Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 34 (2014) 587 – 592
Procedia Computer Science

International Symposium on Emerging Inter-networks, Communication and Mobility (EICM-2014)

A Semantic Approach for Service Adaptation in Context-Aware Environment

Hatim Guermah[a,*], Tarik Fissaa[a], Hatim Hafiddi[a,b], Mahmoud Nassar[a], Abdelaziz Kriouile[a]

[a] *IMS Team, SIME Lab, ENSIAS, Mohammed V Souissi University,BP 713, Agdal Rabat, Morocco*
[b] *ISL Team, STRS Lab, INPT, Rabat, Morocco*

Abstract

With pervasive computing, Service oriented architectures become more mobile and distributed and are executed in varying contexts. Context is understood as any information about the user needs and operating environment which vary dynamically and have an impact on design choices. Context aware Services oriented systems have emerged as an underlying design and development paradigm. An important challenge in such systems is context modeling, reasoning and adaptating services behavior. Using ontologies and semantic web services to deal with context aware services has attracted a lot of interest. In this paper, we present an architecture for the development of context aware services based on ontologies. We highlight our context metamodel and discuss about reasoning process. we also present our semantic approach for service adaptation in context aware environment.

Selection and peer-review under responsibility of Conference Program Chairs
Keywords: Context, Context Awareness, Pervasive Computing, Model Driven Engineering, Semantic Web Services;

1. Introduction

In pervasive computing, context-aware service oriented systems refer to applications that use so-called context information to provide appropriate services to the user. They have emerged in the ambition to provide access to the system anytime, anywhere and an adapted content. To adress the service adaptation issue, it is important to have an efficient mechanism to capture and manage context, an appropriate reasoning mechanism and also a mechanism to adapt services according to the user's context.

Many approaches have been proposed to deal with Context Aware Services (CAS), but most, only offer weak support for knowledge sharing and context reasoning. Ontologies have proved to be the most suitable model for representing and reasoning on context information for the following reasons [1]: (i)Ontologies enable knowledge sharing in open, dynamic systems and enable service interoperability.(ii)Ontologies with well-defined declarative semantics allow efficient reasoning on context information.

* Hatim Guermah. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: Hatim.Guermah@um5s.net.ma

Selection and peer-review under responsibility of Conference Program Chairs
doi:10.1016/j.procs.2014.07.077

In addition, context-aware systems, as well as having a suitable context model, must understand context and adapt to it. To achieve this goal, it is necessary to reason about context at semantic level to interpret this information and obtain knowledge from it. According to this knowledge, the system must adapt its behaviour to meet the needs of users within ever-changing context. In this work, we aim to propose in first stage an overview of an ontology based architecture for the development of context aware services and in the second stage we present our contribution in semantic context-aware service by extending OWL-S[3].

The remainder of this paper is structured as follows. Next we present a motivation scenario that concerns context-aware E-health system. Section 3 discuss some background and definitions. In section 4 we describe the proposed architecture. Section 5 present our proposal for service adapation by extending OWL-S. Finally Section 6 presents some related work in context aware applications. The last section gives rise to some concluding remarks and plans for future works.

2. E-health Scenario

In this scenario, the underlying idea of the telemonitoring is to allow elderly people, chronic or high risk patients to stay at home and to benefit from a remote and automated medical supervision.

Mr John is Diabetic and despite medication and treatment monitoring, he still suffers from diabetic crises. Thus, when the blood sugar is at low level (hypoglycemia), an alarm service is triggered to reports the problem to Mr John and inform him that he should take his medication. Otherwise, hypoglycemia can cause dangerous crises. In the contrary, when the blood sugar is at high level we talk about hyperglycemia. Normally, notification service must be invoked to prevent Mr John to take insulin or his medication.

Thus, the situation of Mr. John requires a vigilance to detect crises. To adress those issues, recently Mr. John has been provided with a tele-monitoring context-aware application able to monitors diabetic patients.

For the development of a such E-health system first, we have to choose the necessary context information. Next step is context modeling using ontologies to provide formal semantics to context. Then, it is important to reason about context at a semantic level to interpret this information and obtain interesting knowledge. According to this knowledge, the system adapts its behavior to meet the user's needs.

