CORe S: Context-aware, Ontology-based Recommender system for Service recommendation

André C. M. Costa[1], Renata S. S. Guizzardi[2] Giancarlo Guizzardi[1,3], and José Gonçalves P. Filho[1]

[1] Department of Computer Science, UFES, Vitória-ES, Brazil
[2] ITC-irst, Trento-Povo, Italy
[3] Laboratory of Applied Ontologies (ISTC-CNR), Trento, Italy

Abstract. Advances in telecommunications and information technology have allowed the proliferation of mobile and multifunctional devices and their incorporation more and more into physical objects, making new information and services available. A consequent problem of this new scenario is information overload, i.e. users face vast and distributed information sources, and have difficulty in selecting those that satisfy their needs and interests. To help users, recommender systems can be applied. Moreover, context-aware systems may collaborate with recommender system, improving recommendations, thus benefiting users with more personalized and contextual results. This work explores the synergy between recommender systems and context-aware computing, describing the development of CORe S (Context-aware, Ontology-based Recommender system for Service recommendation). This recommender system broadens the capabilities of the Infraware context-aware platform, by making service offer more efficient, personalized and proactive.

1 Introduction

Advances in telecommunications and information technology have transformed the use and the shape of computers, coming from the traditional static workstations to the proliferation of mobile and multifunctional devices [1], [2], [3]. Moreover, the computer tends to incorporate more and more into physical objects, such as electronics, furniture and other artifacts of general use, providing more functionality to these artifacts and becoming transparent to the user [3]. This context inaugurates new ways of accessing and providing information and services.

A problem inhering in this new setting is information overload. Users are left with the difficult task of selecting from a growing and distributed information repository, those that satisfy their immediate interests and needs. To alleviate this overload, one can apply recommender systems, which are generally defined as systems that assist users in selecting items of their interest or need from a big set of items, helping users to overcome the overwhelming feeling when facing

a vast information source, such as the web, an organizational repository or the like [4], [5], [6].

Besides recommender systems, research on context-aware computing seems promising. Context-awareness allows software applications to use information beyond those directly provided as input by users [7]. Examples of such information are location, date and time [1], [2]. This information becomes important in an environment where applications are accessed through mobile and ubiquitous devices that communicate with each other. More specifically, automatic sensing of context information facilitates the use of these applications by minimizing user intervention, besides increasing their independency in gathering and processing relevant information.

This work has a general goal of facilitating services selection, exploring the synergy between these two research areas: recommender systems and context-aware computing. As basis for this initiative, we aim at using the results of a previous work [8] on a particular recommender system. In addition to that, this project will benefit from the use of a context-aware services platform named Infraware [2]. More specifically, this paper describes the development of a recommender system called COReS (Context-aware, Ontology-based Recommender system for Service recommendation), which uses the capabilities of the Infraware platform to support services selection, satisfying the needs of the user in a particular context. This way, COReS broadens Infraware's capabilities, making service offer by this platform more efficient, personalized and proactive.

The remaining of this paper is organized as follows: section 2 provides some background information along with the main motivations behind our work; section 3 proposes COReS, clarifying the gains it brings to the Infraware platform; section 4 describes COReS architecture and how its components interoperate with the other Infraware's components; section 5 focuses on COReS's recommendation techniques; section 6 describes related work; and at last, section 7 concludes this paper.

2 Background and Motivation

This section provides background information on the two main focus areas in our research, namely recommender systems (2.1) and context-aware computing (2.2). Moreover, section 2.3 presents our motivations to integrate results of these two areas, and justifies the application of domain ontologies to support context-aware service recommendation.

2.1 Main Recommendation Techniques

A recommender system produces customized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options [4]. So, one of the goals of a recommender system is to help the user in finding interesting objects, based on interests, preferences and personal characteristics.

Such systems perform some similarity measure between object-object, object-user and user-user, with the purpose to determine a set of objects to be recommended to a specific user. The result can a prediction, i. e., a numeric value given to each object that expresses likeliness of an object for the user; or a recommendation, i. e., a top-N list of objects [9].

Recommendations may be based on similar items to those a given user has liked in the past (content-based recommendation); or on items owned by users whose taste is similar to those of the given user (collaborative recommendation). Combining content-based and collaborative recommendations originate hybrid approaches, which are commonly used, considering that both types of recommendations may complement each other [10], [6].

Besides the difference given by the aforementioned recommendation approaches, recommender systems are also differentiated by [5]: the items they recommend (systems have been developed to recommend web pages [10], movies [11], etc.); the nature of the user profile they use to guide the recommendations (e.g. history of items accessed by the user, topics indicating user interest, etc.); the recommendation techniques (mainly, how the user model is represented, what kinds of relevance mechanisms are used to update the user model, and which algorithm is used to generate recommendations); and the recommendation trigger, i.e. whether the recommendation is started by the user or by the proactive behavior of the system.

