TOWARDS INTELLIGENT DISCOVERY OF ENTERPRISE ARCHITECTURE SERVICES

By Nico Brehm, Tariq Mahmoud, Jorge Marx Gómez, and Ammar Memari

Abstract
In the past few years there has been a major leap in the size of data that exist on the Web. Especially the launch of Web 2.0 websites like Wikipedia.org or YouTube.com had its effect on the presented information complexity as well as its actuality. There has been also a steady rise of B2B transactions enabled by the exchange of data between companies. The actual acceptance of e-commerce applications increased the amount of transactions in the B2C segment as well. As these changes already affected the structures of today’s Enterprise Architectures (EAs), future architectural concepts in this context will have to pick up challenges which are caused by the dynamic availability of a huge amount of services as variable elements of a next generation of enterprise software systems and system landscape. The article focuses on the problem of the dynamic discovery and invocation of appropriate Web Services in EA environments to fulfill this purpose. The proposed model is based on a decentralized network of Web Service providers and uses software agents which are responsible for the intelligent discovery and management of Web Service-Endpoints suitable to the needs of the consuming entity. The model considers preferences of users as dynamic criteria. Furthermore the reputation of service providers is seen as collective knowledge of a user community whereas past experiences of neighbored users influence the autonomous acting of user agents. Thus, relevant Web Services will be semantically described and discovered in a collaborative manner. The proposed architectural model is based on an open network architecture which abandons central elements for the management of service offerings in order to increase the openness of conventional architecture approaches. This article shows what current SWS technology offers in respect to the requirements needed by future EA scenarios and will point out the potentials of current SWS technologies in the EAI and B2B context. Furthermore SWS frameworks will be investigated according to their potential as regards to a dynamic discovery and invocation in the given context.

Keywords
Semantic Web Services, Enterprise Application Integration, Peer-to-Peer networking, multi-agent systems, business integration, service discovery.

INTRODUCTION
Searching for a needle in a hay stack has always been a dilemma. By virtue of this problem we can imagine that searching for specific information in the digital world nowadays is quite difficult. Since the digital market is growing drastically and the vendors are controlling this market, there is a need for new approaches to deal with this complexity. The task is to find this information easily and in a semantic manner.

In this article the approach of an IDEAS component is shown, which employs principles developed in already existing Semantic Web Services architectures, and in addition it involves the approaches of multi-agent systems in a Peer-to-Peer network, to demonstrate their ability in enterprise architectures.

The IDEAS component is aiming to decrease levels of complexity, dependency and monopoly of existing digital markets by distributing activities over a Peer-to-Peer network, in which, every node is an agent that has the
responsibility of providing, discovering, consuming, and binding available services which fit into the enterprise scope.

The major focus in this article is to point out the potentials which IDEAS solution offers for enterprise architectures. This analysis includes some challenges for the IDEAS component to fully enable dynamic discovery in EA. It further concludes that it is already possible in intra-Enterprise Application Integration (intra-EAI) scenarios under certain assumptions; furthermore, we demonstrate the applicability of it inside enterprise architectures to exemplify the activities within the lifecycle of such architecture.

The Future of the Web
Most Web pages will themselves contain hyperlinks to other related pages and perhaps to downloads, source documents, definitions and other Web resources. Such a collection of useful, related resources, interconnected via hypertext links, is what was called a "web" of information. Making it available on the Internet created what Tim Berners-Lee first called the World Wide Web (Berners-Lee, Calliau, 1990). Due to the excessive growth of information available over the Web, a need for a new approach has emerged. Web 2.0 is defined as a trend in the use of the World Wide Web that is meant to increase creativity, information sharing, and collaboration among users (Hult, Ketchen, 2006). So we can state the mentality behind Web 2.0 is about helping users finding the best ways to solve a problem inside the forest of all possible ways, and the continuous optimization of those ways. But according to the inventor of the World Wide Web, Tim Berners-Lee, the term "Web 2.0" is a "piece of jargon", "Nobody really knows what it means", and he stated that the Web was originally to connect people (Berners-Lee, 2006), this leads to the idea that having people connected to each other enables some of them to benefit from the others' experiences, which means basically that humans are the analyzing power that is forging best solutions. This can be considered as a weakness in Web 2.0 since machines have much more processing capabilities than humans, when the knowledge is represented in a correct way to them. The revolution towards a new way for utilizing machines in the daily consumption of information led to the idea of the Semantic Web.

