Design and Implementation of an Interaopreable and Extednable Smart Home Semantic Architecture using Smart-M3 and SOA

Haitham S. Hamza, Enas Ashraf,
Azza K. Nabih, Mahmoud M.
Abdallah, Ahmed M. Gamaleldin
SECC-ITIDA, MCIT
Giza, Egypt
{hhamza, inas, azkamal, mmabdallah,
agamal}@itida.gov.eg

Alfredo D'Elia
ARCES, Alma Mater Studiorum,
Università di Bologna
Bologna, Italy
adelia@arces.unibo.it

Hadeal Ismail, Shourok Alaa,
Kamilia Hosny, Aya Khattab,
Ahmed Attallah
ANSR Lab, Cairo University
Giza, Egypt
{hismail, salaa, khosny, akhattab,
aattallah}@ansr.cu.edu.eg

***Abstract* — Smart homes attempt to automate interaction with the environment and existing appliances to optimize resources while maintaining user convenience. *Interoperability* is a key feature, as well as a main challenge in developing smart systems. This is due to the wide interactions among appliances, users, and the surrounding environment, which are heterogeneous by nature. Moreover, smart systems may evolve over time by integrating new actors, devices, and applications, making *extendibility* another key challenge in designing such systems. Accordingly, developing a framework that that provides both interoperability and extendibility for smart systems is of a great interest in practice. Accordingly, this paper proposes a framework based on the semantic and service oriented architecture (SOA) technologies for smart homes. The proposed architecture makes use of open source SOA and Smart-M3 framework that provide the core technologies to enable interoperability and extendibility. Ontology is designed and used to enable semantic middleware for integration. The proposed framework is demonstrated using a simple smart home scenario.**

Keywords-Smart home; SOA; Semantic middleware; Smart-M3; SIB; Smart space; Ontology; Interoperability; Extendibility

I. INTRODUCTION

A smart environment is a context-aware environment that is able to interact with its inhabitants through autonomous devices embedded all around the physical world [1]. Three main keywords define smart environments: *Context Awareness*, *Interoperability*, and *Extendibility*.

Context Awareness is the ability of devices to be aware of the context and changes that take place within the environment [2]-[6]. Being aware is one aspect; however, being reactive to the expected changes is the main goal of context awareness.

Interoperability refers to the ability of the system ability to interact based on a common information language. The distributed and heterogeneous nature of smart environments requires interoperability support in many levels. The following three interoperability levels are identified in [7] and [8]: (1) device interoperability to enable communication among devices, (2) service interoperability to exploit the information originating from heterogeneous devices as services for various end-user applications, and (3) information interoperability across application domains. However, not all smart environments are able to ensure interoperability at these three levels.

Extendibility refers to the ability of the smart space to adapt to configuration changes such as adding, removing, and updating hardware or software parts of the environment.

Achieving interoperability and extendibility in smart home solutions have been demonstrated in the literature using Service Oriented Architecture (SOA) or semantic exist, but rarely both [9][10][21][22]. However, we argue that, to achieve better interoperability and extendibility at both the middleware and application layers, an integrated SOA-semantic architecture is needed. Accordingly, this paper proposes a solution that exploits both SOA (using open source SOA technology) and semantics (using Smart-M3 framework, to be elaborated later). The proposed solution makes use of ontologies to enable interoperability among the various devices to allow interoperability as well as extendibility at both the services and application levels. It is worth noting that the proposed architecture is demonstrated in the context of smart homes; however, its structure and operational concept can be adopted to develop services in various domains, such as healthcare, which have received an increasing interest in adapting smart environment solutions in recent years [16]-[20].

The rest of this paper is organized as follows. Section II summarizes related work. Section III presents the proposed architecture. The use of the proposed architecture in a simple smart home prototype is presented in section IV. Section V discussed how the proposed architecture provides extendibility and interoperability; Conclusions are given in Section VI.

