Rule-based Reasoning Mechanism for Context-aware Service Presentation

Suparna DE, Klaus MOESSNER
Centre for Communication Systems Research
University of Surrey, Guildford, GU27XH, United Kingdom
Tel: +44 1483 683429, Fax: + 44 1483 686011, Email: {S.De, K.Moessner}@surrey.ac.uk

Abstract: With universal usability geared towards user focused customisation, a context reasoning engine can derive meaning from the various context elements and facilitate decision-taking for applications and context delivery mechanisms. The heterogeneity of available device capabilities means that the recommendation algorithm must be in a formal, effective and extensible form. Moreover, user preferences, capability context and media metadata must be considered simultaneously to determine appropriate presentation format. Towards this aim, this paper presents a reasoning mechanism that supports service presentation through a rule-based mechanism. The validation of the approach is presented through application use cases.

1. Introduction

Recent progress in mobile device capabilities, communication technologies and different media formats lend credence to the vision of ubiquitous services. This implies that the system should deliver services to the user in a manner that is most relevant in any given situation. Realisation of such a networked environment requires that the supporting framework components should be semantically enabled to make context aware decisions for service delivery.

The use of Semantic Web technologies for context modelling and exchange not only offers a common, formal structure to the heterogeneous information inherent in current communication environments, but also enables automated application reasoning about service offerings. Ontology-based context formalisms can provide semantic enrichness and also allow machine-enabled automated reasoning. Especially, the OWL-DL representation language provides expressiveness and computational completeness through the underlying DL logic. The OWL-DL knowledge base can be extended with inference rules to enforce more general first-order logic constraints.

With an OWL-DL context model describing the ambient environment, a number of challenges remain for a reasoning component formulating media delivery decisions. This paper focuses on two key issues: user preferences, capability context and media metadata must be considered simultaneously to determine appropriate presentation format [1]. Additionally, the recommendation algorithm must be in a formal, effective and extensible form to deal with a wide variety of context information.

2. Objectives

This paper presents a reasoning mechanism that formulates recommendations on the best suited device for content presentation. The context reasoner ensures a composition or selection of context sources relevant to the requesting entity's application domain. The reasoning engine has been designed to meet the following requirements: matching context source descriptions to application parameters, retrieval and selection of context sources and derivation of relevant context information by combining multiple context sources.

The designed context reasoner includes the query mechanisms that are employed to extract a subset of the context data collected in the domain model. The reasoner also takes into account static user preferences, which are also detailed in this paper.

The mechanisms have been designed to support a large set of possible use cases and scenarios. The choice of enabling technologies means that the resultant semantics can facilitate provisioning of dynamic services.

3. Related Work

User oriented customisation has formed the focus of a number of research initiatives. The Device Collaboration System (DCS) architecture [2] probes the ambient environment for usable devices that can be used for service presentation by matching device profiles with service requirements and user history. Context information considered by the reasoner consists of ambient context (time, position, and brightness), user profile and mobile device profile. It requires fine-grained specification of preferences by a user, e.g. preferred monitor size when brightness levels are at certain values and position is indoor/outdoor. Also, it does not take into account the software elements of a device (e.g. file formats supported by the device monitor); only the hardware elements are considered during resource recommendation. Context-aware media recommendations form the focus of the reasoning methods used in the COMER architecture [1]. Context elements considered include user preference, terminal capability and multimedia context. A hybrid Bayesian classifier and rule-based reasoner is employed for recommendations. In contrast with the approach in this paper though, the focus of the COMER reasoner is to specify most suited content type, e.g. whether the content should be presented as video (high bandwidth conditions) or image (less terminal capability/ low bandwidth). The DIADALOS project [3] implements a middleware for providing personalised service discovery in pervasive environments. It implements a two-phased query procedure for service discovery: one using standard discovery protocols (UPnP, SLP), followed by semantic queries to filter the results returned by the first step. Filtering procedures are then applied based on stored user preferences.

