Ontology-based Context Inference and Query for Mobile Devices

Suparna De and Klaus Moessner
Centre for Communication Systems Research
University of Surrey
Guildford, United Kingdom
{S.De, K.Moessner}@surrey.ac.uk

Abstract—The vision of service personalization for mobile communication environments entails context sensitive service provisioning. The realization of such customizable smart spaces necessitates acquisition and processing of modality context information from a variety of devices in the ambient environment. The heterogeneity of available device capabilities and description formats brings new challenges for a context reasoning engine that formulates content delivery decisions. Specifically, to ensure interoperability with existing application logic, the enabling components should support semantic queries. Secondly, situations where variously formatted context input may not provide enough information to answer queries, should be intelligently handled. Towards this aim, this paper discusses a context reasoning and query interface component as part of a Service Context Manager (SCM) framework that supports semantic querying and service personalization in different planes; with ubiquitous computing purported to be focused on disparate device interoperation and universal usability geared towards user-focused customization.

Realization of such proactive systems brings up the related research issues of context definition, acquisition and reasoning. Multimodal device capability information modeling and processing is an important precursor to customized service delivery in such systems. The applicability of ontologies for context modeling and facilitating application reasoning has been well researched [2] - [3]. Specifically, the OWL-DL (Web Ontology Language - Description Logics) formalism, rooted in the decidable fragment of first-order logic, provides a powerful platform for a formal and machine-processible structure to context information collated from diverse sources.

With an ontology-based context model providing a common, formalized structure, a number of interconnected components are required to provide a generic mechanism for context querying and reasoning. This paper focuses on two key issues: context querying – the framework should support semantic querying, which may be variously worded by application logic that uses the context model. For example, a search query for image display capability could be expressed as a query for ‘image modality’ or ‘image display service’. Moreover, as pointed out in [4], the diversity of context information formats means that not enough information may be available to answer queries; and the framework must adjust accordingly.

To address these issues, this paper presents a context reasoning and query interface component as part of a Service Context Manager (SCM) framework that supports a generic mechanism for querying contexts and handles incomplete context information through a rule-based mechanism.

The paper is organized as follows: enabling technologies and the current state of the art are analyzed in section II. Section III presents the SCM framework with its various functional modules. The context reasoning and querying mechanisms form the focus of sections IV and V, respectively. This is followed by showcasing of implementation aspects, with a reference application scenario, in section VI. It also includes a discussion of the results and mechanisms employed. The paper concludes with a summary and recommendations for future work in section VII.

II. RELATED WORK

Ontologies and rules can be integrated to achieve dynamic service oriented architectures [5]. With OWL-DL taking advantage of the underlying DL logic for computational completeness and expressiveness, the knowledge base can be extended with inference rules to enforce more general first-order logic constraints. The resultant semantics can facilitate provisioning of dynamic services. Towards this end, SWRL (Semantic Web Rule Language) [6] is a W3C submission aimed at combining OWL and an inference rules language based on RuleML (Rule Markup Language). SWRL makes use of pattern-directed invocation of procedures from assertions [7]. It also provides a common language for the context model and the inference mechanism. A comparison of SWRL and RuleML features is given in [5].
Enabling technologies for effective ontology query include SPARQL (SPARQL Protocol and RDF Query language) [8] and SQWRL (Semantic Query-enhanced Web Rule Language) [9]. SPARQL is a RDF-based query language and offers limited support for querying OWL models, as noted in [10]. SQWRL is a library extension to SWRL. It is based on the fact that a rule antecedent can be viewed as a pattern-matching mechanism, i.e., a query [10]. It allows queries directed at OWL classes, individuals and properties and also honors the SWRL rules during query execution.

Ontology-based context management approaches have been employed in research projects for various aims, including media content recommendation [3], activity-based m-learning [11] and command and control systems for the battlespace domain [5]. The context information service architecture in [4] for managing distributed context sources employs an ontology driven mechanism for answering application queries. Each context-aware application provides its own ontology to the context service to describe its hosted services. The context service maintains an ontology mapping repository to translate between ‘equivalent’ words in submitted queries to generate a response from the registered services. Queries can be formulated in SQL, XQuery or RDQL (RDF Data Query Language). However, this approach assumes that queries are expressed in terms of one of the registered ontology concepts and does not account for incomplete context information during ontology formation.

The e-learning repository implementation in [12] consists of a learning resource ontology and SWRL rules to offer recommendations on study methods to learners. A semantic query interface is designed by collecting synonyms of the ontology concepts, calculating a similarity coefficient and storing these in a ‘synonymy list’. Learner’s queries are parsed into tokens that are associated with the ontology concepts using the synonym list, through a breadth-first or depth-first search.

The research project presented in [7] aims to build a smart home environment for its inhabitants. The context model is based on an OWL ontology to provide a representation of the smart home. The inference layer is implemented using SWRL rules to issue orders on the event-driven bus that drives the actuators.

