

Situational Enterprise Services

Paul de Vrieze

SAP Research, Switzerland

pdevrieze@acm.org

Lai Xu,

SAP Research, Switzerland

laixu@acm.org,

Li Xie

GuangDong Polytechnic Normal University, China

xielimorning@gmail.com

ABSTRACT

The ability to rapidly find potential business partners as well as rapidly set up a collaborative business process is desirable in the face of market turbulence. Collaborative business processes are increasingly dependent on the integration of business information systems. Traditional linking of business processes has a large ad hoc character. Implementing situational enterprise services in an appropriate way will deliver the business more flexibility, adaptability and agility.

Service-oriented architectures (SOA) are rapidly becoming the dominant computing paradigm. It is now being embraced by organizations everywhere as the key to business agility. Web 2.0 technologies such as AJAX on the other hand provide good user interactions for successful service discovery, selection, adaptation, invocation and service construction. They also balance automatic integration of services and human interactions, disconnecting content from presentation in the delivery of the service. Another Web technology, such as semantic Web, makes automatic service discovery, mediation and composition possible. Integrating SOA, Web 2.0 Technologies and Semantic Web into a service-oriented virtual enterprise connects business processes in a much more horizontal fashion. To be able run these services consistently across the enterprise, an enterprise infrastructure that provides enterprise architecture and security foundation is necessary.

The world is constantly changing. So does the business environment. An agile enterprise needs to be able to quickly and cost-effectively change how it does business and who it does business with. Knowing, adapting to different situations is an important aspect of today's business environment. The changes in an operating environment can happen implicitly and explicitly. The changes can be caused by different factors in the application domain. Changes can also happen for the purpose of organizing information in a better way. Changes can be further made according to the users' needs such as incorporating additional functionalities. Handling and managing different situations of service-oriented enterprises are important aspects of business environment. In the chapter, we will investigate how to apply new Web technologies to develop, deploy and executing enterprise services.

INTRODUCTION

Service-oriented computing paradigm is transforming traditional enterprise systems from a close, centralized control system into a dynamic information exchange and flexible business process system. Traditionally enterprise applications are defined as software designed to integrate all aspects of a company's operations and processes such as accounting, finance, human resources, inventory control, manufacturing, marketing, sales, and distribution, and resource planning. Advanced enterprise applications provide linkages with customers, business partners, and suppliers (Markus & Tanis, 2000). Normally enterprise applications are complex. There are mission critical applications which developed and deployed by central IT with long development deployment cycle and dedicated IT budget.

Currently, there is increased pressure to build enterprise applications quickly in order to respond to situational needs of the business. Many of these applications for reflecting situational business needs

never get delivered because they are too difficult to write, too costly to implement, and too brittle to customize and maintain once deployed. As a result, many of the needs are addressed by business people who have some knowledge on IT techniques together often inadequate solutions using tools like Excel, Access and Visual Basic.

With a growing number of services on the Web, these needs can now be satisfied more easily and effectively. These development and deployment services, combined with a "situational" mindset and methodology, can offer significant advantages. Unlike traditional enterprise applications, situational enterprise applications are relatively simple. They are not mission critical for organizations. Lots of them developed at the point of need short development cycle under central IT control with little or no recognized budget.

Situational enterprise applications being addressed will not replace core business applications, such as ERP (Enterprise Resource Planning), SCM (Supply Chain Management), CRM (Customer Relationship Management) etc. They address different needs which are built for just handful of users, situational enterprise applications that are used for only a few weeks or months, or situational applications address a small piece of functionality. For example, within the perimeter ERP applications, departmental operation solutions, such as vacation scheduling, seminar and presentation management, purchase procedure management within a work unit, etc, normally are not included in a organization ERP system. However, they can be desired by department staffs. These are typical situational applications for the department staffs who manage those matters on a daily basis.

Types of situational enterprise applications can be divided into data-oriented applications and process-oriented applications. Enterprise widgets, gadgets, pipes and mash-ups belong to data-oriented applications. Lightweight process-oriented applications are currently under research, the EU project SOA4All (<http://www.soa4all.eu/>) aims to provide a platform to build process-oriented applications for end users (non-technical users).

The target audience for situational enterprise applications/services is an educated professional (e.g., accountant, HR personnel) with modest computer literacy (and interest) that mostly includes the Web and MS Office. They have basic computer experience like using a wizard to generate something new; interacting with spreadsheets, documents, and forms; and using drag and drop to rearrange items on the screen.

