A use case of Glue WSMO Discovery Engine

Emanuele Della Valle and Dario Cerizza

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

Abstract. Glue[1] [1] is a WSMO compliant discovery engine [2][3] that aims at developing an efficient system for the management of semantically described Web Services and their discovery. In this document we provide a use case of mediator centric Discovery for eHealth we have developed in the COCOON project[2] [4] and we deployed both in COCOON project and in Nomadic Media project[3].

1 A case of mediator centric Discovery for eHealth

In this use case of Web Service Discovery we describe an interaction between a General Practitioner (GP) and Glue WSMO Discovery Engine with the intent to find medical *advice* [4] and *teaching* services offered by specialists organized in communities of practice (CoP). In particular Glue WSMO Discovery Engine takes responsibility for enabling on demand access to services for arranging virtual meeting; while the actual arrangement and the subsequent meeting is supported by external collaboration services.

The general criteria for matching a GP goal against the description of the services offered by a CoP are on the correspondence between the GP's problem and CoP's expertises and on the matching between the GPs' date–time preferences and the nominal availability of each CoP.

Advice will predictably be the more frequent reason for a GP for starting an interaction, as it normally could be triggered by facts happening during the practice time (e.g. questions by patients). Most of the times the *clinical capabilities* of the CoP may be the ones the GP wishes. But, sometimes (e.g. for more difficult and rare patient cases) *research capabilities* may be sought.

Teaching, on the other hand, will be predictably less frequent and a reason to start a request to the system could happen in the 1-hour/week time that is normally reserved for contacting peers[5], as it normally could be triggered by GP's reflection on his/her week's practice.

[1] http://glue.cefriel.it

[2] COCOON 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.

[3] The Nomadic Media project resides under the Eureka/ITEA program. Within this project we addressed the problem of out to provide mobile access to healthcare services.

[4] a second opinion focused on a particolar medical problem

[5] As reported in a february 2005 national survey of Italian GPs.

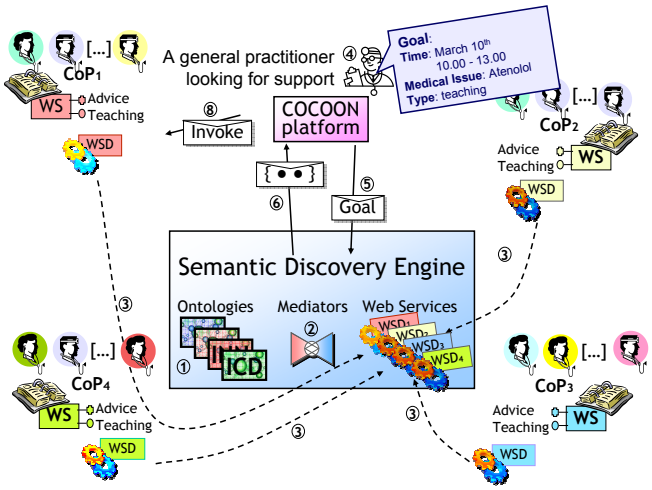

Fig. 1. A case of Service Discovery that enables a general practitioner to find the most appropriate advice/teaching service.

In order to facilitate the understanding of this scenario, in figure 1, we show Glue WSMO Discovery Engine surrounded by a set of CoP (which are provider entities) and a requester entity (named COCOON platform). Each of the CoP exposes the functionality of arranging the two types of 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:

- Set up time:
  1. the service provider and requester entities **agree on the ontologies to use** for modeling pathologies (e.g. they may agree on using the ICD), drugs (e.g. they may choose International Nonproprietary Names for Pharmaceutical Substances – INN[6]), advice services, date–time, etc.;
  2. **if they cannot agree** on the use of a specific set of common ontologies, the use of **mediators 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 service is available on Thursday afternoon and Friday morning), whereas the requester entity prefer to express users' preferences using a *Gregorian calendar* (e.g. is the service available on April, 9th from 10.00 to 12.00?);
- Publishing time:
  3. each CoP provider entity can then **register** in the Glue WSMO Discovery Engine its Web Service for arranging a meeting describing the clinical

