Journal of Biomedical Informatics 41 (2008) 848–859

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Combining Semantic Web technologies with Multi-Agent Systems for integrated access to biological resources

Francisco García-Sánchez [a,*], Jesualdo Tomás Fernández-Breis [a], Rafael Valencia-García [a], Juan Miguel Gómez [b], Rodrigo Martínez-Béjar [c]

[a] Dpto. Informática y Sistemas, Facultad de Informática, Campus de Espinardo, Universidad de Murcia, 30100 Espinardo (Murcia), Spain
[b] Dpto. de Informática, Escuela Politécnica Superior, Campus de Leganés, Universidad Carlos III de Madrid, 28911 Leganes (Madrid), Spain
[c] Dpto. Ingenieria de la Información y las Comunicaciones, Facultad de Informática, Campus de Espinardo, Universidad de Murcia, 30100 Espinardo (Murcia), Spain

ARTICLE INFO

Article history:
Received 31 August 2007
Available online 23 May 2008

Keywords:
Bioinformatics
Ontologies
Semantic Web
Multi-Agent Systems
Semantic Web Services
Data heterogeneity

ABSTRACT

The increasing volume and diversity of information in biomedical research is demanding new approaches for data integration in this domain. Semantic Web technologies and applications can leverage the potential of biomedical information integration and discovery, facing the problem of semantic heterogeneity of biomedical information sources. In such an environment, agent technology can assist users in discovering and invoking the services available on the Internet. In this paper we present SEMMAS, an ontology-based, domain-independent framework for seamlessly integrating Intelligent Agents and Semantic Web Services. Our approach is backed with a proof-of-concept implementation where the breakthrough and efficiency of integrating disparate biomedical information sources have been tested.

1. Introduction

There is currently a huge volume of biomedical and genomic data Internet-available [1]. However, data are distributed into heterogeneous biological data sources, with little or even none information organisation. Therefore, integration and exchange of data within and among organisations is a universally recognised need in bioinformatics [2]. One of the major obstacles for integration efforts in bioinformatics is that relevant information is widely distributed, both across the Internet and within individual organisations. Besides, it can be found in a variety of storage formats, including structured and semi-structured ones.

The Semantic Web [3] provides a common framework that enables for data integration, sharing and reuse from multiple sources. Particularly, the use of ontologies [4] for domain knowledge representation can help bioinformatics in solving the heterogeneity problem. On the other hand, goal-driven architectures and service-based, loosely-coupled infrastructures have proved to be very useful when dealing with data and functionality integration from different and disparate sources. Web Services are the leading technology for developing service-oriented architectures. They provide the means for universal, remote execution of functionality. The breakthrough of adding semantics to Web Services leads to the Semantic Web Services paradigm [5]. Semantic Web Services technology enables the automation of service discovery, composition, invocation and monitoring. However, potential users might deter from using Semantic Web Services since their underlying formalisation and unease of use hampers its use from rich user-interaction perspective.

Agent technology has been broadly studied over the last 30 years. This field emerged due to the promising benefits of having applications with a technology that allows systems to decide by themselves what they need to do in order to satisfy their design objectives [6]. The usefulness of agent technology in a Semantic Web Services environment is 2-fold. They can act as the autonomous software entities that discover, compose, invoke and monitor services without human intervention, thus preventing users from carrying out this kind of arduous, time-consuming tasks. Besides, agents possess the ability to adapt to changing situations and handle the dynamism of Semantic Web Services environments.

In this paper, we present an ontology-based framework that successfully combines both Agent and Semantic Web Services technologies to enable the integrated access to biological data sources. The main goal is the seamless integration and application of these technologies in such a way that their deficiencies are overcome and their utility maximised. Our contribution is an overall solution, based on a fully-fledged architecture and proof-of-concept implementation that serves as a meeting point for both service consumers and service providers through a user-friendly, web-based graphical interface. Once the foundations and main concepts of the referred framework are pointed out and its

* Corresponding author. Fax: +34 968 36 4151.
E-mail address: frgarcia@um.es (F. García-Sánchez).

1532-0464/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2008.05.007

architecture detailed, we show how the framework can help in solving the integration and heterogeneity problems in the challenging field of bioinformatics.

The rest of the paper is organised as follows. In Section 2, the fundamentals of the technologies employed in our research are described. A summary of some of the existent resources on biomedical information throughout the Internet is presented in Section 3. In Section 4, a framework for effectively integrating Agent and Semantic Web Service technologies is formulated. In Section 5, details of how this framework can be applied to the biological domain are provided. Finally, conclusions and future work are put forward in Section 6.

2. Background

The aim of this section is to provide a detailed state-of-the-art of the technologies utilised in this work and a description of the application domain (i.e. Bioinformatics).

2.1. Multi-Agent Systems and Semantic Web Services

2.1.1. Agent technology

The Intelligent Agent (IA) and Multi-Agent System (MAS) area has received ever-increasing attention by researchers over the last few years. This field emerged due to the promising benefits of having applications with a technology that allows systems to decide for themselves what they need to do in order to satisfy their design objectives. A common accepted definition of the term 'Agent' determines that an agent is a computer system situated in some environment and capable of autonomous action in this environment in order to meet its design objectives [6]. Wooldridge also highlights that an agent has to fulfil some properties in order to become intelligent: *reactivity* (i.e. the ability to perceive its environment and respond to changes in it in a timely fashion), *pro-activeness* (i.e. the ability to exhibit goal-directed behaviour by taking the initiative), and *social ability* (i.e. the ability to interact with other agents). This is what Wooldridge and Jennings called the weak notion of agency [7].

Agents can be useful as stand-alone entities that are delegated particular tasks on behalf of a user. However, in the majority of cases agents exist in environments that contain other agents, constituting MAS. A MAS can be seen as a system consisting of a group of agents that can potentially interact with each other [8]. MASs present several advantages over isolated agents, such as reliability and robustness, modularity and scalability, adaptivity, concurrency and parallelism and dynamism [9].

2.1.2. Semantic Web technologies

The Semantic Web aims at adding semantics to the data published on the Web, so that machines are able to process these data in a similar way a human can do [3]. For this, ontologies are the backbone technology. In this work, we have adopted the following definition of ontology: "an ontology is a formal and explicit specification of a shared conceptualisation" [10]. In this context, formal refers to the need of machine-understandable ontologies, which eventually enable automatic reasoning. In this work we use OWL (Web Ontology Language) [11], the ontology language recommended by the World Wide Web Consortium (W3C).

Traditionally, the Web has been conceived as a distributed source of information. The emergence of Web Service Technology has permitted to extend it to a distributed source of functionality. Web Services make applications available on the Web in a standard way so that they can be accessed regardless of the operating system they are deployed on and the programming language they are implemented in. However, as the Web grows in size and diversity, there is an increased need to automate aspects of Web Services such as discovery, execution, selection, composition (that comprises both choreography, which concerns the interactions of services with their users, and orchestration, which defines the sequence and conditions in which one Web Service invokes other Web Services in order to realise some useful function) and interoperation. The problem is that current technology around UDDI, WSDL and SOAP provide limited support for all that [5].

The joint application of Semantic Web and Web Services in order to create intelligent Web Services is referred to as Semantic Web Services. Semantic Web Services consist of describing Web Services with semantic content so that service discovery, composition and invocation can be done automatically by, for example, the use of intelligent agents able to process the semantic information provided. The W3C is currently examining various approaches with the purpose of reaching a standard for the Semantic Web Services technology: OWL-S[12], WSMO[13], SWSF[14], WSDL-S[15], and the recently proposed as W3C recommendation, SAWSDL[16]. The first three approaches propose an ontology that semantically describes all relevant aspects of Web Services. On the other hand, WSDL-S and SAWSDL identify some WSDL and XML Schema extension attributes that support the semantic description of WSDL components.

2.2. Bioinformatics

Bioinformatics involves the application of information technologies to solve biological problems at molecular level, so providing for tools and resources needed to favour biomedical research. Bioinformatics has become essential in the emerging "omics" disciplines such as Genomics, Proteomics or Transcriptomics to facilitate the understanding of the gene structure, interactions, regulation, expression, etc. With all, bioinformatics can be said to be an interdisciplinary area of research that serves as an interface among several disciplines, including Biology, Information Technologies and Medicine. Bioinformatics permits integrating data and information contained in these different areas. It also facilitates the discovery of new knowledge as well as the development of global perspectives from which unifying principles in several areas can be derived. Thus, the collaboration between the areas of medicine and biology makes bioinformatics the integrating tool and the technological support for these areas [17].

