Semantic Medical Devices Space: An Infrastructure for the Interoperability of Ambient Intelligent Medical Devices

Safdar Ali, Stephan Kiefer

Abstract—**In this paper, we show our early efforts to incorporate Web Services and Semantic Web technologies into the field of medical devices and pervasive computing to build a new breed of medical devices, named *Ambient Intelligent* (AmI) medical devices. Having implemented the infrastructure suggested in this paper, named *Semantic Medical Devices Space* (SMDS), the AmI medical devices will be able to semantically interoperate not only with each other, but also with the legacy Hospital Information Systems (HISs) and Laboratory Information Systems (LISs) as well. The SMDS has been partially implemented and tested on VPA IV mobile device.**

I. Introduction

In the present electronic healthcare era, the medical devices, i.e. diagnostic devices use vendor specific proprietary protocols to communicate with other medical devices, Hospital Information Systems (HISs) and Laboratory Information Systems (LISs). Most of the times, these devices interoperate only with the HISs/LISs from the same vendor or the devices from the same vendor, and can't interoperate with other HISs/LISs or the devices from other vendor(s) running on different communication protocols standards on *ad-hoc* basis. There is one exception that almost all PACS (Pictures Archiving and Communications System) systems use DICOM (Digital Imaging and Communication in Medicine) [1] standard for the storage and communication of medical images (i.e. X-rays). Also, IEEE 1073 family protocols [2] have laid down the foundation for standardizing the communication among medical devices, but have rarely been implemented by the medical devices' manufacturers.

In more advanced scenarios, a new breed of medical devices, the *Ambient Intelligent* (AmI) medical devices will communicate seamlessly and transparently with the HISs/LISs, irrespective of their location, while taking care of the privacy of the patient's medical data. These medical devices will also be able to collect patient's medical information from heterogeneous HISs/LISs in order to provide meticulous analysis and better interpretation of the results to the intended user (i.e. patient or health professional) of the device. Such scenarios can include all Point-Of-Care (POC) driven environments such as clinical laboratories, hospitals, physicians' offices, as well as patient's self-testing at home or mobile. Moreover, these AmI medical devices will communicate with each other to transfer the measurement results of an assay in order to provide *multi-parametric* analysis functionalities. Such devices will also help to provide *pervasive healthcare* services (anytime, everywhere) to the elderly patients being monitored at home or mobile.

This paper is organized as follows: Section II gives brief background information; Section III explains the architecture suggested for the AmI medical devices; Section IV explains the application and benefits of the architecture suggested in this paper; Section V describes the interoperability among AmI medical devices and legacy HIS/LIS. Section VI explains the implementation details, while related work and conclusion are given in Section VII and Section VIII respectively.

II. Background Information

In recent years, the *Web Services* [3] technology has been emerged as a set of standards for publishing, discovering, and composing independent services in an open network. Web Services are a family of XML based protocols to achieve interoperability among different networked applications. The benefits of Web Services include *loose coupling, ease of integration* and *ease of accessibility*. Web Services involve *three* interactions, namely publishing, finding and binding. In the *publishing* operation, the service provider publishes the service to a service registry in order to make it possible for a service requestor to *find* and access it. A UDDI [4] registry is normally used for this purpose. In *finding* operation, the service requestor *retrieves* a service description by inquiring the service registry. A WSDL [5] documents is used for this purpose. In the *binding* operation, the service requestor uses the binding details in the service description to *locate, contact* and *invoke* the service at runtime. All these transport function are performed using SOAP [6].

We are inspired by the *Semantic Web* [7] technology, which helps computers and people to work better together by giving the contents *well-defined* meanings. The Semantic Web offers a united approach to knowledge management and information processing by using standards to represent *machine-interpretable* information (including Resource

This work is partially supported by European Commission within the context of SmartHEALTH IP Project, contract 016817, priority FP6-2004-IST-NMP-2; http://www.smarthealthip.com

Safdar Ali (corresponding author), Department of Health Telematics, Fraunhofer Institute for Biomedical Engineering, St. Ingbert, Germany (phone: +49-6894-980-211; fax: +49-6894-980-400; e-mail: safdar.ali@ibmt.fraunhofer.de).