II. RELATED WORK

Smart environments refer to any type of user environment (homes, offices, universities, etc.), where technology is utilized to provide the user with the maximum possible automation, convenience, safety, and comfort. There are common challenges exist in all smart environments including the ability to integrate a wide variety of different objects, with each of which having its own operating standards, programmability, sensing ability, etc. In addition, smart systems do need to evolve over time to meet changing and increasing demands of their users and environments, and hence, extendibility is another key challenge that needs to be addressed in designing such systems.

In addition to the above, smart homes, and environments in general, need to provide the best services to its users while maintaining low deployment and operational costs.

Integrated SOA and semantic architectures/platforms provide good approach to tackle the above challenges. SOA technologies support interoperability and extendibility at the service and application layers; whereas semantics enables interoperability and extensibility at the devices (physical) layer. Research efforts that demonstrate the integration of both semantic and SOA technologies have been reported in various domains (e.g., [16][17]). However, most of these solutions have limited usability as they are tuned to tackle domain-specific issues and challenges.

In [18] and [19], a framework for wearable devices is proposed and demonstrated for use in monitoring sportsman/sportswoman. The framework is shown to be effective; however, it is optimized for use in this particular domain with limited applicability. For instance, services in this framework (sensor agents) are running on the sensor nodes themselves, which makes sensor powerful; but at the cost of an increased complexity. Such complexity (and hence, cost) makes this framework expensive to deploy in smart home applications, where several sensors with much lower complexity and cost are needed.

The framework proposed in the Chiron ARTEMIS project [20] makes use of the SOFIA smart space architecture [23]. A key feature in this framework is the registration of the Semantic Information Broker (SIB) as a service on the SOA. However, this design approach may result in an increased coupling that can affect the extendibility of the framework in dynamic environments with several changes.

Our proposed architecture, on the other hand, suggests an information and service interoperability platform that is highly de-coupled. This is achieved by adopting the concept of Enterprise Service Bus (ESB) to compose and publish services, having a single software component representing the low level services as a single integration point. This design approach would allow for the use of multiple semantic interoperability platforms (middleware), where each middleware is used for implementing and publishing the low level services keeping the high level ones on the ESB intact. Primitive services will be the ones accessing the data in the semantic repository; Semantic Information Broker (SIB) .

III. SYSTEM ARCHITECTURE

The proposed architecture aims at introducing a middleware that is flexible to be used with a wide range of scenarios and devices. To achieve this, interoperability and extendibility are the key design features that the developed middleware should address at both devices and application levels. Interoperability and extendibility are achieved at the devices and application layers by adopting, respectively, Smart-M3 and SOA technologies.

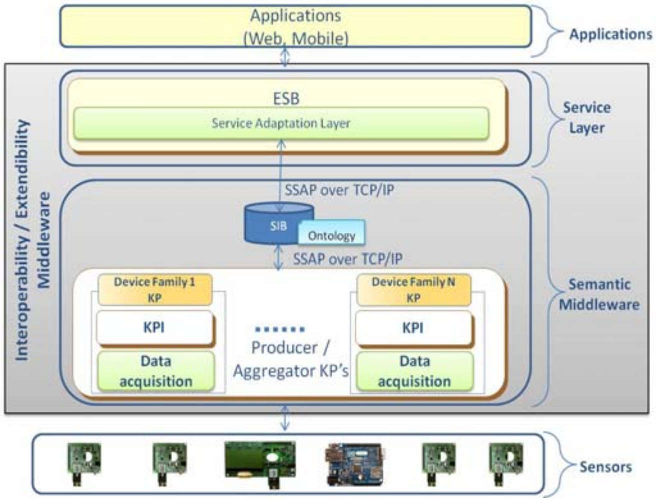


Figure 1. System Layered Architecture

Figure 1 depicts the proposed architecture. As show in the figure, the key components in the proposed middleware are: the SIB, the context ontology, and the SOA/ESB. The following subsections briefly explains these components.