2.2 Context-awareness

According to [7], context is any information that can be used to characterize the situation of entities (person, place or object) that are considered relevant for the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects.

Ubiquitous Computing [3] suggest the development of new context-aware applications, context-aware, created to use contextual information to dynamically select or execute action that better assist the user in his needs. Interaction among user and application is minimized with the perception and use of contextual information. Different from traditional applications, context-aware applications consider, in it's decision taking and processing, not only the inputs provided explicitly by the user, but also implicit inputs, related to computational and physical context of users and environments.

2.3 Context-aware recommender systems supported by domain ontologies

Collaborative filtering recommender systems usually use the whole user profiles (the whole set of rated objects) to calculate similarities among them. In situation where the recommender system should recommend objects in various domains, this may lead to mistakes, once the system assume that if two users have the same preferences in a certain domain, so in others domains their preferences will

also resemble. But, in many cases, this is not true. For example, suppose that two people like the same kind of sports, music and movies, but differentiate their taste for food. While one likes Arabian food, the other prefers Italian. In traditional collaborative recommender systems, while the latter intends to go out to dinner, he may be surprised with a suggestion to go to an Arabian restaurant. This happens because all the information in the user profile is considered to calculate similarities between users, independent of domain and context. Such mistake could be avoided if the recommender system became aware of the user context, going out to dinner in this case, and connected this with a specific domain, i.e. culinary.

This benefit is not unilateral. Context-aware platforms may be also benefited by the use of recommender systems, mainly to support recommendation of the services offered by the platforms. In this case, the user would not need to manually request the platform for the services in which he is interested. These services would be automatically recommended based on the user's context and profile. It is important to realize that the central goal of context-aware systems and recommender systems is the same, i.e. providing users with relevant information and/or services selected from a potentially overwhelming set of choices [12]. The difference is that in the former one, the selection is based on the user's context while in the latter, it relies on the user's interest. This suggest that these kinds of systems are rather complementary than competing, hence motivating their integration.

Going back to the previous example, while the contextual information (i.e. going out to dinner) can be provided to the recommender system by a context-aware platform, the domain information (i.e. culinary) may come from domain ontologies. An ontology is a conceptual model that can be applied to describe a domain of discourse, modeling it as a set of concepts and relations [13]. Ontologies are generally used to provide a uniform conceptualization of terms. Concerning recommender system, it can be applied in support to both collaborative and content-based recommendations. For collaborative recommendations, as in the example we have just saw, it is used in the description of user profiles, representing their interests, needs and personal characteristics. Profiles built under the same domain ontology avoid problems of synonym and homonym. Analogously, domain ontologies may also be useful to establish a common conceptualization to describe or classify recommended objects in content-based recommender systems. In such systems, if the words used to describe objects are syntactically different, but semantically equivalent, some of these objects will not be recommended. And on the other hand, unwanted objects are likely to be recommended in case the words are syntactically the same, however semantically different. We may say that domain ontologies enable semantic matching of objects and profiles, instead of a simple keyword-based matching.

The use of domain ontologies can augment and enrich the description of contextual information, besides supporting the use of suitable reasoning mechanisms. An example of the use of ontologies in context-awreness may be found in theInfraware system [14], [2].As an illustration, take for instance the diagno-

sis service prototyped in the platform. After consulting with a doctor, the user needs to do a computed tomography to understand the reason behind a backache. Holding his PDA, he submits his request to Infraware. The user would also like that the clinic is located in a 5 km radius from his present location, and that it has private parking. And finally, the user must inform his health insurance of how he intends to pay for the exam. The system, through the healthcare domain ontology, infers that computed tomography is a sort of image diagnosis. Then the system looks through all clinics that perform images diagnosis, and selects those which comply with the user's distance and parking preference, supported by the contextual information ontology. At last, the payment ontology supports the user in choosing the type of payment.

3 COReS (Context-aware, Ontology-based Recommender system for Service recommendation)

Our purpose is to develop a service recommender system to function in integration with the Service Manager component of the Infraware context-aware service platform [14], [2]. Infraware is a middleware based on Web Service technology with architectural support for the construction and execution of context-aware mobile applications. The platform conceptual architecture is an extension of WASP platform [15] and is being developed at LPRM (Multimedia and Network Research Laboratory) in the Computer Science Department of the Federal University of Espirito Santo, Brazil.

Figure 1 illustrates Infraware's general architecture.