[6] http://www.who.int/medicines/organization/qsm/activities/qualityassurance/inn/orginn.shtml

capabilities the CoP holds and the date–time intervals the CoP is normally available. For instance, a CoP provider entity may register its CoP as "*one that delivers intervention based on alpha and beta blockers with nominal availability on Monday, Tuesday and Friday in the afternoon for advice and teaching*";

– Discovery time:
  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 term of the available ontologies. For instance the GP asks "*a teaching session 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 requester entity **submits the GP goal** to the Discovery Engine;
  6. the Semantic Discovery engine uses the ontologies and the mediators for **matching** the GP goal against the descriptions of the advice services offered by each CoP; then it returns a list of references to Web 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** one of the CoP till he/she finds one arrange a meeting with.

2 Putting Glue WSMO Discovery Engine at work

We modeled following the WSMO approach the use case illustrated in the previous section. 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 Glue with some tens of realistic descriptions of Web Services for arranging meeting with a CoP. Finally, we developed two user interface one for Web Browsers in use within COCOON project and one for mobile devices[7] demonstrated within Nomadic Media project.

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

medical 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., `specialist`, `cardiologist`, etc.) and the relations among them (like beta blockers `control` hypertension, cardiologists `deal with` heart, hypertension `affects` heart).

The *advice ontology* describes the domain of advice and teaching services defining the concept of clinical, research and teaching capabilities for a Community of Practice.

– *Clinical Capabilities* describes the CoP in terms of
  - `hasClinicalSpecialist`: the list of the kind of specialists grouped by the CoP (e.g. Cardiologist, Urologist, Pneumatologist, Dermatologist, etc.),

[7] publicly available at http://glue.cefriel.it/glueclient

  - `managesDiseases`: the list of diseases managed by the CoP as ICD codes (e.g. Diabetes – ICD9CM 250.00), and
  - `deliversIntervention`: the list of the diagnostic / therapeutic / preventive interventions (including pharmaceuticals) delivered by the CoP;
- *Clinical Research Capabilities* describes the CoP in terms of
  - `hasResearchSpecialists`: the list of the kind of specialists grouped by the CoP (e.g. Statistician, Social worker, Psychologist),
  - `studiesDiseases`: the list of diseases which are actively researched by the CoP (e.g. Gastric ulcer [ICD10–K25] Prevention), and
  - `studiesIntervention`: the list of the diagnostic / therapeutic / preventive interventions (including pharmaceuticals) which are actively researched by the CoP; and
- *Teaching Capabilities* describes the CoP in terms of
  - `hasTeachingExpertise`: the list of teaching roles that the CoP can fulfill (e.g. Teacher, OnlineTeacher, Tutor, OnlineTutor, etc.)
  - `hasAuthoringExpertise`: states the availability of online/offline collaborative working tools (i.e. for teaching) within the CoP (e.g. NetMeeting, Skype, Messenger, etc.)

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-based 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 following the WSMO approach the capabilities of the class of **Web Services** for arranging a meeting with a CoP. We distinguish between *a generic class of services* that provide the possibility of arranging a meeting and the specific advice / teaching meeting arrangement classes of services. In the former case we describe the post condition of the class of services asserting that it provides support (cf. `providesSupport`) with a given CoP and in a given set of date–time intervals. Then, if a CoP offers only advice support, we restrict the support to advice using `providesAdvice` and, in a similar manner, if a CoP provides only teaching support, we state that using `providesTeachingSupport` instead of `providesSupport`. In this way we define a class hierarchy of Web Service descriptions with the generic meeting arrangement class of services on top and the ones for arranging advice / teaching meeting below.

The description of the generic meeting arrangement class of Web Service asserts that:

- 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 meeting the CoP is available for: it can offer support that regards its capabilities and it can provide support 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 a fragment of the internal f–logic syntax of a Web Service description instance of generic service for arranging a meeting. It shows how WSMO concepts (like `capability` and `postcondition`) are mixed with concepts from the advice ontology (like `providesAdvice`), from medical ontology (like `betaBlockers`) and from the week-based date–time ontology (like `monday_afternoon`).

```
capability-of-foo_SDCoP-ClassOfWS:capability[
 postCondition -> fooSupport:providesSupport[
  byCoP->fooCoP:coP[
   hasClinicalCapabilities ->> foocc:clinicalCapabilities[
    hasClinicalSpecialists->>{cardiologists},
    managesDiseases->>{hypertension},
    deliversIntervention->>{alphaBlockers, betaBlockers}]],
   availableIntervalSet-> fooIntervalSet:intervalSet[
   values->>{monday_afternoon, tuesday_afternoon, friday_afternoon} ]]].
```