III. SCM ARCHITECTURE

The work presented in this paper forms part of the SCM framework. The SCM framework aims to facilitate context sensitive service provisioning in mobile communication environments. This section presents a brief overview of the SCM structure to provide a background for the context reasoning module. A more detailed description of the SCM components and implementation is presented in [13]. Fig. 1 provides an illustration of the SCM architecture.

The device and service discovery function detects device presence in the ambient environment through explicit discovery messages and also by listening on the multicast channel for device advertisements. UPnP protocol is used for this discovery step as well as for the retrieval of the XML device descriptions. The components within the Transformation Framework map the device modality context information into a common, formal structure based on the defined ontology stored in the OWL Facts Base. The OWL Facts Base represents the domain vocabulary in terms of classes and properties and thus forms the TBox (concept box). The mapped descriptions constitute the ABox (assertion component), embodying knowledge of real-world objects in TBox-compliant statements.

The generated ABox forms the input to the reasoning subsystem that references the SWRL Rules base. The Reasoner module performs the first stage of context inference by applying rules from the Rule Base that assert missing context information. This pre-processing step is transparent to the user and speeds up the next stage of context filtering. The Context Filtering module then applies the rules relating to the stored user preferences and matches incoming content metadata to the processed device modalities’ ABox to facilitate content presentation with the best possible combination of modalities. The Service Repository Management module maintains an interface to context-aware applications for updating and querying semantic service information.

IV. CONTEXTUAL INFERENCE RULES

The context reasoning module references the SWRL Rules base to infer facts from the ontology instance (ABox) that has been populated from the device and service context information. The rules have been designed to take into account the standard device-service context description templates. Due to this heterogeneity of context information, there may be gaps in the generated ABox. Hence, at a first instance, the modeled rules assert links between the physical device and hosted software services. The second stage of context filtering serves to filter out a subset of possible device modalities based on the defined user preferences and content metadata. For instance, stored preferences may indicate which device the user would like to ‘see videos on’. The content metadata relating to content type and other factors such as resolution, frame rate (for media content) etc. serve to match modalities to content.
SWRL is grounded in first-order logic and provides more expressive power than DL. SWRL provides only first-order logic rules that make use of logical connectors such as ∧ (AND), ¬ (IMPPLY) and ¬ (NEGATION). However, extensions have been proposed to make use of more advanced mathematical and string operators such as comparison operators (swrlb:lessThan) through the SWRL built-in library. Built-ins take any number or combination of OWL datatyp or object property values. The object property arguments are in effect OWL individuals. A SWRL rule contains an antecedent part, called the body and a consequent part called the head. This implies that if all the atoms in the antecedent are true, then the consequent must also be true, i.e., antecedent ⇒ consequent

The rules can be written in terms of OWL classes, properties and individuals. Fig. 2 shows some of the forwardchain rules (that infer about axioms) defined on the domain ontology.

Fig. 2 SWRL forwardchain rules

| Rule-1: | Device(?x) ∧ ScreenOutputModality(?m) ∧ canbeInterfacedVia(?x, ?m) ∧ graphicsEnabled(?m, true) ∧ hasService(?x, false) → hosts(?x, DisplayService) ∧ isLinkedTo(DisplayService, ?m) |
| Rule-2: | Device(?x) ∧ Speaker(?m) ∧ canbeInterfacedVia(?x, ?m) ∧ hasService(?x, false) → hosts(?x, AudioService) ∧ isLinkedTo(AudioService, ?m) |
| Rule-3: | Device(?x) ∧ ScreenOutputModality(?m) ∧ canbeInterfacedVia(?x, ?m) ∧ hasResolution(?m, ?r) ∧ width(?r, ?w) ∧ height(?r, ?h) ∧ swrlb:add(?y, ?w, ?h) → sqwrl:select(?x, ?y) ∧ sqwrl:orderByDescending(?y) |

Rule 1 defines a device with a graphic-enabled screen to offer a display service. The display service has already been defined in the domain ontology to support text output, visual and image display service offerings. Rule 1 reads as: if there is a Device ‘x’, which can be interfaced through a ScreenOutputModality ‘m’, where ‘m’ is graphics-enabled, then Device ‘x’ hosts a display service. In addition, the link between this asserted display service and the screen modality is also asserted in this rule for completeness. This rule also illustrates the work-around for the absence of atom negation in SWRL. Due to SWRL’s monotonicity, it is not possible to query directly the absence of a property assertion. So, since the presence or absence of a hosted service cannot be ascertained, a Boolean valued hasService property is queried to affirm that the rule applies to only those devices where the device-service link is not already present. This takes care of cases where the device description is available in CC/PP [14] or UAProf [15] format, where only the hardware modality information is available.

Rule 2 is similar to rule 1 and defines a device with audio output modality to offer an audio service.

Rule 3 is an example of pre-processing of available modalities where the devices are ordered according to decreasing screen resolution. This rule also illustrates the use of the SQWRL query function for the actual ordering function.