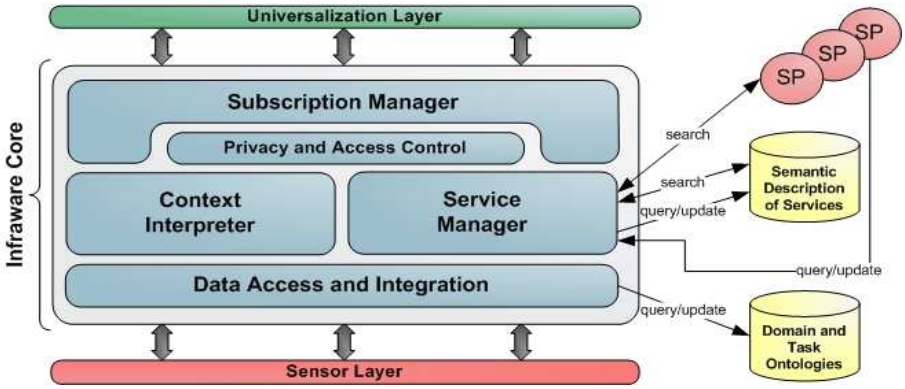

Fig. 1. The Infraware platform's general architecture

The Infraware's architecture uses Semantic Web technologies, like OWL and RDF markup languages [16], [17], and introduces a new approach based on task and domain ontologies to discover, select and compose services. Figure 2 shows

a piece of the healthcare domain ontology used by Infraware. Besides, Infraware has a specific layer that receives and treats subscriptions coming from the applications, and a component specially concerned about privacy and access control. Additionally, Infraware solves the problem of heterogeneous data access and integration via a dedicated infrastructure, and infers new information through context interpreter.

The Service Manager is the component responsible for publication, discover, selection and composition of services, carried out by the Secure Context-aware Service Discovery Protocol (SCaSDP) [18] which implements the Service Matching Algorithm [19]. At present, every time a user wants to use a service offered by Infraware, he must submit a request to the platform. In other words, the Service Manager function in a reactive way, i.e., it only attempt to find and offer a service when the user makes a request. In some situations, it would be desirable to let the system proactively assist the user, anticipating his needs and offering services alternatives related to the user's context, instead of waiting for a service requisition. This functionality may be handled by a recommender system coupled with the Service Manager, thus providing this component with proactiveness. In service repository of Infraware may exist a great number of services advertisement, so depending on the user's request, he could receive, as a result, a large list of services, once all services that match the request are selected. So another advantage of including a recommender system in the Service Manager is to help the user in the choice of the adequate service, showing him only interesting services, according to his profile and other users' profiles, instead of the whole list.

Many recommender systems have been developed for specific domains, like movies [11], music [20], books [21], news [22] and others. However, those systems do not consider context in their recommendation process, which makes the use of such systems in Infraware unfeasible. Another strong requirement of our system, is the use of domain ontologies to enable efficient services recommendations, as highlighted in section 2.3.

These reasons led to the proposal of COReS (Context-aware, Ontology-based, Recommender system for Service recommendation). COReS is a hybrid system that besides combining content-based and collaborative recommendations, apply some other techniques, such as stereotypes and knowledge-based approaches. This combination has the main purpose of eliminating some problems existing in each individual technique and improving the quality of the recommendation.

Another novelty of COReS is the adoption of compartmentalize the user profile according to different domains, selected in time of prediction, based on the user's context. Section 2.3 explains this idea in detail and presents the problems that happen as a consequence of a unified view of the profile, as adopted in many content and collaborative-based recommender systems. Our proposal applies some results of the work on foundational ontologies [23], [24]. The full description of this feature is out of the scope of the present paper and should thus come forward in future publications.

