Semantic Web Services for Smart Devices in a "Global Understanding Environment"

Vagan Terziyan

Department of Mathematical Information Technology, University of Jyvaskyla,
P.O. Box 35 (Agora), FIN-40014 Jyvaskyla, Finland
vagan@it.jyu.fi

Abstract. Various Web resources and services are usually assumed to be used and accessed by human users (current Web) or by software agents on behalf of human users (emerging Semantic Web). However industry emerges also a new group of "users", which are smart industrial devices, robots or any other objects, which can be adapted to the (Semantic) Web environment. They would need special services for e.g. online condition monitoring, information provisioning, remote diagnostics, maintenance support, etc. The goal of this paper is to specify main requirements to Web services that automatically follow up and predict the performance and maintenance needs of field devices. Semantic Web enabled services form a Service Network based on internal and external service platforms and OntoShell software. Concepts of a "Global Understanding Environment" and a "mobile service component" suppose that any component can be adapted to Semantic Web environment and executed at any platform from the Service Network, including service requestor side. This allows delivering not only a service results but also a service itself. Mobile service component within an OntoShell (agent) can move to a field device's local environment (embedded agent platform) and perform its activities locally. Service components improve their performance through online learning and communication with other components. Heterogeneous service components' discovery is based on semantic P2P search.

1 Introduction

The intersection of the following three domains have very recently started drawing enormous attention throughout academia and industry [3]:

- Web Service Technology (manifested through SOAP, WSDL and UDDI);
- Semantic Web Technology (manifested through ontology languages);
- Enterprise Integration (manifested through Enterprise Application Integration and E-Commerce in form of B2B Integration as well as B2C).

The promise is that Web Service Technology in conjunction with Semantic Web Technology (Semantic Web Services) will make Enterprise Integration dynamically possible for various enterprises compared to the "traditional" technologies, e.g. Electronic

Data Interchange or Value Added Networks. Enterprise Integration will become more reliable as well as easier to achieve without the low-level implementation problems.

The Semantic Web is an initiative of the World Wide Web Consortium (W3C), with the goal of extending the current Web to facilitate Web automation, universally accessible content, and the "Web of Trust". Current trends on Web development leading to a more sophisticated architecture: Semantic Web; Device independence; Web Services. Tim Berners-Lee [2] has a vision of a semantic web, which has machine-understandable semantics of information, and trillions of specialized reasoning services that provide support in automated task achievement based on the accessible information. Web-enabled languages and technologies are being developed (e.g. RDF-Schema, DAML+OIL, OWL, DAML-S). The success of the Semantic Web will depend on a widespread adoption of these technologies.

Management of resources in Semantic Web is impossible without use of ontologies, which can be considered as high-level metadata about semantics of Web data and knowledge. Ontologies are content theories about the sorts of objects, properties of objects, and relations between objects that are possible in a specified domain of knowledge [4]. DAML-S (DAML for Services [5]) provides an upper ontology for describing properties and capabilities of Web services in an unambiguous, computer interpretable markup language, which enables automation of service use by agents and reasoning about service properties and capabilities.

There is a growing interest in the use of ontologies in agent systems as a means to facilitate interoperability among diverse software components, in particular, where interoperability is achieved through the explicit modeling of the intended meaning of the concepts used in the interaction between diverse information sources, software components and/or service-providing software [12]. The problems arising from the creation, maintenance, use and sharing of such semantic descriptions are being highlighted by a number of recent large-scale initiatives supporting the interaction of heterogeneous systems (e.g. Agentcities, Grid computing, the Semantic Web and Web Services). A common trend across these initiatives is the growing need to support the synergy between ontology and agent technology.