3. Background

3.1. Context Awareness

In general, context awareness refers to the ability of an application to discover and take advantage of contextual information, such as user location and nearby devices. Dey[4] considers an application as context aware if it uses contextual information to provide relevant information and services to the user, where relevance depends on the user's task.

3.2. Semantic Web Services

Several approaches have been developed for semantic web services among them OWL-S[3]. OWL-S is an ontology for the description of semantic Web services expressed in the Web Ontology Language (OWL). OWL-S defines an upper ontology for semantically describing Web services along three main aspects:

- The Service Profile describes 'what the service does' in terms of inputs, outputs, preconditions and effects (IOPEs).
- The Service Model describes 'how a service works' in terms of a process model that may describe a complex behaviour over underlying services.
- The Service Grounding describes 'how the service can be accessed', usually by grounding to WSDL.

Thus the main objective of our approach is to use Semantic Web services specifically OWL-S by extending it with context information to meet the users needs.

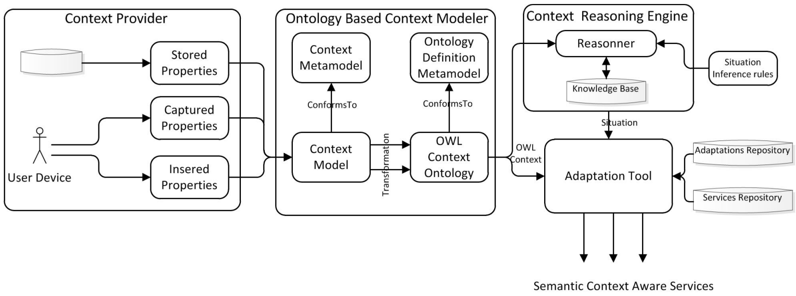

Fig. 1. Overview of the proposed architecture

4. Architecture Overview

To deal with architectural challenges of context-aware services oriented applications and meet the clients requirement, we present our ontology based architecture for context aware services(see Figure 1). The details of the proposed architecture are described in the next sections.

4.1. Context Modeling

To model context in generic and abstract way, we aim to propose a metamodel that defines the context (see Figure 2)

This metamodel is based on the following specifications:

- A context decomposes into sub contexts;
- A sub context can be, recursively, decomposed into categories for its structuring;
- A context, a sub context and a category are constituted of context properties;
- A context property is gathered by sensors: SensedCtxProperty, or derived from other context properties: DerivedCtxProperty, or stored in the database: StoredCtxProperty.
- Each property has a context validity.
- A derivedCtxProperty is obtained by derivation from a set of properties based on derivationFunction.
- A StoredCtxProperty is obtained through the recovery features of stored properties (e.g. RecoveryFunction)
- A sensedCxtProperty is obtained and characterized by the context source SourceCxtProperty and the acceptance value of context property QualityofProperty
- Each sub context has a specific type of access.

Let's project it on the case of the E-health system. The context is composed of the following sub contexts.

- Environment: represent user's location,time...etc
- Medical Information : contains user's health properties(Blood Sugar, Blood pressure) and his medical history.
- User : contains properties describing the user (Blood groupe, age, user's contact, preference)
- Device : contains parameters that describe the entity Device (e.g. medicals device, mobile phones, PDA...etc)

4.2. Context Metamodel Transformation to OWL

Models transformations provide a mechanism for automatically creating or updating target models based on information contained in existing source models. Formally, a simple model transformation has to define the way for generating a model Mb, conforming to a metamodel MMb, from a model Ma conforming to a metamodel MMa. In our case, the source metamodel corresponds to a view from our context metamodel and the target metamodel corresponds to the OWL metamodel.