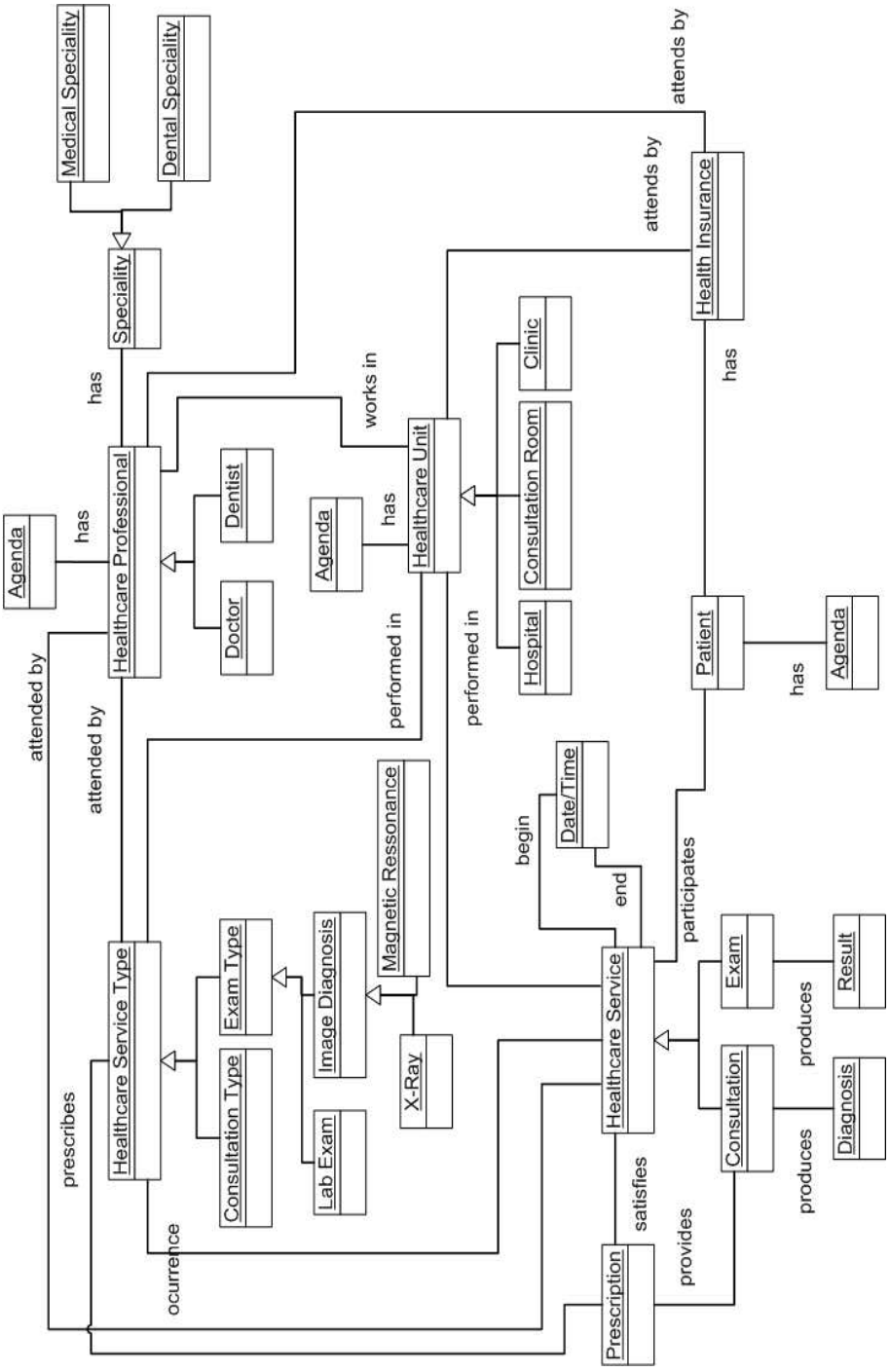

Fig. 2. The healthcare domain ontology used by Infraware

Domain ontologies are used in COReS to assist in the recommendation process and in the comparison of user profiles. In the former, they are used to infer services that may be recommended according to the user's context. In the latter, they provide additional information to the recommender when the user does not have information related to the service in his profile.

User profile is composed of a set of domain ontologies that express user's interests and include his preferences and documents related to each concept. Figure 3 shows the profile of two users.