Stephan Kiefer, Department of Health Telematics, Fraunhofer Institute for Biomedical Engineering, St. Ingbert, Germany (phone: +49-6894-980-156; fax: +49-6894-980-400; e-mail: stephan.kiefer@ibmt.fraunhofer.de)

Description Framework (RDF) [8] and Web Ontology Language (OWL) [9]).

The vision of *Ubiquitous Computing* was introduced in the early 90s by Weiser [10] and now is giving rise to an extensive research. While ubiquitous computing was considered as quite futuristic in the early 90s, the combined effect of advances in the areas of hardware and network technologies, and of the universal usage of Internet-based technologies and wireless phones makes ubiquitous computing almost a reality. The vision is now termed as *Ambient Intelligence* or *Pervasive Computing* to emphasize that it does not solely rely on ubiquitous computing (i.e., useful, pleasant and unobtrusive presence of computing devices everywhere) but also on *ubiquitous networking* (i.e., access to network and computing facilities everywhere) and on *Intelligent aware interfaces* (i.e., perception of the system as intelligent by people who naturally interact with the system that automatically adapts to their preference). The *distinction* between ambient intelligence and pervasive computing systems relates to the end-users who are more specifically targeted: ambient intelligent systems are aimed at consumers and are hence oriented towards infotainment applications, while pervasive computing systems target professional users and thus desktop applications [11]. Another related term that is used within the context of ambient intelligence for the devices is *Context Awareness*, which means that the devices have information about the circumstances under which they operate and react accordingly.

Universal Plug and Play (UPnP) [27] is a set of computer network protocols, which allow the devices to connect seamlessly and to simplify the implementation of networks in the homes and corporate environments. The UPnP architecture offers *pervasive* peep-to-peer network connectivity of PCs, intelligent appliances and wireless devices. The UPnP architecture supports *zero-configuration*, *invisible networking* and *automatic discovery* for a breadth of device categories from a wide range of vendors, whereby a device can dynamically join a network, obtain an IP address, announce its name, convey its capabilities upon request, learn about the presence and capabilities of other devices, and can exert control over the devices in the network. All these functionalities are performed via SOAP [6] messages.

III. Semantic Medical Device Space Infrastructure

In this section, we describe the initial results of our research, how we can achieve the interoperability among different AmI medical devices and the legacy Hospital Information Systems (HISs) and Laboratory Information Systems (LISs). Our ultimate goal is to suggest an architecture for the medical devices to make them ambient intelligent in a way that they could adapt to the changes in the environments, and could adjust themselves for the communication with other medical devices as well as other HISs/LISs, irrespective of their location. For this purpose, the idea of *Semantic Medical Device Space (SMDS)* has been introduced. SMDS is a pervasive computing infrastructure that exploits Semantic Web and Web Services technologies to enrich the medical devices with ambient intelligence and semantic interoperability capabilities. It also supports the explicit representation, expressive querying and flexible reasoning of contexts as well. The idea of context querying and reasoning has been studied and partially borrowed from *Semantic Spaces* [12], which is a pervasive computing infrastructure that exploits Semantic Web technologies to support explicit representation, expressive querying, and flexible reasoning of contexts in smart spaces. Also, a study has been investigated in [13] how the Semantic Web can solve some of the critical problems in a ubiquitous computing environment.

We propose the use of *Semantic Web Services (SWS)* [14] to expose the functionalities of the medical devices as well as the functionalities of HISs/LISs, and to resolve the interoperability issues on each end. By exposing the various functionalities as Web Services and advertising them via UPnP, our architecture supports the AmI medical devices to discover the services available in a hospital, laboratory or a clinic wherever they are physically present. Finally, the semantic descriptions of the Web Services provided by the AmI medical devices will automatically enable them to select, compose and execute the desired composite task.