These solutions on demand will help businesses slash expenses and reduce cycle times by more effectively supporting how people work, address challenges and make business decisions. Situational enterprise applications/services will allow also the business to be more innovative and competitive by supporting new processes more effectively, increasing overall productivity, and facilitating new ways for sharing information.

In this chapter, we introduce background information of situational enterprise applications, general description of Service-Oriented Architecture (SOA), Web services, and enterprise services in Section 2. The needs and benefits of situational enterprise services describe in Section 3. Design principles of situational enterprise services present in Section 4. Section 5 explains applying issues of situational enterprise services. Conclusions and future research direction are depicted in Section 6.

BACKGROUND

Relation among SOA, web services, and enterprise services

Service-oriented architecture (SOA) is a design paradigm which designers use loosely coupled services for building complex services or for incorporating them into applications. An ideal level of abstraction is required for aligning business needs and technical capabilities, to create reusable, coarse-grain business functionalities. SOA is not just architecture of services seen from a technology perspective, but the policies, practices, and frameworks by which it is ensured that the right services are provided and consumed. Enterprise SOA has been revolutionized the design of business

applications, enabling the rapid composition of business solution.

A Web Service is a software service designed to support interoperable XML based machine-to-machine interaction over the Internet. It has an interface described in a machine-processable format specifically Web Service Definition Language (WSDL). (W3C, 2004). Web Services are self contained and self describing application functionalities that can be processed through open Internet standards. SOA is currently adopted massively by many enterprise software vendors (Cardwell, 2007).

Enterprise services as smaller functionality components are introduced by different enterprise software vendors. Enterprise services are highly-integrated web services combined with business logic and semantics that can be accessed and used repeatedly to support a particular business process (Fremantle, Weerawarana, & Khalaf, 2002). The following characteristics differentiate enterprise services from regular web services. First enterprise service contains business semantics. For regular/general Web services, business semantics are not required. Enterprise services are structured according to a harmonized enterprise model based on business objects, process components, and data types. They are defined using common business rules. Second, enterprise services require quality and stability for future using. Regular/general Web services do not need to have such strict rules. Enterprise services safeguard a stable interface for backward compatibility. Their behavior, prerequisites, dependencies of usage and configuration possibilities need to be documented. Third, enterprise services are based on open standards according their application areas, i.e. B2B enterprise services are defined in compliance with e-business standards. Regular/general Web services for different purposes, they will either follow certain standards or do not have any standard need to be followed. The interfaces of enterprise services are described as e.g. WSDL or REST.

A semantic annotation in a document is additional information that identifies or defines a concept in a semantic model in order to describe part of that document. The annotation links the concept to an ontology that is described externally to the document and is used to describe the semantics of concepts for broad use. In Semantic Annotations for WSDL (SAWSDL), semantic annotations are XML attributes added to a WSDL or associated XML Schema document, at the XML element they describe. Semantic annotations are of two kinds: explicit identifiers of concepts, or identifiers of mappings from WSDL to concepts or vice versa. (W3C, 2007)

Difference between enterprise services and situational enterprise services

Situational enterprise services can be web services, enterprise services, widgets, gadgets, pipes, feeds, or mash-ups which can support to build situational enterprise applications. There are not a strict line among Web services, enterprise services and situational enterprise services. Enterprise services are certainly Web services. Situational enterprise applications can be perimeter enterprise applications. Being able to build a perimeter enterprise application, situation enterprise application could consume some related enterprise services. Ideally, situational enterprise applications are support by an end user programming environment. In the programming environment, situational enterprise services are defined not only including semantic annotated Web services and enterprise services, but also involving semantic annotated widgets, gadgets, pipes, feeds and mash-ups. These web-based resources are also important resources for building situational enterprise applications. Business users should easily express their need using a lightweight business process language. Discovery, selection, composition resources such as web services, widgets can be (semi-)automatically done.

Definition of Situational Enterprise Services

We define situational enterprise services are semantic annotated web services , enterprise services and semantic annotated web-based resources, such as annotated widgets, gadgets, pipes, feeds and mash-ups. The situational enterprise services are used to build situational enterprise applications in a end user programming environment.

