Semantic Web Enabled Web Services: State-of-Art and Industrial Challenges

Vagan Terziyan, Oleksandr Kononenko

Industrial Ontologies Group, MIT Department, University of Jyvaskyla,
P.O. Box 35 (Agora), FIN-40014 Jyvaskyla, Finland
vagan@it.jyu.fi, olkonone@cc.jyu.fi

Abstract. Semantic Web technology has a vision to define and link Web data in a way that it can be understood and used by machines for automation, integration and reuse of data across various applications. Ontological definition of every resource as it is assumed in Semantic Web, along with new techniques for semantics processing and new vision Intelligent Web Services is expected to bring Web on its new level. At present, Web Services technology is stressed by the search of a right way for further development. Combination of Semantic Web and Web Services concepts may address many of difficulties of existing technology. It is not a question of whether Semantic Web is coming or not, but a question of when it will come. However without mature standards, proof and actually working industrial cases Semantic Web has small chances to be adopted by industry. In this paper a survey of Web Services recent needs is made, state of the art of Semantic Web technology is discussed in the context of EAI and e-business solutions. Some new challenges brought by Semantic Web were observed and the industrial maintenance case of these challenges implementation was considered.

1. Introduction

The general picture of Web nowadays shows that almost half a billion users are accessing more than 3 billion pages of Internet resources. Serious problems emerge in information search, extraction, representation, interpretation and maintenance because no efficient support in processing this information is provided. The possible impact of resolving problems in Knowledge Management, Enterprise Application Integration and e-commerce draws the best minds and research groups to active efforts, which will bring Web to qualitatively new level of service [Fensel & Musen, 2001].

Appearance of Web Services as a technology is tightly connected with initiatives to create e-commerce systems based on Internet and Enterprise Application Integration problem. "Web Services" term refers to available programmatic interfaces that are used in the World Wide Web for application-to-application communication.

The W3C's Metadata Activity was tightly connected with Knowledge Management problems and has grown from idea of having machine-understandable information in the Web. Metadata Activity has provided approach for metadata labeling of web content. Further, the idea has developed into the Semantic Web vision of having data-oriented web with metadata and links between resources to provide effective discovery, integration, automation and interoperability across various semantic-aware applications. The primer goal of Semantic Web Activity is development of mature comprehensive standards and technologies for future Web, provision with building blocks that will assist in addressing of critical issues concerning interoperability in the Web, and thus, Web Service technology.

Additionally, closely related to the intersection of Semantic Web and Web Services technology, is an Agent Technology. Recently it has started to draw a considerable attention of both research community and industry because of extreme importance as well as relevance of conjunction of these three technologies both to computer science and the business applications.

Since Web Service Technology built upon Semantic Web Technology makes strong promises ("Intelligent Web Services", [Fensel et al., 2002(c)]) a series of questions arise [Bussler et al., 2003]. To what extent have these different technologies already been integrated today? How does the combination of those technologies look like? How does this combination make problems like Enterprise Application Integration, Distributed Knowledge Management systems development, easier to solve and the solution more reliable?

The objectives of this work are divided into groups:

1. Analysis of current state of Web Service and Semantic Web technology;
2. Problems of Web Services and Semantic Web as an approach; Intelligent Web Services concept.
3. Challenges and technical issues regarding semantic-aware services. What kind of work has to be done and which tools are required? What are the first steps towards Semantic Web enabled systems in industry?

The rest of the paper is organized as follows. Chapter 2 contains survey of existing technologies and standards around Enterprise Application Integration, Web Services and Semantic Web. Chapter 3 covers questions about problems of Web Services and about Semantic Web enabled solutions. Analysis of introduced by Semantic Web benefits and challenges are presented and comparison between traditional and semantic-enabled (via Semantic Web) technologies is given. Chapter 4 contains brief description of *OntoServ.Net* framework being developed for industrial maintenance services network as a real case. Conclusions are in chapter 5.

2. EAI, Web Services and Semantic Web

EAI solutions provide an integrated approach to connection of the different components of IT infrastructure: people, applications, platforms and databases to enable secure, intra- and inter-enterprise collaboration. EAI enable an organization to integrate business processes internally and externally and allows creating dynamic environments that support business requirements, thereby creating a global organization. The EAI architecture provides services such as application development tools, repository management, routing, publish/subscribe services, data flow, data transformation, security services, recoverability and workload balancing.

In a broad meaning, *web services* belong to a model in which tasks within e-business processes are distributed and accessible throughout a global network. From another point of view, web services are a stack of emerging standards that describe service-oriented, component-based application architecture. Web Services connect computers and devices with each other using the Internet to exchange data and combine it in new ways. Web Services can be defined as software objects that can be assembled over the Internet using standard protocols to perform functions or execute business processes. The key to Web Services is dynamic service composition using independent, reusable software components. [Fensel & Bussler, 2002].