Web Services are mostly described by WSDL documents, which is a nice and simple interface description language, well suited for developers and developer-class tools. WSDL is easy to integrate with popular programming languages such as C#, Java and Perl and it's really trivial to compose and invoke Web Services if someone knows about WSDL, but WSDL doesn't provide extensive and structured documentation of the service, nor does it support automatic composition. UDDI tries to cover some of this gap but it targets developers, not the end users. DAML-S [15] and its successor, OWL-S [16], supply Ontologies for describing Web Services so that they might be discovered, explained, composed and executed automatically. OWL extends the RDF Language with powerful modeling constructs sufficient for naturally describing many subject domains. Built on the metadata of RDF, OWL effectively describes all manner of web resources for both human beings and the software programs.

Fig. 1 shows the architecture of an AmI medical device, which can be used to realize the idea of SMDS. It comprises three main components, namely *Context Awareness Management (CAM), Device Access/Communication Management (DACM),* and the *Device Manager (DM)*. Detailed information about each of the components is given below.

A. Context Awareness Management (CAM)

Being a constituent part of the Ambient Intelligence, an AmI medical device must have context-awareness capability, so that it could adapt itself to the rapidly changing situations. The various types of contextual information that can be used

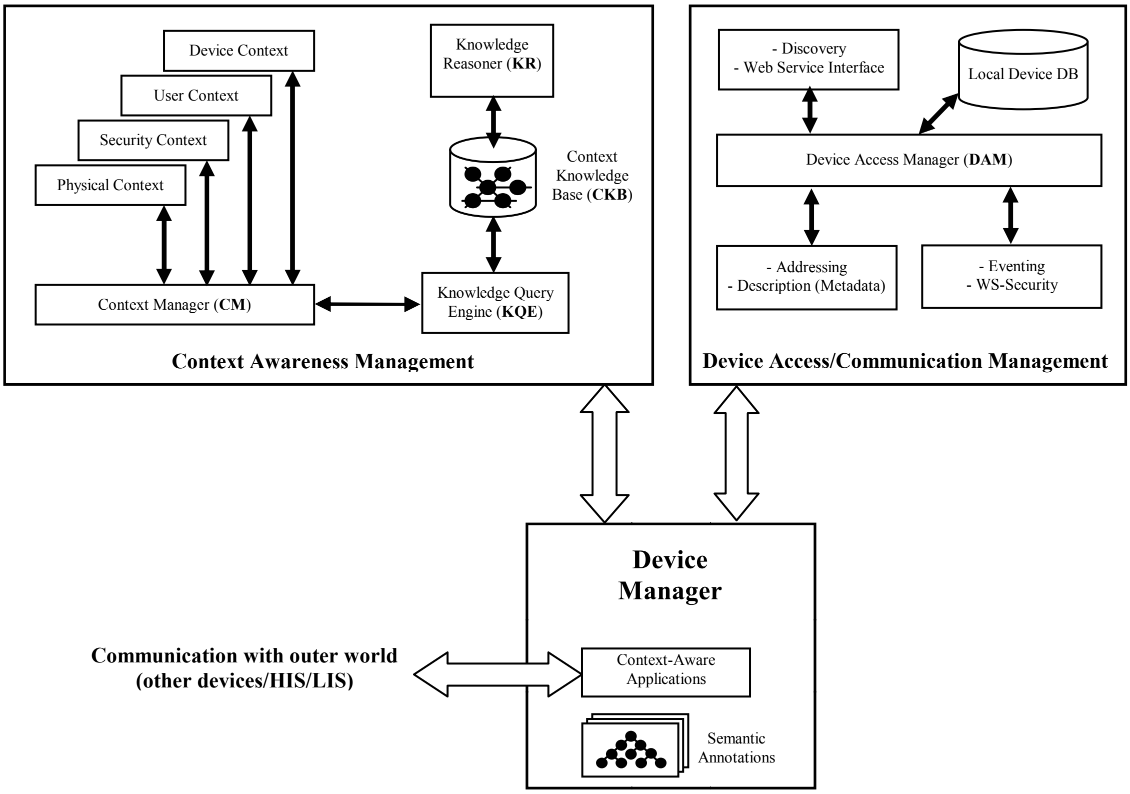

Fig. 1. Architecture of an AmI medical device.

in the environment must be well defined so that different medical devices have a common understanding of the context. Also, there must be mechanisms for the medical device users to specify how different applications and services should behave in different contexts.

