COCOON Glue: a prototype of WSMO Discovery engine for the healthcare field

Emanuele Della Valle and Dario Cerizza

CEFRIEL - Politecnico of Milano, Via Fucini 2, 20133 Milano, Italy
dellavalle@cefriel.it, cerizza@cefriel.it

Abstract. COCOON[1] is a 6th Framework EU integrated project aimed at setting up a set of regional semantics-based healthcare information infrastructure with the goal of reducing medical errors. In order to enable a seamless integration of many complex technologies with the existing regional eHealth services, Web Services technology was selected. But this raised the problem of cataloguing, managing and maintaining these Web Services. In this paper we present COCOON Glue, an early attempt in employing Semantic Web Services technologies in eHealth sector. In particular we focus on a prototype of WSMO Discovery engine and we present how we are testing it to support the arrangement of an advice-meeting between a General Practitioner and one community of practice among many available ones.

1 Introduction

Medical error is one of the most important issues in the European healthcare system to be faced off in the coming years (see communication 356(2004) of EC [1]). Improving patient safety by reducing medical errors has become a business necessity [2]. Therefore, safety concerns are driving investments in a new set of advanced clinical information systems, such as electronic patient files, the related computerized physician e-prescription facilities, the virtual meeting platform for giving remote advice, etc.

Presenting the right knowledge to the right medical personnel in the right place and at the right time is of paramount importance in making critical medical decisions. Driven by this paradigm, COCOON [3][4] concerns with the development of a semantics-based healthcare information infrastructure designed to offer interactive support to general practitioners and healthcare professionals, to assist them in handling efficiently complex medical cases, so that diagnosis and treatment errors would be minimized, potentially saving lives, economizing on the rising costs of healthcare and avoiding malpractice suits. Therefore, COCOON solution must provide: the necessary information from the appropriate clinical guidelines; the relevant and updated research evidence; the information regarding available medical services, technologies and medications, their efficiency and

[1] http://www.cocoon-health.com

side effects; possibly even other medical experts' advice and experience in other similar cases.

From a technological point of view, this means that COCOON project builds both on developing innovative services and integrating those embedded in already running systems (mainly general practitioners' electronic patient file and the regional eHealth services). In order to enable a seamless integration of so many complex technologies, all the interfaces of the services developed within COCOON project are exposed as Web Services. Moreover COCOON will be deployed as a component of the pilots' healthcare information system (e.g. the SISS[2] in Lombardy - Italy) that are already populated by hundreds of Web Services (e.g. e-prescription). So, this raises the problem of cataloguing, managing and maintaining these Web Services. To this purpose COCOON Glue [5] aims at developing an efficient system for the management of the Web Services that would include publishing, discovery and composition of services.

Being widely recognized that "Semantic differences remain the primary roadblock to smooth application integration[3]", we chose to base COCOON Glue on the current Semantic Web Services efforts. We recognize that OWL-S[4] and WSMO[5] are the two major initiatives that aim at facilitating the automation of Web Service tasks. And we chose WSMO because it is a strong conceptual model based on a sound separation between ontologies, goals, web services and mediators, and because it is founded on the two clear principles of strong decoupling and strong mediation.

COCOON Glue, in the 42 months of duration of the project (started on January 2004), will address the integration problem starting from Service Discovery and ending up with Service Composition. In this first phase of the project we focused on Service Discovery that, among the problems addressed by the Semantic Web Services community, is the one around which a set of realistic solutions is already available (e.g. the NTT UDDI Business Registry extended with OWL-S profile [6]).

This paper is structured as follows: in section 2 we give an overview of Service Discovery problem and we emphasize the role of semantics in addressing it; in section 3 we introduce a use case that requires Service Discovery in the healthcare field; the description of a prototypical implementation of WSMO discovery engine that we are developing in COCOON Glue follows in section 4, while an application of COCOON Glue, intended for field tests, is presented in section 5; finally in section 6, we draw some conclusions and we describe our future work.