Based on the review of current state of the art, a number of challenges still remain to be addressed for a framework supporting personalised service delivery: semantic representation and querying must be employed to counteract the heterogeneity of context data. The system must allow user preference statements but should recognise that these can be incomplete or not stated altogether. Content metadata dependence must also not be too high, e.g. [3] assumes that service description would include requirements on the user interface. Finally, these factors must be considered simultaneously to come up with effective recommendations for users.

4. Methodology

The reasoning engine takes as input stored user preferences, incoming content metadata and device modality context. It applies rules relating these sets of context information to facilitate content presentation with the best possible combination of modalities. It forms part of a Service Context Manager (SCM) framework [4] that facilitates relevant and best-suited service presentation.

User preferences offer a way of influencing the state of the service when interacting with the user. The proposed design involves representation of preference configurations that link device types with content metadata. The context in which these stored configurations are activated is provided by the discovery of ambient devices. This design automatically initiates firing of the appropriate rules when the matching device types to an incoming content type, are discovered. The configurations are expressed in OWL-DL. First, the concept of user preference relating to different content types (image, audio or video) is identified. Properties are then defined linking these to the device type concept which also characterises the discovered devices. The device type concept follows the MPEG-21 standard for device type [5]. Thus, the same concept (of device type) is used in the one hand, to place a restriction on the possible classes of devices and also to provide a link to the content type. Device type is an enumerated class and is defined as:

$$DeviceType \leftrightarrow \{PDA, Laptop, PC, MobilePhone, DigitalCamera\}$$

The configuration definition of user preference also includes a weighted measure between 0 and 1 to order preferences, with 0 representing 'unavailable'. This allows the user to express a range of device preferences for a particular content type. The user preference schema is written as:

$$(isLinkedToDeviceType\ \forall\ \mathrm{DeviceType}) \cap (hasWeight \forall float)$$

The universal restriction, $\forall$, on the isLinkedToDeviceType property means that individuals of the user preference class will only have this relation to individuals that are members of the class DeviceType. The UserPreference class is further differentiated into image, audio and video preference concepts.

Each content resource has metadata information that includes multimedia and file format attributes as follows: (1) Image (picture documents): jpg, gif, png etc. (2) Audio files: wma, mp3 midi etc. (3) Video files: wmv, 3gp, mp4, avi, mov etc. The content type along with the format information constitutes a two-level hierarchy of content specification during formulation of content delivery decisions. The content metadata information is provided by the content provider.

The device context information is obtained by the device and service discovery function of the SCM that supports physical device discovery followed by retrieval of associated context information (device descriptions). This context information is then mapped to the defined domain ontology (i.e. the domain vocabulary or TBox) by the transformation module that implements an automated mapping of heterogeneous context information (since device descriptions can have different templates, e.g. UAProf profile or UPnP description) to a common, formalised structure that can be reasoned upon. The design of the multimodal domain ontology is detailed in [6]. Briefly, each device is defined with both its software hosting (e.g. image display service) and hardware (e.g. screen description) capabilities. Each modelled service has an associated service type and function that defines the inputs, outputs and formats permitted. The hardware description adds a layer of refinement and constraint to the software description, e.g. screen size and resolution. Device description also includes network interface definition that provides access to the hosted software services.

5. Technology Description

The reasoning methods employed are based on SWRL (Semantic Web Rule Language) which is a W3C submission combining OWL and an inference rules language based on RuleML (Rule Markup Language). A comparison of SWRL and RuleML features is given in [7]. The implementation of an OWL-SWRL base for context modelling and reasoning follows the assertion that ontologies and rules can be integrated to achieve dynamic service oriented architectures [7]. Rules are written in terms of OWL classes, properties and individuals. These are forwardchain rules that infer about axioms and take the form A^B→C. This implies that if all the atoms in the antecedent (A^B) are true, then the consequent (C) must also be true. The implemented reasoning engine employs the Jess rule engine and SWRL APIs to manipulate SWRL rules programmatically. The API retrieves the defined SWRL rules into the program, which can then be input into the Jess rule engine. In addition, rule creation methods allow dynamic rule creation during runtime and these can then be inserted into the underlying OWL model.