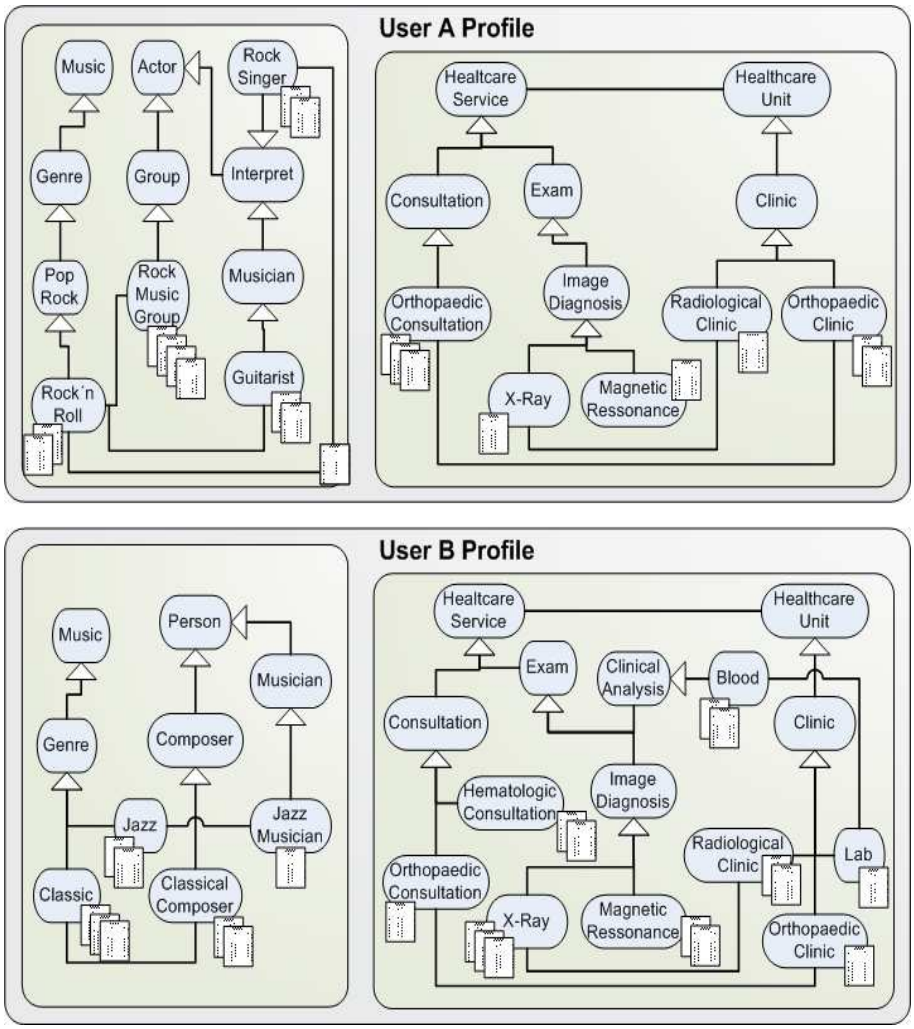

Fig. 3. User Profile

4 Interoperating CORe-S and the Infraware Platform

COReS's recommendations result from mapping user profile and context into services. Since the system is being developed to integrate Infraware, this platform remains responsible for gathering user context, besides inferring the user's implicit needs and wants of the moment. To gather input information and provide its functionalities, COReS mainly interacts with the Service Manager. Hence, Figure 4 presents interactions of COReS with the Service Manager's sub-components. For purposes of simplification, this figure disregards the interactions of the Service Manager with other Infraware components.

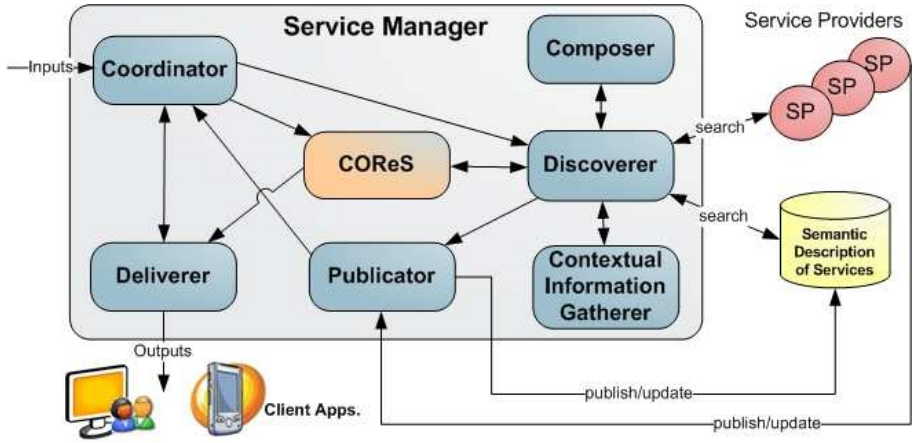

Fig. 4. Service Manager Components

The Service Manager originally had six sub-components: the Coordinator, the Discoverer, the Composer, the Contextual Information Gatherer, the Publisher, the Deliverer. COReS is now added as a new sub-component.

The Coordinator is responsible for receiving all inputs (requests, events, tasks, etc.) coming from other Infraware modules, later delivering these inputs to the adequate component. If an input can not be immediately attended, it can be stored in the Coordinator to be handled in the future. The Coordinator also notifies the Deliverer about a new input, or sends a task to be executed by that component.

The Discoverer manages to whole process of service discovery, including searching for services in the providers and the service repository, requesting for contextual information and handling requests from the Composer.

The Composer is responsible for the semantic composition of services. If the Discoverer does not find a service that fulfills a request, this component sends to it to the Composer. Upon receiving the request, the Composer first analyzes the request, breaking it into many tasks (small services) necessary to orchestrate the

whole service, then asks the Discoverer to find these tasks, and finally composes the service.

The Contextual Information Gatherer connects to the Data Access and Integration module and collects some contextual information that the Discoverer needs to accomplish the service discovery.

The Publisher carries out the publication and update of service descriptions, either requested by the Service Providers or by the Discoverer. In the former, Service Providers send the description directly to the Publisher, while in the latter, the Discoverer requests the publication of the description of a service just composed by the Composer. In both cases, the Publisher validates the description being published.