The use of bioinformatics techniques has produced a massive explosion in the amount of available biological information and how this information can be exploited (i.e. approaching molecular populations as "wholes"). However, data are distributed into heterogeneous biological data sources, with little or even none semantic structure, so that both the organisation and integration of such data have turned out to be strategic and crucial. This has become more important since computational methods' performance for (biological) data analysis depends on how data are organised; this target can be said to be even critical for data interpretation tasks. In fact, integrated exploitation of existing biological data would be extremely beneficial for research purposes. This need for information integration mechanisms is not only a challenging task in bioinformatics, but in every domain.

In the last years, different approaches for integrating biological information resources have been proposed. On the one hand, tools such as Entrez (www.ncbi.nlm.nih.gov/Entrez/) do not provide for an intelligent integrated access to the information but a friendly navigational mechanism to explore the contents of interrelated biological databases. When querying the Cross-Database Search (http://www.ncbi.nlm.nih.gov/sites/gquery), the query is forwarded to the databases regardless their suitability for answering the query or the database structure and content.

When facing information integration, the following issues have to be addressed: (1) identification of common objects and concepts; (2) integration of data represented in different formats; (3) resolution of conflicts among resources; (4) data synchronisation; and (5) unified visualisation. Recent integration attempts combine data warehouses technologies, use of indexed flat files, relational database management systems, individual searches on resources, wrappers associated to resources. In the last years, different approaches for integrating biological data and services can be found: federated systems, ontology-based mediation systems, data warehouses or workflows [18]. These attempts range from using an individual model or parser for each resource to representing available resources in a common format. Examples of such approaches are Biomart [19], the efforts in Neuroscience [20], or for bridging between phenotype and genotype in [21], for service reconciliation such as the AutoMed system [22].

However, the use of heterogeneous schemas for data storage, that are designed primarily to ensure optimisation of storage space, makes it extremely difficult for the users to query data sources in an integrated manner. More recently, semantic services-oriented solutions have been provided to facilitate integrated access to bioinformatics resources. An example is BioMoby [23], which defines an ontology-based messaging standard for automatic discovery and interaction with task-appropriate biological data and analytical service providers. In this approach, manual manipulation of data formats is not required because data flow from one provider to the next. The Semantic Web provides for a common framework that enables data integration, sharing and reuse from multiple sources. The heterogeneity of biological information resources has led to the increasing use of ontologies, which may help then to represent what is known about a particular domain in an explicit and complete manner. Semantic Web Services play then a vital role to allow resources to be accessed in an integrated way, since each resource defines its services by using a common ontology that can be understood and processed by querying systems. In addition to this, humans can get rid of the task of searching for the available services in order to obtain relevant information by delegating this task to intelligent agents.

3. Biological information and knowledge resources

There is currently a great quantity of independent and heterogeneous data resources, accessible through the Internet, covering for instance genome information and other areas of molecular biology and medicine. Most of this information is stored in databases, although in the last years some ontologies have been developed to harmonise the terminology used in the different databases and to promote integrated access and interoperability among different information resources and applications. In this section, biological databases and ontologies will be discussed, paying special attention to the ones that will be used in this work.

3.1. Biological databases

According to the 2007 update of the Molecular Database Collection [1], there are currently about 1000 biological databases. Biological databases can be classified based on different properties, such as the type of information they contain (information about proteins or genes), the origin of the information (primary databases—experimental results— and secondary databases—results obtained from analysing primary databases—), or the way information is structured (flat files or databases). Traditionally, biological databases were represented by using indexed flat files, which provided fast access to particular records. Currently, the interest is not only the retrieval of individual records but the combination of information from different records, tables and databases. For such purpose, relational databases are more effective and most biological databases have currently relational support. However, many of them still offer their content for downloading in structured flat or XML files.

Usually, biomedical institutions possess a catalogue of different databases. This is the case of the National Center for Biotechnology Information (NCBI, http://www.ncbi.nlm.nih.gov/) or the European Bioinformatics Institute (EBI, http://www.ebi.ac.uk/). In this work, the collection of databases offered by the NCBI has been used. This collection can be uniformly queried by making use of the Entrez interface (www.ncbi.nlm.nih.gov/Entrez/). However, this unique interface forwards the query to each individual database and shows the results for each database. Amongst the databases offered by Entrez, the Gene and Protein ones have been used in this work. The Protein database contains sequence data from the translated coding regions from DNA sequences in GenBank, EMBL and DDBJ as well as protein sequences submitted to Protein Information Resource (PIR, http://pir.georgetown.edu/), SWISS-PROT (http://www.expasy.org/sprot/), Protein Research Foundation (PRF, http://www.proteinresearch.net/), UniProt (http://www.uniprot.org/) and Protein Data Bank (PDB, http://www.pdb.org/) (sequences from solved structures). On the other hand, the Gene database provides a unified query environment for genes defined by sequence and/or in NCBI's Map Viewer.

3.2. Bio-ontologies

The increasing number of independent and heterogeneous databases produced high inefficiency when research groups wanted to search for information across different databases. So, finding relevant information was rather difficult due to the terminological diversity. Given this situation, different entities that had developed their own databases worked together to define a common vocabulary for modelling how gene products might be annotated. This was the origin of the Gene Ontology (GO) [24], which offers information about the attributes of gene products. GO provides a way to resolve the semantic heterogeneity associated to the annotations of genetic products in diverse databases: annotations of different databases are linked to the same GO term. The main GO components are the terms and their relations. Each term has a unique identifier apart from the name of the term. GO is divided into three independent ontologies: molecular functions, biological processes and cellular components. Molecular functions describe the basic molecular role of gene products. Each biological process is comprised of different molecular functions and describes its role at conceptual level. The cellular component ontology represents the structure of the eukaryotic cell. Two types of relations exist in this ontology: *IsA*, which describes that a term is a subclass of a higher class, and *part of*, that means that a child is a subprocess or a structural component of its parent. GO can be explored through several tools, the web interface AmiGO being the most relevant browser. Versions of these three ontologies are available in ontological languages like OWL and OBO, apart from being exportable into MySQL. In this work, information about proteins and genes that can be retrieved from the Gene Ontology annotations has been used.

In the last years, the creation of methods for defining and maintaining shared domain models within biology was considered to be a critical issue [2]. The design, development and use of ontologies in bioinformatics is rapidly increasing and turning more and more significant (see for instance [25]). Currently, ontologies have proven to be successful in biology for tasks such as information access. In this context, the Open Biomedical Ontology (OBO) project (http://obo.sourceforge.net/) was proposed as an attempt to have a portal containing ontologies as well as links to controlled vocabularies for shared use between medical and biological domains. The ontologies found in OBO are partially overlapped (effort is

being conducted to obtain a non-overlapping subset of OBO); they can be combined between themselves adding relations and give rise to new ontologies. Researchers in the OBO project have also developed the OBO language for representing biomedical ontologies. However, the language recommended by the W3C to exchange semantics on the Web is OWL, which provides better support for management, reuse, sharing and exploitation of semantics-rich contents. In fact, there are currently some efforts to express OBO contents in OWL.

It is clear then that bioinformatics, and many other research fields, must face two major challenges: data heterogeneity and information distribution. A further problem in these environments is that of dynamism; new information and services can be made available at any time whereas already existing ones might disappear. In this work, we propose a technological solution to these problems that combines Web Services, Intelligent Agents and Semantic Web concepts into an integrated platform. Data heterogeneity is partially managed using ontologies as the knowledge representation methodology. Web Services tackle information distribution by providing a high degree of interoperability across platforms and operating systems. Finally, agent systems, due to their properties of autonomy, reactivity and pro-activeness, are able to deal with data and service dynamism, and help users avoid the burden of performing service discovery, composition and invocation.