The queries have been formalised through the SQWRL (Semantic Query-Enhanced Web Rule Language) language, [8] which is a library extension to the SWRL rule language. It is based on the fact that a rule antecedent can be viewed as a pattern-matching mechanism, i.e., a query. It allows queries directed at OWL classes, individuals and properties. SQWRL queries can operate in conjunction with the SWRL rules and thus can be used to retrieve knowledge inferred by the rules. The result of SQWRL queries is effectively a two-dimensional table. The working of the context reasoner is illustrated in figure 1.

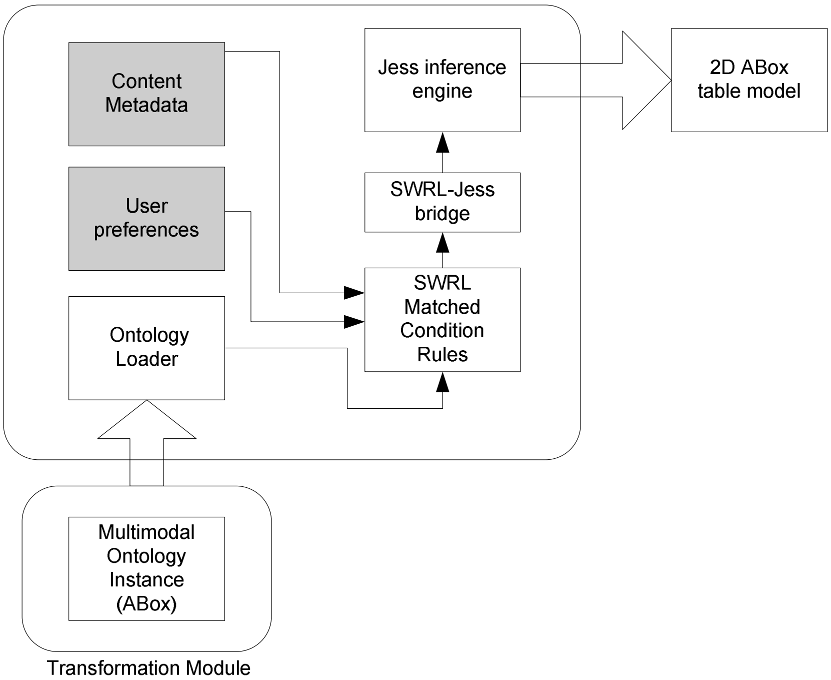

Figure 1. Context Reasoning components

The reasoner performs its processing in two stages. In the first stage, rules are defined for relating user device preferences to content and for ranking discovered devices based on these. Some of these *forwardchain* rules are presented below:

Rule-1:

ImagePreference(?ip) ∧ isLinkedToDeviceType(?ip, ?dt) ∧ hasWeight(?ip, ?wt) ∧
swrlb:greaterThan(?wt, 0) ∧ Device(?d) ∧ hasType(?d, ?dt) ∧ hosts(?d, ?s) ∧
hasSType(?s, ?stype) ∧ swrlb:containsIgnoreCase(?stype, "imag") ∧
ScreenOutputModality(?m) ∧ canbeInterfacedVia(?d, ?m) ∧
Resolution(?res) ∧ hasResolution(?m, ?res) ∧ width(?res, ?w) ∧ height(?res, ?h) ∧
swrlb:add(?reso, ?w, ?h)
→ sqwrl:selectDistinct(?d, ?wt, ?reso) ∧ sqwrl:orderByDescending(?wt, ?reso)

Rule-2:

ImagePreference(?ip) ∧ isLinkedToDeviceType(?ip, ?dt) ∧ hasWeight(?ip, ?w) ∧
swrlb:greaterThan(?w, 0) ∧ Device(?d) ∧ hasType(?d, ?dt) ∧ hosts(?d, ?s) ∧
hasSType(?s,?stype) ∧ swrlb:containsIgnoreCase(?stype, "imag") ∧
hasFunction(?s, ?sfn) ∧ hasFormat(?sfn, ?format) ∧
swrlb:stringEqualIgnoreCase(?sdo:format, "jpg")
→ sqwrl:selectDistinct(?d, ?w) ∧ sqwrl:orderByDescending(?w)

Rule-1 ranks the devices that can support image display capabilities according to the defined user preference and secondly, in decreasing order of screen resolution. It can be read as: consider all defined image preferences ‘ip’ which link to a specific device type ‘dt’ if the weight ‘w’ of this preference is currently greater than 0. Simultaneously, from the list of discovered devices ‘d’, consider the subset of devices that have the same device type ‘dt’ as stated in the image preference. Moreover, if the device hosts a service ‘s’ which has a service type ‘stype’ of image, then retrieve the resolution of the associated screen. Finally, output the ranked list of conforming devices, using the ‘select’ query, ordered first by the weight associated with the image preference and then by the screen resolution of the device.

Rule-2 ranks the devices that can support image display capabilities according to the defined user preference, but only those that support the jpg format. This rule is produced here only for the jpg format, but at runtime, the format information (jpg/png/gif etc.) is passed in as a string variable to complete the rule. This allows dynamic run-time matching with the format of the image. It is similar to rule-1, but here, if one of the supported service formats ‘format’ equals ‘jpg’, then, the list of conforming devices is output, ordered by the weight associated by the user preference.

Similar rules have been defined for audio and video content types. Thus, the device ranking mechanism showcased takes into account a number of variables in a descending hierarchy: the user preferences, content metadata and screen resolution.

It has been recognised in the literature that users can be lazy to spell out their preferences or not even have any clearly defined ones [9]. Hence, to minimise the dependence of the reasoning mechanism on user preferences, a second processing stage has been defined. This is executed if the first stage of rules does not return any results. This takes into account the situation where the user has not defined any content – device related preferences and also when no device types conforming to the stated preferences have been found in the ambient environment. The rule related to image content type is presented below:

Rule-3

Device(?d) ∧ hosts(?d, ?s) ∧ hasSType(?s, ?stype) ∧
swrlb:containsIgnoreCase(?stype, "imag") ∧ hasFunction(?s, ?sfn) ∧
hasFormat(?sfn, ?format) ∧ swrlb:stringEqualIgnoreCase(?format, "jpg") ∧
ScreenOutputModality(?m) ∧ canbeInterfacedVia(?d, ?m) ∧ hasResolution(?m, ?res) ∧
width(?res, ?w) ∧ height(?res, ?h) ∧ swrlb:add(?reso, ?w, ?h)
→ sqwrl:selectDistinct(?d, ?reso) ∧ sqwrl:orderByDescending(?reso)

Rule-3 returns those devices that can display images with the ‘jpg’ format and ranks the selected devices according to their screen resolution. Similar to rule-2, the jpg format specification in the rule has only being presented here for rule completeness; during rule-execution, it is passed in as a parameter. So, this rule will seamlessly select all devices that have jpg listed as one of the formats supported for their image display service.

6. Applications

This reasoning subsystem has been integrated into the SCM framework and validated through a proof of concept demonstrator. This section aims to show its working through a selected use case that highlights the working of the various modules of the framework through the messages exchanged between them.

The scenario involves a user who is in his office and working on image files on his laptop. When he moves to the meeting room and starts the slideshow, the SCM prompts him to display it on the projector screen as the best suited output method. In this scenario, the reasoner computes an ordered list of recommended devices suited to the content request, based on which, delivery decisions can be formulated and executed. The sequence diagrams in Figures 2-4 show the various steps involved in this scenario. These diagrams show only the SCM components involved; the interaction with other system entities, viz. the Content Delivery interface, is not shown here.