The Deliverer is responsible for getting in contact with client applications to send them the result of a discovery or to execute a task coming from the Context Interpreter, for example. Note that the Context Interpreter is an Infraware component which is external to the Service Manager.

Finally, COReS is added to aggregate more functionalities to the Service Manager. In order to accomplish that, this recommender system interacts with three other components: the Coordinator, the Discoverer and the Deliverer. To enable a clear understanding of their interactions, consider the following scenario: a patient is leaving the doctor who has just prescribed him with some x-rays and magnetic resonances. Aware of that, COReS can recommend some radiological clinics with some free schedules, even before the user requests that service. Figure 5 shows how COReS and the remaining Service Manager components deal with this situation.

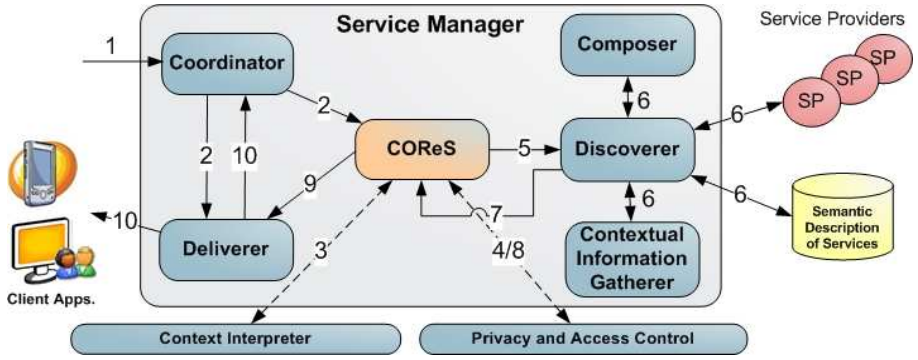

Fig. 5. COReS interacting with Service Manager components to provide recommendations to the user

The Coordinator receives an input (1) and recognizes that this input is an event caused by a change in the context of the user, thus forwarding this event to COReS and notifying the Deliverer about a new input (2).

In order to identify the nature of this event, COReS interacts with the Context Interpreter, which provides it with the present context of the user (in this example, leaving the doctor), and with any other information derived from this context and from the domain ontology (Fig 2) (for instance, a medical prescription) (3). The healthcare domain ontology describes that a consult provides a prescription.

After that, COReS interacts with the Privacy and Access Control component of the platform to gain the authorization to access the prescription (4). Services that satisfy a prescription can be infered from the diagnosis provided by the consultation, and they are those that belong to the service type prescribes by the prescription, that are performed in a healthcare unit or by professional specialized in this kind of service, and that attend by the user's health insurance.

So, in the next step, COReS creates a service request and sends it to the Discoverer (5) to find the suitable services (6) (in this case, radiological clinics). The result of the discovery is sent back to COReS (7) and the system then interacts again with the Privacy and Access Control component to get users profiles (8). The comparison of user profiles takes into account the interests, preferences and documents related to a specific domain (in this case healthcare domain), and tries to find a correlation between the concepts described. Considering the Figure 3, users A and B profiles are similar in the healthcare domain, but they are different in the music domain. COReS uses user profiles and evaluation of other users, and some contextual information like his location and agenda, to rank the results with the purpose of helping the user in his choice.

The system then submits the recommendation results to the Deliverer (9), which finally delivers them to the user (10). Next to this, the Deliverer also notifies the Coordinator about the accomplishment of the recommendation.

5 Recommendation Technique

The recommendation technique implemented in COReS is based on a previous work on a recommender system named KARe (Knowledgeable Agent for Recommendations) [8]. KARe has been originally proposed for knowledge sharing, through the exchange of textual information items. The system architecture and algorithm may however be adjusted for services recommendation. What makes this adjustment viable are the advances of the past few years in Web Services research. Web Services are represented and described using metadata (i.e. textual information) [25], thus allowing the use of the same techniques applied for the recommendation of textual information items in general.

Characteristics of the system which motivate the extension of its capabilities for service recommendation are the following:

- Having a distributed architecture, based on software agents and the peer-to-peer model;
- Using taxonomies to enrich with semantic information the description of recommended items. Taxonomies in KARe are important not only for the

organization and visualization of items but especially for providing flexibility in the recommendation of these items;
- Working both reactively and proactively. Reactively by satisfying the user's request for items of his interest, and proactively by anticipating user needs for certain items, considering even contextual information (location of the user);
- Having been prototyped for mobile devices.

The experiment developed to validate KARe's recommendation algorithm shows that it presents considerable benefits when compared with traditional recommendation algorithms, besides providing us with some directions of improvement [8].