[2] http://www.crs.lombardia.it/
[3] as Larry Ellison, Oracle Chairman and CEO, said in a question-and-answer session, when asked about the future of Web services.
[4] http://www.daml.org/services/
[5] http://www.wsmo.org/

2 Addressing the Discovery problem with semantics

What really distinguishes Web Services as technology is the fact that it is a way to enable a Services-Oriented Architecture (SOA). In a SOA, if a requester entity wishes to interact with a provider entity and it does not already know what provider agent to engage, then it uses a *discovery service* to find out a suitable candidate. As reported in Web Service glossary [7], discovery is "*the act of locating a machine-processable description of a Web Service that may have been previously unknown and that meets certain functional criteria*". UDDI [8] provides a general purpose model for service discovery by gathering metadata about a collection of services and making the information available in a searchable way (white, yellow and green pages). However, even with UDDI, the requester entity *must know* the interface; it requires and searches for a service that meets its functional requirements or that is provided by a well-known partner. As stated in section 1.4.5 "Overview of Engaging a Web Service" of W3C Working Group Note on Web Services Architecture [9], all this requires initial knowledge about both service existence and location. Currently, the most common approach to obtain such knowledge is through e-mail exchanges or word of mouth.

But technologies, which support (and at some degree automate) knowledge sharing are available. In particular, ontologies, with their ability to interweave human understanding of symbols with their machine processability, can play a key role in Service Discovery. Several initiatives[6], including OWL-S, WSMO, IRS[7] and METEOR–S[8], have shown that adding semantics to Web Service descriptions can be achieved by using ontologies. The common idea is that, at publishing time, relevant domain specific ontologies can be used to describe services' capabilities and users' goals, whereas, at discovery time, the Discovery engine, having access to the knowledge modeled in the ontologies, is not limited to syntactic matching techniques, but it can take also semantic matching techniques into consideration.

Among Semantic Web Services efforts, we believe that WSMO is making a step forward, in the direction of better clarifying how to use semantics for discovery purpose, by pointing up the importance of using also *mediators*. As a matter of fact, complete consensus is difficult to achieve, so a way to bypass such unrecoverable lack of agreement is necessary. Mediators in WSMO play exactly this role: "*[they] address the handling of heterogeneities that naturally arise in open environments*". WSMO proposes a classification of mediators according to their role in WSMO conceptual model. It proposes Ontology to Ontology mediators (ooMediators), Goal to Goal Mediators (ggMediators), Web Service to Goal Mediators (wgMediator), etc. For lack of space, we do not provide here a complete list of them and we prefer to refer to WSMO D2 [11] for a detailed explanation of their usage.

[6] For further understanding of Semantic Web Service Discovery we recommend reading WSMO Discovery [10] that presents, in the first part, an up-to-date survey of the state-of-the-art in semantic discovery techniques.

[7] http://kmi.open.ac.uk/projects/irs/

[8] http://lsdis.cs.uga.edu/Projects/METEOR-S/

3 A case of Semantic Discovery in eHealth

The lack of ontologies is one of the major obstacles in showing that semantic techniques can provide a solution for Service Discovery that is better than UDDI. As a matter of fact, building ontologies from scratch is difficult, but COCOON project is set in one of those knowledge intensive domains – the healthcare – that already investigated the use of ontologies in the past. In the healthcare field ontologies have been developed for a decade (e.g. GALEN[9]) and large standard terminologies, such as the International Classification of Diseases (ICD[10]), International Nonproprietary Names for Pharmaceutical Substances (INN[11]) and SNOMED clinical terms[12], etc., are widely employed to gain interoperability among eHealth solutions.

Moreover, most of times Service Discovery seems a back-end problem, but actually, if Semantic Web Services efforts will succeed, the use of Service Discovery tools in the future will be as frequent as using search tools today. Therefore in COCOON project, beside envisioning a clear back-end use of Service Discovery (which is not described in this paper), we also envision a realistic case of its daily use.

For instance, a typical usage scenario describes interaction between a General Practitioner (GP) and COCOON platform, which must provide access to the advice services offered by specialists organized in communities of practice (CoP). This access must be enabled on demand and it must allow the GP to select a CoP to arrange the meeting with. The selection criteria depends:

- on the correspondence between the problem the GP is looking advice for and the topics each CoP can offer advice on, and
- on the matching between the GPs' date-time preferences and the nominal availability of each CoP.