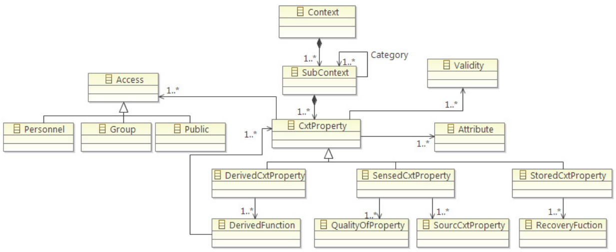

Fig. 2. Context metamodel

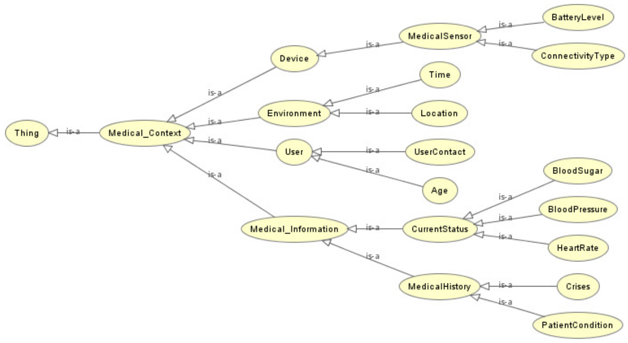

Fig. 3. E-health ontology

After the transformation we obtain an OWL ontology that consists on a set of classes, individuals, properties and relations that describe the various context properties (see Figure 3).

4.3. Context Reasoning Engine:

An inference engine is a software that tries to infer knowledge from a given ontology. Reasoning about Context consists on analyzing the repository of Context in order to derive high level context from low level one according to the context model and the inference rules.

The context reasoning engine is composed of: a Reasoner that allows to infer new situations from relevant context properties, an Adaptation Inference Rule : contains a set of rules ans a Knowledge base.

In our application scenario, context includes data describing the context of a patient. For instance, the low-level context should be converted to high-level context according to the context model and inference rules (see Table 1)

Table 1. Example of rules for generating high level context.

Low level context	High level context	Rules
Blood sugar crisis PatientCond	Normal	If(Disease surveillance=diabete)∧ (Patient Condition=Normal) ∧ (frequency of crisis=Normal)∧ (blood sugar level=Normal)
	Hypoglycemia	If(Disease surveillance=diabete)∧ (Patient Condition=diabetics)∧ (frequency of crisis>Normal)∧ (blood sugar level=Abnormal low)
	Hyperglycemia	If(Disease surveillance=diabete)∧ (Patient Condition=diabetics)∧ (frequency of crisis>Normal)∧ (blood sugar level=Abnormal High)

For Instance, let's take the case of diabetics, these patients should be monitored permanently and the system must detect any type of problems caused by high blood sugar level. Thus when the blood sugar level is in abnormaly diminished level (hypoglycemia) we send a warning alarm to prevent the patient. On the contrary, hyperglycemia may reveal diabetes or Prediabetes, we send a remaining alarm for the patients to take medication. We illustrate different situations according to the blood sugar level. For Hypoglycemia blood sugar level: $< 0.7g/l$, Normal case blood sugar level: $0.7 - 2.5g/l$ and Hyperglycemia blood sugar level : $> 2.5g/l$

5. Service Adaptation

to take advantage of context-awareness (low level context and/or high level context), it's important to have an efficient mechanism to adapt services according to the generated situations and initial context. Thus we propose an extension to OWL-S based on the situations generated by the inference engine to detect the appropriate services to an active situation. Figure 4 illustrate our Semantic Context Aware Service (SCAS). The SCAS is based on the following specifications:

- The SCAS has a ServiceModele, ServiceProfile and ServiceGrounding.
- The service model can be viewed as a process.
- The Process containts AtomicProcess, CompositeProcess or SimpleProcess.
- The ServiceProfile is related to a contextual Situation.
- A situation contains SimpleSituation and CompositeSituation.
- CompositeSituation is composed of other Situation
- Each SimpleSituation is related to contextual parameters: ContextProperty
- Each ContextProperty contains Contextual Attributes : Attribute.
- Each AtomicProcess contains an Adaptation.
- For a given AtomicProcess, a set of AdaptationRule is associated;
- An AdaptationRule can involve the execution of an ordered set of Adaptations.
- An Adaptation is related to a relevent Situation.

For instance, in the E-health motivating scenario (c.f. Sect. 2), the service ControleGlycemia can provoke adaptation by SendingAlarm (i.e. Adaptation) whenever this the situation HyperGlycemia is active. This situation requires three low level context Properties (Contact, BloodSugar and crises).

5.1. Adaptation Tool

To build context aware services, we need to define mechanisms for the selection of appropriate services and the adaptation of their behavior according to the current context situation. The adaptations are eventually conditioned by the existence of relevant situations to the current context. Thus, services must be able, in our case to dynamically adapt its behavior to different situations generated by the inference engine by exploiting only relevant context information.

