A Case Based Reasoning Framework for Service Selection and Adaptation in Mobile Networks

Vasileios Baousis

University of Athens, Greece

bbaous@di.uoa.gr

Konstantinos Tzannetakos

University of Athens, Greece

std0002@di.uoa.gr

Elias Zavitsanos

University of the Aegean, Greece

izavits@iit. demokritos.gr

Vassilis Spiliopoulos

University of the Aegean, Greece

vspiliop@ aegean.gr

Stathes Hadjiefthymiades

University of Athens, Greece

shadj@di.uoa.gr

Abstract

Service selection and adaptation is of paramount importance in contemporary mobile networks. Many diverse parameters should be taken into account (e.g., user context, terminal and network capabilities) for the selection of the appropriate service or the required service adaptations. In this paper we propose a framework for service selection and adaptation. A Case Based Reasoning System (CBRS) is used to select the most appropriate service. Services are modelled using formal semantics. The CBRS retrieves the most appropriate service by comparing previous cases with the current service request. This comparison is performed using similarity metrics. We elaborate on the different aspects of the discussed architecture and provide indicative examples to illustrate the versatility of the proposed scheme.

1. Introduction

The increasing user demand to access services anywhere, anytime, and irrespective of the network and terminal capabilities creates new requirements for service discovery, selection and provision. Moreover there is a growing demand for service personalization, and users require services not only to adapt to their current context but also to have a specific appearance according to their preferences. Furthermore, new services are introduced into mobile networks that will reach a growing number of potential users, which implies that user/device heterogeneity will increase as well. The appropriate service selection and adaptation to the user context is of major importance in the device independent and mobile environments. Main parameters that should be considered in order to explicitly determine the user's context are: her service preferences/profile, device currently used and her dwell network. The key issue on service selection and adaptation is the description of the contextual data with semantic information, which unambiguous describe them.

In this paper we propose a novel framework for service selection and adaptation in mobile networks that considers user preferences and device capabilities along with the service requests in order to select the most appropriate services from the available ones. All the considered data is expressed (or appropriate mapped/enhanced) in unambiguous manner with the use of formal semantics and a CBRS is used to select the most appropriate service after processing this data.

The rest of this paper is structured as follows: section II provides some background knowledge about the implementation technologies, and section III presents an overview of the proposed architecture. Section IV a use case example of the proposed framework is presented, and section V discusses relevant prior work. Finally, Section VI concludes the paper.

2. Background Knowledge

CBR is a methodology that "solves a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation" [2]. CBR resembles nature human reasoning, where effort is put in resolving new problems by referring to similar past cases. In CBR, the primary knowledge source is not some generalized or explicit rules but a memory of stored previous cases. Expertise is integrated in a data base of past cases, with each case containing a description of the problem, a solution and the outcome. The reasoning process used to reach a solution is not recorded, but is hidden into the solution. Knowledge acquired during CBR task cycle is integrated within the new cases inserted to the case base. CBR task cycle problem solving includes (Figure 1): a. Similar Case Retrieving process using problem matching (against the cases in the case base), b. Reusing the retrieved case(s) by combining/integrating parts of these cases and providing a solution suggestion c. Revising and/or adapting the retrieved solution(s) to the current problem in order to provide a realistic solution and d. Retaining the new solution as a new case.

The aforementioned tasks are mapped respectively to the following processes: a. Case representation, b. Case matching and retrieval, c. Case adaptation, and, d. Case-base maintenance. For each such process a plethora of approaches could be adopted.