In order to facilitate the understanding of this scenario, in figure 1, we show a Semantic Discovery engine surrounded by a set of CoP (which are provider entities) and a COCOON platform (which is a requester entity). Each of the CoP exposes the functionality of arranging a meeting as a Web Service. The process that enables a GP to arrange a meeting with the most suitable CoP can be broken down in the following tasks:

1. the CoP provider and requester entities **agree on the ontologies to use** for modeling pathologies, drugs, advice services, date-time, etc.;
2. **if they cannot agree** on the use of a specific set of common ontologies, the definition of **ooMediators is required**. In this scenario, for instance, the CoP providers and the requester entities cannot agree on the use of a common date-time ontology. The CoP provider entities prefer to express the nominal availability of each CoP using a *week-based calendar* (e.g. the advice

[9] http://www.opengalen.org/
[10] http://www.who.int/classifications/icd/en/
[11] http://www.who.int/medicines/organization/qsm/activities/qualityassurance/inn/orginn.shtml
[12] http://www.snomed.org/

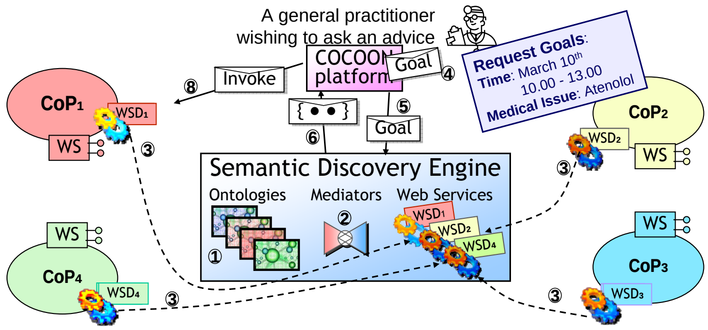

Fig. 1. A case of Service Discovery that enables a general practitioner to find the most appropriate advice service.

service is available on Thursday afternoon and Friday morning), whereas the requester entity prefers to express users' preferences using a *Gregorian calendar* (e.g. is the service available on April, 9th from 10.00 to 12.00?);

3. each CoP provider entity can then **register** in the Semantic Discovery engine its Web Service for arranging a meeting describing the medical issues the CoP deals with and the date–time intervals the CoP is normally available. For instance, a CoP provider entity may register its CoP as "*developer of alpha and beta blockers with nominal availability on Monday, Tuesday and Friday in the afternoon*";
4. similarly, a GP can discover the most suitable CoP by using a GUI, provided by the requester entity, in order to **express his/her goal** in terms of the available ontologies. For instance the GP asks an advice on "*the use of Atenolol preferring the meeting to be arranged on June 8th from 10.00 to 13.00 or on June 9th from 13.00 to 16.00*";
5. the **GP goal is submitted** to the Semantic Discovery engine;
6. the Semantic Discovery engine uses the ontologies and, if necessary, the mediators for **matching** the GP goal against the descriptions of the advice services offered by each CoP; then it returns a list of services for arranging a meeting, ordered by decreasing relevance;
7. the requester entity **displays the results list** to the GP;
8. the GP interactively **selects** a CoP and uses a GUI provided by each CoP provider in order to **invoke** the service and book the meeting;

4 Prototyping a WSMO Discovery engine

In order to have a proof of concept of the use of Semantic Web Services in COCOON, we started developing, within COCOON Glue [5], a WSMO–compliant discovery engine.

Since WSML[13] efforts in providing a language for formalizing WSMO ia a work-in-progress and because the the lack of tools for translating the abstract

[13] http://www.wsmo.org/wsml/

syntax into concrete syntaxes, we are developing COCOON Glue in f-logic [12]. Actually, we believe that f-logic offers the best trade-off between the recommendations proposed by W3C (namely RDF and OWL) and the efficiency we aim at in COCOON Glue since it can be translated into the deductive database language Datalog [13] allowing efficient query answering and easy implementation of rules on top of the ontology (this target, as we describe below, is crucial in our approach to discovery). Moreover, as WSML working group shows in [14], a subset of OWL (named OWL$^-$) can be translated into f-logic.