Software applications can be accessed and executed via the Web based on the idea of interoperated Web services. A fundamental step toward this interoperation is the ability of automatically locating services on the bases of the functionalities that they provide. Location of web services is a semantic problem because the similarity and difference between services should be recognized on a semantic level. Current Web Services technology based on UDDI and WSDL does not make any use of semantic information and therefore fails to address the problem of matching between capabilities of services. In [13] the previous work on DAML-S was expanded, that describe service capabilities within DAML-S, can be mapped into UDDI records providing therefore a way to record semantic information within UDDI records. It was shown how to use this encoded information to perform semantic matching. Also efforts are made to improve DAML-S vision by inheriting some features from multi-agent systems community. For example in [7] it was mentioned that DAML-S approach does not separate the domain-neutral communicative intent of a message (considered in terms of speech acts) from its domain-specific content, unlike similar developments from the multi-agent systems community. To overcome this, an ontologically motivated Web Services system was presented the multi-agent systems techniques are discussed in the context of DAML Services model. The key component of the system is the central broker, which mediates the interaction between the other system components. Data consumers use the broker to find sources; the data sources register a service advertisement via broker; the broker responds with the matching services; the consumers then communicate directly with the data sources.

The key to Web Services is on-the-fly software composition through the use of loosely coupled, reusable software components [6]. Still, more work needs to be done before the Web service infrastructure can make this vision come true. Among most important European efforts in this area one can mention the SWWS (Semantic Web and Web Services, swws.semanticweb.org) project, which is intended to provide a comprehensive Web Service description, discovery and mediation framework.

Usually a Web Service is expected to be accessed by human users or by software agents or applications on behalf of human users. However there already exists and growing a new group of Web Service "users", which is smart industrial devices, robots or any other objects created by an industry and equipped by an "embedded intelligence" There is a good reason to launch special Web Services for such smart industrial devices. Such services will provide necessary online information provisioning for the smart devices, allow the heterogeneous devices to communicate and exchange data and knowledge with each other and even support co-operation between different devices. There are many questions to be answered within this research area. What is now the state of market for the smart devices and what are the trends and emerging needs of this market? Who should and who might be interested to launch appropriate Web Services? What might be a procedure, technologies, tools and resources to create such services? What kind of ontologies would be needed? In what extent Semantic Web – based approaches and technologies would be appropriate? How to manage data and knowledge of such services? How to manage multiple appropriate services? How to combine FIPA and W3C standards and apply it in Web Services to provide reasonable communication platform for smart devices?

In this paper we are trying to discuss the way of implementing emerging Semantic Web and Web services technologies to a real industrial domain, which is field device management. The goal of this paper is to specify main requirements to Web services that automatically follow up and predict the performance and maintenance needs of field devices.

The rest of the paper organized as follows. Chapter 2 describes the domain of field device management and maintenance and the concept of implementing agents in it in the framework of "Global Understanding Environment". Chapter 3 presents the main requirements to the Web service network for smart devices based on integration of Semantic Web services' and multiagent technologies. Main challenges of our approach are also summarized. Chapter 4 concludes.

2 GUN (Global Understanding eNvironment) Concept

The expectations from smart field 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. The emerging agent and communication technologies give new possibilities also in this field. Field device management in general consists of many areas of which the most important are:

- Selection
- Configuration
- Condition monitoring
- Maintenance

Valuable information is created during each phase of device management and it would be beneficial to save it into single database. This information can be utilized in many ways during the lifetime of the devices, especially as life cycle cost (or lifetime cost) of all assets is getting nowadays more and more attention. Accordingly the concept of life cycle management of assets has become very popular [14].

Field Agent is a software component that automatically follows the "health" of field devices. It is autonomous, it communicates with its environment and other Field Agents, and 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. Field device performance has a strong influence on process performance and reliable operation in more distributed process automation architecture based on FieldBus communication. In this situation, 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.

Among known cases to use agents in the maintenance domain, aircraft maintenance [17] supposes providing decision support for human mechanics. In an agent-supported process, a mechanic carries a wearable computer as he completes his maintenance tasks. When he encounters a discrepancy in his inspection, the mechanic fills out a form on his computer. The system analyzes the form and seeks out relevant information from agents. The system then displays the processed information for future use. The advantages of wearable computers with agents include automatic location and retrieval of information, utilization of historical repair data, and reduction in average time for repair.