The Needs of Situational Enterprise Services

Motivating scenario and issues

As a motivational scenario take the setting of a small work unit (small company or department of a larger one) that performs specialized work. As part of the specialized work, there is a frequent need for specialized material purchases. The quantities are not sufficient for a complete purchasing process to be followed, so currently the materials are ordered ad hoc. As these materials are ordered on a regular basis by all team members, the manager wants to integrate the various suppliers' catalogs into the purchasing process whereby the team members can easily order items from an up-to-date catalog of all the offerings of a supplier. At the same time, the manager wants to achieve a better usage of resources by blocking the purchase of small quantities of items that are regularly used.

The scenario described above is a straightforward case of supplier integration and is generally supported within enterprises for large suppliers. As in the traditional way of integration the effort required to integrate a catalog is not insignificant, smaller suppliers, or needs of small parts of an organization are often not attended to. It is a typical process-oriented situational perimeter ERP application which we mentioned in Introduction section.

On www.programableweb.com, a leading mashup directory, there are 3865 registered mashups on the web site. Every month, about 100 new mashups are added. Descriptions of feeds can be obtained, for example, from social bookmarking web sites like www.syndic8.com (Barr & Kearney, 2001), it has about 562,488 feeds. There are too many web-based resources for end users to manually discover, select or compose. There are needs to provide a common platform beyond current mashup environment to facilitate end users to build their applications not only data-oriented, but also process-oriented.

Requirements of building situational enterprise applications

The key requirement of situational enterprise applications is that their initial development until in a working stage is reasonably simple and cheap. This means that little time must be spend in the development, and the knowledge of the “developer” on the framework is limited. The developer is often someone in the line of business with a certain degree of computer skills, but mostly significant knowledge of the actual business processes.

For the developers of situational enterprise applications the development is not their main professional activity. As such for when combining services, it must be easy to find and use the component services. It can not be expected that the developers have prior knowledge of available components or their proper usage, or are willing to invest significant time in learning about this. The retrieval and usage of the components is generally facilitated by rich descriptions using semantic technologies.

Design Principles of Situational Enterprise Services

Service-Oriented Architecture (SOA) and Service-orientation Principles

Service oriented architecture is primarily an approach to information system design. Within the SOA approach the information system consists of loosely coupled components that are interconnected through (Web-) services (Papazoglou & Georgakopoulos). SOA has a number of goals and characteristics:

- **Module independence:** The individual modules can be independently modified or deployed.
- **Alternatives:** Service interfaces can be provided by different components, allowing for a choice based on criteria outside the service definition.
- **Distribution:** As service technologies are naturally decoupled, each component of an SOA system can trivially be deployed on a different computer.
- **Clear separation:** Individual modules are independent processes with clear interfaces. This

allows easy access enforcement and parallelism.

Enhancing SOA with Semantic Technologies

Web services, the enabling technology of SOA are the latest evolution of remote procedure calls (RPC) (Birrell & Nelson, 1984). Web service interfaces are generally described using the Web Service Description Language (WSDL). WSDL, while being based on Web services is only a limited extension to traditional Interface Definition Languages (IDL). WSDL as well as other IDL languages describe the available functions and their parameter and return types. These descriptions, while providing sufficient information for the correct syntax for the invocation of the services, do not describe the semantics of the services.

Traditionally semantic information has been provided by means of textual documentation. This textual information however has disadvantages when used as exclusive source of semantic information. Textual information is generally not machine readable, therefore the information system can not use this information.

Semantic descriptions of Web services, such as provided through SA-WSDL or WSML, can be used for various purposes. Semantics can be used to strengthen the description of services and have stronger verification of correct invocation. Semantics can also be used for service matching, allowing services to be automatically selected for use. A third use is to use the semantic description of the services to facilitate the authoring of information system components that use the services.

Enhancing SOA with the Web Principles

Services as Web resources

The Web has had a big impact on information system design. Many information systems are now provided as Web applications, providing access without requiring a specific client on the user computer. The user interface is further universally provided by the browser, providing a high degree of consistency and platform independence.

The Web is characterized by the fact that it is very easy to get access to the original information through the source view (and easy editing), easy linkage between systems through URLs, as well as easy access to related information through URL editing. Although SOAP, the most used Web services technology, is generally provided over HTTP, it is far from straightforward. The SOAP HTTP binding basically uses HTTP as a tunnel through which an information system is accessed. This conflicts with the design philosophy of HTTP and has a number of caveats. Lately this conflict in philosophy, as well as the general complexity of SOAP, have lead to the growing popularity of the RESTfull approach to Web services (Fielding, 2000). The REST approach follows the HTTP philosophy which allows for example easy access by Web browsers and scripts without a translation layer. This also eases debugging of systems that use Web services as the services themselves can be easily invoked with a browser and their results verified.