Main Layers of the Web Services Computing Stack ([Sycara, 2003]) are as follows:

SOAP (Simple Object Access Protocol)**.** SOAP [Mitra, 2003] is an XML based lightweight messaging protocol intended for exchanging structured information between applications in a decentralized, distributed environment.

WSDL (Web Services Description Language). WSDL provides description of connection and communication with a particular web service [Sankar et al., 2003].

*UDDI***.** It [UDDI] stands for Universal Description, Discovery and Integration and represents a set of protocols and was directed to providing of public directory for the registration and real-time lookup of web services and other business processes.

E-Speak is an example of service architecture developed by Hewlett-Packard. The goal of e-Speak is to perform transaction between e-services. E-speak engines run on participating client machines and e-speak service platforms that can exchange XML based information to solve problem of integration of simple services into more complex ones [Sliwa, 2002].

ebXML (http://www.ebxml.org/) stands for Electronic Business XML. It is a project to standardize the exchange of business data. The core infrastructure specifications of ebXML are the messaging service (ebMS specification), the registry and repository (ebRS specification), and the collaborative partner protocol (ebCPP specification). The ebXML Framework allows a Trading Party to express via CPP supported Business Processes and Business Service Interface to other ebXML compliant Trading Parties.

RosettaNet (http://www.rosettanet.org) is a consortium of the world's leading companies in the fields of electronics, IT-sector, semiconductor manufacturing and solution providers. RosettaNet is dedicated to creation, implementation and promotion of open e-business standards. The ultimate goal of RosettaNet is development of standards for common e-business language and open e-business processes, aligning processes between trading partners, which will provide measurable benefits to the evolution of the global, high-technology trading network.

Semantic Web [Berners-Lee et al., 2001] is the presentation of machine-processable semantics of *data* on the Web. It is a collaborative effort led by W3C Consortium with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework (RDF) and new web languages such as Web Ontology Language (OWL), DARPA Agent Markup Language (DAML), which integrate a variety of applications using XML for syntax and URIs for naming. RDF and RDF Schema provide basic features for information modeling and a simple knowledge representation mechanism for Web resources. DAML+OIL is an ontology description language manifested as RDF Schema extension for expressing far more sophisticated classifications and properties of resources than RDFS [Connolly et al., 2001]. The newest part of the growing stack of W3C recommendations related to the Semantic Web is Web Ontology Language (OWL), which been designed to meet needs for a Web Ontology Language and incorporates lessons learned from the design and application of DAML+OIL [Dean et al., 2002].

3. Semantic Web Enabled Web Services

The next-generation Web Services will transform the web from static content, human-oriented and dependent e-services to a distributed computational system in which intelligent web services complemented by scalable mediation infrastructure to bring on top the performance of the Web. To facilitate full potential of Web Services, appropriate framework is about to be developed [Fensel & Bussler, 2002]. The emerging concept of Intelligent Web Services is shown in Fig. 1.

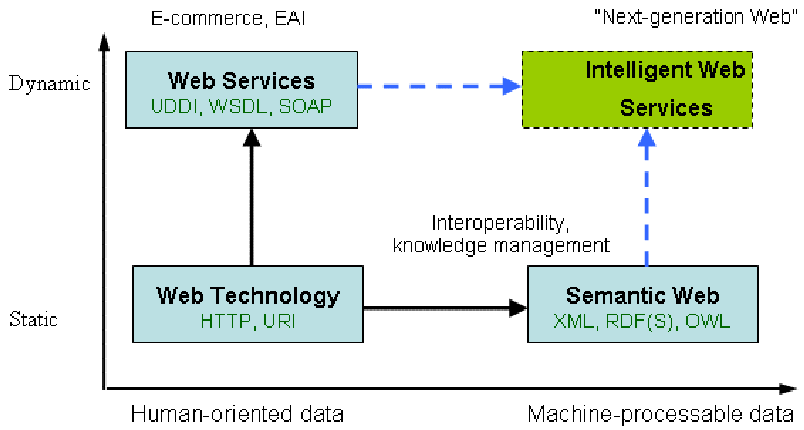

Fig.1. Bringing Web on top of performance with Intelligent Web Services (adopted from [Fensel et al., 2002(c)])

The main objectives of Semantic Web enabled Web Services development:

- Provide a comprehensive Web Service *description* framework.
- Define a Web Service *discovery* framework.
- Provide a scalable Web Service *mediation* platform.

3.1. Requirements to Web Services

Web Services technology nowadays is based on UDDI and WSDL which do not make any use of semantic information, hence, failing to meet the problem of matchmaking between provided capabilities of services and service requestors' needs [Sycara, 2003]. This sought functionality cannot be achieved just on a basis of keyword searches and vocabularies of service types.

But still, though they only partially address requirements sought by the Web Services vision, some lessons have been learnt from UDDI-WSDL-SOAP design. In [Fensel & Bussler, 2002] necessary to scalable web service discovery, mediation and composition elements were identified as:

- *Document types,* which describe the content of business documents.