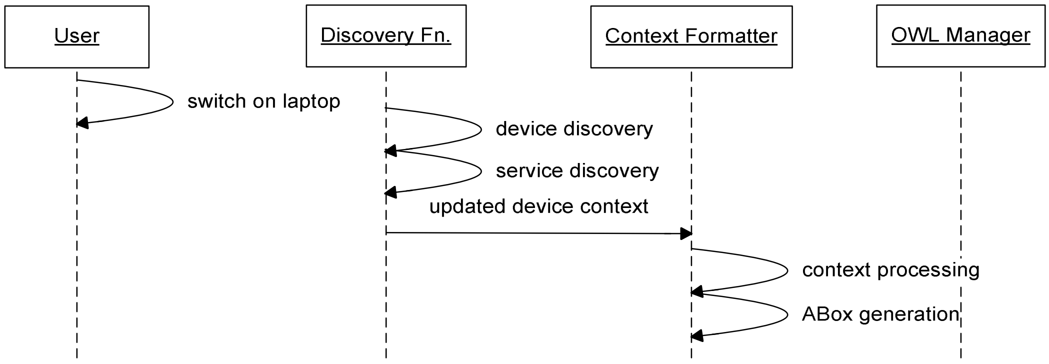

Figure 2. Sequence flow – device available

When the laptop is switched on, it sends out a device presence advertisement message. This is multicast on the network and is received by the discovery function. The extended UPnP protocol [10] is used for this discovery step as well as for the retrieval of the XML device descriptions. The descriptions, which consist of a main file and separate XML description files for each hosted service, are retrieved through HTTP GET commands. The retrieved context is processed to generate the ontology-based instance data representing ambient context (hence forming the assertion component or ABox). The transformation of the varied context data into a semantic form is accomplished by applying scripts matched to the input description format to transform the XML context to an OWL ABox. XSLT (XSL Transformation) is employed for the mapping. The ContextFormtatter and OWLManager represent this step in Figure 2.

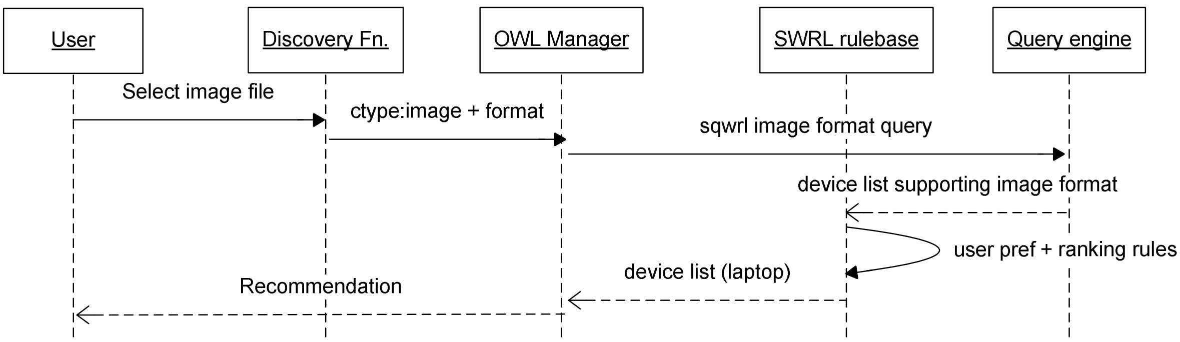

Figure 3. Sequence flow – content selection

The sequence of processing steps performed when the user selects a content type (image file) is shown in Figure 3. The content type (image) and its format information are passed on to the OWLManager. The OWLManager interfaces to the ABox. The content metadata is then input into a SQWRL query which is presented to the query engine. The result of this query, which is a list of the available devices supporting the content, is handed over to the SWRL rule engine. Here, the device list is ranked according to the stored user preferences and other criteria (e.g. device monitor resolution). This re-ordered list is handed back to the OWLManager, which presents it to the user on the GUI.