Services for the Web

Web services, especially RESTfull Web services, are based on Web technology. Web browsers, as well as technologies for server side Web programming such as PHP, ASP and Servlets provide easy access to Web resources. Given this good match between services access technologies it is not surprising that Web mash-ups have become popular as a means to expose services, or a combination of services, to end users.

Integrating SOA with the Web2.0 and Semantic Web

Web 2.0 is a label put to recent developments on the Web. In Web 2.0, the Web is participatory. The information is created collaboratively. The experience of a Web 2.0 site improves with user participation.

The application of situational enterprise services

Situational enterprise services can be used to address the issues of the motivational scenario. In the

solution, a number of components come together that contribute to a sufficient solution for the work unit:

- The supplier provides electronic access to his information. The product catalog is available for electronic querying. Ideally this is in the form of a Web service (SOAP or REST based), but most traditional Web pages would suffice when coupled with a Web-scraping (Pan, 2002) module. For simplicity we assume that the catalog is exposed as Web service.

The second part is electronic ordering. The supplier allows electronic ordering of items from the catalog. The business unit has an account with the supplier that allows for proper authentication of purchase orders with a fixed delivery address. A Web service (or if need be, a fax) is used to submit the orders accompanied by an authentication token. The supplier receives the order and optionally provides an order confirmation by email to the department head and / or the employee ordering the item.

- The company has a workflow system that allows for machine interaction through Web services. When an employee creates an order through the situational purchasing portal, the order, complete with links to Web pages describing the ordered items is submitted to the workflow system.

The workflow system allows for locally specified workflows, and the manager of the work unit has created a lightweight workflow that is used for the approval of purchases. Some purchases are automatically approved or rejected, while some are sent to the manager for approval. The workflow system verifies that these procedures do not violate corporate policy by for example going above the approval limit of the manager.

- The Web interface of the workflow system allows the manager to review the pending approval requests. The requests are accompanied by all information about the goods to be purchased. The approval request contains sufficient information for subsequent ordering by the purchasing situational application. When the request is approved, the request is sent to the purchasing situational application, that then executes the purchase by invoking the Web service of the supplier with the information in the request.
- The situational purchasing application connects all aspects together. There are two parts of the application. The first part is the catalog access. Using the Web service that exposes the catalog information from the supplier, a custom catalog is created that exposes the items for purchase. Some items or quantities may have been blocked from purchase and will not be available. An employee can select which items he wants to purchase in which quantities. When the employee then confirms his selection, the application composes a purchasing request and submits it to the existing workflow system for execution with the custom workflow for the application.

When the purchase request is approved, the workflow system then forwards the approved request to the situational purchasing application. The situational purchasing application takes the information in the request and uses it to submit a purchase request using the appropriate Web service of the supplier.

The easy creation of this application has a number of requirements. The fulfillment of these requirements needs the various composite technologies and defines the nature of the system as a situational enterprise service. First of all, the catalog actually needs to be semantically annotated to allow a generic catalog framework to make use of it. This semantic annotation comes in the place of a standard protocol where the semantics have been predefined. It is difficult to expect services from external partners to implement a standard. The advantages of a semantic annotation are that the semantic technology can help mapping between various notions in the protocols, as well as the fact that annotations can be provided by parties other than the supplier.

The submits orders to the workflow system. The interface to the workflow system is specified semantically (to account for different workflow system interfaces). The submission is also semantically annotated such that the request is recognized as a purchase request and the system can handle the request appropriately and still enforce corporate policy. Similarly, the custom workflow for the system is semantically annotated (and probably derived from a standard template) such that its correctness and fitting with corporate policy is ensured.

The workflow system uses the workflow description leading to an automatic or manual approval or rejection. The system must have some level of support for custom requests. Basically this means that the system is able to display custom documents.

FUTURE RESEARCH DIRECTIONS

The area of situational enterprise services is very new. There are a lot of topics that still remain to be researched. Below we sketch a number of the topics that are particular importance.