- *Semantics,* which is introduced as semantic descriptions to be interpreted correctly by the service requesters and providers.
- *Transport binding,* which is an agreement between service requestor and service provider on the transport mechanism to be used for service requests.
- *Exchange sequence definition,* which is transport-level communication protocol to follow in inherently unreliable data communication networks.
- *Communication process definition,* a manifestation of business logic in terms of the business messages exchange sequence.
- *Security.* Data contained in the messages between service requester and service provider should be private and unmodified as well as non-reputable.
- *Syntax.* Documents can be represented in one of syntaxes available.

In UDDI only Transport binding, Exchange Sequence Definition and Communication Process Definition elements' requirements are partially fulfilled via general UDDI architecture, SOAP and WSDL, and provide limited support in automated service recognition and comparison, configuration, combination and automated negotiation. In addition to UDDI, WSDL and SOAP, there are standards such as WSFL, BPSS, XLANG, ebXML, BPML, WSCL and BPEL4WS, WS-Security and WS-Routing, which are intended to fill up other parts of the stack**.** But they are numerous, overlap each other in addressed problems and have been developed by individual web-services industry players (like IBM, Microsoft, HP and etc.) often for own innovations. It is evident that consistent solution cannot be achieved without combined efforts of industrial leaders and research communities.

3.2. Service Description Framework

Management of resources in Semantic Web is impossible without use of ontologies, which can be considered as high-level metadata about semantics of Web resources [Fensel et al., 2002(d)]. DAML-S is an upper ontology for describing properties and capabilities of Web Services. DAML-S provides an unambiguous, computer interpretable markup language, which enables automation of service use by agents and reasoning about service properties and capabilities [Ankolenkar et al., 2001].

Approaches to defining things followed in RosettaNet and ebXML frameworks are very alike to that in DAML-S. The differences are in the extent and specific of described process. DAML-S follows Semantic Web's line and uses ontology as a foundation for every description. RosettaNet and ebXML are e-business oriented frameworks whereas DAML-S stays aside of any specific service domain. The strength of DAML-S based service description is in adopted from Semantic Web having ontology as the schema for metadata provided. ebXML's meta-models are similar to ontology used in DAML-S, though in less general sense and they are dedicated mostly to business process description; tModel, vocabularies and dictionaries in UDDI, e-Speak and RosettaNet are more schemas for description rather then basis for semantic annotation of web services. ebXML and RosettaNet e-commerce frameworks are given here to admit that proposed by DAML-S expressiveness is potential enough to become an important part of semantic-enabled Web Services that will be essential part in EAI, e-business. Fig. 2 depicts the relations between DAML-S, WSDL, UDDI, RosettaNet, ebXML and related frameworks/languages.

UDDI doesn't provide facilities for service descriptions except keyword and industrial service type categorization. Without sharing common definitions and understanding of the concepts, without shared metadata and semantics associated with particular web service, an interaction between UDDI client and web service cannot be performed in the correct manner. Because DAML-S provides no framework for discovery (just syntax for descriptions) and UDDI has a lack of description potential, that make some minds thinking over extension of possibilities proposed by UDDI with DAML-S [Paolucci et al., 2001] to get the best of the two worlds: support from the popular industry standard framework and expressiveness from the Semantic Web.

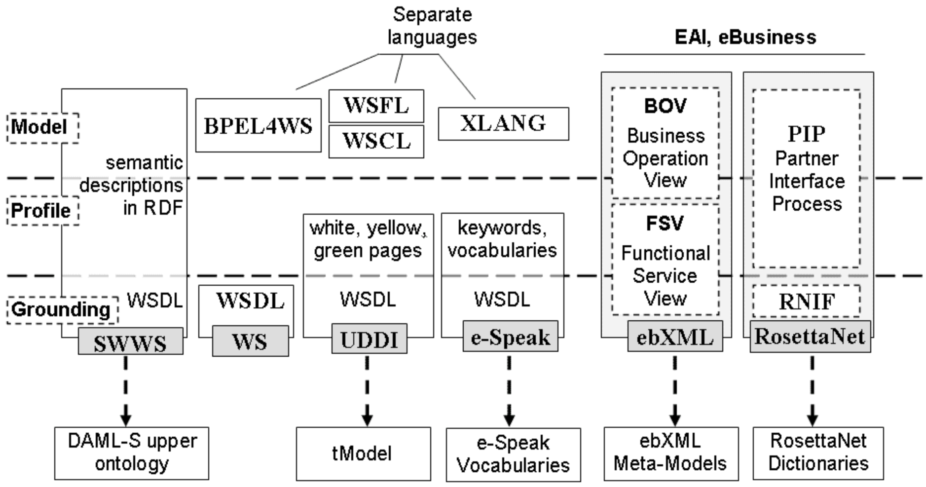

Fig. 2. Technologies and description languages concerning Web Services. Correspondence to *ServiceModel, ServiceProfile* and *ServiceGrounding* parts of DAML-S service description

Will DAML-S become a substantive for WSDL in UDDI framework or new mediation framework initially designed to be semantic-aware will be developed, depends on market, solution providers and adoption of Semantic Web approach.