Semantic Web is an evolving extension of the World Wide Web in which the semantics of information and services on the Web are defined, making it possible for the Web to understand and satisfy the requests of people and machines to use the Web content (Berners-Lee, Hendler, Lassila, 2001).

Web Technology as Part of Future Companies
Academia and industry have realized that the Semantic Web can facilitate the integration and interoperability of intra- and inter-business processes and systems, as well as enable the creation of global infrastructures for sharing documents and data, make searching and reusing information easier (Cardoso, Hepp, Lytras, 2007). That encouraged companies to move their entire information infrastructure towards the Web platform, offering a unified and standardized access for customers, suppliers and employees to the information and services the companies offer. Companies tend to an inter-company integration of their information system to achieve strategic alliances. That gives a big importance to a vital factor which is enterprise integration. "Business to Business" (B2B) Integration, referred to as inter-EAI, specifies the automated and event-driven information exchange of various systems between companies (Bussler 2003). This integration can be applied through various aspects in different management systems like Enterprise Resource Planning (ERP), Customer Relationship Management (CRM) and Supply Chain Management (SCM). Currently one of the concepts involved in EAI is Service-Oriented Architecture (SOA), with the advent of the Semantic Web more concepts from its vision will be involved in forming the future version of integration among enterprises. These concepts are (but not limited to): Semantic Service Oriented Architecture (SSOA), Agent and Multi-Agent Systems and Peer-to-Peer-networks.

Multi-agent Systems and Peer-to-Peer Networks
Semantic Web agents are programs that act on behalf of the user. They are not executed or invoked for a specific task; rather they are autonomous and activate themselves. They collect Web contents from different sources, process information, and contact each other to exchange results, experiences and trust degrees of the knowledge sources etc. Since agents are going to act on behalf of humans, studies have been conducted to equip agents with human
logic. The most popular approach for this domain is the Believe-Desire-Intention model (Bratman, 1987). The vision of the Semantic Web goes beyond the view of agents working in isolation; it goes to considering all the agents as a society of communicative individuals. An agent in such a society can act in a social way by interacting with other agents in order to cooperate, coordinate, negotiate, advise and ask for advice. This kind of system is called a Multi-Agent System (MAS) (Marx-Gómez, Memari 2008a). Agents are categorized into different "species". Among these species is the mobile agent, and the intelligent agent, the proactive agent and the learning agent etc. In general, intelligent agents are not mobile since, usually, the larger an agent is, the less desirable it is to move it (Buckle, Hadingham, Poslad, 2000). An exception to this statement is the "Swarm Intelligence" in which the agents collaborate to perform complex tasks, which they are unable to solve individually due to their limited intelligence.

Peer-to-Peer networks are systems of nodes (peers) having equal rights which are interconnected without an isolatable central control instance and share resources among each other (Brehm, Marx-Gómez, Rautenstrauch, 2006). Peer-to-Peer applications employ distributed resources to perform a function or a task in a decentralized manner (Kalogeraki et al., 2002). A peer can act as a client as well as a server so it can request and offer resources within its network. These resources are exchanged directly between the peers which means that there is no central control instance responsible for the provision of resources. This responsibility is shared between all peers. Each peer can itself decide which resources are offered to other peers. All peers act autonomously on their own responsibility as regards to the provision and use of resources. However all nodes of a Peer-to-Peer network have to attend their duty by means of a cooperation with other participants of the network in order to enable this network to organize itself. A general task in this context is the forwarding of search requests to neighbored peers in case of search strategies based on the flooded requests model.

**Enterprise Architectures and Semantic Services**

EA projects mainly aim at the integration of processes via third-party applications and legacy systems in order to minimize the transformation effort when there is a connection between two systems. Since the semantics of heterogeneous systems must be integrated at one point, most of software-supported EA projects may fail if this scenario is considered.

Despite of the fact that software-supported EA projects involve a set of scenarios to enable integration by passing instance data, the process of integrating the semantics of the underlying systems seems to be difficult. The problem is that no formal interface definition is available. This implies that there is a need to build consistent knowledge between the applications in the overall integration project. Web-Service-enabled SOA solutions based on standards like UDDI, WSDL, SOAP, try to decrease the integration effort. This model is based on the idea that business functionality is separated and published as services which can be composed as a process.