V. QUERY INTERFACE

Since SWRL is built on top of OWL, it shares OWL’s Open World Assumption (OWA) where every fact can be thought of as true unless explicitly stated to be false. For instance, two individuals cannot be assumed to be automatically distinct unless explicitly stated to be so with the owl: differentFrom restriction. However, the SWRL Query library (SQWRL) that functions as a query language on the SWRL rule-set has an approximation to Closed World Assumption (CWA). CWA follows the presumption that what is not currently known to be true is false. This grounding in CWA allows the query to apply to the formalised TBox and the resultant ABox only and allows effective query of the knowledge base. This also prevents undecidability.

With the context inference rules in place, the device-service context of the discovered devices can be queried for available services. Fig. 3 shows some of the sample queries.

Fig.3. SQWRL queries

| Query-1: | hosts(?x, ?m) ∧ ModalityService(?s) ∧ hasSType(?s, "image") → sqwrl:select(?x) |
| Query-2: | hosts(?x, ?s) ∧ ModalityService(?s) → sqwrl:select(?x, ?y) |

Query 1 searches for devices that offer an image display service. It can be read as: for the asserted property ‘hosts’ that links a device ‘x’ to its hosted service ‘s’, where ‘s’ is a modality-related service and the service type of ‘s’ is ‘image’, output all conforming ‘x’-s (or devices).

Query 2 searches for all devices that host any manner of modality service (text/image/audio/video) and outputs the hosted service as well.

VI. IMPLEMENTATION

This section presents the results of the implementation of the reasoning and query subsystem. To validate the approach, context sources available in two standardized formats, viz. UAProf and UPnP[1], were input into the SCM framework.

Fig. 4 shows a sample UAProf profile of a mobile phone. The mobile phone is modeled to support a screen that is color-capable and graphics-enabled (image capable). Thus, this...
The SWRL plug-in to the Protégé IDE was used for entry and editing of SWRL rules. To perform the actual inference, the Jess [17] rule-based inference engine was employed as a plug-in to the Protégé SWRL tab. However, it should be noted that the rules are represented independently of the inference engine. Moreover, all instances in the ABox get validated against the rules during execution. The queries have been designed through the SQWRL built-in library mechanism.

The results of executing queries 1 and 2 are shown in the result pane snapshots in Figs. 6 and 7, respectively.

The second context source models an extended UPnP description [16] of a mobile phone, specifying the offered ‘image display’ service and associated screen modality. The associated context source file fragment is shown in Fig. 5.

A. Application Scenario

The potential of the approach can be demonstrated with an example scenario illustrating service personalisation. The devices in the environment host their own descriptions in XML format, which are retrieved following device discovery. The use case consists of two devices: a UAProf mobile terminal offering a display service (Fig.4) and a mobile phone advertising its image service through UPnP descriptions (Fig. 5). The devices are discovered by the discovery module. The use case employs UPnP protocol for this step. The retrieved XML descriptions are processed and then transformed by the transformation component into an ontology instance (ABox). A more detailed breakdown of the implementation steps involved in transformation is available in [13]. The generated ABox forms the input to the reasoning subsystem. The result sets in Figs. 6 and 7 validate the working of Rule 1 which asserts the associated modality service for physical device descriptions. This first stage of context inference thus handles incomplete context information, which is not taken care of in [4]. The second reasoning stage then applies rules to match incoming content metadata to the processed ABox. This serves to filter out a subset of possible device modalities. For an example case where the content type is audio, it does not match the offered modalities in the ambient environment (image display on both available devices). This generated semantic information can be then input to an Adaptation Manager functionality that decides what adaptation mechanisms may be applied based on the delivered processed modality context. In this case, the context source provides only hardware modality information, with no information on the related software service (formats, or supported types).
Adaptation Manager decides that an audio to text adaptation is necessary and it is executed through the available adaptation mechanisms. The adapted content is then handed to a content delivery system that streams the content to the selected device.

This scenario walk-through demonstrates the potential applications of the proposed approach. The framework thus forms a pluggable input to content adaptation and content delivery systems. When deployed as part of a wider ubiquitous environment, the end-user will thus be presented with a more automated and proactive system.

B. Discussion

The approach presented in this paper 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 domain and then express particular behaviors. Also, Jess supports insertion of the inferred knowledge back into the OWL-DL ABox, which allows subsequent query answering. However, as also identified in [5], extensions to model uncertainty and probability in real-world contexts are needed.

VII. CONCLUSION

The proposed context inference approach indicates that ontologies and rule-based reasoning can help to achieve automated and personalized service delivery in mobile communication environments. The rule base is easily extensible to encompass varied context scenarios and supports semantic querying through the developed query interface.

A possible extension includes a learning mechanism that can serve to improve the reasoning quality for incomplete context information.

ACKNOWLEDGMENT

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. Fully detailed technical reports on this research are available to Industrial Members of Mobile VCE.

REFERENCES