3.3. Requirements to Service Description

Each of popular frameworks (e-Speak, UDDI, ebXML, RosettaNet) uses own mechanism to make descriptive advertisements about services. The basic requirements to service description language in [Trastour et al., 2001], formulated as:

Requirement 1: High degree of flexibility and expressiveness

Requirement 2: Ability to express semi-structured data

Requirement 3: Support for types and subsumption (categorization)

Requirement 4: Ability to express constraints

If to consider these requirements and comparing proposed by Semantic Web ontological descriptions (written in DAML-S) with other mentioned frameworks, the following conclusions become clear:

- in DAML-S: RDF layer as a representation basis covers requirements 1 and 2, RDFS layer covers 3rd and refines 1st , whereas DAML layer meets 4th requirement;
- in UDDI tModels have no classification or data structures organization (Req. 3);
- tModels only provide a tagging mechanism and only a first level filter is provided. Further discrimination is done in communication with service provider (see Req. 2);
- searching is only done by string equality matching on some fields such as name, location or URL (see Req. 3 and 4);
- there is no possibility to extend the description schema (see Req. 1 and 2).
- ebXML framework is very focused on defining business processes and business documents payload; the Core Component vocabulary meta-model does not look very rich and do not provide support for semi-structured data (Req. 1), inheritance (Req. 3) and constraints (Req. 4);
- neither e-Speak, nor RosettaNet seem to provide anything beyond a basic ontology definition (vocabularies and dictionaries can be seen as primitive ontologies with limited capabilities).

ProcessModel in DAML-S description of service provide description of workflow within service. There are at least two reasons for that. The first reason is to enable monitoring of service execution stages; this can be used for complex transactions management with many services involved, where execution of services can be stopped due to some conditions. The second reason is to provide additional service semantic that will be used for better service matching. For instance, if specified that service at first "makes lookup" in "address database" repository for text from inputs annotated as "person name", and then find "map" for "address", then this service has more chances to be used than service with "person" as input and "map" as output if relation between "person" and "map" is not specified as "map-of-the-location-of-the-person" Note, that in traditional UDDI/WSDL description we'd have description that service has string as input and string (url) as output, and also keywords "location map person" that describe service along with some unique service identifier (tModel) which has to be known to service requestor a priori to use it for service discovery.

Hence, DAML-S provides better the means for a web service to advertise its functionality to potential users of the service. The detailed process description of the service enriched with ontology features, thus leads to more accurate matchmaking.

RDF-based serialization (RDF vs. SOAP). SOAP message consists of the SOAP envelope for expressing *what* is in a message; *who* should deal with it, and *whether* it is optional or mandatory. The SOAP encoding rules define a serialization mechanism and a convention that can be used to represent remote procedure calls and responses.

SOAP standard matches perfectly initial idea of exchange instances of application-defined data types in heterogeneous distributed environment, but there are some limitations of SOAP to be a base standard of universal messaging framework for Web Service technology:

- SOAP message formats are provided as a part of higher level standards, e.g. WSDL, hence communication requires a-priori agreement between Web Services on message format and protocol;
- SOAP standard has no communicative speech acts: there is no way to determine intention of the message sender or what the message trying to achieve (semantic of message is not introduced explicitly).

From the point of view of Semantic Web enabled Web Services approach, SOAP is not suitable as container language for semantic-aware mediation since it, first, has no semantic and, second, scores low on possibility to be used in situation when there is no a priori message format are defined.

It is possible to use RDF payload in SOAP (as a first step from SOAP to RDF messaging) or even SOAP-less pure-RDF messaging system. Corresponding ontology support and mediation framework are required. RDF can be chosen as a messaging language for Web Services because:

- it is not *structure-oriented* as SOAP, but *semantic-oriented*; there is a resource description model behind the RDF which binds assertions (RDF statements) in the message to ontology and there is XML Schema behind SOAP which only restricts XML structure of the message;
- it is easy to parse (as easy as SOAP since both are XML based), less strict, since statements' order in RDF is not important, and more flexible, since parts of RDF can represent virtually any kind of message;
- it supports knowledge representation for service description and any other asserts (e.g. about preferences, security etc.), allowing inference on such information;
- it will be widely used for resources description and developed tools will be reused for web service if appropriate web service ontology exist;
- RDF and ontologies in Semantic Web are going to be universal semantic description framework and their adoption will be a crucial point in the future knowledge management technologies, so accepting it in advance is reasonable.

From above statements two conclusions become obvious:

1) SOAP needs semantics "injected" in it or to be superseded by another semantic-enabled standard;
2) RDF and "mediation" ontology for Web Services are possible substitutes proposed by Semantic Web.

