OTM 2009 Workshops. Vilamoura, Portugal, 2009

FleXConf: A Flexible Conference Assistant Using Context-Aware Notification Services

Nikos Armenatzoglou, Yannis Marketakis, Lito Kriara, Elias Apostolopoulos, Vicky Papavasiliou, Dimitris Kampas, Alexandros Kapravelos, Eythimis Kartsonakis, Giorgos Linardakis, Sofia Nikitaki, Antonis Bikakis, and Grigoris Antoniou

Institute of Computer Science, FORTH, Crete, Greece, and
Department of Computer Science, University of Crete, Greece
{armenan,marketak,kriara,ilapost,papavas,kampas,kapravel,kartson,linard, nikitaki,bikakis,ga}@csd.uoc.gr

Abstract. Integrating context-aware notification services to ubiquitous computing systems aims at the provision of the right information to the right users, at the right time, in the right place, and on the right device, and constitutes a significant step towards the realization of the Ambient Intelligence vision. In this paper, we present *FlexConf*, a semantics-based system that supports location-based, personalized notification services for the assistance of conference attendees. Its special features include an ontology-based representation model, rule-based context-aware reasoning, and a novel positioning system for indoor environments.

Keywords: context awareness, location-based services, notification services, context ontology, rule-based reasoning.

1 Introduction

Context awareness and notification services have recently gained a lot of attention among researchers, and have been used in various application domains, including Ambient Intelligence. In brief, context awareness refers to the idea that computers can both access context information through sensors, and react to certain context changes based on policies or an intelligent stimulus [1]. On the other hand, notification systems use alert services in order to inform users about specific events or changes in the user's context in a timely manner.

The aim of Ambient Intelligence systems is to provide the right information to the right users, at the right time, in the right place, and on the right device. In order to achieve this, a system must have a thorough knowledge and, as one may say, *understanding* of its environment, the people and devices that exist in it, their interests and capabilities, and the tasks and activities that are being undertaken. All this information falls under the notions of *context*.

In this paper, we describe *FlexConf*, a flexible Ambient Intelligence system, which combines context-awareness and notification services to cover the needs of conference organizers and attendees. Exploiting the advantages of Semantic

R. Meersman, P. Herrero, and T. Dillon (Eds.): OTM 2009 Workshops, LNCS 5872, pp. 108–117, 2009.

Web technologies, such as Web ontology languages and rule languages and systems, the described system integrates various types of context information that is relevant to the organization of a conference, and provides personalized context-aware notifications via e-mail about upcoming events. Ontology languages have been argued to be the perfect choice for context representation in the Ambient Intelligence domain [2], mainly because they offer enough representational capabilities to develop a formal context model that can be shared, reused, extended, but also combined with data originating from various diverse sources. Moreover, the development of the logic layer of the Semantic Web has recently resulted in expressive rule languages and powerful rule systems that enable reasoning with the user's needs and preferences and with the available ontology knowledge. Rule languages provide a formal model for reasoning on the context data, while rules are easy to understand and widespread used.

Overall, the main contribution of the paper is twofold: (a) to demonstrate how the combination of Semantic Web technologies and context-aware services can effectively support the needs of conference assistance systems; and (b) to highlight the advantages of this approach including its flexibility, scalability and extensibility. We have already implemented a prototype of *FlexConf*, which operates in the premises of FO.R.T.H. (Foundation for Research and Technology-Hellas) research facilities. In this prototype, location sensing is enabled by CLS [3,4], a novel positioning system exploiting the existing IEEE802.11 wireless infrastructure. However, the flexibility in the system design enables deploying *FlexConf* anywhere that a location sensing subsystem is available to provide the necessary location information.

The rest of the paper is organized as follows. Section 2 discusses related work on context-aware notification systems. Section 3 describes a use case scenario that highlights the desired functionality of a conference assistance system. Section 4 describes an ontology-based context representation model and a rule-based model that enables reasoning with the available knowledge. Section 5 provides a description of the system architecture, while Section 6 summarizes and discusses plans for future work.

2 Related Work

Several recent works have focused on systems that offer context-aware notification services to the members of a specific community. The Mass Notification System described in [5] aims at the assistance of students in a university campus through recommendations based on user preferences and user location, task reminders, and support for collaborative applications. The Library SMS Alert Service developed in the Hong Kong Institute of Education [6] integrates mobile phone SMS and Google Calendar technologies to provide simplified versions of important library notices, such as availability of requested items and overdue reminders. *eyeJot* [7] is a context-aware campus information system that supports information posting for news, activities and schedules using Short Message Service (SMS). All these approaches share a common deficiency. The lack of a formal model for modeling and reasoning about the relevant context information influences their flexibility, as the decision-making processes of those systems are hardcoded, and therefore difficult to adjust.