We are considering case when (predictive) maintenance activities can be performed not only by humans but also by embedded automatics controlled by agents. We also assume that newest Semantic Web and Intelligent Web Services concepts can be applied to the problems of interoperability among field devices and will result to essential improvement of field device maintenance performance.

The concept of GUN (Global Understanding eNvironment) assumes an adaptation of every object from physical world to a Semantic Web environment. GUN Adapter is represented by integrated software/hardware components, which on the one hand implement object-specific functionalities and on the other hand – the common for whole Semantic Web environment functionalities. The Adapter translates interaction activities from device-specific format to a Semantic Web one and vice versa. Adapter also supplements real-world object with agent functionality, implicit purpose of the object becomes explicit goal of an agent (see Fig. 2).

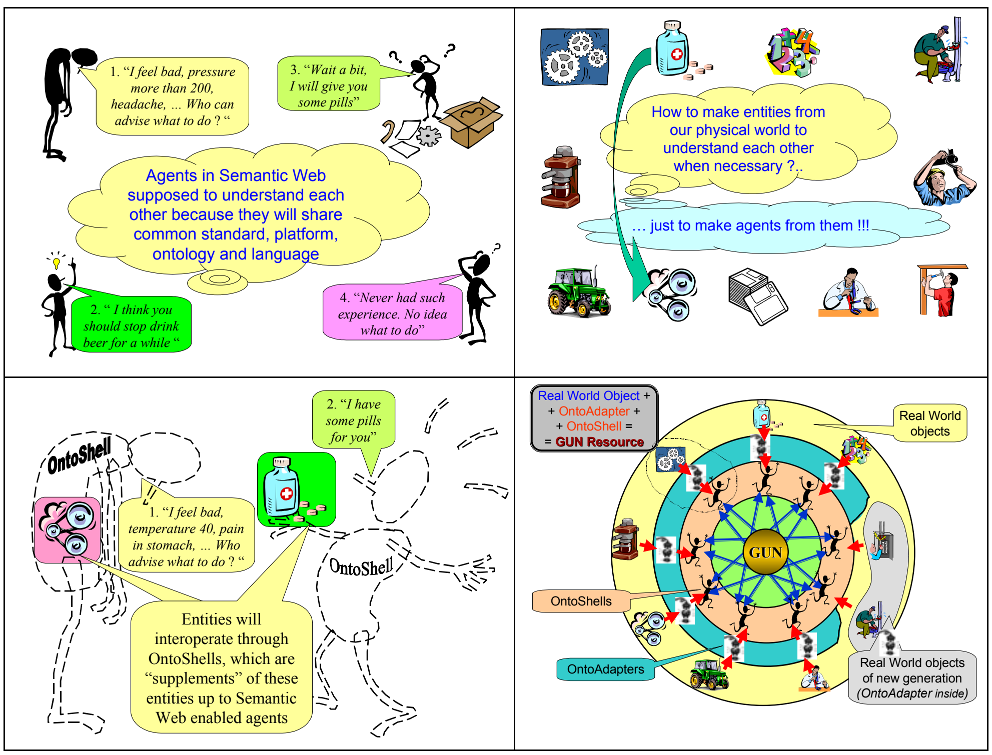

Fig. 1. The concept of Global Understanding eNvironment illustrated

The ideal GUN Adapter must adapt to a specific object automatically. The set of GUN agents can be joined into cluster (OntoShell) and the cluster will be represented for external world as a single entity. Example: industrial plant GUN agents (adaptive field devices) are joined into a cluster and other plants consider it as a single entity. As an example of implicit (“hidden”) purpose of an object we can take “pills”: they were manufactured for certain diseases and have strict application instructions. There are usually behind the scene producer and supplier of this product, some store, method, price and scope of delivery, business description etc. If to supplement the pills to the GUN agent and place it in some environment that supports such agents, then “owners” of the pills can forget about taking care of this object because an agent will take care about it. Most of present Web resources don’t have their purpose explicit: who can find it, what should be noticed, etc. An OntoShell is an active resource; an OntoAdapter supplements a passive resource with active functionalities. As a result a Semantic Web will be populated by active, goal-oriented agents. We consider a field agent concept and functionality related to field device monitoring and maintenance in the context of a GUN concept (Shells and Adapters).