Of course, it is clear that Semantic Web Services are harder to build comparatively to SOAP services. Especially because there are already powerful tools developed for traditional services technology that supports SOAP (like Microsoft .NET, for instance), but no any more advanced then pilot implementation in some projects tools for Semantic Web software development. And it's clear why it is so; Semantic Web and future technologies with it are being just developed, there is a gap between ideal and reality, but it will be filled soon [Ohlms, 2002]. Semantic Web Services require efforts at the outset, but make it more likely that services will stay longer and play well with others.

3.5. Service Composition

Composition of web services that have been previously annotated with semantics and discovered by a mediation platform is another benefit proposed by Semantic Web for Web Services. Composition of services can be quite simple sequence of service calls passing outputs of one service to the next and much more complex, where *execution path* (service workflow) is not a sequence but more sophisticated structure, or intermediate data transformation is required to join outputs of one service with inputs of another. Within traditional approach such service composition can be created but with limitations: since semantics of inputs/outputs is not introduced explicitly, the only way to find matching service is to follow data types of its inputs and/or know exactly what service is required. This approach works for simple composition problem but fails for problems required for the future Web Services for e-commerce.

As an example of composition, suppose there are two web services, an on-line language translator and a dictionary service, where the first one translates text between several language pairs and the second returns the meaning of English words. If a user needs a *FinnishDictionary* service, neither of these can satisfy the requirement. However, together they can (the input can be translated from Finnish to English, fed through the English Dictionary, and then translated back to Finnish). The dynamic composition of such services is difficult using just the WSDL descriptions, since each description would designate strings as input and output, rather than the necessary concept for combining them (that is, some of these input strings must be the name of languages, others must be the strings representing user inputs and the translator's outputs. To provide the semantic concepts like language or French, we can use the ontologies provided by the Semantic Web.

Service composition can also be used in linking Web (and Semantic Web) concepts to services provided in other network-based environments [Sirin et al., 2002]. One example is the sensor network environment, which includes two types of services; basic sensor services and sensor processing services. Each sensor is related to one web service, which returns the sensor data as the output. Sensor processing services combine the data coming from different sensors in some way and produce a new output. These sensors have properties that describe their capabilities, such as sensitivity, range, etc., as well as some non-functional attributes, such as name, location, etc. These attributes, taken together tell whether the sensor's service is relevant for some specific task. An example task in this environment would involve retrieving data from several sensors and using relevant fusion services to process them via SOAP calls. As an example, the data from several acoustic and infrared sensors can be combined together and after applying filters and special functions, this data may be used to identify the objects in the environment. In this setting, we need to describe the services that are available for combining sensors and the attributes of the sensors that are relevant to those services. More importantly, the user needs a flexible mechanism for filtering sensor services and combining only those that can realistically be fused.

In DAML-S *ServiceGrounding* part of service description provides knowledge required to access service (where, what data, in what sequence communication goes) and *ServiceProfile* part provides references to the *meaning* what service is used for. Both these pieces of information are enough (as it supposed by Semantic Web vision) to be used by intelligent mediator (intelligent agent, mediation platform, transaction manager etc.) for using this service directly or as a part of compound service.

The implementation of service composer [Sirin et al., 2002] have shown how to use semantic descriptions to aid in the composition of web services-- it directly combines the DAML-S semantic service descriptions with actual invocations of the WSDL descriptions allowing us to execute the composed services on the Web. The prototype system can compose the actual web services deployed on the Internet as well as providing filtering capabilities where a large number of similar services may be available.

4. Industrial Case of Semantic Web Enabled Web Services Application

We are developing a framework for industrial semantics-enabled maintenance services organized in peer-to-peer network of services platforms embedded into maintained devices and specific maintenance centre nodes. *OntoServ.Net* (see Fig. 3) is based on Web Services and Semantic Web technologies and meant to provide solution for building large-scale industrial maintenance networks.

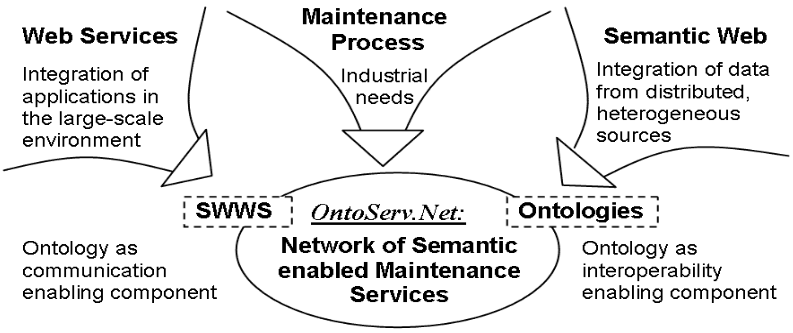

Fig. 3. OntoServ.Net concept

4.1. Industrial Maintenance

Maintenance of complex industrial machines such as paper-machines, mills, turbines, etc. is a complicated and important task. Maintenance activities include condition monitoring, preventive maintenance, tuning, repair works. Unlike condition monitoring systems, preventive maintenance is directed to analysis of current device state with the object to reveal some *possible* (not detected post facto) emerging problems, thus preventing failures via adjustment of parameters, change of parts, tuning etc. beforehand, and it leads to lower expenses for device maintenance (because failures can damage devices very hard sometimes) More advanced techniques are used for prediction of faults. To recognize some dimensions of the device state and derive useful patterns from this information, which can be considered as “symptoms” of the device “health” historical data, online learning and prediction techniques are used within preventive maintenance activities.