There is also a number of recent works that exploit Semantic Web technologies to support context-aware services. [8] presents an extended survey of such systems. The most relevant to our work is the semantics-based meeting alerting system of [9], which integrates RDF [10], and Semantic Web rules in Defeasible Logic for making context-dependent decisions, GPS technology for location sensing and SMS message delivery. Though the underlying representation and reasoning models are similar to those used in *FlexConf*, the specific system is aimed at different types of applications and outdoor environments.

Finally, to our knowledge, the *Conference Assistant* prototype described in [11] as an application of Dey and Abowd's *Context Toolkit* is the only implemented system that provides context-aware services to conference attendees. Compared to *FlexConf*, it also takes into account user's interests and profile information to recommend specific events that take place during a conference, without however supporting notifications / alerts to conference attendees.

3 Use Case

In this section an imaginary use case scenario is presented, in order to explore the functionalities that a context-aware conference assistant should support.

Consider a Knowledge Representation Conference, which takes place in the FORTH conference area. This area includes several meeting, seminar and other types of rooms that can be used to host paper presentations, invited talks, demonstrations and discussions.

Consider also a visitor, Mark, that has registered to attend the conference, and enters the conference area for the first time. Mark updates the conference assistant with personal information by filling up his profile (i.e. name, role in the conference, research interests, physical disabilities, group memberships). Mark states that he has movement disabilities (he uses a wheelchair). He also states that his research interests include semantic web (SW).

According to the conference programme, a presentation on RDF concepts is scheduled to take place in Seminar Room VI at 12.15 pm. The system should inform Mark in time for this presentation (as it matches with his research interests) and guide him towards Seminar Room VI taking into account his disabilities. To accomplish that, the system provides a floorplan of the whole conference area indicating the users current position, the exact location of room VI, as well as how the room is accessible to people using wheelchair.

Consider, finally, another conference attendee, James, who is a member of SW-Group. James should also receive an alert for the same presentation due to common interests of SW-Group members.

4 Context Representation and Reasoning

In this section, we describe the ontology-based context representation model and the rule-based reasoning methods that we employed to enable context-based decisions.

4.1 Ontology-Based Modeling

For the representation of people, events, interests and other conference related concepts, we designed an ontology (*ConfOntology*) in RDFS [12]. *ConfOntology* can be used as a standard format for representing and exchanging information, for the domain of Conference Assistance Systems. Figure 1 illustrates the proposed ontology schema and a hypothetical instantiation.

The upper part of Figure 1 depicts the schema of *ConfOntology*, while below the schema there is a possible instantiation, which we use as the running example throughout the rest of the paper and which is based on the use case scenario that we described in Section 3. The RDF classes are represented as boxes. The labeled arrows stand for properties having as domain the class connected to the beginning of the arrow, and as range the class connected to its end. The label of an arrow stands for the name of the respective property. Unlabeled bolded arrows denote subclass relations - the class connected to the beginning of the arrow is a subclass of the class connected to its end. Finally, dashed arrows denote class instantiations.

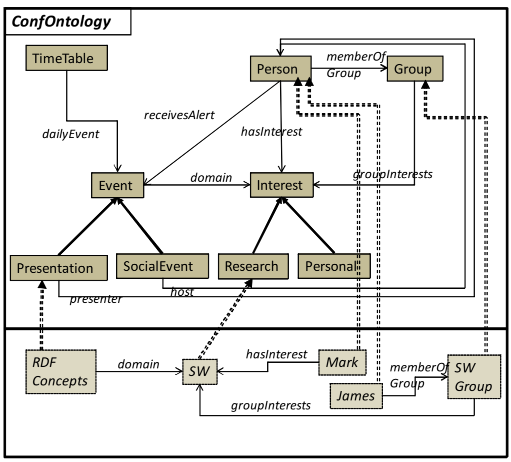

Fig. 1. The ConfOntology Schema and instantiation

The main ontology classes are *Person*, *Event*, *Group*, *Interest* and *TimeTable*. The *Person* class models conference attendees. For each person, various types of relevant information are retained. One such type is a person's interests (in terms of both personal and research interests), which are modeled through *Interest* class, and are associated to a person through the *hasInterest* property. The *Event* class is used to represent events related to the conference program, such as presentations, lectures, demonstrations or talks. It is also used for other types of activities such as excursions or museum visits. The *Group* class is used for groups of people that share common interests, which are linked to groups through the *groupInterests* property. For instance, according to Figure 1, both Mark and James will receive an alert for the *RDF Concepts* presentation, as Mark has explicitly included *Semantic Web*, which is the domain of the event, in his research interests, while James is member of *SWGroup*, which is also linked to *Semantic Web* through *groupInterests*. Finally, the *TimeTable* class is used to group events and deliver their schedule to the users.