The *CAM* component manages the context awareness behavior of the AmI medical device. It includes *Context Manager (CM)*, which retrieves the contextual information from the sub-components, i.e. Device Context, User Context, Security Context and the Physical Context. The *device context* provides information about the device (i.e. status, battery power etc.); the *user context* provides information about the user of the device (i.e. patient/health professional, personal prefers.); the *physical context* provides information about the present environment (i.e. hospital, clinic, laboratory, home etc.); and the *security context* provides information about the required and provided security level for a particular environment (i.e. a health professional must provide his user identity (i.e. smart card, eToken) to send or receive patient's information from/on the device etc.). These sub-components provide basic contextual information in the form of *context markups* (i.e. an RDF graph), which support the CM not only to retrieve the contexts from Context Knowledge Base (CKB) through the Knowledge Query Engine (KQE), but also to *infer* higher-level contexts, with the help of Knowledge Reasoner (KR).

The *CKB* provides persistent knowledge storage, in the form of an *extended context ontology* for a particular environment (i.e. hospital, laboratory etc.) and the context markups that are given by the users or gathered from the basic context provider components (device context, physical context etc.). The CKB links the context ontology and markups in a single *semantic model* and provides interfaces for the KQE and the KR to manipulate correlated contexts. The KQE provides an abstract interface to the CM for extracting desired contexts from the CKB. To support expressive queries, any RDF Data Query Language can be used as context query language, because it supports querying, using declarative statements, over semantic models based on triples (<**subject**, **predicate**, **object**>) patterns. The KR infers abstract, higher-level contexts from basic contextual information. Whenever, an application running on an AmI medical device needs certain higher-level contexts, it submits a set of rules to the KR, which applies them to infer higher-level contexts on the application's behalf.

B. Device Access/Communication Mechanism (DACM)

It manages different interaction patterns and behavior of the AmI medical device. The integral part of this component is *Device Access Manager (DAM)*, which provides different interaction patterns, i.e. addressing, discovery, description, controlling, and eventing; with the help of Addressing, Discovery, Description (Metadata), Web Service Interface, and Eventing components respectively. The *discovery and* *description* of the medical devices must be *semantic* in order to discover appropriate medical devices to which one device wants to communicate. Thus, we suggest that the profiles of the AmI medical devices must be described by using existing ontologies, i.e. FIPA [17] or CC/PP [18], or by further specializing these ontologies for medical devices. The FIPA ontology specifies a *frame-based* structure to describe devices, and is intended to facilitate agent communication for purposes such as content adaptation. On the other hand, CC/PP is an RDF-based framework for describing software and hardware *profiles* of the devices, specifically to facilitate the decision making process of a server, on how to customize and transfer web content to a client device in a suitable format.

The DAM stores the measurement results, performed by the AmI device, on the local device Database (DB). The DAM also uses the *semantic annotations*, available on DM, to interpret the messages received from the other AmI medical devices or HISs/LISs. These messages can include parts of *EHR* (Electronic Health Record) of the patient, measurements performed by other AmI medical devices, as well as status information, alerts, and other events information. Additionally, the DAM uses Security component to provide different security functionalities, i.e. authentication, authorization, confidentiality, and integrity.

C. *Device Manager (DM)*

DM controls both of the management components explained above. The Context Manager (CM) and the Device Access Manager (DAM) use the semantic annotations available on DM to semantically annotate the contexts and the messages respectively. The DM launches the appropriate application as per the requirement of the present context, making these applications *context-aware applications*.