Consider an example. Assume that the state of field device is described by 3 parameters and the dynamics of their change is it is shown in Fig. 2a.

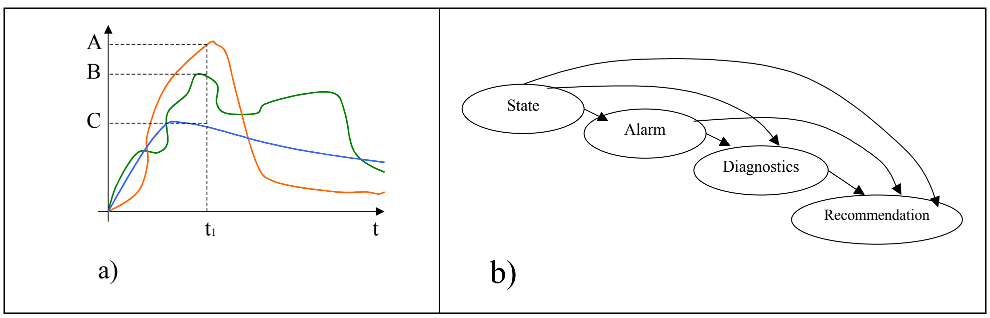

Fig. 2. The dynamics of data and its processing stages

Assume that in some time point t_1 we retrieved the values of the parameters: A, B, C. OntoAdapter integrated with the software, which provides these values, will wrap A, B and C with semantic metadata, basing on correspondent ontology of field device parameters. But what if there is some additional software that can determine the logical interval correspondent to the current value of field device parameter. The explicative example: the temperature of some field device varies in the interval from –10 C to +100 C. This interval is divided into 3 logical zones that can be recognized by alarm system: cool (-10 to +10), normal (+10 to +60), hot (+60 to +100). So, the outputs of alarm system can be also wrapped into semantic data based on additional ontology of intervals.

Fig. 2b shows possible data flows between software processing this data on 4 different levels. Alarm systems would need information about current state of field device. Diagnostic software can make decisions both on base of information about state and alarm condition.

Adapters can be really useful when many heterogeneous systems with different standards are integrated together. So there will be no need to change the components.

However it is also true that sometimes it is more economically reasonable to implement a new Semantic Web compatible system by rebuilding the components instead of developing OntoAdapters for the old ones.

According to the GUN concept, every peer in Peer-to-Peer network of GUN resources is represented by correspondent OntoShell, which generalizes and hides its internal structure. Such peer-OntoShells are interconnected with each other as neighbours forming Peer-to-Peer network.

If to assume that each OntoShell accumulates knowledge about optimal query routes (routing information is semantically enriched), then one day a group of peers can make a decision about rearrangement. Such rearrangement means formation of peers cluster: peers are joined in a group according to some common features. For instance, peers, which provide different maintenance services for Control Valves can form a cluster 'Control Valve Services'. And this cluster generalizes the features of its members generating meta-profile for a group as for a single entity. Inside the group the nodes can be rearranged into centralized topology for more efficient query routing. The duties of a central (representative) node can be delegated to some of the internal peers. This peer will be an entry point to this cluster from the outside; it will possess the functionality of an OntoShell (see Fig. 3).