A major goal of the maintenance process is to perform the most appropriate maintenance procedures (not just repair works, but preventive maintenance activities also) at the right moment in the most efficient way and in the shortest time. Since maintenance-related processes rely on relevant information, comprehensive and timely information delivery to the individuals involved in the maintenance can significantly benefit the process. This makes automated maintenance system, which can integrate maintenance-related information from many sources, highly desired in order to give appropriate maintenance support. The typical lifecycle of maintenance activities is shown in Fig. 4.

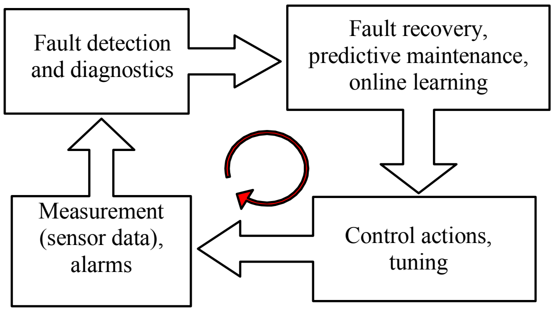

Fig. 4. Lifecycle of maintenance activities

4.2. Principles of OntoServ.*Net*

Growing interest to machines with embedded intelligent maintenance capabilities lead to a special kind of industrial products – smart-devices (machines). The expectations from smart devices include advanced diagnostics and predictive maintenance capabilities. The concerns in this area are to develop a diagnostics system that automatically follows up the performance and maintenance needs of field devices offering also easy access to this information.

Field Agent concept is used for a software component that automatically follows the "health" of field devices. Field Agent component is autonomous and communicates with its environment and other Field Agents; it is capable of learning new things and delivering new information to other Field Agents. It delivers reports and alarms to the user by means of existing and well-known technologies such as intranet and e-mail messages.

Easy on-line access to the knowledge describing field device performance and maintenance needs is crucial. There is also growing need to provide automatic access to this knowledge not only to humans but also to other devices, applications, expert systems, agents etc., which can use this knowledge for different purposes of further device diagnostics and maintenance. Also the reuse of collected and shared knowledge is important for other field agents to manage maintenance in similar cases.

In any case history data, derived patterns and diagnoses can be stored and used locally however there should be a possibility to easy access this information and also to share it with other maintenance platforms for reuse purposes.

Appropriate field agents should communicate with each other (e.g. in peer-to-peer manner) to share locally stored online and historical information and thus to improve the performance of the diagnostic algorithms, allowing even the co-operative use of heterogeneous field devices produced by different companies, which share common communication standards and ontologies. Maintenance centres supported by machine manufacturers or by some other parties will provide entry points to a maintenance network and play role of mediator of the maintenance networking (see Fig. 5). Communication between nodes in the maintenance network is to be built as web services communication. Maintenance centers mediate such communication providing service discovery capabilities and provide own services that can compose web services to deliver complex ones to embedded maintenance platform of smart-devices.

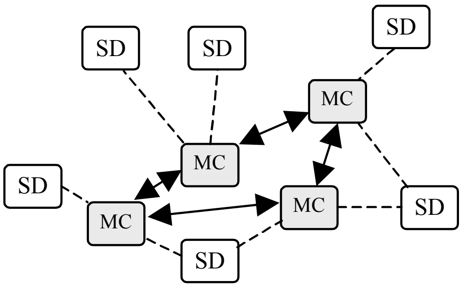

Fig. 5. Smart-devices and maintenance centers in OntoServ.Net

Ontological descriptions in OntoServ.*Net* play role of enabling technology that will provide efficient service discovery and automated services use in such environment. DAML-S will be used for web services descriptions. RDF serialization of data is to be used. Most of interactions is done in form of *semantic queries*, so appropriate communication ontology is required for exchanging such queries and other communication messages.

In order to provide interoperability in information exchange between nodes in OntoServ.*Net*, passed data has to be annotated using some common ontology for all nodes this data will be delivered. Since virtually any part of embedded maintenance platform can use network resources (access maintenance web services and provide own services), it is required to have data annotated immediately after its creation and process it with semantic-aware applications in the embedded platform. General maintenance process follows structure and data flows as it is shown in Fig. 6.