IV. Interoperability of AmI Medical Devices

In advanced scenarios, the healthcare facilities (hospitals, clinics, laboratories etc.) and the patients' houses will be having complex, dynamic and *intelligent* environments around. The configuration of the medical devices must change as activities change, or as the medical devices/sensors enter and leave in the environment. In the following sections, we describe the interoperability of AmI medical devices with other AmI medical devices as well as with the legacy HISs/LISs.

As the research in wearable computing [19] is advancing, more and more applications of wearable sensors are envisaged and introduced to the health care community, such as LiveNet [20] project. In our case, suppose that a patient is wearing sensors (i.e. ECG) embedded in his/her clothes, an AmI medical device would *collect* the medical data from these sensors wirelessly and then send this data to the remote HIS. The configuration management is very challenging in such scenarios because:

1) New medical devices/sensors may enter, which have never seen before.
2) The medical devices need to discover and collaborate with other medical devices/sensors automatically.
3) The medical devices are heterogeneous and autonomous.

Before the two autonomous medical devices can interact with one another, they need to know what *interfaces* each of them supports and what *protocols* or commands they understand. In an ambient intelligent environment, this can't be known in advance. New medical devices/sensors may enter in the environment at any time, and they have to interact with the existing medical devices and the HISs. The interaction must be based on common, well-defined *concepts*, so that there is no misunderstanding between the medical devices and the HISs. The medical devices must have a common understanding of the various terms and concepts used in the interaction.

Fig. 2 shows one of the scenarios of SMDS in the field of hospitals and clinics. Two different medical devices, implemented with the architecture of AmI medical device perform distinct measurements. After the measurements are done, the results are stored locally on the devices for a particular span of time. Both of these devices offer functionalities to retrieve the measurement results, through Web Services.

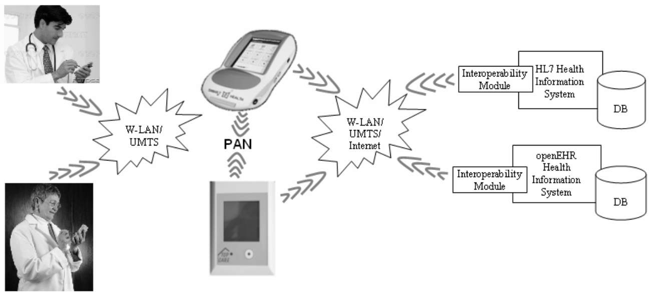

Fig. 2. Realization of SMDS in Hospital/Clinic. The health professionals can view the measurement results remotely on their PDAs by connecting to the AmI medical devices. In such scenarios, the AmI medical devices sometimes act as a bridge between health professionals and HISs/LISs.

The health professionals (HPs) have their PDAs, with the client software installed on them, to interact with these medical devices. Whenever the measurement is finished, an alert message is sent to the desired PDA of the HP. The HP can remotely query the measurement results of the desired medical device.

On the other hand, the AmI medical devices can also interoperate with existing legacy systems, being operated on different health standards, i.e. HL7, OpenEHR etc. As shown in Fig. 2, a component with the name *Interoperability Module* has been shown, which enriches the legacy systems with the capabilities of *discovery, advertisement, subscription,* and *eventing*, as well as *Web Services* through which, it could offer a number of services. We suggest that this module should be developed for each of a particular health standard (i.e. HL7 v.2.3) compliant system in a language that can be executed on a number of platforms without its recompilation. The obvious choice for this purpose is Java [21], because the runtime environments to execute Java byte code exist for a number of platforms (software/hardware). Once this module has been developed for a particular health standard with the afore-mentioned capabilities, the AmI device can easily discover this HIS/LIS and can query the functionalities that it provides and communicate with it seamlessly by understanding the semantic meanings of the functionalities that it offers.

V. REALIZATION OF SEMANTIC MEDICAL DEVICES SPACE

By implementing the architecture of AmI medical device, the vision of SMDS can be applied and realized in different healthcare facilities, covering a variety of healthcare scenarios; some of them are given below.