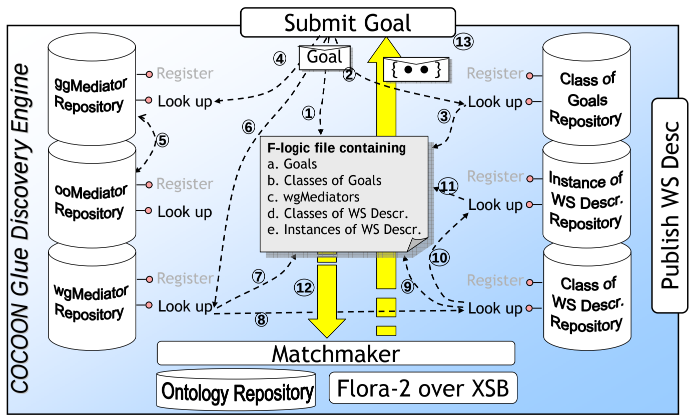

Fig. 2. This collaboration diagram depicts the components of COCOON Glue Discovery engine and their role in supporting the discovery process. The arrows illustrate the interactions required for building the f-logic file (shown in the center) from user goal. The numbers are provided to explain which information in the f-logic file is looked up in which component. For instance, in order to have the instances of Web Services descriptions in the f-logic file (cf. line e.), at first the goal is used to look up a set of appropriate wgMediators (cf. interaction 6), then the selected wgMediators are used to look up the corresponding classes of Web Services descriptions (cf. interaction 8), afterwards, these classes are used to look up the instances (cf. interaction 10), finally, the instances are included in the f-logic file (cf. interaction 11).

We built COCOON Glue around an open source f-logic inference engine called Flora-2[14] that runs over XSB[15], an open source implementation of tabled-prolog and deductive database system. Flora-2 provides only the reasoning support, while COCOON Glue wraps the inference engine and builds a WSMO infrastructure around it (see figure 2).

The basis of COCOON Glue infrastructure is a set of facilities for registering and looking up WSMO components (ontologies, goals, Web Services descriptions and mediators). Using these components we implemented a matching mechanism that relies on wgMediators as envisioned in section 4.2.1 of WSMO Primer [15]:

[14] http://flora.sourceforge.net/
[15] http://xsb.sourceforge.net/

> *[...] a service may be linked to certain goals that are solved by the service via a special type of mediators, named wgMediators. These links are useful in the Service Discovery phase.*

In order to use wgMediators, we distinguish between classes of goals (or classes of Web Services descriptions) and instances of these classes. In our approach requester entities must register a class of goals in order to be able to submit a goal. Similarly, in our approach provider entities must register a class of Web Services descriptions in order to be able to publish a Web Service description. The rational behind this choice is twofold. At set up time, a COCOON glue administrator can develop a wgMediator by using f-logic rules to assert the similarities that link a class of Web Services descriptions to a class of goals. At discovery time, it enables the use of a simple look up mechanism for selecting the most appropriate wgMediators for the submitted goal and the use of such mediators to match a goal instance against numerous Web Service description instances. *The discovery mechanism, then, becomes a composite procedure where the discovery of the appropriate wgMediator and the discovery of the appropriate service is combined.* Below we provide the pseudo-code of the procedure implemented in our discovery engine.

```
 1. ListOf<WebServicesReference, MatchingDegree> discover(Goal g) {
 2.   ListOf<WebServicesReference, MatchingDegree> results;
 3.   SetOf_Goal goals;
 4.   goals.add(g);
 5.   SetOf_ggMediator ggms;
 6.   ggms = lookupGGMediators(g);
 7.   foreach ggm in ggms
 8.     goals.add(mediate(g,ggm));
 9.   foreach goal in goals
 10.    SetOf_wgMediator wgms;
 11.    wgms = lookupWGMediators(goal);
 12.    for each wgm in wgms
 13.      results.add(match(g,wgm));
 14.  return results;  }
```

This `discover` procedure expects a goal as input and provides, as result, a list of pairs in which the first element is a reference to a Web Service and the second element is the degree of matching (exact, subsumed, plug-in and intersection as proposed in WSMO D5.1 [10]). The first part of the procedure (see lines 1-8) uses ggMediators to expand the submitted goal to a set of goals. Then (see lines 9-14), for each goal the appropriate wgMediators are looked up and used for matching the set of goals against the Web Services descriptions registered in COCOON Glue. The lookup procedures (e.g. `lookupWGMediator`) refer to a simple URL based query to the WSMO components' repositories within COCOON Glue. Finally the `match` procedure asks Flora-2 to evaluate the similarity rules embedded in the description of each wgMediator and returns references to the discovered Web Services and the degree of matching as list of pairs.