For the creation of the ontology, we used the ontology editor Protégé[1]. Protégé, besides the editor, provides a suite of tools that support the creation, visualization, and manipulation of ontologies in various representation formats.

4.2 Rule-Based Reasoning

Ontology languages provide some limited forms of reasoning through specially designed query languages. However, these languages can not cover the needs of large-scale context aware systems, as they are primarily focused on information retrieval. More expressive forms of reasoning are required for making context-dependent decisions. To this end, we designed a separate rule-based reasoning subsystem. The rules that we use follow the form described below.

$$[R : \{list\ of\ predicates\} - > conclusion]$$

R denotes the *id* of the rule. The predicates used in the body of the rule are matched with RDF triples in the knowledge base. Finally, the conclusion in the head of the rule is used by the system to determine the appropriate actions, as described in Section 5. Below, we describe two representative examples.

$$[R_1 : (?presentation : Presenter?person) \\ \rightarrow (?person : receivesAlert?presentation)]$$

$$[R_2 : (?presentation : PresentationDomain?interest), \\ (?person : HasInterests?interest), (?person : wantsAlerts'yes') \\ \rightarrow (?person : receivesAlert?presentation)]$$

R_1 is used to alert presenters of presentations, while a similar rule is used to alert hosts. R_2 is used to alert people that their interests match with the domain of a presentation, and who have stated that they wish to receive system alerts. Similar rules are used for alerting people about events with domains that match with the common interests of the groups that those people are members of.

5 Technical Description

In this section, we describe the system architecture and provide important implementation details. A component diagram of the system architecture is depicted in Figure 2. Overall, the system comprises the following components: (i) the *User-Interface*, through which users users can interact with the system; (ii) *SWKM*,

[1] http://protege.stanford.edu/

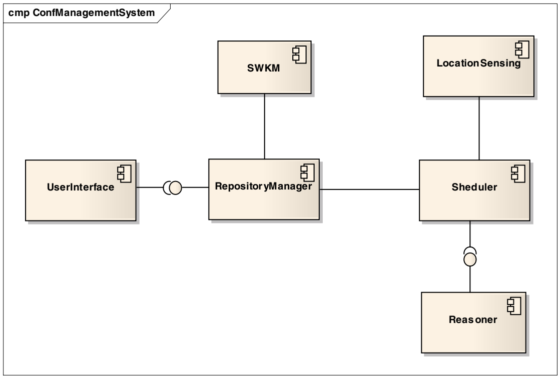

Fig. 2. The Architecture of the system

which is a persistence storage RDF mechanism, and *RepositoryManager* that acts a mediator between SWKM and the rest of the system; (iii) the *Reasoner*, which is used to execute the rules and determine about the appropriate system actions; (iv) the *Location Sensing Subsystem*, which provides the exact position of the user in the physical space; and (v) the *Scheduler*, which is responsible for the management of notifications. In the rest of the section, we describe each component in detail.

5.1 User Interface

The UserInterface constitutes the interaction means between the users and *FlexConf* (see Figure 3). The UserInterface does not require any specific software; it is a web application, and thus is accessible through any web browser. *FlexConf* identifies and supports two types of users; common users and system administrators. Through UserInterface, a common user may import or update information about personal preferences and groups memberships. He can also have access to the conference program and view details about specific events. Finally, he has the option to disable alerts, so that the system does not include him in the list of the potential notification recipients, and stops positioning him in the

Fig. 3. User Interface

conference area. Administrators are users with appropriate privileges that allow them to perform additional actions, such as creating or updating information about events, groups, or interest categories.

5.2 SWKM and Repository Manager

The Semantic Web Knowledge Middleware (SWKM)[2] is a persistence storage RDF mechanism. It provides a set of services that aim at the manipulation of a database containing RDF data. The basic set of services consist of *import*, *export*, *query* and *update* services, through which the user can import/export RDF schemas and RDF files and also query/update the RDF knowledge base.

The benefits of adopting such a SW-based repository is that we can exploit its validation mechanisms, and therefore ensure that the ontology is consistent with the RDF standard and the instances are valid according to the schema, and its declarative query and update languages. For example, in the future one might want to extend the core ontology with a class *Accommodation* to support information about the hotels that accommodate the conference attendees. Moreover, let's assume that this class is added manually (and not using a tool like Protégé). During import time the extension will be examined and in case it is valid, it will be immediately imported in the ontology schema. Otherwise, the user will be prompted for the errors in the RDF schema.

