Semantic Web Service Composition in the NeP4B Project: Challenges and Architectural Issues* (Extended Abstract)

Federica Mandreoli[1], Wilma Penzo[2], and Antonio Massimiliano Perdichizzi[1]

[1] DII
Università di Modena e Reggio Emilia
Modena, Italy
{mandreoli.federica, antonio.perdichizzi}@unimo.it
[2] DEIS
Università di Bologna
Bologna, Italy
wpenzo@deis.unibo.it

Abstract. Semantic Web service discovery and composition frameworks proposed so far assume for the most part a centralized registry that holds information of all the Web services available at any given time. This solution does not well cope with the scalability and flexibility requirements of dynamic, fast changing contexts. As part of the NeP4B project, in this paper we propose an alternative peer to peer architecture based on the Goal concept.

1 Introduction

Service Oriented Architectures (SOA) and Web services (WS) as a way to realize the formers have been in both the industrial and scientific focus for many years. They provide a new perspective to look at the Internet and at its potentials for supporting business.

Currently the Internet is mainly a collection of information but does not yet support processing this information. Recent effort around the Web services try to lift the Internet to a new level of service enabling full cooperation and integration between users. The ultimate vision is to discover, invoke and compose Web services to create new complex services fully automatically.

Proposed standardization of basic WS capabilities, i.e. communication (SOAP), description (WSDL) and discovery (UDDI)[?], only address part of the overall stack that needs to be available in order to eventually achieve large scale interoperation of Web services. Fundamental to cope with this issue is the need to make such services computer interpretable, that is to create a Semantic Web of

* This work has been partially supported by the Italian Council co-founded FIRB Project: “NeP4B: Networked Peers for Business”.

services whose properties, capabilities, interfaces and effects are encoded in an unambiguous form [?]. Semantic Web Services (SWS) combine Semantic Web research efforts with the Web services world. Their strength lie in being machine-accessible, as they are founded on the Web Service technology, and machine-understandable, as part of the Semantic Web vision.

The NeP4B project is an Italian Council co-founded FIRB project whose main aim is to investigate and design a technology as the basis for creating a network of semantic peers, providing advanced data-driven semantic services for B2B applications, where companies can classify their own profiles, offers, services and other features so as to gain public visibility to potential customers and partners. These semantic peers will cooperate on a P2P basis to deliver the targeted semantic services to all the users of the semantic peer infrastructure. As part of the NeP4B project, we are interested in defining a SWS scalable and flexible architecture to support large scale service interoperation, and particularly composition, in a broad and heterogeneous environment, where each user may as well be a provider of his own services and a consumer.

Starting from this context definition (Section **??**), we aim to propose a possible architectural solution to face the identified challenges in the NeP4B project (Section **??**). Finally, we provide conclusions and future work (Section **??**).

2 Motivations

In this Section, we first introduce the architectural challenges to be faced in the NeP4B Project for supporting large scale service interoperation. We then present the core concepts of our proposal.

2.1 The NeP4B Scenario

To compete in the global market context, ICTs are a key asset to gain and sustain competitive advantage. Yet, for small and medium-sized enterprises (SMEs), the high cost level of an IT project and the high risk rate involved in such projects [?], represent a quite insuperable barrier to technological innovation.

Taking this into account, NeP4B aims to develop an advanced technological infrastructure to support SMEs by enabling them to search for partners, exchange data, negotiate and collaborate without limitations and constraints, regardlessly of nature, size and geographic location. In order to do this, NeP4B relies on the concept of semantic peers constituting a virtual network (see Fig. **??**) of intelligent, trusted and distributed data-driven services, with high added value.

In this context, peers can be single SMEs as well as mediators representing groups of companies. Each semantic peer may as well be a provider of its own services and a consumer. It is fully autonomous of participating to the network and exposing Web services on the basis of its own internal and external business needs.