Wrapping up COCOON Glue process can be broken into three phases:

- *set up time*: provider and requester entities agrees on the ontologies to use. If they cannot reach an agreement on the ontologies to use (e.g. some likes SNOMED, other ICD) then ooMediators are required[16]. At this point provider and requester entities can formally describe (using the agreed ontologies and, if necessary, mediators) the classes of services and goals they are respectively going to use for publishing and querying COCOON Glue. But a final step is required in order to enable service discovery: the description of criteria for asserting similarity between a class of services and a class of goals. This is done writing f-logic rules in a wgMediator.
- *publishing time*: provider entities publish Web Services descriptions by the means of *instantiating a* description of a specific *class* among those previously registered.
- *discovery time*: requester entities ask COCOON Glue to discover services by instantiating a goal from a class of goals previously registered. The discovery mechanism, then, uses a composite procedure that combines the discovery of the appropriate wgMediators and the use of the similarity rules coded in them to discover the appropriate service.

5 WSMO at work in the healthcare field

In order to prove COCOON Glue concept and to test the discovery engine, we modeled in WSMO the use case illustrated in section 3. We used f-logic to describe the ontologies, the classes/instances of Web Services description, the classes/instances of goals and the wgMediators. Then, we populated COCOON Glue with some tens of realistic descriptions of Web Services for arranging meeting with a CoP. Finally, we developed a user interface[17] for running field tests within COCOON knowledge management platform (i.e. the component of COCOON architecture with responsibility for presenting COCOON services to the GPs). The CoP are hosted by the two implementation of such platforms[18] and real virtual meetings can be performed using the respective off-the-shelf collaboration tools.

The **ontologies** necessary to support this use case are the COCOON ontology, the advice ontology and two calendar ontologies.

COCOON ontology is a demonstrative ontology of hypertension and breast cancer domains derived from ICD-10 and INN. It contains the definition of a hundred concepts (like `disease`, `hypertension`, `breast neoplasm`, etc., `medication`, `beta-blockers`, etc., `part of the body`, `heart`, etc., `medical doctor`, `cardiologist`, etc.) and the relations among them (like beta blockers `control` bypertension,

[16] At this point of development of COCOON Glue we do not provide an environment for defining mediation in a declarative way, but only a facility to register a service that does the mediation.

[17] At http://cocoon.cefriel.it/COCOONGlue/Discovery/GUI/SDCoP.aspxFor readers can try out the user interface of Semantic Discovery of CoP.

[18] Livelink (http://www.opentext.com/) or MERMIG (http://www.mermig.com/).

cardiologists `deal with` heart, hypertension `affects` heart). Below we provide a small subset of the internal f-logic syntax.

```
disease[ affects=>>{bodyPart} ].
hypertension::disease[ affects=>>{artery, heart} ].
medication[ controlsDiseases*=>>{disease} ].
blockers::medication.
alphaBlockers::blockers[ controlsDiseases*=>>{hypertension} ].
atenolol::betaBlockers.
tenormine::atenolol.
```

The *advice ontology* describes a Community of Practice in terms of which kind of specialists are members of the CoP (cf. `coP`), which diseases are studied by the CoP and which medication are developed by the CoP. Then a service that provide the possibility of arranging a meeting with a CoP provides advice (cf. `providesAdvice`) with a given CoP in a given set of date-time intervals. Similarly, when GPs request an advice (cf. `getAdvice`), the medical issue and a set of preferable date-time intervals are required.

```
coP[
 hasSpecialists=>>specialist,
 studiesDiseases=>>disease,
 developsMedications=>>medication ].
providesAdvice[
 byCoP=>coP,
 availableIntervalSet=>intervalSet ].
getAdvice[
 forMedicalIssue=>medicalIssue,
 preferredIntervalSet=>intervalSet ].
```

Finally, two *calendar ontologies* are necessary in our use case to express the date-time intervals. One express date-time according to Gregorian calendar, which is useful for expressing the date-time preferences of the GP (e.g. April, 9th, 2005 from 10.00 to 12.00). The other one express date-time according to a week-based calendar, which is useful to describe nominal date-time availability of each CoP (e.g. every Monday afternoon and Friday morning).