A. In the field of Hospitals and Clinics

1) To develop solutions for the realization of smart hospitals.
2) To provide mobile access to the patient's Electronic Health Record (EHR).
3) To enable the AmI medical devices to send parts of the EHR (i.e. measurement results), alerts, status information etc. to the PDA of the physician (see Fig. 2)
4) To develop AmI medical devices that inherently support the interoperability among each other.

B. In the field of Laboratories

1) To develop laboratory automation solutions for the realization of intelligent laboratories.
2) To enable the AmI medical devices to perform multi-parametric analysis, which means that they can perform more than one type of measurements if they receive few parameters from other sources (device/HIS/LIS).
3) To develop *Hubs* to make two different medical devices interoperable, working on two different eHealth communication standards.

C. In the field of Home Care services

1) To enable the AmI medical devices to send alerts (i.e. SMS) to the handheld devices (i.e. mobile, pager) of the caregivers of the patients, if something goes wrong with the patient.
2) To develop solutions that provides pervasive healthcare services (everywhere, anytime) to the patients, whether staying at home or mobile.
3) To develop solutions for *ambient assistant living* for the elderly patients (i.e. medication reminders, cognitive assistance etc.)

VI. IMPLEMENTATION DETAILS

In this section, we describe how we have partially implemented the SMDS infrastructure within the context of SmartHEALTH project [22]. We have also investigated some other approaches i.e. [12], [23], [24], [25], being adopted by the research community in the field of ambient intelligence, context awareness and home automation. These approaches used Jini™ [26] or UPnP [27] for the *capability publishing* and *capability discovery* of the devices. Some approaches used Salutation [28], Ninja [29], SLP [30] etc. as well. Each of the approaches has its own advantages and disadvantages regarding the particular application domain area. Additionally, a future release named as Devices Profile for Web Services (DPWS) [31] has been studied, which is proposed by *Microsoft*. It has the same advantages as UPnP but additionally it is fully aligned with Web Services technology. DPWS is actually expected to form the foundation for the next major upgrade of UPnP (referred to as UPnP v.2), but for reasons of market strategy related to the lack of backwards compatibility, no date is set for this transition. It is further worth noting that Microsoft's next-generation Windows platform *Longhorn* will natively integrate DPWS.

We suggest a new approach, based on Web Services technology, for the interoperability of AmI medical devices and HISs/LISs. For our experiments, we have chosen MDA IV mobile from Vodafone, with CPU of Intel (R) PXA 270, with the speed of 520 MHz and RAM of 64 MB. The operating system used is PocketPC/Windows Mobile 5.0. We have developed the Web Services in C# and used Intel's UPnP SDK [32] for the development of *addressing, discovery, eventing* and *publishing* capabilities; the WS-Security (WSS) specification [33] to implement the *security* functionalities i.e. signing, verification, encryption, decryption etc. To transfer the semantics of functionalities offered by Web Services and the detailed information about the AmI medical devices, we are planning to use WS-MetaDataExchange specification [34]. The local device database is developed using Microsoft SQL Server CE. The CKB, KQE, and KR will be implemented by using Jena2 Semantic Web toolkit [7]. In order to query the KQE, we have investigated a number of RDF query languages given in [6], and we suggest using SPARQL, because it is inherently supported by Jena2 framework. We suggest sending the messages among the AmI medical devices and the HISs/LISs as *RDF graphs*, so that each of the system could understand the meaning of the message on machine level, by using appropriate Ontologies.

VII. RELATED WORK

Our work builds heavily on the convergence of the Semantic Web and Web Services, particularly as supported by the OWL-S. So far, we have seen only Task Computing Environment [35], which is partially similar to our approach as far as the adoption of technologies is concerned. Our architecture is enriched with context querying and reasoning, which is not supported by Task Computing Environment.

VIII. Conclusion

We believe that Semantic Medical Devices Space (SMDS) is a promising application of Semantic Web and Web Services technologies, especially in the field of pervasive computing and medical devices. What is unique about our architecture is that we have introduced an end-to-end system from discovery, to composition and eventually execution of services, all driven automatically by the AmI medical devices, rather than manually by the users. In this paper, we have discussed some of our design choices and implementation, especially the *context-awareness* mechanism has been explained in great detail.