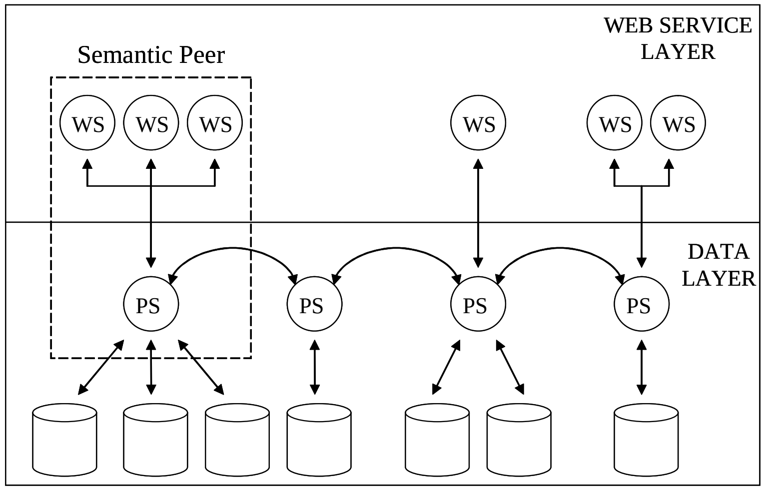

Fig. 1. NeP4B Architectural Layers

Semantic Web Services are powered by the underlying data, which are collections of structure, semi-structured, unstructured and multimedia data and which are described by the peer's schema (PS). In such a cooperative context, data can also be shared among peers in order to allow services, particularly information providing services, to collect information by spanning over the network. To this end, mappings between schemas are provided locally between pairs of peers. In Figure **??**, this layer is referred as the data layer. On top of this layer, there is the Web service layer consisting in the data-driven SWSs that a given peer offers. Each SWS is marked up by adopting an ontology language such as DAML/OWL-S [**?**] and refers to the underlying data by using the local dictionary of the peer's schema.

In this distributed and heterogeneous context, services have to be searchable to meet users' requests and invocable both for a stand alone execution or an automatic composition process [**?**,**?**]. The composition issue has a crucial role: Being able to automatically re-use existing services to generate new services would allow scalability, flexibility and effectiveness of the network, reducing human effort otherwise needed. To support it, in this paper we provide a broad framework to overcome the heterogeneity of dictionaries and the lack of shared service knowledge due to the autonomy of peers in our dynamic environment.

2.2 Our contributions

Much of the work done on Web service architectures proposes solutions based on centralized registries, such as UDDI [**?**], where every Web service coming on line advertises its capabilities and functionalities with the registry. Centralized control of published services allows to ease discovery and composition of services but it suffers from the traditional weaknesses of centralized systems, namely single point of failure, and performance bottlenecks.

An alternative to this approach is provided by P2P computing, where Web services interact with each other dynamically, without any centralized control. In such a context, there have been several proposals for Semantic Web service discovery (e.g. [**?**,**?**]). However, this alternative does not well support automatic composition of services because it does not provide a known and definite service space. There have been several proposals to overcome this problem by either trolling both the construction of the overlay network and the location of data within the system, i.e. structured systems as Chord (e.g. [**?**]), or only defining its topology ex-ante (e.g. [**?**]).

Finally, it should be noted that both approaches rely on a common service dictionary for composition to take place.

Considering all of the above, we propose a hybrid approach which exploits the advantages of centralized registries for service discovery and composition, as well as the dynamism of non-structured P2P networks that ensure the peers' full autonomy. The main idea is that, while not centralizing the knowledge of what specific services are available in the system, we keep centralized knowledge of what objectives may be satisfied within the network, namely *Goals*. A Goal is the conceptualization of a domain of services whose ultimate aims are identical or similar. For example, all services that, in the same geographical area provide driving directions from a departure point to a destination. Some could allow the requester to choose among different route options, for example the shortest or the most economical one, while others would only provide the fastest. Even if they may have different reference dictionaries, specific requirements or functionalities, they do answer the same requestor's need: To provide driving directions within a certain geographical area. Therefore they would be well represented by the same goal. Each Goal specifies therefore a subnetwork of specific services, and it is stored in an appropriate repository, called *Goal Repository*.