4. A framework for Intelligent Agents and Semantic Web Services integration

Despite their ability to provide a high degree of interoperability across platforms and operating systems, Semantic Web Services infrastructures possess neither enough degree of autonomy nor ability to automatically adapt to changing situations. In these settings, agents can contribute to make systems more autonomous and dynamic, thus maximising their perceived utility. In line with this, the framework presented here comprises both Intelligent Agents and Semantic Web Services working cooperatively in the same environment. Ontologies act as the facilitating technology thus making it possible to exploit the advantages these technologies offer separately.

This solution stems from a basic underlying hypothesis: Intelligent Agents and (Semantic) Web Services were conceived for quite different purposes, and so, they must lie on different abstraction layers. The main idea behind Agent Technology was not for Intelligent Agents to be able to provide services, but to act as autonomous entities that incorporate intelligence and cognitive capacities, which allow them to show a goal-oriented pro-active behaviour and to establish, either competitive or cooperative, interaction processes with other software entities in order to satisfy their design objectives. On the other hand, Web Services involved a further evolution in the Distributed Computing field and their only purpose is to provide worldwide-accessible functionality. These conceptual differences between Agent Technology and (Semantic) Web Services Technologies lead to the need to have both technologies working in an integrated environment and glimpse the advantages of their combination in the development of complex systems.

Next, the foundations of the referred framework are presented and the main elements of the architecture enumerated.

4.1. Ontology-centred view

Ontologies are the paramount technology in our approach as they operate as the 'glue' that binds the remaining components of the architecture [see Fig. 1].

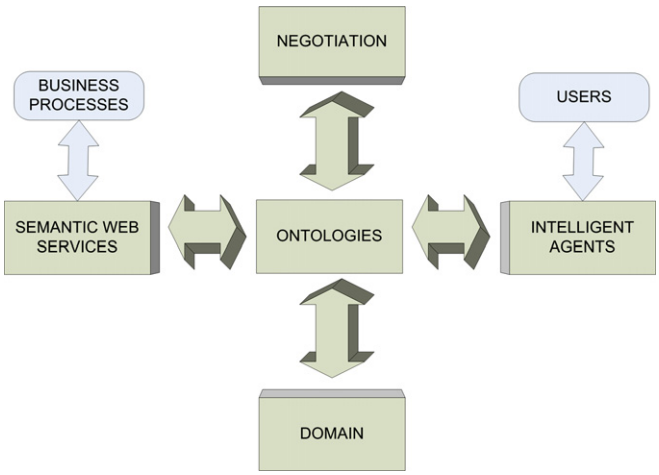

Fig. 1. Ontology-centered approach.

First of all, ontologies function as universal vocabularies so that Web Services and agents share the same interpretation of the terms contained in the messages that they exchange.

From the Web Services point of view, ontologies are useful to semantically describe Web Services' capabilities and processes. This semantic description can then be automatically processed by software entities, so that Web Service discovery, composition, selection, execution and monitoring can be done without human intervention. On the other hand, from the perspective of the agents, the local domain-related knowledge of each agent may be extracted from, or built upon, the application domain ontology. Moreover, inter-agent communication may be carried out by means of a common vocabulary based on an agreed ontology.

Finally, in a similar way to that presented in [26], the negotiation processes between agents may take place in accordance with protocols represented in an ontology and by using the related strategies also stored in an ontology. It is at run-time, and depending on the problem they face, that agents with the desire to cooperate agree upon the protocol and strategy they are going to use.

Summing up, in the architecture proposed here, there is a need for a number of different and disparate ontologies: domain ontology, application ontology, Web Services ontologies, agents' local knowledge ontologies, negotiation protocol ontology and negotiation strategy ontology. Thereby, it becomes of utmost importance to devise a means for ontology reconciliation and mediation. For this purpose, a brokering mechanism has been modelled that is carried out by agents, which take over all the mediation issues (not only data but also process and functional mediation when needed). A more detailed description of the ontology mediation solution is presented in further sections.

4.2. Multi-layered model

The framework presented here is based on a multi-tier architecture that is composed of four different layers [see Fig. 2], namely, Business Logic Layer, Semantic Web Services Layer, Intelligent Agents Layer and Application Layer.

The lower layer, namely, the *Business Logic Layer* provides the most specific operations. It comprises the internal and private business processes that constitute the companies' *engine*. Thus, a set of web services have been implemented that account for some elements of these business processes. Those public services along with the semantic description of their capabilities lay on the second layer, namely the *Semantic Web Services Layer*. Adding semantic annotations to Web Services capabilities enables software entities to automatically interact with those services in a dynamic

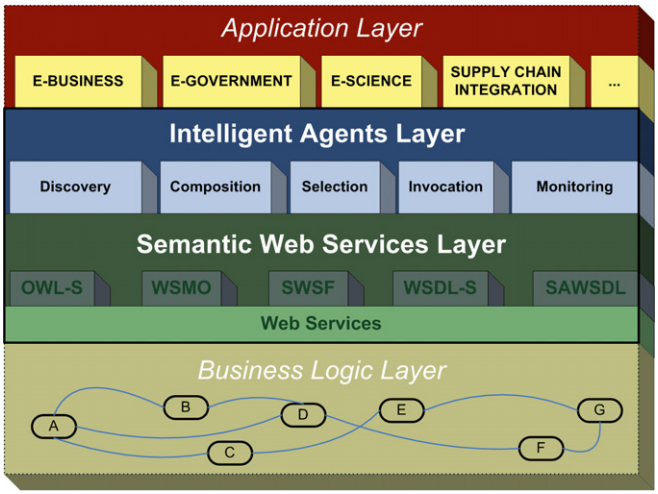

Fig. 2. Four level model.

way and without human intervention. In particular, new services can emerge and others may change their functionality or even disappear at run-time, but the system would keep on working and the changes would be reflected on the application instantly. These sophisticated software entities, namely Intelligent Agents, that interact with and take advantage of the offered services are located in the *Intelligent Agents Layer*. Intelligent Agents make use of the semantic annotation of services capabilities to automatically discover, compose, invoke and monitor Web Services. They are also able to exhibit and dynamically propagate the changing functionality provided in the lower layers. Finally, the *Application Layer* is responsible for organising (orchestrate and coordinate) agents to actually perform useful activities for the users. In this way, depending on the agents available in the system and the way they interoperate different user-tailored applications can be obtained.

4.3. The SEMMAS framework

The SEMMAS (*SEMantic web services and Multi-Agent System*, http://www.semmas.com) framework comprises two of the aforementioned layers, the Intelligent Agents and the Semantic Web Services layers. As a result, the framework becomes independent from both the domain and the actual application in which it is to be applied. In order to develop a specific application for a concrete domain, programmers only need to set the appropriate domain ontologies and decide on what agents to instantiate and which services to access. Thus, the framework can be considered as a reference architecture for several business scenarios and complex, dynamic and open environments such as eGovernment, eScience, eBusiness or Supply Chain Management.

4.3.1. The SEMMAS architecture

The components that constitute the framework are shown in [Fig. 3]. The system architecture is composed of three main elements: a set of intelligent agents that constitute a MAS, four ontology repositories, and three different interfaces for interacting with the external actors that have been identified, namely, service providers, service requesters and software developers.

In the platform proposed to run the system, there are seven types of agents. Agents are grouped in three main categories: agents that act on behalf of service owners (‘*Provider Agent*’ and ‘*Service Agent*’), agents that act on behalf of service consumers (‘*Customer Agent*’, ‘*Discovery Agent*’, and ‘*Selection Agent*’), and agents that perform management tasks (‘*Framework Agent*’ and ‘*Broker Agent*’). Those acting on behalf of service owners manage the access to services and ensure that the contracts are fulfilled. On the other side, the agents that act on behalf of service consumers have to locate services, agree on contracts, and receive and present results. Management agents have a 2-fold purpose: to avoid system resources becoming overloaded and to monitor the status of all the interactions. This should not be understood as a centralised control mechanism that hampers the decentralised vision of MAS. Instead, our aim was to enable the system to both manage unexpected errors and improve its performance. Thus, the ‘*Framework Agent*’ does not control the activity of the remaining agents, but makes sure no errors block the functioning of the system. A description of the seven types of agents present in the framework is given in [Table 1].