Therefore, following WSMO approach, an agreement between provider and requester entities is nomore necessary: the CoP providers can keep expressing their nominal availability using the week-base calendar, while the GPs can express their date-time preferences using a Gregorian calendar, as long as a **mediator** is used to bypass the heterogeneity problem. In particular an *ooMediator* can be employed in translating the date-time from the Gregorian calendar to the week-based one. In our implementation, this ooMediator is realized by a Java program exposed as a Web Service.

Having these ontologies, we were able to describe in WSMO the capabilities of the class of **Web Services** for arranging a meeting with a CoP.

- the *pre-conditions* are: the input has to be the information about an advice request, the general practitioner has to ask an advice on one of the medical issues treated by the various CoPs, and the booking date has to be after the current date;

- the only *assumption* is that the general practitioner has the right to use the advice service;
- the *post-conditions* describe the possible advices the CoP can offer: it can answer to questions that regards its medical capabilities and it can provide advice only during its nominal available times;
- the *effect* is that the agendas of both the GP and the specialists in the CoP are updated with a reference to the scheduled meeting.

In the following, we provide an excertpt of the internal f-logic syntax. It shows how WSMO concepts (like `capability` and `postcondition`) are mixed with concepts from the advice ontology (like `providesAdvice`), from COCOON ontology (like `betaBlockers`) and from the week-based date-time ontology (like `monday_afternoon`).

```
capability-of-foo_SDCoP-ClassOfWS:capability[
 postCondition -> fooAdvice:providesAdvice[
  byCoP->fooCoP:coP[
   hasSpecialists->>{cardiologists},
   studiesDiseases->>{hypertension},
   developsMedications->>{alphaBlockers, betaBlockers}
  ],
   availableIntervalSet-> fooIntervalSet:intervalSet[
   values->>{monday_afternoon, tuesday_afternoon, friday_afternoon} ]]].
```

In a similar manner we were able to describe in WSMO a class of **goals** that asserts GP need of finding a CoP that can provide advice on a given medical issue in the date-times intervals the GP prefers. As above, we provide a fragment of the internal f-logic syntax that exemplifies an instance of goal.

```
fooGoal:goal[
 requestsCapability->fooGoal_capability[
  postcondition->fooGoal_getAdvice[
   forMedicalIssue->fooGoal_medicalIssue[askFor->atenolol],
   preferredIntervalSet->fooGoal_intervalSet[
values->>{thursday_lateMorning, friday_earlyAfternoon}] ]]].
```

Because of the fact that the date-time intervals are not expressed using the same ontology, we also defined a class of goals that express the GP goal in terms of the week-based calendar ontology and we used a **ggMediator** for translating instances of goal from one class to the other. This ggMediator, when invoked, simply rewrites the goal formulated by the GP using Gregorian dates (e.g. June, 8th 2005) translating it into days of the week (e.g. Wednesday) through the ooMediator illustrated above.

Finally, as we described in section 4, we expect COCOON Glue administrators to encode in a set of wgMediators the similarity rules that make the discovery engine able to match a class of goals against a class of Web Services descriptions. For instance, the rule that perform an exact match between the medical issue, expressed in the GP goal, and the medical capabilities of the CoP, described in Web Service description is:

```
exactMatchMedicationWithCoP(I,C) :-
 I[askFor->M], (
 (C[developsMedications->>M]);
 (M[controlsDiseases=>>D], C[studiesDiseases->>D]);
 (M[controlsDiseases=>>D], D[affects=>>B],
   C[hasSpecialists->>S], S[deals=>>B])).
```

This rule sais that there is an exact matching if M is a medication developed by the CoP C, or if M controls a disease that is studied by a CoP C, or if M controls a diseases which affects a part of a body which is studied by the specialist in the CoP C, etc. The rules for subsume and plug-in matching mainly differ from the one presented above because they broaden the search space to subconcepts and superconcepts respectively.