An important question is how web services, enterprise services, widgets, gadgets, pipes, feeds and mashups can be annotated in a unified way. Can all web resources be seen as services? How is a mechanism designed that glues services together automatically and enhances interoperability?

What is the right way to abstract processes (mapping between an activity and a service). When activities are too abstract they do not provide sufficient information for mapping a service onto them, and when activities are too detailed they do not allow flexibility, and the usage of the detailed activities is highly complex.

As situational enterprise applications are flexible and easy to extend, it is interesting to see how well the technology supports scientific work-flows, for example to support genome research.

CONCLUSIONS

The topic of situational enterprise services encompasses concepts such as semantically annotated web services and enterprise services, as well as more lightweight concepts such as widgets, gadgets, pipes, feeds and mashups. Within the moniker of situational enterprise services service concepts are applied to all web resources.

Situational enterprise services includes semantic annotated web services, enterprise services and other web-based resources such as widgets, gadgets, pips, feeds and mashups. It is applying service concepts to all web-resources. Situational enterprise services can extend current data-oriented applications into process-oriented applications.

REFERENCES

- Barr, J. and Kearney, B (2001) <http://www.syndic8.com>
- Birrell, A. D., & Nelson, B. J. (1984). Implementing remote procedure calls. *ACM Transactions on Computer Systems*, 2 (1), 39–59. doi:10.1145/2080.357392.
- Cardwell, G. (2008). The inuence of enterprise architecture and process hierarchies on company success. *Total Quality Management & Business Excellence*, 19 (1–2), 47.
- Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. Unpublished doctoral dissertation, University of California, Irvine.
- Fremantle, P., Weerawarana, S., & Khalaf, R. (2002). Enterprise services. *Communications of the ACM*, 45 (10), 77–82. doi:10.1145/570907.570935.
- Markus, M., & Tanis, C. (2000). The enterprise systems experience-from adoption to success. In R. W. Zmud & M. F. Price (Eds.), *Framing the domains of it management: Projecting the future through the past* (pp. 1–46). Cincinnati, Ohio: Pinna ex Educational Resources, Inc.

- Pan, A., Raposo, J., Álvarez, M., Hidalgo, J., & Viña, Á. (2002). Semiautomatic wrapper generation for commercial web sources. In *Proceedings of the ifip tc8 / wg8.1 working conference on engineering information systems in the internet context* (pp. 265–283). Deventer, The Netherlands, The Netherlands: Kluwer, B.V.
- Papazoglou, M. P., & Georgakopoulos, D. (2003). Service-oriented computing. *Communications of the ACM*, 46 (10), 24–28. doi:10.1145/944217.944233.
- W3C (2004). *Web Services Glossary*. <http://www.w3.org/TR/ws-gloss/>.
- W3C (2007). *Semantic Annotations for WSDL and XML Schema – Terminology*. <http://www.w3.org/2002/ws/sawsdl/spec/#Terminology>.

