Users, MIH Function (MIHF), and the lower layers. The MIH Users layer are IP, Mobile IP, applications and so on that contact the MIHF in order to fetch information or make decisions. The MIH protocol defines the messages that are exchanged between MIHF in different hosts, which do not necessarily have to be sharing the same media and can communicate over L3 with other entities. The MIHF also communicates with the lower layer which is the corresponding access technologies used (802.11, 802.16, 802.3 and so on). Some services are defined in MIHF that can be used by upper layers such as the Media Independent Event Service (MIES), which provides link related characteristics, the Media Independent Command Service (MICS), which allows control over link behaviour and the Media Independent Information Service (MIIS), which provides information about the network and surrounding networks and services offered by them such as QoS, network security, roaming agreements or connecting costs. To query this information two approaches are possible, a binary and a XML/RDF-based approach. The XML/RDF queries are made in the standard fashion using SPARQL.

RDFS provides enough means for structured information for a basic ontology but is not good enough for reasoning and composed workflows. Usually a service is wrapped in a workflow of services linked to it that might be dynamically instantiated: E.g. I want to buy a movie. The current approach would imply that one would find that a certain network provides a service called iTube that most ”probably” will enable us to download a video. However, when one connects to the network they happen to provide a video format which is to big for your handheld and need to by transcoding services additionally and look for them manually. An approach to discover a workflow linked to a goal you want to accomplish would have resolved this problem with a simple query offering us the best price taking into account the whole chain of service interactions.

VI. Service-oriented Network selection

As it was explained in previous sections a telecommunication market trend that has been starting to gestate for a few years now is the provisioning of value added services directly from the network provider, which offers them in some cases as a bundle. However, relations between the different parts of the value-chain are static and so is the relation between the network user and the network provider. A flexible value chain where services at each level can be composed on-the-fly would increase range and number of supplied services. A mechanism through which a customer can select services to use directly before connecting to the network would incentivate and facilitate the consumption of these services.

RDF is a good language for basic description of knowledge, however its capabilities are very limited when it comes to description of ontologies in the wider sense, and specially and most important to enable reasoning and knowledge inference. OWL extends RDF-S and providing more description expresivity and reasoning, since it enables to express disjointness, transitiveness, inversal properties, and so on. 802.21 provides extensibility allowing a basic standardized schema and a extended schema that is individually designed by each network operator taking into account the peculiarities of their networks and downloaded upon request by the user. XML allows very easy extensibility and the use of OWL in this context is straight forward.

OWL defines a good framework to create ontologies and to enable reasoning but some additions must be made for the concrete case of the description of services. To this end, OWL-S provides a specific ontology to describe services. Simple atomic services as well as complex composite services are contemplated. A service is described by a service profile, which describes what the service does; a service model, which details how to use the service and what is needed before the service can be carried out and what happens afterwards; a service grounding describes how to access the service in terms of e.g. protocol, message formats, ports. One of the drawbacks of OWL-S is that the grounding has to be expressed in WSDL imposing that all services need to be Web Services or at least that they need to have a Web Service interface. This restricts greatly the range of services that can be used.

On the other hand, thanks to the alignment of efforts among several European research projects, the Web Service Modeling Ontology (WSMO) [18], presents an ontology for semantic web services supported by a development framework and a formal language, the Web Service Modeling Language (WSML). WSMO is based on four top level elements: ontologies, services, goals and mediators. Ontologies provide the terminology to describe a conceptualized domain. Goals describe the wish of the service requestor regarding the functionality of a service. Mediators are used to cleanly resolve incompatibilities at either data, protocol or process level. At last, services, are the services that can be provided or requested by service requesters or providers. One advantage of WSMO is that the concrete grounding service used is not restricted to Web Services, the type of service used is kept transparent. In addition, composed services are seen as any other service and there is no distinction between atomic and composed services such as in OWL-S, which enables several layers of composition such as the ones in Fig. XX. Although traditionally Web Services are as the concrete grounding of the services, it does not always seem to be the case in the area of general services provided by a service provider through a network provider. The use of this framework in 802.21 is not as straight forward as with OWL-S, however it still can be expressed in XML.

As mentioned before in this work, users' consumption of resources implies much more than the use of simple atomic services. Usually the service is linked to complementary services, that are needed in order for it to work. Such as purchasing a video stream implies network access, a certain QoS to that supports the streaming characteristics a payment method, and so forth. These different services can be modeled as a virtual service network (VSN, in IRMOS project [17] terminology) that describes the relation between the services that are necessary to accomplish a certain purpose. The user does not necessarily need to be aware of all the details of the network at the time of querying, most of the details should be abstracted and hidden to the end user. At this point one should distinguish between the network of services that a user is able to query and the instantiated network of services, which describes the relation of the different services mapped to physical characteristics of their deployment.