Cases can be expressed in a formal way e.g. through Ontologies. Each stored case is part of an ontology that unambiguously specifies its components. Ontologies facilitate

knowledge sharing among heterogeneous systems, through explicit formal specifications of the terms used in a knowledge domain and relations among them. Currently, several languages have been proposed for Web Service description. In our work we adopt OWL-S [3], since it has well defined specifications and is widely accepted by the scientific community. OWL-S implicitly defines message types (as input/output types of processes) in terms of OWL classes, which allows for a rich, class-hierarchical semantic foundation. Specifically, OWL-S models the services via three sub-ontologies: (i) a service profile describes what the service requires from users and what it gives them, (ii) a service model specifies how the service works, and (iii) a service grounding provides information on how to use the service.
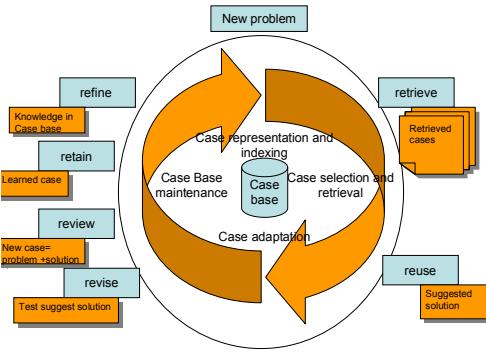
Figure 1: CBR task cycle.

In the proposed framework, the capabilities of the devices utilised by the users to access services are described in unambiguous manner. This is achieved by adding semantic annotation to the standardised means of device capabilities description. In mobile networks, the Composite Capabilities/Preference Profiles (CC/PP) [4] has been proposed by the World Wide Web Consortium for the device description. CC/PP is a description of the device capabilities and user preference, and is used to reveal the user's context in order to allow service providers to adapt their service content to this context. CC/PP is based on Resource Description Framework (RDF) and the main branches that the CC/PP profile tree describes are the terminal software and hardware and application specific information (such as browser).

Finally, the proposed framework considers services that are expressed using formal semantics. According to the Parlay X Web Services specifications [5], web services in mobile networks are described with the following four document types: a. the Type Definition Document that contains data definitions (optional), b. the Shared faults document containing the fault definitions (optional), c. the Service interface document contains the message and interface definitions expressed in WSDL (Web Service Description Language), and finally d. the Service binding document contains both the binding to be used and the service definition. The specifications in [5] cover all aspects of service description and invocation. In order for the framework to process these services, the description of these services should be transformed to OWL-S.

3. Framework Architecture

The discussed framework provides the functionality of enhancing the WSDL service descriptions to ontology based description using OWL-S. Furthermore, when all services descriptions and service requests are semantically enriched, the framework is capable of providing matchmaking among service requests and available services. The most representative techniques used for matchmaking [6], are: Semantic Capability Matching, Multi-Level Mapping, DL Matching with Service profile Ontologies, Information Retrieval Based and Graph-Based approaches.

Figure 2 presents the outline of the proposed system architecture. Subsequently, we provide a short description of the system functionality. The system consists of the following entities:

- **Network**: Provides the network capabilities profile like QoS parameters, capacity, and security protocols available.
- **Users**: Perform service requests while providing their service profile and their device capabilities.
- **Service providers**: Register their services to the service registry (expressed in OWL-S), provide their services to the users and provide feedback to the CBR system regarding service invocations.
- **Ontology creation**: Uses the service query to create an instance of a service ontology.
- **Ontology mapping**: Provides ontology mapping to service user profile and the terminal capabilities description.
- **Case Based Reasoning System (CBRS)**: Performs service selection based on previous cases and provides the result to the user.

In the proposed framework, the network capabilities are revealed to the CBRS and appropriate mapping is performed in order for the produced ontology to refer to the same concepts such as Quality of Service, network technology, bearer rates and network type. Available services are published to the service registry. Services are expressed in OWL-S with an appropriate annotation from the WSDL. It should be noted that both network capabilities publication to the CBRS and service registration to the service registry are performed offline and not at runtime (i.e. on every service request). Each user requests a service, providing his user profile/ preference and terminal capabilities. User profile/ preferences are expressed in OWL, terminal capabilities in CC/PP and service request is in simple text (e.g., "How is the weather in Rome now" or "How long does it take to travel from Rome to Paris by train?"). Subsequently, ontology creation for service request takes place and ontology mapping for user profile and terminal capabilities is performed. The CBR receives the service requests and finds the most suitable service descriptions that yield a best match. The matching procedure is performed in two steps: a. Firstly searching the User DB for a similar service request from the same user and b. Secondly, retrieving similar past cases from the Case base, (the procedure is further analysed in 3.6). The service description with the best score is provided