According to this, a Web-Service-enabled SOA offers a solution to the standards problem by avoiding the central point of integration; also it reduces the number of point-to-point adapters because each interface is based on WSDL representation. What it doesn’t solve is the problem of creating unambiguous specification of such interfaces as regards to the functionality of services (Mahmoud, Marx-Gómez 2008b). By extending the concept of SOA with semantics, a formal description of the Web Service functionality will be provided. Determining the semantics for interfaces means to make a definition of the concepts as well as the relationships between them through an ontology language like Web Service Modeling Language (WSML) (Bruijn et al. 2006). Also semantics offer the possibility of binding services dynamically by discovering them at runtime (Bussler, Fensel, 2002).

**BACKGROUND INFORMATION**

**Service Oriented Architecture and Web Services**

A Web Service as defined by the W3C consortium is "a software system designed to support interoperable machine to machine interaction over a network" (Booth et al., 2004). It is the most commonly discussed approach to implement SOAs. In concept, there are three main components in SOA architecture (Mahmoud, Marx-Gómez, 2008a):
• **Service Provider**: It creates a Web Service and possibly publishes its interface and access information to the service registry.

• **Service Broker**: Also known as service registry, it is responsible for making the access information of both Web Service interface and implementation available to any potential service requester, and categorizing the results in taxonomies. The Universal Description Discovery and Integration (UDDI), defines a way to publish and discover information about Web Services.

• **Service Requester**: The service requester or Web Service client locates entries in the broker registry using various find operations and then binds to the service provider in order to invoke one of its Web Services.

Figure 1 illustrates the mechanism of publishing, discovering and binding Web Service in SOA environment.

Towards Semantic Service Oriented Architecture

With the use of Semantic Web markup languages, data structures passed through Web Service interfaces are expressed in ontologies creating a distributed knowledge base. It becomes a lot easier to understand what the service actually can be used for. And this provides the means for agents to reason about the Web Service description and to perform automatic Web Service discovery and invocation (Bruijn et al., 2007).

Web Service Modeling Ontology (WSMO) is a formal model for describing various aspects related to Semantic Web Services, and it is based on the Web Service Modeling Framework (WSMF), which is depicted in Figure 2. The objective of WSMO is to define a consistent technology for Semantic Web Services by providing the means for semi-automated discovery, composition and invocation of Web Services which are based on logical inference mechanisms. WSMO applies WSML as the underlying language based on different logical formalisms (Bussler, Fensel, 2002).

WSMO defines four main modeling elements for describing several aspects of Semantic Web Services (Bruijn et al. 2007):

- **Ontologies**: Are formal explicit specifications of a shared conceptualization (Gruber, 1993). They link machine and human terminologies.
- **Goals**: Provide the means to express a high level description of a concrete task.
- **Web Services**: Define various aspects of a capability.
- **Mediators**: Bypass interpretability problems.

In a semantically-enabled SOA there is a need to redefine the three main concepts of the traditional SOA in the following way:

- **Service Provider**: WSDL can still be in use as a universally accepted interface language, but additionally there is a need to provide other Web Service descriptions like WSMO, OWL-S or WSDL-S etc. compliant service descriptions. By doing so requester will be able to discover services based on formally-defined goals instead of searching only through the directory service and selecting the suitable one.
- **Service Broker**: The functionality of it remains the same. The only difference between it and traditional SOA service broker is that it stores semantic service descriptions instead of references to technical WSDL descriptions.
• **Service Requester:** Has to publish the desired functionality as a semantic goal in a semantic framework instead of the traditional way in SOA.

**Adaptive Multi-Agent Systems**

As defined in biology: adaptation is the change in organisms that allow them to live successfully in an environment (Freeman, Herron, 2007). The environment around the applications of the Semantic Web is in constant variation, and in order for a system to perform a good and suitable action in such a dynamic environment, it has to adapt to the new circumstances. That has to be done dynamically, autonomously and in runtime; today's accurate decision might be inaccurate tomorrow, the system must be capable of "learning from" and "adapting to" changes in the environment. The necessity of such adaptation on the Web comes from the fact that desired information often remains unfound, because it is hidden in a huge amount of unnecessary and irrelevant data that is increasing more and more. But automatic information processing capabilities are also increasing, thus there are and will be great potentials to make use of the automation capabilities in order to extract the information and services relevant to the user from the overflow of the Web on an ad-hoc basis (Marx-Gómez, Memari, 2008b).