As explained in [35], We also see SMDS as a business opportunity for both device manufacturers and IT solution providers. For device manufacturers, for example, the AmI medical devices can be enriched with a number of interfaces, each for distinct eHealth standard, so that they could behave like an *intelligent bridge* to transfer a specific eHealth standard message to the other system working on different standard. The selection of appropriate interface will be on the disposal of AmI medical device. IT solution providers, for example, can take advantage of Web Services technology on the AmI medical devices to resolve interoperability issues with other medical devices and eHealth systems, and can provide interesting applications in different health care scenarios, some of them have already been explained under Section V.

References

[1]Digital Imaging and Communication in Medicine-DICOM; http://medical.nema.org/

[2]IEEE 1073, Standard for Medical Device Communication; http://www.ieee1073.org/

[3]W3C, Web Services Architecture, W3C Working Draft, 2002; http://www.w3.org/2002/ws/

[4]Universal Description, Discovery and Integration *specifications;* http://www.uddi.org/specification.html

[5]*Web Services Description Language 1.1*, W3C Note: http://www.w3.org/TR/wsdl

[6]*SOAP 1.2 Part 1*, W3C Working Draft: http://www.w3.org/TR/soap12-part1/

[7]T. Berners-Lee; *The Semantic Web*, Scientific Am., May 2001.

[8]Resource Description Framework (RDF);http://www.w3.org/RDF/

[9]Web Ontology Language (OWL); http://www.w3.org/2004/OWL/

[10] Weiser, M. (1993). *Some computer science problems in ubiq. computing*; Communications of the ACM, 36(7).

[11] Arts, E. et al. Ambient Intelligence. In The Invisible Future: The Seamless Integration of Technology into Everyday Life. McGraw Hill Professional.

[12] X. Wang, J. S. Dong, C. Y. Chin, S. R. Hettiarachchi, D. Zhang; *Semantic Space: An Infrastructure for Smart Spaces.*

[13] R. E. McGrath, A. Ranganathan; *Investigations of semantic interoperability in ubiquitous computing environments.*

[14] Semantic Web Services; http://swws.semanticweb.org/

[15] DAML; http://www.daml.org/services/

[16] OWL-S; http://www.daml.org/services/owl-s/

[17] FIPA Device Ontology Specification; http://www.fipa.org/specs/fipa00091/XC00091D.html

[18] Composite Capabilities/Preferences Profile (CC/PP); http://www.w3.org/Mobile/CCPP/

[19] Wearable Computing; http://www.media.mit.edu/wearables/

[20] LiveNet Project; http://hd.media.mit.edu/livenet/

[21] Java Technology; http://java.sun.com/

[22] SmartHEALTH IP Project; http://www.smarthealthip.com

[23] Steven E, Y. Zaho; *An Architecture for a Secure Service Discovery Service*

[24] F. Zhu, M. W Mutka, L. M. Ni; *Service Discovery in Pervasive Computing Environments*.

[25] T. Gu et.al.; *Towards an OSGi-Based Infrastructure for Context-Aware Applications*

[26] Jini™: *Jini Network Technology*; http://www.sun.com/software/jini

[27] *Universal Plug-and-Play* (UPnP); http://www.upnp.org

[28] Salutation Network Technology; http://www.salutation.org

[29] Ninja Software Infrastructure; http://ninja.cs.berkeley.edu/

[30] Service Location Protocol (SLP); http://www.openslp.org/

[31] *Devices Profile for Web Services* by Microsoft Corporation; http://specs.xmlsoap.org/ws/2005/05/devprof/devicesprofile.pdf

[32] Intel UPnP SDK; http://www.upnp.org/resources/sdks.asp

[33] Web Service Security; http://www.oasis-open.org/committees/wss/

[34] WS-MDE; http://xml.coverpages.org/WS-MetadataExchange.pdf

[35] R. Masuoka, B. Parsia, Y. Labrou; *Task Computing – The Semantic Web meets Pervasive Computing*