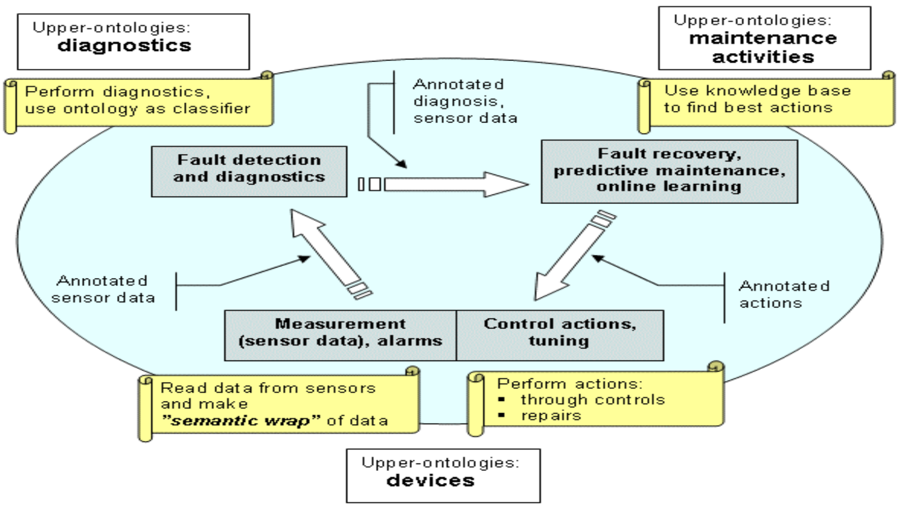

Fig. 6. Schema for ontological support for automated maintenance system

Major ontology providers of the network (groups of manufactures) organize ontology management in OntoServ.Net. If there are several groups (as it is naturally so) ontology mapping between different *"maintenance contexts"* have to be supported by them when appropriate.

5. Conclusions

One of Semantic Web promises is to provide intelligent access to the distributed and heterogeneous information and enable mediation via software products between user needs and the available information sources.

Web Services technology resides on the edge of limitation of the current web and desperately needs advanced semantic provision oriented approach. At present, the Web is mainly a collection of information and does provide efficient support in its processing. Also the promising Web services idea to allow services to be automatically accessed and executed has no yet facilities to efficiently discover web services by those who need them. All service descriptions are based on semi-formal natural language descriptions and put limits to find them easily. Bringing Web services to their full potential requires their combination with approach proposed by Semantic Web technology. It will provide automation in service discovery, configuration, matching client's needs and composition**.** Today there are much less doubts both in research and development world, than few months ago, whether Semantic Web approach is feasible within at least 5-10 years.

The importance of Web services has been recognized and widely accepted by industry and academic research. However, the two worlds have proposed solutions that progress along different dimensions. Academic research has been mostly concerned with expressiveness of service descriptions, while industry has focused on modularization of service layers for usability in the short term [Sollazzo et al., 2002].

Web services technologies are rapidly changing, and a long list of additional features and functionality is required to complete the vision. The basic Web services standards (SOAP, WSDL, and UDDI) are immediately useful for many applications, such as publishing interfaces to automated business processes, bridging disparate software domains, and connecting wireless clients to Web functions [Newcomer, 2002]. With UDDI, SOAP, WSDL, industry has made a bold move forward and started initiatives that target the potential benefits of Web services. In contrast to the industry academic research has investigated languages that offer extensible ontology frameworks for the Semantic Web services. The benefits of the integration include increased visibility of Web services, because open ontology frameworks allow for semantically expressive advertising on the Web that may be found by Web crawlers.

After looking at the industry standards, it is obvious that further work is required in following areas:

- Providing a comprehensive Web Service description framework that includes service modeling (it seems to be a weakness in DAML-S).
- Establishing a tight connection to industrial efforts like XML, RDF, WSDL, WSFL and research efforts like, DAML+OIL, OWL, DAML-S etc., popularization of Semantic Web approach in the industry and, finally, pointing to industry needs for new technologies and that they are available soon.
- Defining a Web Service discovery framework that goes beyond simple registration means (like UDDI) and provides advanced ontology-based and metadata driven service discovery (DAML-S currently provides means but doesn't specify how to do it).

- Providing a scalable Web Service mediation framework that is fundamentally based on the P2P approach in order to provide direct connectivity between service requestors and service providers (at the moment only centralized architectures are in use). This framework also includes means for configuration, composition and negotiation.
- Investigate recent trends around the semantic web and web services and their potential in scientific terms (more researches in ontology management and services composition, also, policing issues of the *Web Of Trust*)
- Building a large core consortium for Semantic Web and Web Services related challenges to provide stable standardization process.

A large amount of work exists around this problem that has not found yet its way into real applications and industry. Further efforts have to be concentrated in:
- exploring and extending Semantic Web technology;
- resolving the bottlenecks of Semantic Web technology;
- bring the latest Semantic Web technology to industry;
- applying and improving the existing Semantic Web technology in the real-life applications.