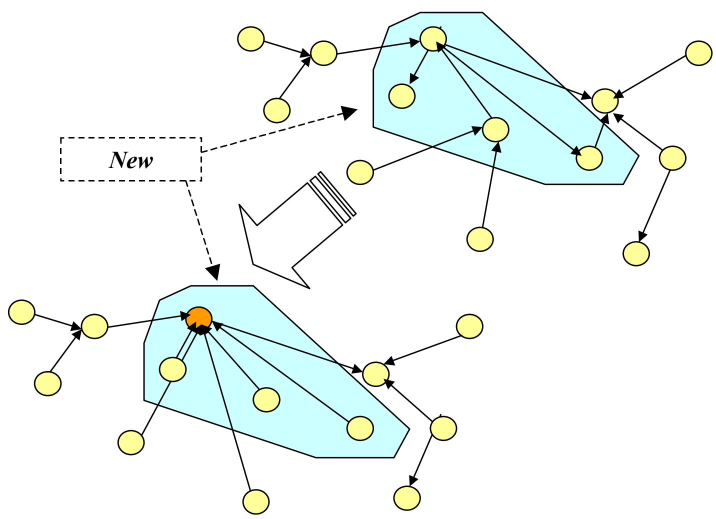

Fig. 3. Group of OntoShells is arranged in a cluster OntoShell

After formation of an OntoShell cluster a central peer can still have links with its former neighbours and even can be a member of another cluster. Such clustering of peers will reduce the unnecessary roams of queries. If the query doesn't match the profile of a cluster it wouldn't match any profile of its members. So the number of matching processes is decreased by number of members of the cluster. The challenge here is a process of generation of cluster meta-profile from the profiles of its members.

The concept of OntoShell can be also used for integration of formerly independent intranets. Each intranet has its own unique message protocol. OntoShell is "put" on each Intranet and every OntoShell translates messages from internal format into common for all OntoShells one. In this case a single node, which implements the functionality of OntoShell, must be created for every Intranet. This will be an entry point of external messages into the Intranet (see Fig. 4).

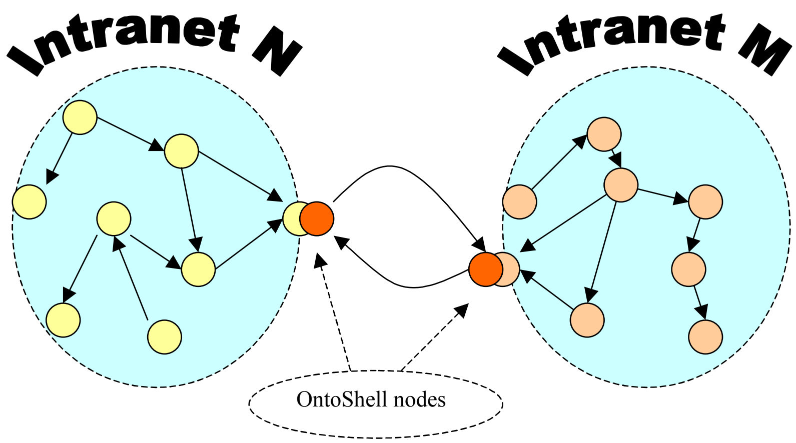

Fig. 4. OntoShells provide interoperability between Intranets

3 Network of Web Services for Smart Devices

Our primer goal is to implement the benefits of the Semantic Web (interoperability based on ontological support and semantic annotations), Intelligent Web Services (modeling, automated discovery and integration) and (Multi)Agent technologies (agents communication, coordination and mobility) according to GUN concept to substantially improve the performance of the Smart-Device Management Process by launching appropriate network of distributed intelligent maintenance services. More specifically the goal is to develop a an advanced global intelligent diagnostics system that automatically follows up and predicts the performance and maintenance needs of field devices, an appropriate multiagent architecture, ontological support for it, pilot implementation and case study.

3.1 Methods and Technologies Used

In this research we are using the most emerging technologies related to Semantic Web, Ontology Engineering, Semantic Annotations, Semantic Search, Intelligent Web Services (Modeling, Discovery and Integration), Multiagent Systems, Mobile Agents, Peer-to-Peer Networks, etc.; standards from W3C and FIPA, various Artificial Intelligence methods.