In the table above, only abstract, high-level descriptions of the tasks that each agent must perform are given. SEMMAS has been designed so that there is not a fixed way each task should be imple-

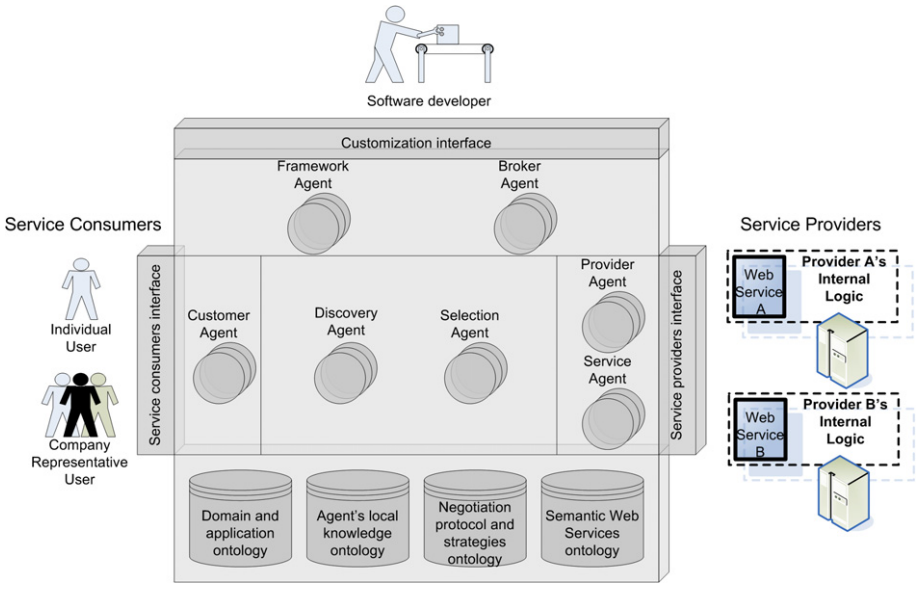

Fig. 3. The SEMMAS framework.

Table 1
Description of the agents in the platform

Agent type	Description
Service owners agents	
Provider Agent	It acts as a service provider representative. The entities set their preferences regarding service execution and these are taken into account during the negotiation process with the service consumers
Service Agent	It acts as a service representative. The service provider establishes a concrete set of preferences regarding a particular service and these are taken into account when negotiating with service consumers (in this case, the Selection Agent)
Service consumers agents	
Customer Agent	It acts as a user representative. First, (individual or collective) users indicate their preferences and specify the goal to be achieved. Then, the goal is carried out and the results given back to the user. The intermediate process happens transparently to the user
Discovery Agent	It is in charge of searching in the Semantic Web Services repository for the service or set of services (i.e. composition) that satisfy the requisites established by the users
Selection Agent	It is in charge of selecting the most appropriate (single or compound) service from the set of services found by the discoverer according to the users' preferences. A negotiation process with the different service representatives (i.e. Service Agent) is carried out for that purpose
Framework management agents	
Broker Agent	It is responsible for solving interoperability issues. Three different levels are considered: data mediation, process mediation and functional interoperability
Framework Agent	It is responsible for monitoring and ensuring a correct functioning of the platform. This type of agent also controls and balances the workload

mented. In fact, it is at run-time (not compile-time) when it is determined which implementation is actually used. For this purpose, the '*role*' concept is introduced. Roles are encapsulations of dynamic behaviour and properties that can be played by agents [27]. We distinguish between roles dealing with service-related issues from those related to the framework management [see Table 2]:

At run-time each agent decides on what roles it *wants* to play depending on the goals the agent pursues. Nevertheless, some of the roles are mandatory for some agents depending on the type of agent. Both factors, the actual implementation of these roles and the agent election of the roles to take, eventually determine the agent behaviour.

Another important constituent element of the architecture proposed here is that formed by a set of ontologies. In order for intelligent agents to successfully carry out their assigned tasks, those ones must have access to various data repositories containing the knowledge that is necessary to fulfil the assignment. These repositories can be either local or external to the system. As data repositories, four kinds of ontologies have been included within the platform (see [Fig. 3]):

1. *Application and domain ontology*: The application ontology contains the knowledge entities (i.e. concepts, attributes, relationships, and axioms) that model the application in which the framework is to be employed. On the other hand, the domain ontology, represents a conceptualisation of the specific domain the framework is going to be applied in. This ontology supports the communication among the components in the framework without misinterpretations.
2. *Agent local knowledge ontology*: It contains, for each agent, the knowledge about the environment it possesses. This ontology generally includes knowledge about the assigned tasks, as well as the mechanisms and resources available to achieve those tasks.
3. *Negotiation ontology*: It comprises both negotiation protocols and negotiation strategies that constitute the negotiation mechanisms agents must use to coordinate their interactions. With this ontology, agents can choose the best mechanism to use for coordinating their actions, which highly depends on the problem under question and the application domain.
4. *Semantic Web Services ontologies*: In this repository (that can be comprised of various ontologies distributed all over the Internet) the ontologies that contain the semantic description of Web Services are stored. The framework does not impose any restriction in terms of the kind of SWS specification (i.e. OWL-S, WSMO, SWSF, WSDL-S or SAWSDL) to be used.

Finally, three different interfaces have been included within the framework architecture. They aim at enabling the interaction with the actors that are external with respect to the framework: service consumers, service providers and software developers. Software developers can, by means of their interface, customise the application by setting up the specific ontologies to be used. They also have to instantiate and configure the core agents necessary for the proper functioning of the system (customer, provider and service agents will be launched as needed during run-time). Once the application has been properly set up, both service consumers and service providers can register in the system and use it as a meeting

Table 2
Description of agent roles

Role	Description
Service-related roles	
Broker role	It represents the functionality needed for solving all kind of interoperability problems (data, process and functional mediation)
Composer role	It allows the achievement of a goal by means of several composed services
Invoker role	It invokes a Web Service once the operation to be executed and the parameters are known
Matchmaker role	It finds the services whose semantic descriptions match the goal that was sent by the user
Monitor role	It ensures that the contracts established for the execution of the services by both service owners and service consumers are fulfilled
Ontology manager role	It includes functionality associated with the access and processing of ontologies
Selector role	It provides the functionality necessary for the selection of a service from a list of services according to a set of preferences
Framework management roles	
Negotiator role	It enacts the actual negotiation process between the parties establishing the basis for the system execution
Platform manager role	It controls and balances the system workload
Provider representative role	It interacts with service providers. At a high level of abstraction, it must be able to enforce the conditions present in the company's business strategy
Service representative role	It acts on behalf of services, participating in negotiations and improving the services offered when possible
Consumer representative role	It interacts with service consumers by, firstly, determining their wishes and, then, returning the expected results
Global monitor role	It monitors the events in the application, detects possible problems and defines the actions to take in case of error

point. Through their interface, service providers can modify the list of services they provide and set the conditions under which a service they provide must be executed. Service consumers, on the other hand, can, by means of their interface, query the system and trigger the execution of one or several Web Services in order to fulfil a particular goal.

4.3.2. The brokering process

Interoperability is a major problem when trying to join together systems or components that have not been designed to interoperate. Three different levels of interoperability issues must be addressed [28]. Firstly, when a system deals with multiple disparate ontologies, *data mediation* is needed. Secondly, a *process mediation* solution is required when the outputs of a component differ from the inputs of the next. Finally, as the requested and provided functionalities do not likely match precisely, *functional mediation* becomes necessary. The framework presented here copes with all these issues.

4.3.2.1. Data mediation. It is the broker role along with the broker agent what provide a complete solution for data interoperability issues. Each agent in the platform can take over the broker role as desired. This would serve as a way to solve some of the misunderstandings the agent faces when communicating with other agents. In case the local broker role of an agent cannot do the job, the broker agents come into the stage. The problem is delegated to the broker agents, which have better means for solving it. It is left to the programmer to decide what techniques to use (e.g. ontology mapping, ontology alignment, ontology merging and so on) to deal with data interoperability.

4.3.2.2. Process mediation. Given that all the communication between service providers and service consumers flows through agents, process interoperability is automatically tackled. Certainly, Intelligent Agents communicate asynchronously by message passing according to a negotiation protocol and a strategy previously agreed upon thus avoiding process interoperability problems.