This approach is very useful to discover a desired service and all the dependent services that are necessary to deploy it. In this case a total interaction flexibility is granted to the services involved, this is a rather static view of the interactions between them. This would be valid in case of a simple video streaming service example where the services involved are the downloading service and the network QoS. However if transcoding and payment were to be involved too, because the streaming needs to be paid for and the device only supports a certain type of codec, a workflow needs to be defined that where the user first has to do the payment, the streaming is started and sent to the transcoding from where the stream is either directly sent to the user or sent back to the streaming server to be sent to the user. The interaction between these services at this level has to be detailed on top of the network of services' description as a second layer that orchestrates the interaction between them needs to be added.

A. Upper ontology

Business models can be described using the Business Process Modeling Notation (BPNM), which is a standard notation for developping business processes and workflows (Modelling tools such as Telelogic system architect, are able to describe business models and sometimes generate code interfaces for the underlying code of each service). In order to be able manage, search or compare the different workflows a upper ontology is needed in order to be able to create annotations. To this end the Business process modelling ontology (BPMO) provides the necessary tools under the WSMO framework, and can be used in combination with BPMN also included in the framework.

B. Network layer ontology

Appart from an upper ontology to semantically describe workflows, a network-oriented ontology is needed that describes the knowledge space of network related user experience in a mobile environment, in relation with the IT services provided. Some of the most important concepts are presented in the following:

a) User context:: a relatively great amount of research has been dedicated to this field in the past few years and the community is well aware of the advantages of modeling them with ontologies [8]. Although terminal capabilities and location are usually understood as context as a whole, the fact is that many disciplines are bundled together in it. User context refers to concepts such as user mood, current user activity, status, food preferences, user schedule, weather conditions, and so forth [7].

b) Terminal:: semantics are needed to model terminal capabilities and terminal context [11]. This enables the provisioning of services in the most usable way according to the physical terminal capabilities and state. CC/PP has been found to not have enough expressivity and OWL-based approaches are currently pursued [12]

c) Location:: also, a part of what is traditionally know as context, spatial location models [13] have been studied for a relatively long time and therefore are probably the most mature. Location context offers not only geographical information but also defines areas of spatial relevance based on the geographical relations, which obviously is precious information in a mobile architecture.

d) Scheduling:: needed in order to allocate resources and enable a new time-based dimension in the discovery of services. As mentioned before it is also needed for the user context. [14] shows a simple ontology for publishing and scheduling events much in the way that it is required in a system like the one subject of this work.

e) Policy:: policy-based management enables the administration of a system based on the enforcement of certain rules that are usually provided by the administrator. A rule could be something like: *just head departments or above are allowed to use a certain service between 09:00 and 21:00 if the available bandwidth is less than "acceptable" and in that case their bandwidth will be limited*. Related work in policy-based network management seeks to convert natural language rules to enforcement actions on concrete parts of the system. Typically the language used in this area is CIM [19].

f) Access control:: means for access control that are closely related with identity management and the policy concept previously mentioned, need to describe different players and the relations between them e.g. groups, individuals, roles, ranks, profiles, type of services, and so forth, defining what an entity can access under what circumstances.

g) SLA:: specially in an system with real-time requirements there are strong needs to have high expresivity to model complex real-time service requirements, and enable negotiation and matching of goals between users and providers. Concepts such as cost of service, QoS, limit on cpu usage, penalties if some part of the SLA is broken or the service is not rendered, are necessary and can be very complex.

h) QoS:: this tackles the reservation of resources to ensure a certain level of quality using a service and can be experienced at different levels such as end user-level or system level. This encompasses not only quality of network services such as jitter, bandwidth an so forth, but also from the SOA perspective, time constrains in which a certain job needs to be started and terminated.

C. The network edge interface

In 802.21 the communication between nodes is contemplated at L2, however a L3 communication is also contemplated, which can be used to communicate with a back-end node that collects and provides unified information of the network and might also contain information about the surrounding networks. The information for knowledge querying can be packed in the MIIS interface and shipped to the central collector (directory) that is aware of available services and should then be able to enact them on demand in case a user requests them. Such a network of service's procedure should occur as follows:

1.- The user connects at L2 with several networks. Some connections might take place simultaneously through several interfaces. If some of them use the same media, virtual interfaces by the client or the process can take place sequentially (which would be faster however than having to establish a L3 connection). Partner networks might include information of each other to minimize the amount of networks to query.