We base our efforts on intersection of Semantic Web, Web Services and (Multi)Agent Technologies, thus our approach inherits some essential properties of all of them:

- The services or service components are represented by agents and inherit following properties of *Agent Technology*: ability to learn, mobility, cooperation, coordination, FIPA standards and others. Network of services itself can be considered as multi-agent system of cooperating or self-interested, heterogeneous distributed agents;
- Inheritance from *Semantic Web Technology* results to the network of semantically annotated resources, which describe network of smart field devices that will be the subject of communication content of the agents.
- Profiles, roles and other properties of agents will be semantically annotated in a similar way as web-service profiles are annotated accordingly to *Web Services Framework* To provide interoperability of main components within such framework we will need appropriate ontological support.

3.2 Main Requirements

Service requestors are smart-devices. The class of service requestors is extended with new group of service users – smart devices. We add semantic-enabled descriptions of services to facilitate automated search and use of services by smart-devices and to enable communication between heterogeneous services and agents acting on behalf of other services or human-user agents.

Services form a Service Network based on internal and external service platforms. Service Platform is an environment for running services. Services can be provided either locally, i.e. by embedding them to smart-device internal platform, or remotely by querying them from a Web-based external platform. Such external platforms form a distributed network of Web services, which is expanded by adding internal services platforms nodes. We assume that an external service can be queried either from Web-based external platform or from another internal platform. External Web service platforms provide more rich services since they are used by many clients and quality of services can be permanently improved according to growing experience. Various interactions between service platforms (internal-internal, internal-external, external-external) can be organized as a P2P-like network.

Service components are held by autonomous intelligent agents. Because of this we can benefit from agent technologies in a following way:

Service components are mobile. Our concept of "Mobile Service Component" supposes that any service component can be executed at any platform from our Service Network, including service requestor side. This allows delivering not only a service results but also a service itself when appropriate. Mobile service component representative (agent) can move to a local environment of service consumer (embedded agent-enabled environment, agent platform) and perform its activities locally. Necessity for such kind of services and their benefits can be shown on examples of services performing actions that demand close, intensive and secure communication with client-side platform. Here are several most typical cases:

- if there is more or less permanent need to use service component then the most appropriate solution is to import service component and use it on a client side;
- activities have to be performed locally for efficiency, business or security reasons;
- heavy data flows between client and service parts (intensive communication with service better be done locally rather then in remote fashion);
- necessity to provide guaranteed service availability if access to services network cannot be guaranteed for technical reasons (e.g. for wirelessly connected peers);
- strict services response time requirements (cross-network delay is too big, so local and fast service component is required).

Integration of mobile data components to create compound document was used in [15], where document is being dynamically composed of mobile agents and can migrate itself over a network as a whole, with all its embedded agents. The key of this framework is that it builds a hierarchical mobile agent system that enables multiple mobile agents to be combined into a single mobile agent. In our approach we are also using mobile components but for creation of a compound service for smart devices.

Service components are able to learn. Service components can improve own performance through online learning during execution on service requestor site or through communication with other services components.

Service components are "Semantic Web Enabled":

- Service components are semantically described;
- Service components are retrievable as a result of semantic search;
- Service components make decisions based on shared ontology;
- Service components can be integrated based on ontology.

Semantic P2P concept for service network management. The concept assumes decentralized management architectures with ontology-based information brokerage for:

- Service certification management;

- Service discovery management;
- Service responsibility management;
- Quality of Service management;
- Trust management
- Privacy and security management.

Existing security strategies are inadequate for the distributed networks, e.g. users in pervasive environments expect to access locally hosted resources and services anytime and anywhere leading to serious security risks and access control problems. In [8] a solution is proposed based on a *distributed trust management* infrastructure to provide a highly flexible mode of enforcing security in a pervasive computing environment. A Client can access the services provided by the nearest Service Manager via short-range communication. The Service Manager acts as an active proxy by executing services on behalf of any requestor.

The critical need to complement current navigational and information retrieval techniques with a strategy of information content and semantics is discussed in [10]. A scalable approach is proposed for vocabulary sharing. The objects in the repositories are represented as intentional descriptions by pre-existing ontologies characterizing information in different domains. User queries are rewritten by using interontology relationships to obtain semantics preserving translations across the ontologies.