Acknowledgements. Authors are grateful to Dr. Jouni Pyotsia and his colleagues from Metso Corporation and Metso business units for useful consultations and materials. Also we would like to thank our colleagues from Industrial Ontologies Group (Oleksiy Khriyenko and Andriy Zharko) for useful discussions within the scope of this paper.

References

[Ankolenkar et al., 2001] A. Ankolekar, M. Burstein, J.R. Hobbs, O. Lassila, D.L. Martin, D. McDermott, S.A. McIlraith, S. Narayanan, M. Paolucci, T.R. Payne, K. Sycara, DAML-S: Web Service Description for the Semantic Web.

[Berners-Lee et al., 2001] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American, May 2001.

[Bussler et al., 2003] C. Bussler, D. Fensel, L. Franzens, Semantic Web Services and Their Role in Enterprise Application Integration and E-Commerce, Special Issue of the International Journal of Electronic Commerce (IJEC), 2003.

[Connolly et al., 2001] D. Connolly, F. van Harmelen, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, L.A. Stein, DAML+OIL (March 2001) Reference Description, W3C Note 18 December 2001.

[Dean et al., 2002] M. Dean, D. Connolly, F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, L.A. Stein, eds., Web Ontology Language (OWL) Reference Version 1.0, 12 Nov 2002. This W3C Working Draft is available at http://www.w3.org/TR/2002/WD-owl-ref-20021112/.

[Fensel & Bussler, 2002] D. Fensel, C. Bussler, The Web Service Modeling Framework WSMF, In: White Paper and Internal Report Vrije Unversiteit

Amsterdam, 2002, available at www.cs.vu.nl/ swws/download/wsmf.paper.pdf, 2002.

[Fensel et al., 2002(a)] D. Fensel, C. Bussler, A. Maedche, Semantic Web Enabled Web Services, In Proceedings of the International Semantic Web Conference 2002, LNCS, Springer, pages 1–2, 2002.

[Fensel et al., 2002(b)] D. Fensel, C. Bussler, Y. Ding, V. Kartseva, M. Klein, M. Korotkiy, B. Omelayenko, R. Siebes, Semantic Web Application Areas, In: Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems, Stockholm - Sweden, June 27-28, 2002.

[Fensel & Musen, 2001] D. Fensel, M.A. Musen, The Semantic Web: A Brain for Humankind, Intelligent Systems, IEEE, March/April 2001.

[Mitra, 2003] N. Mitra (ed.), SOAP: Version 1.2, W3C Recommendation, 7 May 2003, URL: http://www.w3.org/TR/soap12-part0/.

[Newcomer, 2002] E. Newcomer, Introducing Web Services, Addison Wesley, May 31, 2002, URL: http://www.iona.com/devcenter/ NEWCOMER. ch01.qk.pdf.

[Ohlms, 2002] C. Ohlms, A Perspective on the Market Adoption of Semantic Web Technologies, In Proceedings of AIK-Symposium, 2002.

[Paolucci et al., 2001] M. Paolucci, T. Kawamura, T.R. Payne, K. Sycara, Importing the Semantic Web in UDDI, In: Proceedings of Web Services, E-business and Semantic Web Workshop, 2001.

[Sankar et al., 2003] K. Sankar, K. Liu, D. Booth, eds., Web Services Description (WSDL) Version 1.2: Primer, World Wide Web Consortium, 3 March 2003. The editors' version of the Web Services Description Version 1.2: URL: http://www.w3.org/2002/ws/desc/.

[Sirin et al., 2002] E. Sirin, J. Hendler, B. Parsia, Semi-Automatic Composition of Web Services Using Semantic Descriptions, *Accepted to "Web Services: Modeling, Architecture and Infrastructure" workshop in conjunction with ICEIS2003*, 2002.

[Sliwa, 2002] C. Sliwa, From e-Speak to Web Services, Computerworld, December 2, 2002, URL: http://www.computerworld.com/ developmenttopics/development/webservices/story/0,10801,76207, 00.html.

[Sollazzo et al., 2002] T. Sollazzo, S. Handschuh, S. Staab, M. Frank, Semantic Web Service Architecture - Evolving Web Service Standards toward the Semantic Web, In: *Proceedings of the 15th International FLAIRS Conference*, Pensacola, Florida, May 16-18, 2002. AAAI Press.

[Sycara, 2003] K. Sycara. Autonomous Semantic Web Services, In: Proceedings of the 15th Conference on Advanced Information Systems Engineering, Velden, Austria, 2003. URL: http://www-2.cs.cmu.edu/~softagents /presentations/parisseminarcolor.pdf

[Trastour et al., 2001] D. Trastour, C. Bartolini, J. Gonzalez-Castillo, A Semantic Web Approach to Service Description for Matchmaking of Services, In: Proc. International Semantic Web Working Symposium (SWWS), Stanford, CA, USA, July 2001.

[UDDI] UDDI: The UDDI technical white paper, 2000, URL: http://www.uddi.org/.