2.- It is supposed that the user's terminal is shares a semantic vocabulary and ontologies with network to be used. It is not intended that all terminals understand all networks, most of the functionality will be give by add-ons that the terminal would be able to understand for a specific purpose, e.g. Real-Time related ontologies. Otherwise, the on-demand fetching of ontologies could be eventually possible. The user queries the networks using a form of knowledge base query. This query can be really simple or really complicated describing a concrete network of services with very low-level details. A response receives a very response that might be as simple as yes or no or as complicated as the whole network of services to be used, and the address of the directory and a session token. In which case the user might one to connect to the concrete network. The delivery of the token is susceptible to session hijacking and would need to be be correspondingly protected.

3.- Once the user connects to the network in which the services to use are provided, the token is sent to the directory, which is in charge to to instantiate the services for the corresponding identity or IP. In order to match the identity the user needs to be authenticated agains an Authentication Authority such as radius or diameter that would need to link the token and the identity and the directory needs to obtain the identity (which can be virtual) from the authentication authority, offering the token or a hash of it. 4.- The services are instantiated and the workflow enacted in the case that there is a concrete workflow to be carried out. A concrete grounding description of the instantiated services that directly interface with the user endpoint terminal.

VII. Conclusion

This approach is necessary to leverage a useable and commercially-enabled NGN. In such an architecture, knowledge processing needs to be boosted in order to enable a consistent yet flexible architecture in which services from independent providers and nature can be searched for, automatized and composed taking into account a wide range of influent knowledge sources (e.g. context, real-time constrains, roaming and so forth). In order to access this information and compare the different networks and services offered, a L2 technology independent mechanism needs to be used that minimizes the connection establishment time and the connection overhead. The protocol 802.21, currently under development, provides an interface that can easily be extended in order to accomplish this. Part of this architecture is intended to be developed in the ICT project IRMOS where so-called virtual network of services are necessary accomplish certain tasks in a network aware fashion, adding a real time component.

Acknowledgment

This work is supported by the IRMOS project, which is co-funded by the European Commission's IST programme of the 7^{th} Framework Programme.

References

[1] Birnbaum, J.H., No Neutral Ground in This Internet Battle, Washington Post, page D01, Monday, June 26 2006

[2] Wallace, B., 30 Countries Passed 100% Mobile Phone Penetration in Q1, Telecommunications Online, June 9 2006.

[3] Jähnert, J., Villagrá, V., et al: (Eds.), The Akogrimo Mobile Grid Reference Architecture, Akogrimo Whitepaper, *http://www.mobilegrids.org/*

[4] Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services (Daidalos), *http://www.ist-daidalos.org/default.htm*

[5] Cuevas, A, Moreno, J.I., et al: The IMS Service Platform: A Solution for Next Generation Network Operators to Be More Than Bit Pipes", IEEE Communications Magazine, September 2006

[6] Paolucci, T., Kawamura, T., et al: Semantic matching of Web Service Capabilities, ISWC, pages 333-347, June 2002

[7] Wang, X.H., Zhang, D.Q., et al: Ontology based context modelling and reasoning using OWL, Pervasive Computing and Communications Workshop, pages 14-17, March 2004

[8] Strang, T., Linnhoff-Popien, C., A Context Modeling Survey, UbiComp 1st International Workshop on Advanced Context Modelling, Reasoning and Management. Nottingham. September, 2004

[9] IEEE standard 802.21, work in progress, http://www.ieee802.org/21/

[10] Web Service Modeling Language (WSML), http://www.wsmo.org/wsml/wsml-syntax

[11] Korpipää, P., Häkkilä, J., et al: Utilising context ontology in mobile device application personalisation, Proceedings of 3rd international conference on Mobile and ubiquitous multimedia, pages 133-140, 2004

[12] Agostini, A., Bettini, C., Riboni, D., Loosely Coupling Ontological Reasoning with an Efficient Middleware for Context-Awareness, Proceedings of the The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2005

[13] Stuckenschmidt, H., van Harmelen, F., Information Sharing on the Semantic Web, pages 166-183, July 2004

[14] Sayers, C., Letsinger, R., An ontology for publishing and scheduling events and the lessons learned in developing it,

[15] Studer, R., Grimm, S., et al, 2007 Semantic Web Services: Concepts, Technologies and Applications,

[16] OWL-S, http://www.w3.org/Submission/OWL-S/

[17] Interactive Realtime Multimedia Applications on Service Oriented Infrastructures (IRMOS), http://www.irmosproject.eu/.

[18] Web Service Modeling Ontology (WSMO), D2v1.0, http://www.wsmo.org/2004/d2/v1.0/

[19] Common Information Model (CIM) Specification Version 2.3, Distributed Management Task Force, Inc., January, 2005