to the user. The user receives the service description and, either, invokes the indicated service or rejects the suggestion of the CBRS. The event of the suggested service execution or not is provided to the framework either explicitly in case of service execution or implicitly, e.g., when a predefined time period is elapsed and the user has or has not executed the suggested service. In the following paragraphs the framework components are further analyzed.
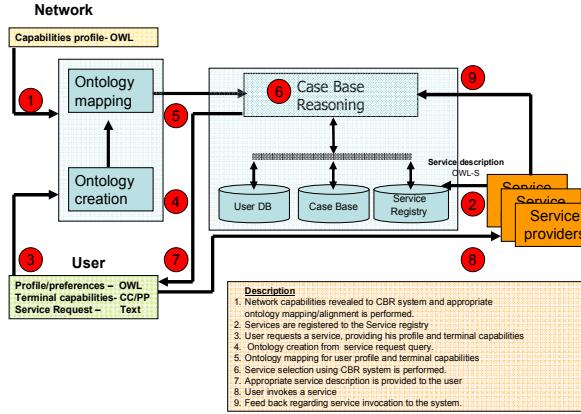

Figure 2 : Service Selection Component

3.1 Network
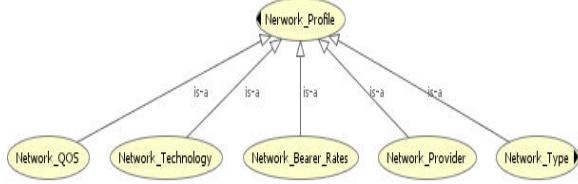
Figure 3: Network profile ontology

Each network is described with a Network Profile, which is expressed in OWL. This ontology contains classes such as the Network Provider, that contains further information such as static network configuration, the Network Type and Technology classes that withhold information regarding the type of the network and access technology, Network Bearer Rates (containing information such as available bandwidth) and Network QoS containing QoS parameters such as capacity, reliability, jitter, latency, response time, security, and cost (see Figure 3).

3.2 Users

We adopted an approach similar to [17] for the modelling of the user service profile, regarding the modules that such a profile should contain. Moreover we represent this profile with a generic ontology that is generated from the user preferences and device description (Figure 4). The user profile ontology contains concepts such as specific user preferences that may influence the service provision, user's subscribed services, personal information and current location, and current device. The device concept is further detailed since it has a considerable impact on the service selection and adaptation process. The user's device capabilities are expressed

in CC/PP and the ontology creation module is responsible to create a respective ontology that describes the device and the Ontology mapping to map similar concepts. The processes of creation and mapping of user profile and device ontologies to the CBR system are performed by the proposed framework with the respective functional components. The device capabilities description and user profile are provided to the framework during user registration and updated when a change occurs. The respective processes of ontology creation and mapping are performed simultaneously with user registration.
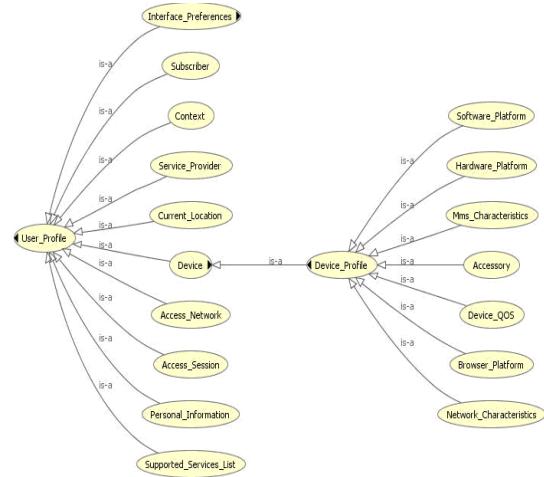
Figure 4: User service profile ontology