The concept a Peer-to-Peer Semantic Web is discussed in [1]. Its realization assumes that DAML+OIL provides a specification framework for independently creating, maintaining, and interoperating ontologies while preserving their semantics, and P2P is used to provide a distributed architecture which can support sharing of independently created and maintained ontologies. The concept facilitates:

- Distributed and autonomous creation and maintenance of local ontologies,
- Advertisement (i.e., registry) of local ontologies,
- Controlled sharing of knowledge base components among users in the network,
- Ontology-driven semantic search of concepts and services,
- Knowledge discovery and exploration of inter-ontological relationships.

An emergent ad-hoc network of wirelessly connected agent platforms was investigated in [9]. Such network provides an environment for testing the multi-agent paradigm's ability to provide interoperability between heterogeneous services. The multi-agent systems are moved into small physical and computational spaces, ensuring that they can discover and communicate with each other in a reliable and maintenance-free manner. Fraggle Rock (http://sprocket.mle.ie) is the smallest node deployed in the Agentcities network and remains interoperable with other FIPA-compliant platforms.

Certification of services on one hand, online learning of a trust network between intelligent distributed service components on the other hand, and finally P2P semantic discovery of service components in wired and wireless networks is the core of our approach to manage maintenance Web services and deal with security and privacy.

3.3 Extending Existing Maintenance Management Technology

The idea of having capabilities of browsing a device internal state extended to a browsing, automatic diagnostics and recovery within a network of maintenance centers is assumed to essentially extend the existing maintenance management technology.

The specifics of automation provided for smart-devices (that includes embedded condition monitoring) results to embedding in devices such components that can detect essential deviations from normal state of the device. For example a FieldBrowser™ [11] condition monitoring system from Metso Automation can detect faulty states and send error warnings (miscellaneous alarms) to e-mail addresses or mobile phones of operators located anywhere in the world. Another example is Sensodec 6S [16], which is the integrated solution to machinery condition monitoring and runnability analysis. The system analyzes the mechanical condition of machinery components. By using sophisticated analysis tools, machine operators and maintenance staff are quickly alerted to mechanical problems in their early phases of development.

Benefits of Condition Monitoring are evident from these examples. They are in:

- Avoidance of major mechanical failures, machinery damage and unplanned downtime;
- Immediate response is prompted by the operator interface;
- Effective preventive maintenance scheduling;
- Identification of exact causes of faults making maintenance quick and decisive.

However the problem still remains, which is lack of automatic diagnostic capabilities in existing solutions. Even such browsing solutions exist only for some types of devices and developed for a certain class of supported devices each. Current technology around UDDI, WSDL, and SOAP provide limited support in mechanizing service recognition, service configuration and combination, service comparison and automated negotiation.

Taking into account the above limitations we are using in our design a maintenance infrastructure with following main types of maintenance service components (agents):

1. *Wrapper components* – for integration with device-dependent (software and hardware) resources, acts as a semantic adaptor, mediator between semantic-enabled and traditional parts of service infrastructure;
2. *Management components* – for management of maintenance activities and distributed resource allocation;
3. *Diagnostic components* – for online discovery of problems within a device based on its state parameters and ontology-based classification of these problems (component is mobile agent);
4. *Recovery components* – for automatic planning and performing appropriate maintenance activities for a discovered diagnosis (component is mobile agent).

Humans can be represented in the maintenance services network as service component agents (human-service) or as a service requestor part (user agent), or both. These kinds of agents allow integration of humans into automated services network transparently for experts' decision support in diagnostic or recovery (maintenance) activities.

3.4 Ontological Support

The minimal set of necessary ontologies for the maintenance domain includes:

- Ontology of smart-devices;
- Ontology of maintenance activities;
- Device diagnostics ontology.