ADDITIONAL READING

- Bravery, A., Cherbakov, L. and Pandya, A. (2008) *SOA meets situational applications, Part 2: Building the IBM Situational Applications Environment*, IBM Developer Works, 10 January 2008.
- Cherbakov, L., Bravery, A., Goodman, B. D., Pandya, A. and Baggett, J. (2007) *Changing the corporate IT development model: Tapping the power of grassroots computing*, IBM Systems Journal, 2007.
- Cherbakov, L., Bravery, A. and Pandya, A. (2007) *SOA meets situational applications, part 1: Changing computing in the enterprise*, IBM Developer Works.
- Tuchinda, R., Szekely, P., and Knoblock, C. A. (2008). Building Mashups by example. In *Proceedings of the 13th international Conference on intelligent User interfaces*. IUI '08. ACM, New York.
- de Vrieze, P., Xu, L., Bouguettaya, A. Yang J. and Chen J. (2009) *Process-oriented Enterprise Mashups*. 4th International Workshop on Workflow Management (ICWM2009), Geneva, Switzerland, 4-8 May 2009.
- Wong, J. and Hong, J. (2008). What do we "mashup" when we make mashups?. In *Proceedings of the 4th international Workshop on End-User Software Engineering* (Leipzig, Germany, May 12, 2008). WEUSE '08. ACM, New York, NY, 35-39.
- Zang, N., Rosson, M., and Nasser, V. (2008). Mashups: who? what? why?. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*. Florence, Italy, April 05 - 10, 2008. CHI '08. ACM, New York,
- Yu, J., Benatallah, B., Casati, F., and Daniel, F. (2008) Understanding Mashup Development. *IEEE Internet Computing*. (12)5, pp. 44-52.
- Benslimane, D., Dustdar, S., and Sheth, A. (2008). Services Mashups: The New Generation of Web Applications. *IEEE Internet Computing*. (12)5, pp. 13-15
- Rosenberg, F., Curbera, F., Duftler, M. J., and Khalaf, R. (2008) Composing RESTful Services and Collaborative Workflows: A Lightweight Approach. *IEEE Internet Computing* 12, 5 (Sep. 2008)
- Braga, D., Ceri, S., Daniel, F., and Martinenghi, D. (2008). Mashing Up Search Services. *IEEE Internet Computing* 12, 5 (Sep. 2008), 16-23.
- Maximilien, E. M., Ranabahu, A., and Gomadam, K. (2008). An Online Platform for Web APIs and Service Mashups. *IEEE Internet Computing* 12, 5 (Sep. 2008), 32-43.
- Silva-Lepe, I., Subramanian, R., Rouvellou, I., Mikalsen, T., Diamant, J., and Iyengar, A. (2008). SOALive Service Catalog: A Simplified Approach to Describing, Discovering and Composing Situational Enterprise Services. In *Proceedings of the 6th international Conference on Service-Oriented Computing* (Sydney, Australia, December 01 - 05, 2008). A. Bouguettaya, I. Krueger, and T. Margaria, Eds. Lecture Notes In Computer Science, vol. 5364. Springer-Verlag, Berlin, Heidelberg, 422-437
- Ennals, R. J. and Garofalakis, M. N. (2007). MashMaker: mashups for the masses. In *Proceedings of the 2007 ACM SIGMOD international Conference on Management of Data* (Beijing, China,

- June 11 - 14, 2007). SIGMOD '07. ACM, New York, NY, 1116-1118
- Nigam, A. and Caswell, N.S. (2003) Business artifacts: An approach to operational specification. *IBM Systems Journal*. 42(3): 428-445.
- Alonso, G., Casati, F., Kuno, H. & Machiraju, V.(2004). *Web Services Concepts, Architectures and Applications*. Springer, 2004
- Anderson, C.(2006). *The Long Tail, Why the Future of Business is Selling Less of More*. Hyperion. July 2006.
- Högg, R., Meckel, M., Stanojevska-Slabeva, K. & Martignoni, R.(2006). *Overview of business models for Web 2.0 communities*, Proceedings of GeNeMe 2006, p. 23-37, Dresden, 2006
- McAfee, A.(2006). Enterprise 2.0: The Dawn of Emergent Collaboration. *MIT Sloan Management Review*, Vol.47, No.3 (pp. 21-28).Spring 2006
- O'Reilly, T. & Musser, J.(2006). Web 2.0 Principles and Best Practices. *O'Reilly radar*, November 2006.
- O'Reilly, T.(2005). *What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>
- Riabov, A. V., Boillet, E., Feblowitz, M. D., Liu, Z., & Ranganathan, A. (2008) *Wishful search: interactive composition of data mashups*. In Proceeding of the 17th international Conference on World Wide Web (Beijing, China, April 21 - 25, 2008).
- Smith, R.(2006). *Enterprise Mashups: An Industry Case Study*. Keynote at the New York PHP Conference & Expo, Manhattan, New York, USA, 14-16 June 2006

TERMS

Mashup: A mini application that is primarily created by composing independent components from various sources. Information and functionality from one source is combined with information or functionality from other sources to create added value. After combination the result is visualized.

Web service: A web service allows remote systems to interact with each other.

Semantic annotation: The annotation of an object, relating its concepts to well defined semantics. This allows computer reasoning about the object. A simple example would be to provide a US zip code as a general zip code. On a higher level, the semantic annotation could express information about the functionality of the service.

Semantic web service: A web service that is semantically annotated. This allows for machine assisted or automatic usage of web services

Enterprise service: An enterprise service is a highly integrated web service that combines business logic and semantics that is used to support a particular business process.

Service oriented architecture: Service oriented architecture is an approach to the design of information systems. Within this approach an information system consists of various independent modules that interact through exposing service interfaces.

ERP system: An ERP system is a system that supports the running of a business by maintaining shared data and functionality for a broad range of business functions, ranging from human resources to production planning and sales.