The relation between adaptation and agents has two different dimensions:

- **The adaptive agent dimension:** From this point of view, agents react on changes in the environment, and in other words they adapt to it. We look at the agent itself as an adaptive application, since the features of the way that agents work can be easily mapped into performing tasks that are adaptivity-related; e.g. agents use knowledge bases and rules to govern their actions and that can easily be mapped into the use of rules to personalize information according to a user model. An adaptive agent is an agent with actions that are flexible and may be learned through interaction (Walton, 2006).

- **Using agents for building adaptive applications dimension:** Looking at the relation from this point of view leads us to consider a higher level of adaptivity, where a MAS can compose a layer or a module in the application which will be responsible for giving the application the adaptive property, in a way that this MAS is used for filtering, rating and optimizing the information according to the preferences of a user (or a set of users). Such a use of MAS can be found in a semantic search engine that constitutes a part of a Web portal, or for a semantic Web Service discovery component as the one proposed in this article. For another example of a multi-agent framework for personalized information filtering see (Kunegis, Lommatzsch, Mehlitz, 2007).

These two dimensions in looking at the relation between adaptation and agents are not mutually excluded, i.e. they can co-exist, so we can have a platform for extracting relevant information from the overflow of the Web that uses a layer of agents as the filtering engine, with these agents being at the same time adaptive applications that can autonomously adapt to the changes in their environment (user preference changes, changes of trust degrees of an information source, reliability changes of a peer etc.). This hybrid approach will be used in the IDEAS scenario.

**Decentralized Service Discovery**

If we consider Web Services as resources of a Peer-to-Peer network, search strategies can be applied to decentralized enterprise networks where each peer (enterprise) can benefit of both the usage and the provision of access points (interfaces) to business application systems or business components (Brehm, Marx-Gómez, Rautenstrauch, 2006) without being dependent to a central control instance. In general, two different scenarios are imaginable to make use of the provision of business applications based on Web Services in a Peer-to-Peer network. Firstly Web Services can provide access to enterprise services as part of a supply chain whereas companies either act as suppliers or as customers and thus they either search for or publish product offerings (e.g. materials) automatically. Secondly enterprises can share professional business functionality in the form of software components which are encapsulated as Web Services (e.g. information resources about the interests of customers or optimization algorithms in the area of production planning and scheduling problems) (Brehm, Marx-Gómez 2007). In both cases the problem of finding a suitable service arises. This can be transferred to the field of Peer-to-Peer networks whereas possible existing solutions for search problems
can help to bring the characteristics of open markets to the Web Service world. Open markets are characterized by the absence of organizational and technical barriers that hinder potential actors to participate in the market. If we consider a market in general as institutionalized place where demands and offerings come together, markets can also be seen as a medium to manage the problem of negotiation between required and available software components (as services) in large information system landscapes inside one enterprise.

PROBLEM DEFINITION

The Problems of SOA and the Benefits of Semantic SOA

- Traditional SOA service discovery problems: In the next few years, Web Services appear to be more widely adopted, allowing much broader intra- and inter-enterprise integration. But developers will require automated systems for service discovery, enabling further Web Service interactions with less human effort. UDDI exists precisely for this reason. However, unless the service requester knows the exact form and meaning of a service's WSDL specification in advance, the combination of UDDI with WSDL and coarse-grained business descriptions is not enough to allow fully automated service discovery and usage.

- Semantic SOA service discovery: As we mentioned earlier, the three main components in SOA architecture are: service provider, registry, and requester. To add semantics to these components' interactions, (Li, Lin, Qiu 2007) proposed adding a semantic matchmaker system to existing UDDI registry, in the other hand, the service provider will add a semantic description to his Web Services descriptions. Additionally, by using ontologies, the requester will send his request as a semantic goal. In this way, the UDDI will receive a semantic goal and make a matchmaking process with available Web Services to give the requester the most related results fulfilling his needs.