4.3.2.3. Functional mediation. Both composer and matchmaker roles are responsible for finding the appropriate services that are necessary to execute in order to satisfy a user request. It depends on the implementation of these two roles how the process takes place. The action flow can be as follows: (1) the composer sends the complete description of the user request to the matchmaker; (2) the matchmaker searches for services that exactly match the request; (3) if the matchmaker finds at least one appropriate service, then it notifies the composer and the process is over; otherwise, the composer is informed that no single service exists satisfying the request; (4) the composer uses a planner to split up the user request into fine-granulated parts and sends these "*sub-goals*" to the matchmaker; (5) the matchmaker searches for services that exactly matches the requirements of each sub-goal; (6) if the matchmaker finds at least one appropriate service for each sub-goal, then it notifies the composer and the process is over; otherwise, the remaining unsolved sub-goals are sent back to the composer, which further divides them into smaller pieces that are then matched again against the available services and so on.

4.4. Discussion

With the aim of tackling data heterogeneity and information distribution problems of complex systems, the proposed platform seamlessly integrates three main ingredients: Ontology, Web Service and Intelligent Agent technologies. We pointed out above several disadvantages derived from the use of Intelligent Agent and Semantic Web Service technologies separately. Summarising, MASs are not able to properly communicate across firewalls over the Internet while Semantic Web Services consumption can be dramatically improved through autonomous software entities able to exploit the semantic descriptions to make use of the services. The framework presented here overcomes these problems by putting them together and using ontologies to enable a seamless interaction. Agents are kept in the same environment (the same platform or in different 'controlled' platforms) so the communication problem is avoided. On the other hand, Web Services capabilities are semantically annotated and agents have no restrictions on how to use them.

The cornerstone technologies of SEMMAS are ontologies. Ontologies lay on the background of every communication process that takes place between all the elements of the architecture, in such a way that they all share the same interpretation of the terms contained in the messages that they exchange. For this to happen, the available ontologies should be fully exploited so making it necessary the use of the existing techniques and tools for ontology management and consumption from the Semantic Web field.

With all, the SEMMAS framework successfully exploits features such as autonomy, dynamism, and pro-activeness of agent systems, data interoperability provided by ontologies and the Semantic Web vision, and platform interoperability supported by Web Services. In the context of SEMMAS, autonomy and pro-activeness are related to the fact that once an agent has been assigned a goal, it tries to fulfil it without human intervention. Dynamism is achieved by the ability of the agent system to automatically adapt to changing situations. In particular, the behaviour SEMMAS shows and the way it responds to the requests depend on the services available in the environment at any time. The use of semantics enables the framework to deal with data heterogeneity. Finally, SEMMAS reach machine to machine interoperation over the Internet by applying Web Service technologies. The combination of these properties makes SEMMAS appropriate for the development of powerful and flexible distributed systems in complex, dynamic, heterogeneous, unpredictable and open environments.

At this point, it is worthy to point out that it was not the purpose of this research to find new solutions for automatic discovery, matchmaking, composition, selection or invocation of Web Services, but to design an environment where these can be tested and utilised in real-world settings.

5. Applying SEMMAS in the biological domain

SEMMAS has been successfully applied to integrate several heterogeneous biological information resources, thus providing users with a user-friendly, single interface through which they can have access to all these resources[1]. The major benefit of applying a MAS-based approach such as SEMMAS in this domain is the support for dynamism. Thus, in contrast to traditional workflow-based approaches, the system is able to make use of the new data and operational services that can emerge in an open and distributed environment such as the Internet. The methodology to customise the framework to the requirements and particularities of the biological domain is as follows: (1) design of a domain ontology, which contains the concepts and relationships we are interested in; (2) development of a Web Service for each resource we aim to access; (3) semantic annotation of the services by making use of the domain ontology; (4) instantiation of a set of agents in SEMMAS to retrieve biological information.

In this section, we first show some insights on the domain ontology that has been used. Then, details about the Semantic

[1] See http://www.semmas.com/prototypes/bioinformatics.html

Web Services that have been implemented are put forward. Finally, the use of the platform in the biological domain is presented.

5.1. The biological domain: oncogenes

In this work, we are interested in accessing information available about oncogenes. An oncogene is a modified gene, or a set of nucleotides that codes for a protein, that increases the malignancy of a tumour cell. Some oncogenes, usually involved in early stages of cancer development, increase the chance that a normal cell develops into a tumour cell, possibly resulting in cancer. There are several approaches for classifying oncogenes, but none is considered standard. For our purposes, the Oncogene ontology[2] was designed and developed. This ontology was developed because no formal, standard representation for both the biological and medical aspects of oncogenes was found in the literature.

The model used to represent knowledge of oncogenes, presented in [29], is based on the representation of multiple and hierarchical restricted. Both taxonomic and merologic relations are covered by this model. Taxonomies classify the knowledge domain, and mereologies indicate the relation between parts and wholes. The set of relations in the model defines a series of axioms that result from the relations: "class has attribute", "is a class of", "is a part of", or "is connected to", among others. Some examples of these relations are "the oncogene is a mutation" and "the DNA is transcribed to mRNA".

In order to share this ontology with the community, it has been represented using OWL (DL variant) since it is the W3C recommendation for exchanging semantic content on the web. A partial view of this ontology is shown in [Fig. 4]. This ontology is mainly a taxonomy, and each class is defined through the following elements: its name, its synonyms, its properties inherited from the taxonomic parent classes, its specific properties, its original data source and its links to external sources. By original data source, we refer to the origin of the information: a database, a lab name, a researcher, etc., whereas external sources link to other databases containing information about this class. The inclusion of a terminological relation such as synonymy is positive, since the ontology will be used to guide natural language-driven inputs. As it can be drawn from [Fig. 4], the higher level classes are cancer, carcinogenesis, cell, cellular cycle, chromosome, gene, mutation and protein. This ontology contains 203 classes, 10 occurrences of the mereological relation, 55 attributes, 6 inverse properties, 202 occurrences of the taxonomic relation and 41 instances of Oncogenes.

5.2. Designing Semantic Web Services for biological resources

As it was aforementioned, we have developed several Web Services in order for the system to be able to access different biological information resources. These services have been implemented in Java and the Apache Axis2 library (http://ws.apache.org/axis2/) has been used. The Web Services were then semantically annotated by making use of the OWL-S ontology and the OWL-S Editor (http://owlseditor.semwebcentral.org/). At this point, it is important to highlight that the SEMMAS platform is not constrained to any particular Semantic Web Services approach. Next, one of the services that have been implemented is briefly commented.

5.2.1. Gene ontology Semantic Web Service

The Gene Ontology Semantic Web Service (GOSWS) uses the Gene Ontology Database view instead of the ontological one. The GOSWS is defined on top of the Gene Ontology Web Service (GOWS), which is available at http://klt.inf.um.es:8080/axis2/services/GO_WebService. The GOWS contains three methods that expose part of the functionality available in the Gene Ontology Web Portal (http://www.geneontology.org/):

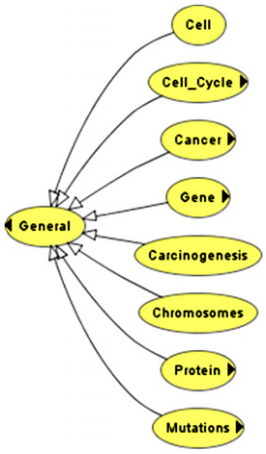

Fig. 4. Oncogene upper-level hierarchy.

- Get information about the term "*term*" (Get_information):
  - http://amigo.geneontology.org/cgi-bin/amigo/go.cgi?action=query& view=query& session_id =864b1175276845&query=***term***&search_constraint=terms
- Get information about the gene "*geneName*" (Get_Gene):
  - http://amigo.geneontology.org/cgi-bin/amigo/go.cgi?action=query& view=query=***geneName***&search_constraint=gp
- Get information about the protein "*proteinName*" (Get_Protein):
  - http://amigo.geneontology.org/cgi-bin/amigo/go.cgi?action=query&view=query&query=***proteinName***&search_constraint=gp

These three GOWS methods return the protein or gene whose identifier has been entered or the identifier of a biological element. Once the GOWS was available, a corresponding WSDL file was developed [see Fig. 5] that syntactically describes the aforementioned operations.