Using a repository of Goals has several advantages that do not come with loss of flexibility or scalability. Firstly, Goals constitute the domain for the composition purposes. Now its dimension is greatly reduced w.r.t. the underlying service level as goals only represent the objectives the network is able to satisfy rather than how they are satisfied. In this way, we move the issues of discovery and defining a composition pattern from service level to Goal level, namely *Goal discovery* and *Goal composition* respectively. Once Goals have been identified, it is possible to limit the search of the most suitable candidates within their own service networks, e.g. for the composition synthesis process [**?**,**?**].

Secondly, the Goal layer acts as a "semantic service integrator" reconciling peers' service models and dictionaries. Goals are indeed described in a common language and consistency of concepts within the Goal space is ensured by referring to a domain ontology. In this way the inherent heterogeneity of an open P2P system is reconciled within a homogenous space allowing the communication of different services through the Goal layer. In the literature, there are several works dealing with the issue of building semantic service integration sy-

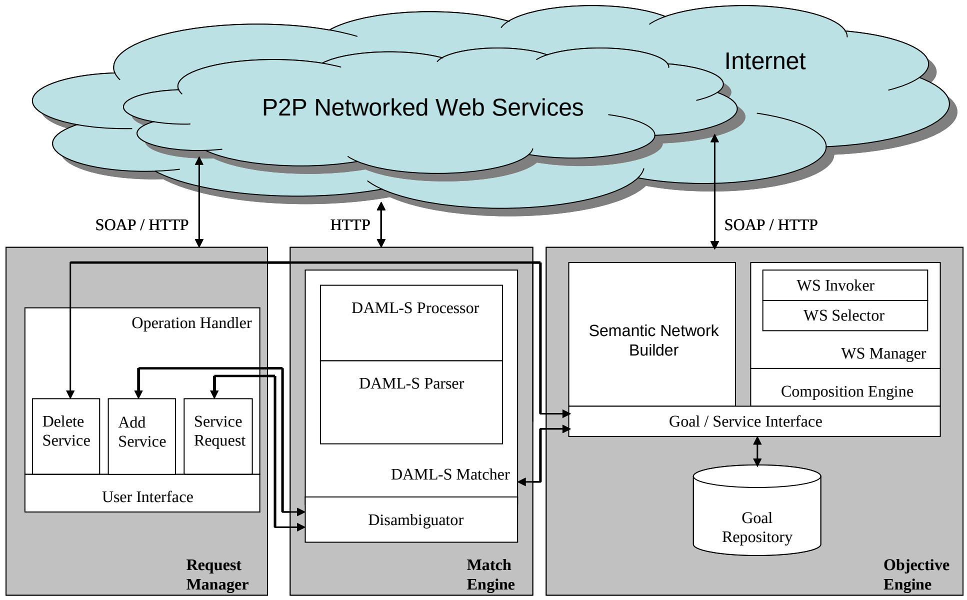

Fig. 2. The FLO^{2}WER Architecture

stems for composition purposes (e.g. [**?**,**?**,**?**]). However, they mainly focus on the translation of service descriptions into an internal form to apply specific techniques for automatic composition. In this paper, rather than focusing on this aspect, we create a proper interoperable environment for such kind of proposals be applicable in our context, regardless of their specificity.

3 The FLO^{2}WER Architecture

A schema of the FLexible GOal-Oriented WEb SeRvice (FLO^{2}WER) architecture we propose is shown in Figure **??**. It is a high level description which is meant to show the basic processes and data flows needed to support automatic semantic Web service handling. It is composed of three main modules, the *Request Manager*, the *Match Engine* and the *Objective Engine*, and it is founded on the community ontology [**?**] described in the Goal Repository.

The user interacts with the system through the Request Manager. Activating the other modules, which are described in the following subsections, it allows to perform three basic operations: Look for, add, or delete a Web service.