The Repository Manager acts as mediator between the system and SWKM. Specifically, through the Repository Manager, SWKM interacts with UserInterface in order to update the knowledge base, and with the Scheduler to retrieve information about users' profile information and scheduled events.

5.3 Reasoner

The reasoner subsystem uses the available context information and rules such as those that we described in Section 4.2, to determine the people that should be informed about certain upcoming events. For the reasoning tasks of our system, we used the Jena2 Java framework[3]. The Jena2 inference subsystem is designed to allow a range of inference engines or reasoners to be plugged into Jena. The primary use of this mechanism is to support the use of ontology languages, such as RDFS and OWL [13], which allow additional facts to be inferred from instance data and class descriptions. Jena2 also includes a general purpose rule-based reasoner, which we used to perform rule-based reasoning for the notification of users, using the rules that we described in Section 4.2.

5.4 Location Sensing

In our system prototype, we used a novel positioning system, the Cooperative Location-sensing system (CLS) [3,4], which exploits the IEEE802.11 network,

[2] http://139.91.183.30:9090/SWKM/
[3] http://jena.sourceforge.net/

mainly due to the wide popularity of the network, the low deployment cost, and the advantages of using it for both communication and positioning. (CLS) employs the peer-to-peer paradigm and a probabilistic framework to estimate the position of wireless-enabled devices in an iterative manner without the need for an extensive infrastructure or time-strenuous training. CLS can incorporate signal-strength maps of the environment to improve the position estimates. Such maps have been built using measurements that were acquired from access points (APs) and peers during a training phase.

CLS adopts a grid-based representation of the physical space; each cell of the grid corresponds to a physical position of the physical space. The cell size reflects the spatial granularity/scale. Each cell of the grid is associated with a value that indicates the likelihood that the node is in that cell. For our needs, we divided the FORTH testbed into certain zones (areas containing multiple cells). CLS computes an estimated position iteratively every 3-4 seconds. In each iteration a file with a predefined name containing the zone ID of the user's estimated position is updated. An empirical evaluation of CLS is presented in [3,4].

5.5 Scheduler

The Scheduler is responsible for checking upcoming events and for notifying the people that are responsible for an event or may be interested in it by sending them alerts via e-mail. It runs as a daemon process that periodically checks whether there are any events in a subsequent time interval. The period of checks and the time interval can be configured based on the needs of a specific conference. For example, wider conference areas impose the need for longer time intervals, while dense conference programs require shorter checking periods.

The responsibilities of the Scheduler include the coordination of the rest of the system components and the management of notifications. Initially, when the process "wakes up", it communicates with the Repository Manager to retrieve information about upcoming events. If there are no events scheduled for the subsequent time interval, it suspends for the predefined period. Otherwise, it contacts the Reasoner to retrieve information about the people that should be alerted, as well as additional information that will be included in the body of the notification e-mail (i.e. information for the event, location of the event). It also contacts the Location Sensing Subsystem to retrieve the user's location. Using this information, it creates a floorplan of the area, marked with the user's current location and the relevant (according to upcoming events and user's interests) conference rooms, and attaches the floorplan to the notification e-mail. After sending the appropriate alerts, the Scheduler suspends for the predefined period.

Back in our running example, the *RDFConcept* presentation has been scheduled to start at 12.15 pm. in Seminar Room VI. Assuming that the time interval has been configured to 60 minutes, and that at 11.30 am. the Scheduler checks for upcoming events, the *RDFConcept* presentation will be included in the list of upcoming events. Based on Mark's interests that include *Semantic Web*, James' group memberships that include *SWGroup*, and the domain of the presentation, which is *Semantic Web*, the system (Reasoner) determines that both Mark and