3.3 Service Providers

Service providers (SPs) offer services to the interested users. Services are described in WSDL. Therefore SPs need to add semantic annotation to their services (i.e., to represent their Web Services with a Service Profile, Service Model and Service Grounding) and generate an OWL-S document. The Service Grounding is equivalent to WSDL while service model and profile can be generated from WSDL using either the ontology creation module (section 3.4). The services with the semantic information are registered to the Service Registry. It should be noted that SPs can register a service with different configurations in order to render it accessible from different devices (e.g., devices with a low screen resolution) and allow the reasoning system to select the appropriate service configuration that best matches the context of a requesting user. Moreover, when a service is selected and provided to a user from the proposed framework, the SP offering this service is notified accordingly. Thereafter, the SP is responsible to provide a response to the CBRS, regarding the specific service invocation in order to record the whole service request, service suggestion and user experience as a new case within the case base.

3.4 Ontology Creation

In the general case, the ontology building module comprises the following tasks for ontology construction: a. Identification of the basic concepts of the ontology, b. Building

of the hierarchical backbone of the ontology, and c. Semantic relation enrichment. The purpose of the final step is to identify other semantic relations that might hold between concepts, besides the inclusion relation.

In the proposed framework, where prior knowledge, as well as sources of information are not available, but the user only provides a string query as input to the ontology building module, there are two well-defined scenarios for ontology construction.

In the first case, a Part-Of-Speech (POS) tagger is used to identify the nouns that appear in the string query. Then, by consulting the WordNet thesaurus [8], the set of the related concepts to these nouns is retrieved. This set comprises the concepts that their names are the lexicalizations of these nouns, as well as the concepts that appear in the vicinity of the nouns along with their semantic relations. The WordNet provides for each entry semantic relations such as hyponymy, hypernymy, meronymy, etc. In addition, it provides glosses for the retrieved concepts, that is, sort descriptions that explain the sense of each concept. In the final step, the retrieved concepts and their semantic relations are transformed into the ontology that represents the query of the user. Hyponymy and hypernymy relations are directly translated into subsumption relations between concepts, while other semantic relations are used to further enrich the taxonomic backbone of the ontology. Let us consider the query example introduced in the beginning of this section: "How is the weather in Rome now". The POS tagger identifies the word "weather" as a noun and the module consults the WordNet in order to retrieve the direct concepts that are related to this term.

In the second case, the system creates a corpus of text documents in order to construct the ontology based on these text sources. The system uses a crawler to interact with the Google API and retrieve documents related to the string query of the user and construct a set of sufficient texts. At this point, hierarchical probabilistic topic models are applied, in order to identify thematic topics that will constitute the concept entries of the ontology and arrange them in a taxonomic manner [9] [10].

The ontology creation component can be used offline in order to model the device/terminal capabilities, as well as to semantically describe the provided web services. By storing the CC/PP descriptions of several users, as well as the WSDL descriptors of all available web services, two corpora are created. The first is focused on the description of terminal capabilities, while the second on the description of the available web services. Thus, two main domain ontologies are created: the device ontology (DevOnto) and the web service-oriented ontology (ServOnto). These ontologies are constructed by following the approach in [12]. In both cases, after a POS tagging in the RDF/WSDL descriptions, a pattern matching for the identification of domain concepts (usually nouns), and the identification of functionalities that are frequently offered in the domain is applied. Following, the ontology building relies on a compositionality-based hierarchical building, where the concepts' lexicalizations reflect the subsumption relations between them. Furthermore, the functionality modelling of OWL-S and WSMO has been followed by including both the verb of the action and a directly involved data element in the functionality (e.g., BookTicket). The final step includes pruning of the irrelevant concepts following a baseline pruning strategy which advocates that frequent terms in the corpora denote domain concepts while less frequent ones lead to concepts that can be safely eliminated from the ontologies [13].