Assured that the world of offered Web Services on the Web is massive in number and variation, any user of Web Services will only use a very small portion of them. The services within this portion have something in common: they are desired by this user. What if we managed to isolate this portion?

Nowadays, desired services and providers are hard to find, and they can remain unfound, because they are hidden under a huge amount of irrelevant services. For finding all the relevant services it is usually necessary to analyze potentially relevant ones from a big number of providers, rate each of them based on several different strategies and deliver only the most relevant results to the user, taking into account the preferences of the requester, the query context, as well as knowledge about sources and ratings provided by other users from previous experiences. When we keep in mind that this forest of possibilities is in constant movement, and that many services are added, removed, or modified every minute, then we can imagine that under these circumstances, discovering the appropriate Web Service for a specific purpose is not a simple task, furthermore, processes themselves are subject to changes, so a static goal description is no more sufficient.

An intelligent agent can continuously keep an eye on this variable environment of available Web Services as well as the variable goals and preferences of the user, and can significantly decrease the response time of the system and increase the accuracy by having a registry of cached Web Services, which tries to expect the portion of Web Services space that conforms to the needs of the user.

This registry will be filled automatically by the agent with service descriptions of the most appropriate services according to user's profile. This profile is updated either manually, or by intelligently observing actions of the user in order to extract new trends and preferences from his behavior, and acting immediately to fulfill the new requirements and preferences.

The Problems of Network Centralization

The centralized character of infrastructures of Web Service-based application systems results from the conventional approach to use only one instance of a repository (e.g. a UDDI registry) in which Web Service offerings can be published.
and searched. In doing so, the provision of such a UDDI registry is dependent on a central management unit. Speaking in terms of an organizational point of view, this instance is a control center for the whole network. Initially the following advantages arise by the consideration of this concept as part of Web Service-based software infrastructures:

- **Common access point**: Both, users and providers of Web Services deal with only one system for publishing and searching Web Services. Thereby the amount of dependencies and communication connections between a number of users and a number of providers is reduced to a minimum.

- **Completeness of information about offerings**: Because only one system is used for the management of all offerings, the effort for the generation of complete search results is reduced to a minimum. Normally only one database is existing which has to be searched through.

Despite of these advantages, the following disadvantages and problems arise:

- **Single point of failure**: A central repository is a component which in case of failover can cause a breakdown of the whole system landscape (Krämer, Papazoglou, Yang 2003). Furthermore such a central instance often limits the scalability of systems (e.g. application networks) in general (Bellwood et al. 2004).

- **Repository provider’s monopoly**: Because the offering of a repository as a service is conducted by only one provider, this market form can be considered as a supplier’s monopoly. This causes the effect that the repository provider is not forced to consider his own effort for the definition of prices to be paid by the ones who require the repository service (in a certain quality) in order to benefit from SOAs in general or Web Service technology in particular. Offerings of concurrent repository providers do not have to be taken into consideration. Thinking in terms of a given effort of profit maximization, a repository provider can negatively influence the general trend of providing and using services within a network. This in turn can reduce the economic values a SOA is expected to create. Those market scenarios which are dependent on a central control instance need to be supervised by other regulatory units as e.g. other enterprise departments or the government. Even if this problem can arise inside enterprises with a central repository for all services, the cross-enterprise organization of such repositories has to be based on open market strategies where the repository providers themselves are competing companies. Here the equality of actors within a market can be brought by Peer-to-Peer network models.

**INTELLIGENT DISCOVERY OF ENTERPRISE ARCHITECTURE SERVICES (IDEAS)**

As the proposed solution, IDEAS, is a component that will reside in the Web Service invocation component as illustrated in Figure 3 on the next page. Its main mission is publishing the Web Services of the enterprise it resides in, and discovering the services of other enterprises, in addition to other tasks concerning the social activities of IDEAS and the interaction with other IDEASs such as advising, asking for advice, cooperating and negotiating.