Also as it was mentioned above, there is also a need of ontologies for describing service components, trust, security, responsibility, quality, etc. profiles, in a way to allow implementation of the semantic P2P management framework

4 Conclusions

The goal of this paper is to specify main requirements to Web services that automatically follow up and predict the maintenance needs of field devices. Semantic Web enabled services form a Service Network based on internal and external service platforms. Concept of a "mobile service component" supposes that any component can be executed at any platform from the Service Network, including service requestor side. This allows delivering not only a service results but also a service itself. Mobile service component carrier (agent) can move to a field device's local environment (embedded agent platform) and perform its activities locally. Service components improve their performance through online learning and communication with other components. Heterogeneous service components' discovery is based on semantic P2P search. The paper contains mostly requirements and challenges related to Web services for smart devices in a Global Understanding Environment. More research and development efforts are needed to proof some of concepts mentioned in this paper.

Acknowledgements. Author is grateful to Dr. Jouni Pyotsia and his colleagues from Metso Corporation and Metso business units for useful consultations and materials. Also I would like to thank my colleagues from Industrial Ontologies Group (Oleksandr Kononenko, Andriy Zharko and Oleksiy Khriyenko) for useful discussions within the scope of this paper.

References

1. Arumugam, M., Sheth, A., Arpinar, B.: The Peer-to-Peer Semantic Web: A Distributed Environment for Sharing Semantic Knowledge on the Web. In: Proceedings of International Workshop on Real World RDF and Semantic Web Applications, Hawaii (2002)

2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, Vol. 284, No. 5 (2001) 34–43
3. Bussler, C., Fensel, D., Sadeh, N.: Semantic Web Services and Their Role in Enterprise Application Integration and E-Commerce, March 2003, available in: http://www.gvsu.edu/ssb/ijec/announcements/semantic.doc
4. Chandrasekaran, B., Josephson, J., Benjamins, R.: What Are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems, (January/February 1999) 20-26.
5. DAML Services Coalition: Ankolekar, A., Burstein, M., Hobbs, J., Lassila, O., Martin, D., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Payne, T., Sycara, K.: DAML-S: Web Service Description for the Semantic Web. In: *The First International Semantic Web Conference (ISWC)* (2002)
6. Fensel, D., Bussler, C., Maedche, A.: A Conceptual Architecture of Semantic Web Enabled Web Services. ACM Special Interest Group on Management of Data, Vol. 31, No. 4 (2002)
7. Gibbins, N., Harris, S., Shadbolt, N.: Agent-based Semantic Web Services. In: Proceedings Twelfth International World Wide Web Conference (2003)
8. Kagal, L., Finin, T., Peng, Y.: A Framework for Distributed Trust Management. In: Proceedings of IJCAI-01 Workshop on Autonomy, Delegation and Control (2001)
9. Lawrence, J., Embedded FIPA Agents. In: Agentcities: Agent Technology Exhibition, Barcelona (2003), Available in: http://www.agentcities.org/EUNET/ID3/documents/exh_program.pdf
10. Mena, E., Illarramendi, A., Kashyap, V., Sheth, A.: OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation Across Pre-existing Ontologies. International Journal on Distributed and Parallel Databases, Vol. 8, No. 2, (2000) 223-271
11. Neles FieldBrowser™ System for Field Device Predictive Maintenance. Metso Automation Techn. Bulletin (2003) Available in: http://www.metsoautomation.com/
12. Ontologies in Agent Systems: Workshop announcement (11 March 2003) Available in: http://oas.otago.ac.nz/OAS2003
13. Paolucci, M., Kawamura, T., Payne, T., Sycara, K.: Importing the Semantic Web in UDDI. In: Proceedings of Web Services, E-business and Semantic Web Workshop (2002)
14. Pyötsiä, J., Cederlöf, H.: Advanced Diagnostic Concept Using Intelligent Field Agents. ISA Proceedings (1999)
15. Satoh, I.: Mobile Agent-Based Compound Documents. In: Proceedings of the 2001 ACM Symposium on Document Engineering, ACM Press (2001) 76-84
16. Sensodec 6C for Paper: Metso Automation Technical Bulletin (2003) Available in: http://www.metsoautomation.com/
17. Shehory, O., Sukthankar, G., Sycara, K.: Agent Aided Aircraft Maintenance. In: Proceedings of Autonomous Agents '99, Seattle, Washington (1999) 306-312