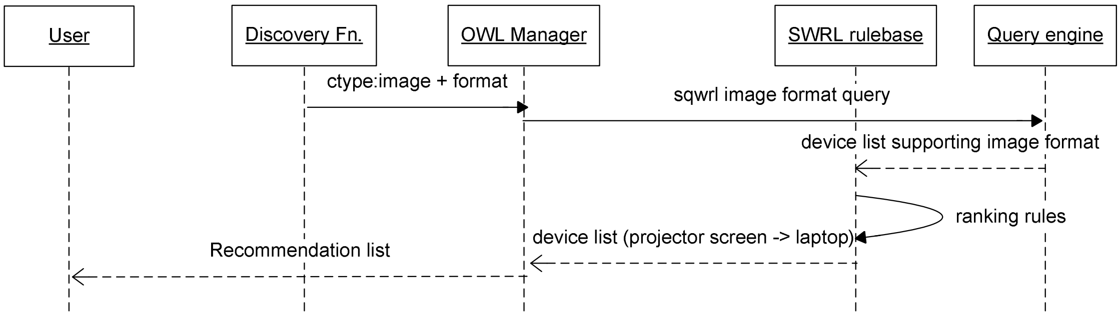

Figure 4. Sequence flow – device recommendation

The sequence of information and control flow when the user enters the meeting room is shown in Figure 4 and is similar to when the laptop is first switched on. The OWL-DL model is updated, content type is matched against the updated ambient context and then the recommendation list is formed by applying the SWRL rules. Since the projector screen has a bigger resolution, it ranks above the laptop screen for image display and is shown as a recommendation to the user.

7. Discussion

The design of the context reasoning module employs the SQWRL query mechanism built on top of SWRL to formulate context sensitive queries. Since SWRL itself is built on top of OWL, the query language offers the twin benefits of querying ontology terms directly and also being cognizant of the defined rules during query execution. On the other hand, queries can be said to be constrained to the domain ontology terms. However, since queries are posed by the application logic that references the *common*, formal structure of the domain ontology, natural language queries need not be taken into account in the context of this work. The combination of OWL and SWRL offers an expressive platform for constructing a generic model of the heterogeneous domain and then express particular behaviors. However, as also identified in [7], extensions to model uncertainty and probability in real-world contexts are needed.

The design principles of the work reported in this paper can be contrasted to other personalised service platforms. The work presented here takes into account user preferences but is not wholly dependent on it for correct execution. This is in contrast to the reasoning mechanism in the DCS project [2] which requires fine-grained stated preferences, as pointed out in section 3. Moreover, the link between the information base and the resource discoverer which searches and gathers sharable resources is not clearly reported in [2]. The Akogrimo project [11] aims at mobile grids as virtual organisations that offer resource sharing and continuous service sessions for a mobile user in different contexts. The scope of contexts refers primarily to different user 'situations', with resource (device) context already being defined and stored. In contrast to this, resource and service context is acquired from the user's vicinity in our work, formalised and then reasoned upon to facilitate personalised media –based services. The reasoning here, is however, focussed to matching content to the 'right device'. In the DIADALOS project, service description is required to include requirements on the user interface (e.g. device screen). In the work presented here, this is inferred as part of the reasoning mechanism. Also, queries in [3] include both a protocol specific (e.g. UPnP) and semantic component. This means that the ambient environment is queried for suitable devices for every user or application generated request. In the here presented work, queries are directed to the ontology context model (ABox) for ambient context, generating fewer number of messages.

8. Conclusions

The proposed approach indicates that ontologies and rule-based reasoning can help to achieve automated and personalized service delivery in mobile communication environments. The reasoning mechanism takes into account a number of variables, viz. the user preferences, ambient device context and the incoming media stream type. The dependence on detailed user preferences and content metadata has also been minimised. The design is generic enough to be further extensible to a number of factors, such as costs associated with network interfaces on a device to decide upon optimum content delivery.