The IDEAS component itself is built to be an agent; which means it will not be activated for a certain task, rather it will activate itself and will be always searching and filtering new found services according to a profile of the owning enterprise. This profile is the logical representation of the enterprise’s knowledge; it contains preferences, trends, requirements and goals. This profile is represented semantically using ontologies.
IDEAS also has to keep an eye on the feedback that comes from the enterprise management system concerning the relevance, efficiency, performance, productivity and cost effectiveness of the suggested services. According to this feedback, IDEAS will update the user profile that it keeps, and will store the knowledge about this experience in its reputation knowledge base. The performance and effectiveness of executed service will affect rating of the service provider for the executing peer. This knowledge about rating of service providers will be shared among IDEAS components in the society, different policies for exchanging this knowledge may
apply here, other peers may have to pay a certain amount of money for this knowledge, or they might have to give a useful experience in return as suggested in "Case Bartering Mechanism" (Plaza and Ontaño, 2003). IDEAS will return the services it provides as a result for a search request when a hit occurs, and will forward this request to all known peers (flood the request), which is one of its tasks as an agent in a typical Peer-to-Peer network.

Using this component in a decentralized network, ad-hoc collaborations between enterprises can be established whereas those relationships can be created without dependence on how a central management instance is capable to broker between the members of potential compounds of companies, which will increase the agility of the enterprise by increasing the ability of adapting to the dynamic nature of the environment surrounding the enterprise. In this decentralized network the availability of offered services will increase when there are redundant service providers offering a similar service available in the open market.

In Figure 4 we can see the following components:

- **Selection Component**: It is responsible of selecting the Web Services that are relevant to enterprise needs based on the knowledge about the past experiences with these services whether they were gained from this enterprise or from other peers on the network. It stores and retrieves these entries using the reputation knowledge base.
- **In UDDI**: This component caches the specifications of Web Services that were filtered by the selection component according to a semantic goal or by using traditional syntactic way.
- **Out UDDI**: It has the role of publishing Web Services that the enterprise provides.
• **Match Maker**: Receives the requests for a Web Service as semantic goals and tries to match them semantically with the capabilities of offered ones. It also has the task of resolving interoperability problems and forwarding this request to other peers.

**CONCLUSION**

In this article we have proposed our IDEAS component as a semantic agent in a Peer-to-Peer environment. We have shown it as a powerful component which allows dynamic discovery of services published in the decentralized network.

Since the automated discovery of Web services is not facilitated by a conceptual model alone we have identified some challenges on Semantic Web Service frameworks and standardization efforts, and by using the IDEAS component we reach the goal of dynamic environments.

Our future work will focus on the development of the policy assertions as logical conditions including security, reliability, etc. in order to apply it on network nodes based on the user rating about the Web Services offered by a specific node; and later on, this feedback will be stored on another node in an encrypted manner. Prototype implementations of the future will have to show the practicability of those concepts.

**AUTHOR BIOGRAPHIES**

**Nico Brehm** studied computer science at the University of Applied Sciences in Wernigerode (Germany) and is currently a research assistant at the working group of business information systems at the University of Oldenburg. His research work focuses on enterprise resource planning (ERP) systems, IT security, and service-oriented architectures (SOA).

**Tariq Mahmoud** studied Information Engineering at Al-Baath University (Syria) and is currently a research assistant at the working group of business information systems at the University of Oldenburg. His research work focuses on applying semantics on SOA-enabled ERP solutions, information security, and Semantic Web.

**Ammar Memari** studied Information Engineering at Al-Baath University (Syria) and is currently a research assistant at the working group of business information systems at the University of Oldenburg. His research work focuses on adaptation and user modeling, multi-agent systems, Semantic Web.

**Jorge Marx Gómez** studied computer engineering and industrial engineering at the University of Applied Science of Berlin (Germany). He was a lecturer and researcher at the Otto-von-Guericke-Universität Magdeburg where he also obtained PhD degree in business information systems with the work “Computer-Based Approaches to Forecast Returns of Scrapped Products to Recycling”. In 2004, he received his habilitation for the work “Automated Environmental Reporting through Material Flow Networks” at the Otto-von-Guericke-Universität Magdeburg. From 2002-2003, he was a visiting professor for business informatics at the Technical University of Clausthal (Germany). In October 2005, he became a full professor of business information systems at the Oldenburg University (Germany). Professor Marx Gómez’s research interests include business information systems, business intelligence, e-commerce, Web intelligence, material flow management systems, life cycle assessment, eco-balancing, environmental reporting, recycling program planning, disassembly planning and control, simulation, and neuro-fuzzy-systems.

**REFERENCES**