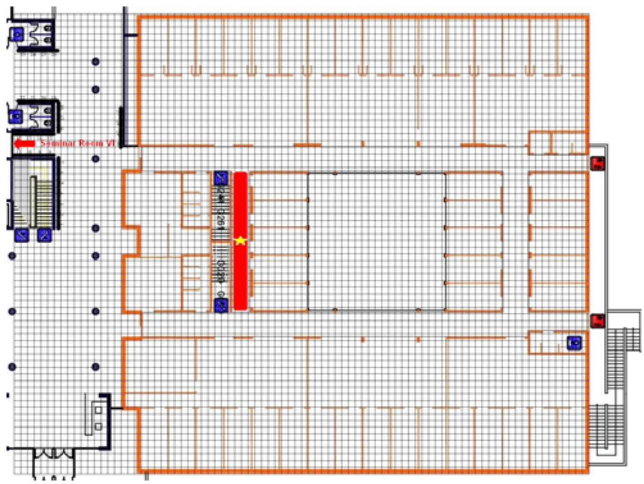

Fig. 4. The floor plan for Mark

James should be alerted about the *RDFConcept* presentation. The next step is to localize Mark and James using the Location Sensing component in order to deliver the appropriate floorplans. Figure 4 presents the floorplan that will be created for Mark. In the floorplan, the cell that Mark is located at is filled with red, and Mark's current position is denoted ny a star. Additionally, the position of the Seminar Room VI is described by a red arrow pointing at the entrance of the room, followed by the name of the seminar room. Finally, the system creates and sends the appropriate e-mail notifications to them.

6 Conclusions and Future Work

In this paper we described *FlexConf*, a flexible conference assistant that integrates Semantic Web technologies, a location sensing system, and relevant context information to support personalized, context-aware notifications to conference attendees. The main features of the system include: (i) a semantics-based knowledge representation and reasoning model, (ii) a flexible design, which enables using the system anywhere that a location sensing system is available and (iii) personalized notifications that are delivered to conference attendees in a timely manner, and are created according to various context parameters.

In the future we plan to extend *FlexConf* in various ways, taking into account the needs of conference organizers and attendees. First of all, we plan to employ the peer-to-peer paradigm for users to communicate. This will allow them to share files, exchange ideas or even use instant messaging through the platform. Furthermore, we plan to continue our work on sending alerts via Bluetooth or SMS to mobile devices e.g. PDAs and mobile phones. A third plan is to integrate navigation services, which will more effectively assist conference attendees to find their way in the conference area. Finally, integrating more context parameters, such as personal calendars, privacy preferences and several types of sensory information will enable us to support more advanced notification, as well as other types of context-based services. Our future plans also include a more complete evaluation of the system. Currently, only the Location Sensing subsystem has been extensively evaluated. Additionally, we plan to evaluate the overall system performance, as well as the usability of *FlexConf*.

References

1. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: Workshop on Mobile Computing Systems and Applications, 1994. Proceedings, pp. 85–90 (1994)
2. Schmidt, A.: Interactive Context-Aware Systems Interacting with Ambient Intelligence. In: Riva, G., Vatalaro, F., Davide, F., Alcaniz, M. (eds.) Ambient Intelligence. IOS Press, Amsterdam (2005)
3. Fretzagias, C., Papadopouli, M.: Cooperative location-sensing for wireless networks. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications, PerCom 2004, pp. 121–131 (2004)
4. Vandikas, K., Katranidou, A., Kriara, L., Baltzakis, H., Papakonstantinou, T., Papadopouli, M.: Empirical-based analysis of a cooperative location-sensing system. In: Proceedings of the 1st international conference on Autonomic computing and communication systems (2007)
5. Mass Notification Systems for College: University & Higher Education Schools by e2Campus: Info On The Go!, Omnilert LLC (2009), `http://www.e2campus.com/`
6. Library SMS Alert Service: The Hong Kong Institute of Education (2007), `http://www.lib.ied.edu.hk/collection/sms.html`
7. Al Takrouri, B., Canonico, A., Gongora, L., Janiszewski, M., Toader, C., Schrader, A.: eyeJOT-A Ubiquitous Context-aware Campus Information System. In: 2nd International Conference on Pervasive Computing and Applications, ICPCA 2007, pp. 122–127 (2007)
8. Bikakis, A., Patkos, T., Antoniou, G., Plexousakis, D.: A Survey of Semantics-based Approaches for Context Reasoning in Ambient Intelligence. In: Constructing Ambient Intelligence, AmI 2007 Workshops Proceedings. CCIC, vol. 3774, pp. 14–23. Springer, Heidelberg (2008)
9. Antoniou, G., Bikakis, A., Karamolegou, A., Papachristodoulou, N.: A context-aware meeting alert using semantic web and rule technology. International Journal of Metadata, Semantics and Ontologies 2(3), 147–156 (2007)
10. Lassila, O., Swick, R.: Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, World Wide Web Consortium (1999)
11. Dey, A.K., Abowd, G.D., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction 16(2), 97–166 (2001)
12. Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, World Wide Web Consortium (February 2004)
13. van Harmelen, F., McGuiness, D.: OWL Web Ontology Language Overview. W3C Recommendation, World Wide Web Consortium (February 2004)