3.1 The Goal Repository

The Goal Repository comprises several aspects: a DAML-S description of Goals and a DAML domain ontology which represents the semantics of the information and data of the application domain. It may also include a service ontology specifying the meaning of the offered operations. To this extent, we adopt two semantic integration approaches [**?**]: A service-oriented approach for Goal descriptions, as they are built taking into account the service available in the underlying P2P network; a client-tailored approach for the domain ontology which is independent from the services available and that, if needed, can be downloaded off the Web.

Here, DAML-S is used to describe capabilities of a Goal as the Service Profile of a Web service. It describes a Goal as an atomic process thus specifying what the Goal is for, the inputs it requires, the outputs it produces and the preconditions that must hold for the Goal to take effect. DAML-S Service Profile also describes the post-conditions, i.e. the service execution effects on the real world. However, for Goal discovery and composition, we are concerned with the knowledge effects (outputs) rather than the physical effects (post-conditions) of executing Web services, which might in turn be relevant for invocation. Inputs, outputs, and pre-conditions refer to the domain ontology and are mapped towards the underlying Goal's subnetwork of services. Thus, we do not maintain mappings from the domain ontology to the peers' schemas, but they can be derived from the mappings between Goals and services.

3.2 The Match Engine

It is invoked by the Request Manager when the user adds or looks for a service and it is composed by the *Disambiguator* and the *DAML-S Matcher*. The Disambiguator takes non-DAML-S user's request and outputs a corresponding DAML-S description disambiguating the information managed against the domain ontology. A disambiguation process is described in [**?**]: Future research on this approach aims to also disambiguate natural language requests within an accuracy range. The DAML-S Matcher takes a DAML-S user's request to match it with the Goal's descriptions. [**?**] proposes a possible implementation of the matcher in [**?**] in an unstructured P2P context. It is composed by the *DAML-S Parser* and the *DAML-S Processor*. The former translates DAML ontologies and DAML-S specifications in a set of predicates, whereas the latter implements a DAML inference engine.

3.3 The Objective Engine

The central component is the *Goal/Service Interface*. Besides mediating between the other components, it allows the automatic creation, activation, deactivation and deletion of a Goal. A Goal is deactivated when its subnetwork is empty, and it is activated again if it matches a new service to be added to its network. There may be several possible policies to adopt for deletion, e.g. timeouts: A deactivated Goal could be deleted after a certain time interval during which it did not match any new added service.

The *Semantic Network Builder* is activated to accomplish a delete or a add service request and drives the integration process necessary to add or delete a service. It maps each Goal to only one service, called the *entry point* of the Goal subnetwork[3]. While a structured network is supposed not to be appropriate in the NeP4B context, we can still control how each subnetwork should evolve, optimizing a trade-off between the cost of establishing mappings and the cost of navigating the network in the search of proper services to be chosen. There are several topologies to compare, such as ring, bus, or Cayley graphs networks like hypercubes or star graph [**?**]. When a new service joins the network, it is matched against existing Goals. If a matching Goal is found, the Semantic Network Builder maps the new service semantically to a service of the subnetwork, namely the *service integrator*, chosen depending on the policies adopted for the network topology. When a service leaves the network, the same process takes place to replace canceled semantic mappings with new ones. Such new semantic mappings are derived by the Goal matching information, along the path of mappings from the entry point to the service integrator, assuming the mapping function to be transitive. At last, when a matching Goal is not found for a new service, or when the last service of a Goal subnetwork leaves the system, the Semantic Network Builder passes this information to the Service/Goal Interface, that provides to create or deactivate the Goal, respectively.

The Composition Engine implements the Goal discovery and composition, identifying Goals suitable to answer a user's request. Once one Goal or a pattern of Goals have been found, the Composition Engine invokes the WS Manager which handles the automatic service composition process. It is constituted by the *Web Service Selector* and the *Web Service Invoker*. The Web Service Selector enters, through the entry point, each of the Goal subnetwork identified, and selects the most suitable service based on different ranking criteria such as reliability, cost, quality of service, trust and reputation and, if available, on its process description[4]. Once Web services have been selected, the Web Service Invoker manages their execution and communication. Successful Goal composition patterns are then stored in the Goal Repository for future use.