Then, when we refine such generic class of services considering advice [teaching], we extend such **class of Web Service descriptions** by specifying in *post-conditions* that the CoP can provide advice [teaching support] with regards to its clinical/research capabilities [teaching capabilities] and it can provide support only during its nominal advice [teaching] available times.

As above we provide a fragment of the internal f–logic syntax for a case of CoP that offers only teaching support. It shows that `providesTeachingSupport` and `availableTeachingIntervalSet` is used instead of the generic `providesSupport` and `availableIntervalSet`.

```
capability-of-footeach_SDCoP-ClassOfWS:capability[
 postCondition -> footeachSupport:providesTeachingSupport[
  byCoP->footeachCoP:coP[
   hasTeachingCapabilities ->> footeachtc:teachingCapabilities[
    hasDomainExpertise->>{cardiologists},
    hasTeachingExpertise->>{onLineTutor},
    hasAuthoringExpertise->>{netMeeting}]],
```

```
availableTeachingIntervalSet-> footeachIntervalSet:intervalSet[
values->>{tuesday_afternoon} ]]].
```

In a similar manner we used WSMO constructs for defining a hierarchy of **classes of goals** that asserts GP's need of finding a CoP that can provide advice/teaching support on a given medical issue in the date–times intervals the GP prefers. We do not repeat here what we say above about the hierarchy of classes of Web Service descriptions, we focus, instead, on the need for mediators. In fact, as described in section 1, in our use case no agreement was reached in the date–time ontology to use. To bypass such heterogeneity we defined also a hierarchy of classes of goals that express the GP goal in terms of the week-based calendar ontology (the one chosen by CoP providers) 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, we expect 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 what the GP ask for and the capabilities of the CoP is:

```
exactMatchMedicationWithCoP(GP,C) :- GP[askFor->M], (
 (C[deliversIntervention->>M]);
 (M[controlsDiseases=>>D], C[manageDiseases->>D]);
 (M[controlsDiseases=>>D], D[affects=>>B],
   C[hasClinicalSpecialists->>S], S[deals=>>B])) .
```

This rule says that there is an exact matching if CoP `C` delivers intervention `M`, or if `M` controls a disease that is managed 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.

Due to the fact that we have two parallel hierarchies of classes we write three wgMediators: one links a generic service for arranging a meeting to a generic goal for requesting support, an the other two link respectively a service for arranging advice meeting to a request for advice and a service for arranging a teaching meeting with a request for teaching support. The rules in the three wgMediators largely overlap, so we defined a sort of library of matching rules and we call them from the three wgMediators. We found useful the possibility of building in WSMO also hierarchies of wgMediators, so that a the two specific wgMediator can be defined by extending a the generic one.

For readability we present only some snapshots of the internal f–logic syntax of WSMO components. The complete f–logic descriptions of the use case are available at http://glue.cefriel.it/.

Acknowledgements

The research has been supported by the COCOON (IST FP6-507126) integrated project and Nomadic Media ITEA (Information Technology for European Advancement) project, financed by the Italian Public Authorities. We thank Prof. Stefano Ceri, Doc. Michele Tringali, Lara Gadda, Maria Rodriguez and Irene Celino for their precious comments and support.

References

1. Della Valle, E., Cerizza, D.: Cocoon glue: a prototype of wsmo discovery engine for the healthcare field. In: Proc. of 2nd WSMO Implementation Workshop WIW'2005. (2005)
2. Keller, U., Lara, R., Lausen, H., Polleres, A., Predoiu, L., Toma, I.: D5.2 WSMO Discovery Engine. Technical report (2004)
3. Keller, U., Lara, R., Polleres, A., Toma, I., Kifer, M., Fensel, D.: D5.1 WSMO Web Service Discovery. Technical report (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)