3.5 Ontology Mapping

As it is expected, it is not realistic to assume that the ontology constructed from the user service request the network and terminal capabilities, and the ontologies stored in the case base are unambiguously defined, i.e., elements with the same intended meaning are treated as equivalent. For this reason, it is necessary that the proposed framework should be able to determine mappings between such elements (e.g., ontology concepts or other generic schema elements). State of the art ontology mapping systems exploit combinations of string similarity and machine learning methods, as well as external lexica (e.g. WordNet) to determine mapping pairs. In the proposed framework, we follow a similar strategy in order to maximize the efficiency of the ontology mapping component by exploiting the SEMA mapping tool [15], which has been shown to, achieve quite high precision and recall. SEMA combines string similarity, semantic and structural based matching algorithms: A semantic matching method exploiting Latent Dirichlet Allocation model (LDA) [14], requiring no external resources, in combination with the lexical matcher COCLU (**CO**mpression-based **CLU**stering) [16] and a matching method that exploits structural features of the ontologies by means of simple rules. This combination of approaches contributes towards automating the mapping process by exploiting lexical, structural and semantic features of the source ontologies, resulting to increased recall and precision.

3.6 Case Base Reasoning System

The CBRS is responsible for receiving a service request and retrieving the most relevant service description from the case base and the service repository and finally provide this service description to the user. The matching procedure is performed in two steps: Firstly, the local User Database (User DB) is searched if the same user has requested a similar service in the past and if there is such an entry, the service description is retrieved from the service registry and Secondly, the Case base is searched for similar cases and the most relevant cases are retrieved.

The service description with the best ranking (either retrieved from the User DB or the Case Base) is either executed or provided to the user (depending on the service itself and the user context). Furthermore, the CBRS collects feedback from the Service providers in order to update the case base with new cases. Each such new case is created from the

service request, the service description provided by the system along with the success of this service suggestion, quantified in a number between zero and one hundred. This number captures the user satisfaction from the service execution in the interval [1,100] (zero is assigned when the user does not use the service and hundred when the user does use the suggested service for a predefined time period specific to each service). The simplest form of capturing the user experience from the service execution is to consider the real service execution time. If the user executes the service for a time period equal or greater than an average typical service execution time then this interaction can be rated with hundred per cent whereas when user does not use the service. All the other cases can be rated considering the actual service execution time and provide a number in [1, 99].

Appropriate service selection is based on the matching between the service request, user context, current network status and the previous cases already recorded in the case base. Each element of the considered data is expressed in unambiguous manner using semantics and specifies the type of similarity metrics that should be considered during the matching process. In this regard, the CBRS may consider different matching approaches for each compared element in order to compute the matching degree. Moreover, in each similarity computation between two elements can be assigned a different weight factor in [0,1], therefore allowing similarity of some elements to have greater or lesser impact on the matching process. Furthermore, and most important, by adjusting properly these weights the matching process can be similar or a combination of the semantic matching techniques described in section 2. Regarding service adaptation, it should be noted that it is mainly performed though the appropriate service configuration that each service provider supplies to the service registry. The CBRS has a comprehensive representation of the user context and therefore is able to select the appropriate service configuration that best matches the user's needs.