4 Conclusions

In this paper, we described a broad framework to achieve a flexible and scalable Web service interoperating environment in an open, dynamic and heterogeneous P2P context, such as it is the NeP4B's. The FLO^{2}WER architecture we propose aims to overcome the heterogeneity of dictionaries and the lack of shared service knowledge due to the autonomy of peers in our network, and allows to create the proper conditions for the application of specific techniques for automatic composition of semantic Web services. In order to do so, we have adopted a hybrid

[3] There may be also more entry points to one subnetwork to avoid single point of failure. What matters is that the Goal does not have to know the whole set of services which constitute its subnetwork.

[4] DAML-S Web service description includes a Process Model, where it is defined what is needed for a proper interaction of services

approach which exploits the advantages of centralized registries for service discovery and composition, as well as the dynamism of non-structured P2P networks that ensure the peers' full autonomy. This is performed by introducing Goals as centralized knowledge of what objectives may be satisfied within the network, which allows to greatly reduce the domain for the composition purposes and to create a semantic service integrator layer.

Due to the peculiarities of the NeP4B project, we believe that to face large scale interoperation of semantic Web services, we first had to address the architectural issues involved. For these reasons, this work might constitute a solid starting point for the NeP4B project as it defines the basic building blocks and execution flows to enable automatic service discovery and composition. In our future works we will focus on the development of the FLO^{2}WER components.

References

1. UDDI: *The UDDI Technical White Paper.* http://www.uddi.org/, 2000.
2. B. Van Ark et al. ICT Investments and Growth Accounts for the European Union. Tech. Rep., European Commission, 2002.
3. D. Berardi et al. Automatic Composition of E-Services that Export their Behaviour. In *Proc. of ICSOC*, 2003.
4. D. Berardi et al. Automatic Composition of Transition-based Semantic Web Services with Messaging. In *Proc. of VLDB*, 2005.
5. D. Berardi et al. Automatic Web Service Composition: Service-tailored vs. Client-tailored Approaches. In *Proc. of AISC Workshop (in conj. with ECAI)*, 2006.
6. D. Wu et al. Automating DAML-S Web Services Composition Using SHOP2. In *Proc. of ISWC*, 2003.
7. F. Emekçi et al. A Peer-to-Peer Framework for Web Service Discovery with Ranking. In *Proc. of ICWS*, 2004.
8. M. H. Burstein et al. DAML-S: Web Service Description for the Semantic Web. In *Proc. of ISWC*, 2002.
9. M. Paolucci et al. Semantic Matching of Web Services Capabilities. In *Proc. of ISWC*, 2002.
10. M. Paolucci et al. Using DAML-S for P2P Discovery. In *Proc. of ICWS*, 2003.
11. M. Pistore et al. Automated Synthesis of Composite BPEL4WS Web Services. In *Proc. of ICWS*, 2005.
12. M. Schlosser et al. A Scalable and Ontology-Based P2P Infrastructure for Semantic Web Services. In *Proc. of P2P*, 2002.
13. R. Hamadi and B. Benatallah. A Petri Net-based Model for Web Service Composition. In *Proc. of ADC*, 2003.
14. F. Mandreoli, R. Martoglia, and E. Ronchetti. Versatile Structural Disambiguation for Semantic-aware Applications. In *Proc. of CIKM*, 2005.
15. S. A. McIlraith, T. Cao Son, and H. Zeng. Semantic web services. *IEEE Intelligent Systems*, 16(2):46–53, 2001.
16. S. Narayanan and S. McIlraith. Simulation, Verification and Automated Composition of Web Services. In *Proc. of WWW*, 2002.
17. S. M. P. Papazoglou and D. Georgakopoulos. Service Oriented Computing. *CACM*, 46(10), 2003.
18. C. Schmidt and M. Parashar. A Peer-to-Peer Approach to Web Service Discovery. *WWW*, 7(2), 2004.