While COReS's architecture is quite well understood (please refer to section 4), most of our current and future work relies on adjusting KARe's algorithm for the use in COReS. Our effort starts by understanding how the user profile should be represented and how it relates to contextual information. Both profile and context refer to information about a particular user, the latter being more dynamic than the former (i.e. contextual information such as time and location are more likely to change in time than personal interest and taste). Our second concern regards the definition of what characterizes a service. To address this, service description languages (such as WSDL, SOAP, etc. [25]) must be investigated. And finally, we aim at verifying how the applied recommendation algorithm may be used to match profile and service description, thus generating service recommendations to the user in need. As appointed in section 2, this present initiative bets on the power of domain ontologies to generate a semantic match. It is thus our intention to understand how these domain ontologies may substitute the previously applied taxonomies in KARe. At a first glimpse, it appears that domain ontologies may enrich the semantic information provided by the system. This deservers a thorough investigation to realize the possible gains this might bring to the generation of recommendations.

6 Related Work

In this section we briefly present some of the research literature related to service recommender systems and context-aware recommender systems.

The situation-aware task-based service recommender system [26] is an extension of the initial task-oriented service navigation system. This service navigation system supports the user in finding appropriate services, including services the user might not have been aware of before. This is possible through the use of a rich task ontology. The user, to use the basic task navigator must specify an activity query and send to the system. After that, a list of tasks that match that query is sent back to the mobile device of the user. Now the most appropriate task can be selected and in turn the corresponding detailed task-model is shown to the user. In a final step, associated services can be invoked by establishing an Internet connection to invoke actual i-mode services. The extension of the system takes the users situation into account and thereby avoids the necessity to input an initial task query. Ontology-based logical reasoning makes use of context data

such as time, location and people in proximity to compute concrete individual situations. Tasks are categorized according to the high-level context concepts given by the ontologies to be able to recommend an appropriate task-list corresponding to the user's actual situation. On request the task engine returns a filtered task list that results from matching the inferred user situation with the task-specific categories.

Other attempt to integrate context awareness with mobile recommender systems is the COMPASS application [12]. COMPASS, which means Context-aware Mobile Personal ASSistant, is a context-aware mobile tourist application that serves a tourist with information and services based on his interests and current context. The application consists of a map showing the user location. Depending on his profile and goal, the system selects nearby buildings, buddies and other objects and shows (recommend) them on the map and in a list. The application is built upon the WASP platform [15] that provides generic supporting services, such as a context manager and service registry. The COMPASS, to show objects on the map, first queries the service registry for search services that are bound to deliver objects related to the user's context. The WASP retrieves services matching the user's context and goal. After that, the relevant search services are queried to retrieve the objects matching the context's criteria, e.g. being in a certain distance from the user. The retrieved objects are then sent to the recommendation engine which scores each object based on the user's interests and contextual factors. The resulting objects and scores are displayed on the map and in the list of objects.

A third approach is the system UbiMate that [27] cite in their work. UbiMate is a mobile recommender system proposed by Annie Chen. Chen proposed a context-aware collaborative filtering system that can predict a user's preference in different context situations based on past user-experiences. The system uses what other like-minded users have done in similar context to predict a user's preference towards an item in the current context [Chen, 2005]. The context information that UbiMate include are: user information (profile and rating history), social environment (alone, friends, family, etc.), tasks (activity), location (coordinates), weather (derived from coordinates) and time (time of interaction). The composition of different context information establishes a snapshot of context. The information may be available or unavailable, so they are stored independently. The collaborative filtering algorithm the user's profile and a snapshot of the current context along with the rating, and use statistical methods to predict the items the user will most prefer.

7 Conclusion

This paper proposed an approach to facilitate service selection, exploring possibly synergy among recommender systems and context-aware computing. It described the development of COReS (Context-aware, Ontology-based Recommender system for Service recommendation) which uses the capabilities of the Infraware platform to support services selection, making service offer by this plat-

form more efficient, personalized and proactive, and thus satisfying the needs of the user in a particular context.

The motivation behind the creation of COReS comes from the fact that most of recommender systems have been developed for specific domains. Furthermore, they do not consider context in their recommendations and they perform simple keyword-based matching instead of semantic matching, enabled by ontologies, which can cause inconsistent recommendations. And, an innovation of COReS is the adoption of compartmentalized the user profiles according to different domains. The suitable profile part for a given recommedation is selected in time of prediction, based on the user's context.

COReS is a hybrid system implementing recommendation technics based on an existing system named KARe. Integrating the Infraware platform, COReS is being developed as a Service Manager's new component. This way, we aim at aggregating more functionalities to this module, like anticipating user needs for certain services, helping him in the choice of a service and assisting him on accomplishing his goals.