Next, the GOSWS was defined. For this, we used the OWL-S Editor Protégé plugin along with the Oncogene ontology (i.e. the domain ontology). For each accessible method in the service, an OWL file was created containing the OWL-S semantic description of the operation. A complete example is shown in the figures below [Figs. 6–9].

In the figures above, the way input and output parameters are bound to the domain ontology is highlighted. By having access to

```
<wsdl:operation name="getProtein">
  <wsdl:input message="axis2:getProteinMessage" wsaw:Action="urn:getProtein"/>
  <wsdl:output message="axis2:getProteinResponse"/>
</wsdl:operation>
<wsdl:operation name="getInformation">
  <wsdl:input message="axis2:getInformationMessage" wsaw:Action="urn:getInformation"/>
  <wsdl:output message="axis2:getInformationResponse"/>
</wsdl:operation>
<wsdl:operation name="getGene">
  <wsdl:input message="axis2:getGeneMessage" wsaw:Action="urn:getGene"/>
  <wsdl:output message="axis2:getGeneResponse"/>
</wsdl:operation>
```

Methods available in the Web Service

Fig. 5. GOWS WSDL file.

[2] See http://www.semmas.com/prototypes/resources/ontologies/OGEN.owl

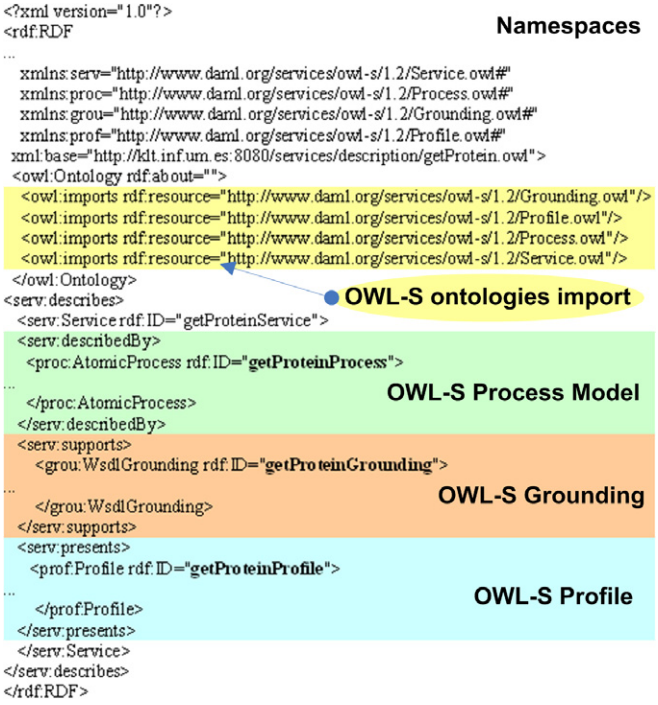

Fig. 6. Extract of getProtein.owl.

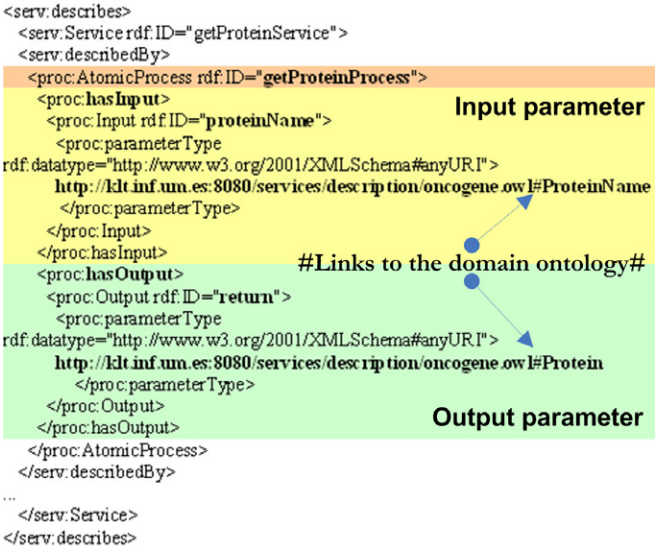

Fig. 7. getProtein.owl—process model.

both the Oncogene ontology and these OWL-S files, every service requester can make use of the service unambiguously.

The Semantic Web Services allow for intelligent agents to interact with biological information resources semantically. For the purposes of this work, it was us who implemented the services and provided the semantic description. Ideally, it would be those who actually provide the service (i.e. service providers) who would generate these.

The implemented services offer a very similar functionality. In these situations, the consumer and provider agents have to negotiate in order to decide which service is to be executed.

5.3. Retrieving biological information with SEMMAS: use case scenario

Let us suppose that a user (hereafter Alice) of the SEMMAS platform wants to obtain the description of the protein with name

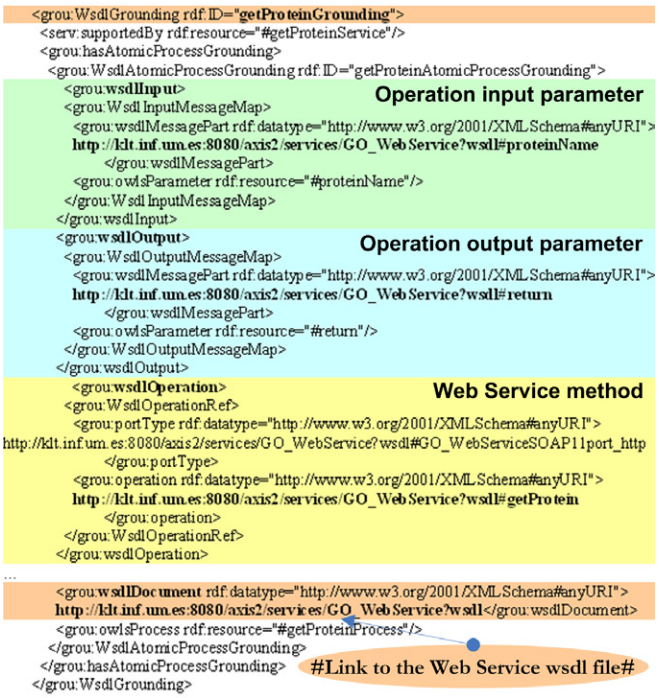

Fig. 8. getProtein.owl—grounding.

Fig. 9. getProtein.owl—Profile.

'*trk2*'. In such a scenario [see Fig. 10], Alice would write a sentence in natural language that might look as follows: "*I want to get the*

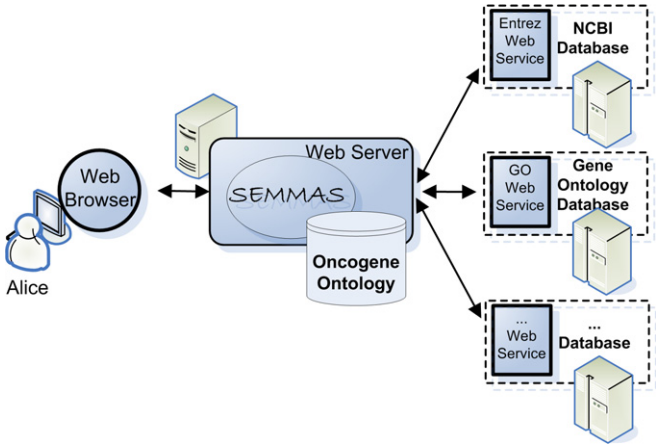

Fig. 10. SEMMAS—use case scenario.

description of the protein with name trk2". Alice would then wait for her personal agent to process the query and return the expected answer.