Figure 6 illustrates the mechanism behind the Adaptation tool. The Request Notifier notifies, in a synchronous or asynchronous mode, the Service Identifier with the active situation in order to recuperate the appropriate services responding to the given situation from the Service Repository. Finally, the interpretation mechanism, operated by the Service Reconfigurer, consists in weaving the required Adaptation Aspects, following a set of AdaptationRules, into the core service to produce the corresponding Context aware Service.

6. Related Work

Managing context is of vital importance for Context-Aware applications architects. SOCAM[5] (Service Oriented Context-Aware Middleware) is a middleware for context-awareness that provides an infrastructure for creating context-aware applications. The adaptation is not taken into account and left to the developers who should implement the adaptations of each service. The work presented in[7] by Ou et al., attempts to obtain complete functional context-aware pervasive systems by following an MDE strategy. To do this, they define a set of metamodels and a Model Driven Integration Architecture to integrate these metamodels and generate CAA implementations. However, they do not give support to the adaptation of service behavior and nor context metamodel is proposed. Vale[6] proposes

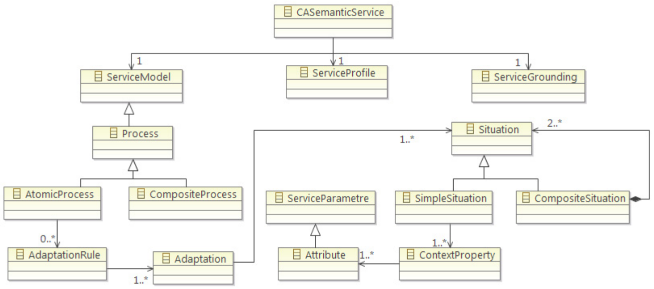

Fig. 4. Semantic Context Aware Serivces Description by extending OWL-S

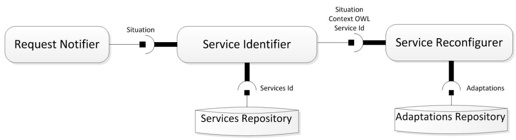

Fig. 5. Adaptation Tool

Model Driven Development to promote reuse, adaptability and interoperability in context-aware applications development.But, they do not give support to the adaptation of service and he is not take in charge the reasoning mechanism in order to profit from the power of using ontologies In our work we propose an ontology-based strategy to give support to the development of context-aware systems. Using Ontologies guarantees a high degree of expressiveness, formality, semantic richness and applying reasoning techniques.

7. Conclusion

In this paper, we present A Semantic Approach for service adaptation in context-aware environment. Thus, we proposed, firstly, a context metamodel which is generic and open to allow its extension to various domains depending on needs. Then we perform a model transformation to generate an OWL ontology that represents context information. The generated OWL ontology is then used to extract high level adaptations situations. Secondly, we proposed a semantic context-aware service, represented by extending OWL-S. Our approach make use of the power of ontologies to represent and reason about context and perform service adaptation to users needs according to their context. Currently we are evaluating our Semantic Approach and we are completing the architecture of our Selection and Adaptation Tool. In the medium term, our objective is to propose a framework allowing the Semantic Context Aware Service development.

References

1. Chen, H., Perich, F., Finin, T., Joshi, A. (2004, August). Soupa: Standard ontology for ubiquitous and pervasive applications. In Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on (pp. 258-267). IEEE.
2. Metamodel, O. D. (2009). Version 1.0. Object Modeling Group.
3. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., ... Sycara, K. (2004). OWL-S: Semantic markup for web services. W3C member submission, 22, 2007-04.
4. Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., Steggles, P. (1999, January). Towards a better understanding of context and context-awareness. In Handheld and ubiquitous computing (pp. 304-307). Springer Berlin Heidelberg.
5. Gu, T., Wang, X. H., Pung, H. K., Zhang, D. Q. (2004, January). An ontology-based context model in intelligent environments. In Proceedings of communication networks and distributed systems modeling and simulation conference (Vol. 2004, pp. 270-275).
6. Vale, S., Hammoudi, S. (2008). Context-aware model driven development by parameterized transformation. Proceedings of MDISIS.
7. Ou, S., Georgalas, N., Azmoodeh, M., Yang, K., Sun, X. (2006, January). A model driven integration architecture for ontology-based context modelling and context-aware application development. In Model Driven ArchitectureFoundations and Applications (pp. 188-197). Springer Berlin Heidelberg.