The heterogeneous device management can allow service and content providers to deliver personalised services to users. The generic, extensible nature of the developed mechanisms can facilitate rapid service creation, agnostic to users' devices.

The mechanisms presented here have been integrated into a system demonstrator for the project [12] of which this work is a part of. This work plugs in as an input to an adaptation system developed within the project and showcases autonomous and dynamic adaptation on behalf of the users for each service/ content request.

Recommendations for future work include learning mechanisms that can provide support for anticipatory responsiveness to changing context.

Acknowledgement

The work reported in this paper has formed part of the Ubiquitous Services Core Research Programme of the Virtual Centre of Excellence in Mobile & Personal Communications, Mobile VCE, www.mobilevce.com. This research has been funded by the Industrial Companies who are Members of Mobile VCE, with additional financial support from the UK Government's Technology Strategy Board (previously DTI). Fully detailed technical reports on this research are available to Industrial Members of Mobile VCE.

References

[1] Y. Zhiwen, Z. Xingshe, Z. Daqing, C. Chung-Yau, W. Xiaohang, and M. Ji, "Supporting Context-Aware Media Recommendations for Smart Phones," *IEEE Pervasive Computing*, vol. 5, pp. 68-75, 2006.

[2] J.-H. Park, W.-I. Park, K. Young-Kuk, and K. Ji-Hoon, "Personalized recommender system for resource sharing based on context-aware in ubiquitous environments," in *Proc. International Symposium on Information Technology, ITSim*, vol. 4, 2008, pp. 1-5.

[3] K. Frank, V. Suraci, and J. Mitic, "Personalizable Service Discovery in Pervasive Systems," in *Proc. Fourth International Conference on Networking and Services, ICNS*, 2008, pp. 182-187.

[4] S. De and K. Moessner, "A Framework for Mobile, Context-Aware Applications," presented at 16th International Conference of Telecommunications (ICT '09), Marrakech, Morocco, May 25-27, 2009.

[5] *Multimedia Framework (MPEG-21) — Part 7: Digital Item Adaptation*, British Standard 21000-7, 2004.

[6] S. De and K. Moessner, "Semantic Description of Multimodal Devices: Modelling and Evaluation," presented at 18th eChallenges (e2008) conference, Stockholm, Sweden, October 22-24, 2008.

[7] S. Stoutenburg, L. Obrst, D. Nichols, K. Samuel, and P. Franklin, "Applying Semantic Rules to Achieve Dynamic Service Oriented Architectures," in *Proc. Second International Conference on Rules and Rule Markup Languages for the Semantic Web*, 2006, pp. 75-82.

[8] M. J. O'Connor, S. W. Tu, A. K. Das, and M. A. Musen, "Querying the Semantic Web with SWRL," in *Proc. The International RuleML Symposium on Rule Interchange and Applications (RuleML2007).* Orlando, FL: Springer Verlag, 2007, pp. 155-159.

[9] C. Doctorow, "Metacrap: Putting the torch to seven straw-men of the meta-utopia," 2001. [Online]. Available: http://www.well.com/~doctorow/metacrap.htm

[10] S. De and K. Moessner, "Device and Service Descriptions for Personal Distributed Environments," in *Proc. 2nd IEEE International Conference on Digital Information Management (ICDIM'07), volume 2*. Lyon, France, 2007, pp. 695-700.

[11] C. Loos, S. Wesner, and J. M. Jahnert, "Specific Challenges of Mobile Dynamic Virtual Organizations," in *Innovation and the Knowledge Economy: Issues, Applications, Case Studies*, IOS Press Amsterdam, 2005, pp. 1209 - 1216.

[12] MVCE, "Ubiquitous Services: Core 4 Research Programme," www.mobilevce.com.