The sequence of steps SEMMAS takes once the query has been stated is described next:

1. Alice's personal agent (Customer Agent) analyses her query by using KAText [30], a natural language processing (NLP) tool that is able to transform the sentence into a lightweight ontology that formally represents Alice's requisites. For this, the tool was previously trained in order to align the resultant ontology with the domain ontology (i.e. Oncogene). The lightweight ontology that results from the analysis of the input query is shown in the next figure [Fig. 11]
2. Alice's personal agent sends the query, now in the form of an ontology, to one of the instances of the Discovery Agent present in the platform. The content of the message is fully understood by the receiving agent because it refers to the common domain ontology. This holds for every agent interaction during the processing of the user query.
3. The Discovery Agent searches for the services that would satisfy Alice's request in the repository of semantic descriptions. In the example under question, the process is very simple since service composition is not required (single, atomic services exist that fulfil the goal). Besides, since both the goal and the semantic description of the services are linked to the same ontology (i.e. Oncogene, the domain ontology), the matching becomes quite straightforward. In a more complex situation, the procedure would be as follows:
   a. First, the agent looks for single services that completely match the query. The discovery process benefits from the fact that both the query and the description of the services are aligned with the Oncogene ontology.
   b. If one or more services are found satisfying the request, the process ends. Otherwise, the agent decomposes the query into several sub-queries and looks for services able to fulfil each of these sub-queries. For this, the system makes use of a planner.
   c. The process goes on until services are found satisfying all the sub-queries.
4. The Discovery Agent returns to Alice's personal agent the collection of sets of services found [see Fig. 12]. Then, the latter sends the list of services to one of the instances of the Selection Agent.
5. The Selection Agent is responsible for finding the best (set of) service(s) satisfying the request. For this, it acts as follows:
   a. Those sets of services that do not fulfil some of the basic requirements Alice has previously indicated are automatically dismissed. This is evaluated on the basis of non-functional properties that might have been introduced in the semantic description of the services.
   b. Then, the Selection Agent asks the Framework Agent to create as many instances of service agents as necessary.
   c. Once a Service Agent has been created for each service in the list of services, the Selection Agent starts a negotiation process by sending a *Call for proposals* to all the Service Agents (the ontology-based negotiation mechanism is yet to be incorporated into the framework implementation).
   d. Each Service Agent makes its proposal according to the preferences of the service provider and the Selection Agent selects the best set of services given their expected utility in accord with the preferences of the user [see Table 3].
6. The Selection Agent returns to Alice's personal agent the sorted list of set of services along with the agreed conditions for the execution of each service.
7. Alice's personal agent shows Alice the ordered list of services and she chooses the one she would prefer to be executed, if any [see Fig. 12].
8. Alice's personal agent interacts with the appropriate Service Agents so that they actually invoke the chosen services. During invocation, the Service Agents have to map user input information with the input parameters of the methods to execute. This task is considerably facilitated by the use of the domain ontology in both the goal representation and the semantic description of the services [see Figs. 7 and 11].
9. The Service Agents return the services results to Alice's personal agent, which is in charge of integrating them into a common data structure that is then sent back to Alice through the Web interface [see Fig. 13].

The simple scenario considered shows the promising benefits SEMMAS, in its current state of implementation, provides to the biomedical community. On the one hand, it acts as an easy-to-use, unique interface to several heterogeneous data sources. End users only have to state a goal or wish in natural language and, then, the system works out a set of services able to fulfil it. Therefore, users with limited computing background avoid the

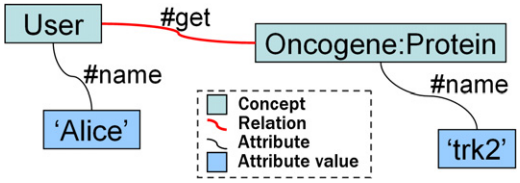

Fig. 11. Resultant ontology.

Table 3
Negotiation parameters and selection

Service	Trust	Delay	Provider proposal	Utility value
Entrez	0.8	0.1	Other parameters hat can be assessed	0.31
Gene Ontology	0.6	0.3		0.39
...	...	...		...
User-defined impact factor	0.3	0.7	...	

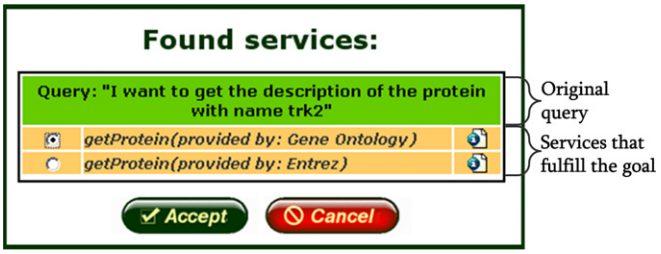

Fig. 12. SEMMAS—service selection.

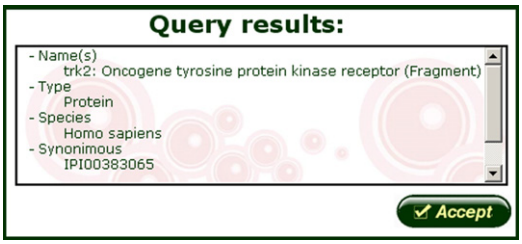

Fig. 13. SEMMAS—query results.

burden of this time-consuming process. On the other hand, the system is aware of the dynamic nature of the underlying environment. Emerging services are dynamically exploited by SEMMAS to respond to users' new enquiries. Besides, this use case serves to describe how SEMMAS carries out tasks such as goal recognition through NLP techniques, and service discovery, composition, selection and invocation. Certainly, once each and every one of the features conceived in the conceptual framework have been implemented, it will be possible to design more complex use case scenarios where semantic heterogeneity is tackled.

6. Conclusion and future work

The concept of bioinformatics refers to the application of information technologies in molecular biology and all the "omics" disciplines. This is a very challenging research field, data heterogeneity and information distribution being two major problems in bioinformatics environments. The application of agents, Web Services and Semantic Web-related technologies has proved to be very useful in this area [23,31,32]. Agent technology promises to enable powerful, flexible and cost-effective distributed systems. However, several problems arose that have prevented Multi-Agent Systems from being applicable to real-world settings. A major flaw of this technology is the use of non-standard proprietary protocols what makes it difficult for agents that have not been designed to work together to interoperate. Web Service technology is the evolution of other solutions in the distributed programming field such as RMI, CORBA or DCOM. Web Services are based on three open Internet standards, namely, UDDI, SOAP, and HTTP. The utilisation of standard protocols enables Web Services to constitute loosely-coupled distributed systems. In fact, a key advantage of this technology is that it allows for dynamic service composition using independent, reusable software components. Semantic Web Services emerged to facilitate the automation of service discovery, composition, monitoring, and invocation. They consist of describing Web Services' capabilities with semantic content, so that autonomous software entities can interact with services without human intervention.

Although it is still subject to great discussion in both the Artificial Intelligence and the Distributed Computing areas, it seems quite straightforward that joining together Agent and Semantic Web Service technologies can lead to the development of new, more powerful applications. Actually, various research projects have been carried out worldwide with the purpose of integrating these two technologies [33,34]. However, the existing approaches suffer from several shortcomings caused mainly by their inability to completely benefit from the advantages of this combination. By considering this fact, in this paper we introduce SEMMAS, an ontology-based framework for Intelligent Agents and Semantic Web Services integration. This framework aims to exploit the striking potential of these technologies while overcoming their deficiencies. For this, we take an ontology-centred approach. Ontologies are the facilitating technology that enables a seamlessly communication between agents and services. Ontologies are present at almost every stage of the interaction process. We use ontologies not only to represent different kinds of knowledge (domain and application knowledge) but also to formally exhibit the services' capabilities. The former is employed by agents to communicate to each other without misunderstandings. The later enables agents to automatically discover, compose and invoke the available services.

The SEMMAS framework consists of: (1) seven different agent types (Broker Agent, Customer Agent, Discovery Agent, Selection Agent, Provider Agent, Framework Agent and Service Agent); (2) four ontology repositories (domain and application ontologies repository, agents' local knowledge ontologies repository, negotiation protocol and strategies ontology repository and Semantic Web Services ontologies repository) that can be either local or remote; and (3) three interfaces (service consumers interface, service providers interface and customisation interface) aimed at enabling the interaction with the three external actors (service consumers, service providers and software developers, respectively). A number of roles have also been conceived that comprise all the system functionality. The roles an agent instance chooses to play will eventually determine the behaviour of the agent at run-time. An agent can dynamically change the roles it plays depending on the goal it has to achieve at any moment.