For lack of space we present only some snapshots of the internal f-logic syntax of WSMO components. Readers can refer to COCOON Glue Web site[19] for more detailed information.

6 Conclusion and future work

The main lesson we are learning from the work we are undertaking in applying WSMO in the healthcare field is that the clear separation between the ontologies used by each entity involved in COCOON simplifies and speeds up the gathering of consensus, which is alway difficult to reach in large groups, but especially in healthcare. This is mainly due to the adoption, in WSMO, of mediators. In particular, wgMediators appear to offer a flexible way for describing similarities between goals and Web Services descriptions, hence for enabling a semantic match between them. The way to transform existing terminologies like ICD into ontologies is still an open issue, but this is a topic that is not obvious to address and that will probably require lot of manual work.

Moreover, from a technical point of view, we believe that the subset of f-logic (OWL-Lite$^-$ together with instance–to–class relations) we are using for describing all the components except the mediators offers a good trade-off between expressiveness and performances. We easily modeled the presented use case and the performances[20] of the discovery engine with 100 Web Service descriptions remains under 2 seconds. Clearly this subset of f-logic is too restricted for describing similarity in the wgMediators in a suitable way. Actually, we have to use f-logic rules to achieve such task, but the layering of these rules over the rest of the language makes the description of similarities between disparate concepts easier.

Finally, the tasks that are currently being the subject of our research within COCOON Glue are:

1. selecting and adjusting the ontologies required for describing the healthcare services offered in COCOON, through the development of (possibly ad-hoc)

[19] http://cocoon.cefriel.it/RD2/
[20] the machine we used for the tests is a 2800MHz P4 processor with 1GB of RAM

mediation services to overcome heterogeneity of the various healthcare related ontologies;
2. extending the test cases of our discovery engine to include the other components still under development in COCOON project;
3. extending the approach to the regional eHealth services (starting from the SISS in Lombardy - Italy); and
4. starting the development of a use case that requires orchestration and choreography of eHealth services.

Acknowledgements

The research has been supported by the COCOON (IST FP6-507126) integrated project. We thank Prof. Stefano Ceri, Lara Gadda and Irene Celino for their precious comments and support.

References

1. Commission of the European Communities: (e-health - making healthcare better for european citizens: An action plan for a european e-health area. com (2004) 356)
2. Johnson, W.G., Brennan, T.A., et al.: The economic consequences of medical injuries. JAMA **267** (2002)
3. Bettoni, S., Gadda, L., Perdoni, V., Savoldelli, A.: COCOON - building knowledge driven and dynamically networked communities within european healthcare systems. In: eChallenges. (2004)
4. Della Valle, E., Cerizza, D., Celino, I., Gadda, L., Savoldelli, A.: The COCOON project. (In: Demos and Posters of the 2nd European Semantic Web Conference (ESWC 2005), Heraklion, Greece, 29)
5. Della Valle, E., Korda, N., Ceri, S., Dori, D.: Gluing web services through semantics: The COCOON project. In: Proceedings of ICCMSE. (2004)
6. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Semantic matching of web services capabilities. In: International Semantic Web Conference. (2002) 333–347
7. Haas, H., Brown, A.: Web services glossary. Technical report, W3C (2004)
8. OASIS: The uddi technical white paper. Technical report, OASIS (2004)
9. Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D.: Web services architecture. Technical report, W3C (2004)
10. Keller, U., Lara, R., Polleres, A., Toma, I., Kifer, M., Fensel, D.: D5.1 WSMO discovery. Technical report, WSML working group (2004)
11. Roman, D., Lausen, H., Keller, U.: D2 web service modeling ontology (WSMO). Technical report (2005)
12. Angele, J., Lausen, G.: Ontologies in f-logic. In: Handbook on Ontologies. (2004) 29–50
13. Dantsin, E., Eiter, T., Gottlob, G., Voronkov, A.: Complexity and expressive power of logic programming. ACM Comput. Surv. **33** (2001) 374–425
14. de Bruijn, J., Polleres, A., Lara, R., Fensel, D.: Owl-. Technical report, WSML (2004)
15. Feier, C., Domingue, J.: D3.1 WSMO primer. Technical report, WSMO working group (2005)