We used the JColibri [11] in order to build the CBR component of our framework. The JColibri is an open source framework for building CBR systems that comprises most of the tasks included in a CBR system lifecycle. The cases in this framework are organized in an internal ontology that describes the case structure, the fields of the case and the similarity functions of each such field. This CBR ontology (termed CBROnto) elaborates an extensive ontology over CBR terminology thus, providing a common language to define the elements that comprise a CBR system. Specifically, the CBR ontology has three classes: a. CBRCASE and CBRDESCRIPTION refer to the case base structure and each case description respectively, and b. CBRINDEX contains the indexing to the structure and content of the case base. Considering the case similarity metrics for case retrieval, we used the similarity functions provided by the JColibri. Specifically, these functions provide similarity comparisons both on local (such as cosine, cousin, and equal similarity measures) and aggregate level where the

former refers to similarity between individuals in a one-to-one fashion whereas the latter refers mainly to averaging values of numeric individuals of a whole case. The provided similarity functions, along with the capability to construct new similarity functions such as K-Nearest Neighbour, were used to implement and test the functionality of our framework.

4. Use Case Example

In a realistic situation, the user might want to be informed about the weather in a place. Let us recall the example of section 3: the user's query is "How is the weather in Rome". The ontology construction module identifies the word "weather" as a possible concept, consults the WordNet and constructs the corresponding ontology for this query. The resulting ontology has a main concept called "weather" that has direct super-concept "atmospheric_phenomenon" and direct sub-concpets "good_weather", "bad_weather", "cold_weather", hot_weather", fair_weather", exactly as retrieved from WordNet. The main concept is further enriched with other semantic relations, such as the one that indicates that belongs to "meteorology". In addition, the ServOnto and DevOnto ontology instances are retrieved from the corpora of WSDL and CC/PP files that describe the services and terminal capabilities respectively. When a new query is posed to the framework an ontology is created and is aligned with the ontology in the Case Base database. This is a necessary step in order to be able to unambiguously define all concepts and to be able to apply similarity algorithms among these concepts. For example, two differently labelled concepts, such as "weather" and "weatherCondition", may have the same intended meaning. In this case, a string similarity mapping method could locate a possible equivalence mapping. Furthermore, a tokenization technique could split the second label into two new ones: "weather" and "condition", and in combination with an external dictionary (e.g., WordNet) could further assess that concepts "weather" and "weather condition" are synonyms, as these sets of terms are defined in the same WordNet synset. The CBRS receives and processes the service request. The processing involves the retrieval of the most suitable service(s) that best match this request. The system searches the User DB for a similar service request from the same user and if there is such a request then retrieves the service from the User DB (if a user A has requested a service with the same parameters e.g "How is the weather in B place" then the service that is recorder in the user DB is retrieved). If the user does not have previously requested a similar service, then the system tries to retrieve a similar past case from the Case base. If there are similar cases retrieved from the Case Base, then the service with the highest degree of match is selected. Depending on the service invocation scheme, the selected service is either executed by the framework and the service results are provided to the user or the service description is forwarded to the user. In the first case the framework is updated with

the service invocation and the whole process details (service query and selected service) are captured as a new case in the Case Base. In the second case, the user receives the service description and either invokes the retrieved service or not. The event of the service execution or not by the user is provided to the framework from the respective SP. The SP provides a user service satisfaction index that is captured either explicitly in case of service execution (e.g. by comparing the time of the user interaction with an average time of service invocation) or implicitly, e.g., when a predefined time period is elapsed and the user has or has not executed the suggested service. The process ends with the creation of a new case in the case base that contains the service invocation parameters.

5. Related Work

In [1] web service execution experiences are modelled as cases that contain the functional and non-functional domain specific Web service properties that are described using semantics. In the proposed framework a Case base reasoner matches the new service request to the cases in the case base using Nearest Neighbour matching and ranking. The reasoner computes the similarity of each property between the new problem and the cases and qualifies the assigned weight of the similarity, and finally the case with the greatest similarity degree is the best- match. In the discussed framework, service requests by user are considered in a monolithic and opaque manner, since users have to explicitly specify service requirements. Our framework considers user specific data, such as user service profile and device capabilities, and service requests that are properly mapped/ aligned using semantic languages in order to accurately retrieve appropriate services from the case base.