In this paper, we also present the application of the SEMMAS framework in the biomedical domain. An ontology about oncogenes has been created, implemented and used. Several services have been implemented as well providing access to disparate biomedical information sources. Then, these services were semantically annotated by making use of the Oncogene ontology and the OWL-S approach. With all these ingredients, SEMMAS allows users to exploit those services, thus getting biomedical and genomics information in an integrated manner. Currently, the system is able to provide integrated access to heterogeneous data sources. It is left for future versions the inclusion of data integration solutions.

In conclusion, the main contribution of our work is a fully-fledged framework that gathers together the best features of three leading technologies: a new level of interoperability between software applications through Web Services, data interoperability by using ontologies for knowledge representation, and dynamism provided by the combination of abilities such as autonomy and pro-activeness of Intelligent Agents. The aggregation of these properties represents a clear improvement over the existing approaches and makes this framework appropriate for the development of powerful and flexible distributed systems in complex, dynamic, heterogeneous, unpredictable and open environments. In particular, the implemented platform is capable of dealing with the heterogeneity that characterises current biomedical information sources. Nevertheless, a number of issues still remain unsolved and must be addressed. In its current state, the platform collects data from heterogeneous biological sources by mapping their contents to the domain ontology through the semantic description of the web services. However, no action is taken to control data redundancy, inconsistency, etc. An improved data integration mechanism is necessary to deal with these issues. Also limiting is the chosen approach for service composition. Classical planning requires complete knowledge of the world, assumes the existence of atomic actions with deterministic effects, and produces only sequential workflows. To meet Web Services requirements, other more sophisticated methods must be investigated that could even involve the interaction with users (semi-automatic solutions).

As further work, we plan to evaluate the framework in terms of its performance and usability on other domains (e.g. eCommerce, eGovernment, etc.). In particular, we aim at finding domains with different properties and challenges so that the whole potential of the framework can be analysed. On the other hand, a number of algorithms for Web Service discovery, composition and invocation will be tested and integrated into the implementation. At the same time, the performance and accuracy of various ontology reasoners and inference engines will be examined, and the most effective among them included in the platform. We also plan to study the inclusion of Grid services in the referred framework and its application for pervasive computing. Currently, the platform is prepared for dealing with stateless Web Services. However, agents are inherently stateful and consequently support for stateful services such as Grid services can be easily provided. Besides, current approaches

towards the semantic annotation of Grid services (e.g. [35]) can facilitate the process.

Acknowledgments

We thank the following institutions for their respective financial support: the European Commission under project ALFA-0447-FA; the Spanish Ministry for Industry, Tourism and Commerce under project FIT-340000-2007-212; the Spanish Ministry for Education and Science under projects TSI2007-66575-C02-02, TEC2006-12365-C02-01, TIN2006-15140-C03-02 and TIN2006-14780; the Centre for Public Works Studies and Experiments (CEDEX) from the Ministry of Public Works under project PT-2006-055-24ICPP; and the Murcian Department for Education under projects TIC-INF 06/01-0002 and BIO-TEC 06/01-005.

References

[1] Galperin MY. The molecular biology database collection: 2007 update. Nucleic Acids Res 2007;35:D3–4.
[2] Altman RB, Valencia A, Miyano S, Ranganathan S. Challenges for intelligent systems in biology. IEEE Intell Syst 2001;16(6):14–20.
[3] Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Sci Am 2001:34–43.
[4] Gruber TR. A translation approach to portable ontology specifications. Knowl Acquis 1993;5:199–220.
[5] Fensel D, Bussler C. The web service modeling framework WSMF. Electronic Commerce Res Appl 2002;1(2):113–37.
[6] Wooldridge M, editor. An introduction to multiagent systems. John Wiley & Sons Ltd.; 2002.
[7] Wooldridge M, Jennings NR. Intelligent agents: theory and practice. The Knowl Eng Rev 1995;10(2):115–52.
[8] Vlassis N. A Concise introduction to multiagent systems and distributed artificial intelligence. In: Brachman Ronald J, Dietterich Thomas G, editors. Synthesis lectures on artificial intelligence and machine learning. Morgan and Claypool Publishers; 2007.
[9] Elamy AH. Perspectives in agents-based technology. AgentLink News 2005;18:19–22.
[10] Studer R, Benjamins R, Fensel D. Knowledge engineering: principles and methods. Data Knowl Eng 1998;25(1–2):161–97.
[11] McGuinness DL, van Harmelen F, editors. OWL web ontology language overview. W3C recommendation. Available from: http://www.w3.org/TR/owl-features/ [10 February 2004].
[12] Martin D et al., editors. OWL Web Ontology Language for Services. OWL-S W3C Submission; 2004. Available from: http://www.w3.org/Submission/OWL-S/.
[13] Lausen H, Polleres A, Roman D, editors. Web service modeling ontology. WSMO W3C submission; 2005. Available from: http://www.w3.org/Submission/WSMO/.
[14] Battle S et al. Semantic web service framework. SWSF W3C submission; 2005. Available from: http://www.w3.org/Submission/SWSF/.
[15] Akkiraju R et al. Web service semantics. WSDL-S W3C submission; 2005. Available from: http://www.w3.org/Submission/WSDL-S/.
[16] Farrell J, Lausen H. Semantic annotations for WSDL and XML schema. W3C recommendation. Available from: http://www.w3.org/TR/sawsdl/ [28 August 2007].
[17] Martin-Sanchez F et al. Synergy between medical informatics and bioinformatics: facilitating genomic medicine for future health care. J Biomed Inform 2004;37(1):30–42.
[18] Goble C. The state of the nation in data integration. WWW2007, workshop on health care and data integration for the semantic web; 2007.
[19] Durinck S et al. BioMart and bioconductor: a powerful link between biological databases and microarray data analysis. Bioinformatics 2005;21(16):3439–40.
[20] Chen H et al. A semantic web framework for integrating relational neuroscience databases. ISWC06, workshop on health care and data integration for the semantic web; 2006.
[21] Sahoo SS et al. An experiment in integrating large biomedical knowledge resources with RDF: application to associating genotype and phenotype information. WWW2007, workshop on health care and data integration for the semantic web; 2007.
[22] Zamboulis L, Martin N, Poulovassilis A. Bioinformatics service reconciliation by heterogeneous schema transformation. Workshop on data integration in the life sciences, Philadelphia, USA; 2007.
[23] Wilkinson M et al. BioMOBY successfully integrates distributed heterogenous bioinformatics web services. The PlaNet exemplar case. Plant Physiol 2005;138:1–13.
[24] Ashburner M et al. Gene ontology: tool for the unification of biology. The gene ontology consortium. Nat Genet 2000;25:25–9.
[25] Bodenreider O, Stevens R. Bio-ontologies: current trends and future directions. Brief Bioinform 2006;7(3):256–74.
[26] Tamma V et al. Ontologies for supporting negotiation in e-commerce. Eng Appl Artif Intell 2005;18:223–36.
[27] Zhao L, Mehandjiev N, Macaulay L. Agent roles and patterns for supporting dynamic behavior of web services applications. In: 3rd international conference on autonomous agents and multi-agent systems; 2004.
[28] Stollberg M et al. A semantic web mediation architecture. Canadian semantic web working symposium; 2006..
[29] Fernández-Breis JT et al. Towards cooperative frameworks for modeling and integrating biological processes knowledge. IEEE Trans Nanobiosci 2004;3(3):164–71.
[30] Valencia-García R et al. A methodology for extracting ontological knowledge from Spanish documents. Lect Notes Comput Sci 2006;3878:71–80.
[31] Gómez JM et al. BIRD: biomedical information integration and discovery with semantic web services. Lect Notes Comput Sci 2007;4528:561–70.
[32] Armano G, Manconi A, Vargiu E. A multiagent system for retrieving bioinformatics publications from web sources. IEEE Trans Nanobiosci 2007;6(2):104–9.
[33] Gómez JM et al. GODO: goal oriented discovery for semantic web services. discovery on the WWW Workshop (SDISCO'06), Beijin, China; 2006.
[34] Shafiq O et al. A first step towards enabling interoperability between software agents and semantic web services: multi agent systems adapting web services standards. IBIS 2006;2(2):97–117.
[35] Corcho O et al. An overview of S-OGSA: a reference semantic grid architecture. J Web Semantics 2006;4(2):102–15.