References

1. Chen, G., Kotz, D.: A Survey of Context-Aware Mobile Computing Research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, USA, 2000.
2. Pereira Filho, J.G., Pessoa, R.M., Calvi, C.Z.: INFRAWARE: A Support Middleware to Context-Aware Mobile Applications (in portuguese). In Proceedings of the 24th Simpsio Brasileiro de Redes de Computadores, Curitiba-PR, 2006.
3. Weiser, M.: The Computer for the Twenty-First Century. Scientific American, pages 94–10, September, 1991.
4. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4):331–370, 2002.
5. Montaner, M., Lpez, B., De la Rosa, J.L.: A Taxonomy of Recommender Agents on the Internet. In Artificial Intelligence Review, 19(4):285–330, 2003.
6. van Setten, M.: Supporting People in Finding Information. Number 016 in Telematica Instituut Fundamental Research Series. Universal Press, 2005.
7. 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 Journal, Volume 16, pages 97–166, 2001.
8. Guizzardi, R. S. S.: Agent-oriented Constructivist Knowledge Management. PhD. thesis, University of Twente, The Netherlands, 2006.
9. Karta, K.: An investigation on personalized collaborative filtering for web service selection. MSc. thesis, The University of Western Australia, 2005.
10. Balabanovic, M., Sholam, Y.: Combining Content-Based and Collaborative Recommendation. Communications of the ACM, 40(3), 1997.
11. Basu, C., Hirsh, H., Cohen, W.,: Recommendation as classification: using social and content-based information in recommendation. In Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-1998), pages 714–720, 1998.
12. van Stetten, M., Pokraev, S., Koolwaaji, J.: Context-Aware Recommendations in the Mobile Tourist Application COMPASS. In Adaptive Hypermedia, vol. 3137 of LNCS, Springer-Verlag, pages 235-244, 2004.

13. Guizzardi, G.: On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV, Olegas Vasilecas, Johan Edler, Albertas Caplinskas (Editors), ISBN 978-1-58603-640-8, IOS Press, Amsterdam, 2007 (forthcoming).
14. Pereira Filho, J.G., Barbosa, A.C.P., Pessoa, R.M., Calvi, C.Z., Carmo, R.R.M., Oliveira, N.Q., Castro, L.R.A. INFRAWARE's Platform Architecture (in portuguese). Technical Report RT-02, Federal University of Esprito Santo, 2005.
15. Costa, P.D.: Towards a Services Platform for Context-Aware Applications. MSc. thesis, University of Twente, The Netherlands, 2003.
16. Web Ontology Language (OWL). http://www.w3.org/2004/OWL/, 2007.
17. Resource Description Framework (RDF). http://www.w3.org/RDF/, 2007.
18. Carmo, R. R. M.: A Service Discovery Protocol for Context-aware Systems (in portuguese) MSc. thesis, Federal University of Esprito Santo, Brazil.
19. Costa, A. C. M.: AlComp: A Context-aware, Ontology-based, Semantic Matching Algorithm for Service Discovery (in portuguese). Graduation project, Federal University of Esprito Santo, Brazil, 2006.
20. U. Shardanand, Maes, P.: Social information filtering: Algorithms for automating word of mouth. In Proceedings of ACM Conference on Human Factors in Computing Systems (CHI 95), pages 210-217, 1995.
21. Mooney, R. J. and Roy, L.: Content-based book recommending using learning for text categorization. In Proceedings of the Fifth ACM Conference in Digital Libraries, pages 195204, 2000.
22. Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. In Proceedings of the ACM Conference on Computer Supported Cooperative Work, pages 175186, 1994.
23. Guizzardi, G.: Agent Roles, Qua Individuals and the Counting Problem. In: P. Giorgini; A.Garcia; C. Lucena; R. Choren. (Org.). Software Engineering of Multi-Agent Systems. Berlin: Springer-Verlag, 2006, v. , p. -, 2006
24. Masolo, C., Guizzardi, G., Vieu, L., Botazzi, E., Ferrario, R.: Relational Roles and Qua Individuals. In: American Association for Artificial Intelligence (AAAI), Fall Symposium on Roles: an Interdisciplinary Perspective, 2005, Arlington. Proceedings of the 2005 AAAI Fall Symposium on Roles, an interdisciplinary perspective, 2005.
25. Graham, S., Simeonov, S., Boubez, T., Davis, D., Daniels, G., Nakamura Y., Neyama, R.: Building Web Services with JavaTM: Making Sence of XML, SOAP, WSDL and UDDI. Sams Publishing, 2002.
26. Fukazawa, Y., Luther, M., Wagner, M., Tomioka, A., Naganuma, T., Fujii, K., Kurakake S.: Situation-aware, task-based service recommendation. In Proceedings of the 4th International Semantic Web Conference, 2005.
27. Helfenstein, S., Beutler, E.: Context-awareness. Seminar of Context-aware Computing, University of Zurich, 2006.