A framework called Personalized Service Framework (PSF) presented in [7], allows users to access and execute Web services on their personal mobile device. Device capabilities and user context are expressed in CC/PP. A specific component, named ServiceSearcher, is responsible to locate web services (expressed in OWL-S) stored within the service database and perform the matching process which mainly relies on comparing the inputs/outputs among service requests and available services.

The Delivery Context: Client Interfaces (DCCI) [18] (under standardization), can provide contextual information (such as device properties and network and user preference) to the service providers, thus allowing for content adaptation and device independence. The DCCI covers the user and network parts in the process of service adaptation.

6. Conclusions

We presented a novel framework for service selection and adaptation in mobile networks that considers user pref-

erences and device capabilities along with the service requests in order to select the most appropriate services from the available services. All considered data is expressed in unambiguous manner with the use of formal semantics and a Case Base Reasoning System matches and selects the most appropriate service, based on this data. To this extent, ontology creation and matching techniques have been exploited for the transformation and matching of the user's requirements to well-defined artifacts that enforce the efficient semantic retrieval of the services. Future work includes testing of the whole framework components and further investigation and studying of methods for capturing user experience gathered by the framework.

7. References

[1] D. Thakker et al, "Semantic-Driven Matchmaking and Composition of Web Services Using Case-Based Reasoning," 5th European Conference on Web Services (ECOWS'07), p. 67-76, 2007

[2] A. Aamodt, E. Plaza "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches" AI Communications 7(1) (1994), S.39-59.

[3] OWL-S: http://www.w3.org/Submission/2004

[4] CC/PP: http://www.w3.org/Mobile/CCPP/.

[5] OSA Parlay X Web Services (2008): Parlay X Web Services Specifications. http://www.parlay.org/en/specifications/pxws.asp

[6] Tsetsos V., Anagnostopoulos C., and Hadjiefthymiades S. "Semantic Web Service Discovery: Methods, Algorithms and Tools", chapter in "Semantic Web Services: Theory, Tools and Applications" (Ed. Dr. Jorge Cardoso), IDEA Group Publishing.

[7] W. Zahreddine, Q. H. Mahmoud. "A Framework for Automatic and Dynamic Composition of Personalized Web Services." In 19th International Conference on Advanced Information Networking and Applications (AINA'05), 2005.

[8] C. Fellbaum. WordNet: An On-Line Lexical Database and Some of its Applications. Cambridge, MIT Press, 1998.

[9] D. Blei, T. Griffiths, M. Jordan, J. Tenenbaum. Hierarchical Topic Models and the Nested Chinese Restaurant Process. NIPS, 2004.

[10] E. Zavitsanos, G. Paliouras, G.A. Vouros, and S. Petridis "Discovering subsumption hierarchies of ontology concepts from text corpora" in International Conference on Web Intelligence, 2007.

[11] J. J Bello et al "JColibri: An object oriented Framework for building CBR Systems" book chapter in Advances in Case-Based-Reasoning, SpringerLink, ISBN-978-3-5400-22882-0, November 2004.

[12] Sabou M. "Learning Web Service Ontologies: an Automatic Extraction Method and its Evaluation" in OLP Workshop, 2005.

[13] A. Maedche "Ontology Learning for the Semantic Web" Kluwer Academic Publishers, 2002.

[14] V. Spiliopoulos, G. Vouros, V. Karkaletsis. "Mapping Ontologies Elements using Features in a Latent Space" in International Conference on Web Intelligence, 2007.

[15] Spiliopoulos, V., Valarakos,.G., Vouros, G. Karkaletsis, V. "SEMA: Results for the ontology alignment contest OAEI 2007" in 2nd Ontology Matching International Workshop, OAEI, 2007.

[16] Valarakos A. et al "A name-Matching Algorithm for supporting Ontology Enrichment". In SETN'04, 2004.

[17] Francois Bougant et al "The user profile for the virtual home environment" IEEE Communication magazine, January 2003, p. 93-98

[18] DCCI: http://www.w3.org/TR/ DPF/