A. Smart-M3 Interoperability Platform

Smart-M3 is an open-source software project that provides an infrastructure for sharing semantic information among software entities and devices. Smart-M3 deploys an information interoperability approach for devices to easily share and access local semantic information. In Smart-M3 the information is represented by using same mechanisms as in semantic web, thus allowing easy exchange of global and local information. Two main components make up the Smart-M3 infrastructure (Figure 2): SIB and Knowledge Processors (KPs).

1) Semantic Information Broker (SIB): The SIB is the access point for receiving information to be stored in the smart space or retrieving such stored information. The information is stored in the form of RDF graph (according to some defined ontology) in order to link data between different domains causing cross-domain interoperability to be much easier.

2) Knowledge Processors (KPs): KPs are the active parts of a Smart-M3 system. It provides the information (producer KP), modify and query it (consumer KP) and take actions based on the information seen in the SIB. This is done via a simple XML-based communication protocol called Smart Space Access Protocol (SSAP) that provides KPs with primitives to access SIB data (insert/add/remove/query/etc.).

B. Context Ontology

Ontologies are used to represent knowledge in machine understandable format. Ontologies represent the knowledge for describing certain domain. The context is all the information used to define the situation of an entity. As detailed in [11], the context can be physical, computational or user context. To completely describe a context, the ontology used should be simple, complete, expressive, and evolvable. As a result, defining an ontology that can be extended with new concepts is a key factor. For example, if the ontology includes a concept for device that is associated with some data, it is a good practice to define a separate class for data that has many types that can be extended later (sensor data, identification data, etc.). However, to achieve better extendibility and interoperability, it is required not only to make ontology concepts extendible, but also to add the extendibility concepts to the ontology itself.

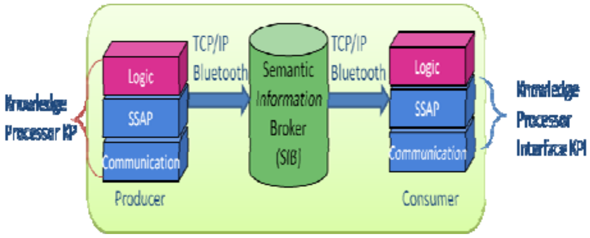


Figure 2. Smart-M3 Component Interaction

C. *Service-Oriented Architecture*

Service-oriented architecture (SOA) is a set of principles and methodologies for designing and developing software in the form of interoperable services. These services are usually representing the high level business functionalities that are built as software components, which can be reused for different purposes. Many techniques and approaches can be used to achieve this, e.g., OSGi framework [12], even SSAP protocol used in Smart-M3 can be considered as a SOA approach that provides low level services that can guarantee interoperability between devices compliant with the same approach [13]. In the proposed architecture, the ESB technology is used to achieve this goal. ESB helps achieving SOA goals through providing a flexible connectivity infrastructure for integrating applications and services.

IV. DEMONSTRATION OF SMART HOME SCENARIO

This section demonstrates the implementation of the proposed architecture (Figure 1) in a simple smart home scenario. The key features of the demonstrated smart home are: (1) providing instruments to edit and apply user preferences related to installed devices, (2) using low cost sensors to detect user presence and identity in order to configure the installed devices, (3) implementing a simple and complete web based interface to manage system behavior and to provide remote monitor and control, (4) detecting and reacting to anomalous situations such as faults and alarms, and notifying the user on their mobile devices, and (5) enabling maintenance companies to register in order to offer their services to fix faults and maintain devices installed in the smart home.

In this scenario, the system should consider the following key aspects: users should be identified; user preferences should be applied upon user identification, and faults are automatically detected and communicated to users via mobile and web applications. The main actors of this scenario are: (1) the administrator who defines the smart home and localize devices into smart home rooms, (2) the owner of the smart home, who is registered to the system using security credentials beside the RFID identification tags, and (3) the maintenance companies who will register to offer their maintenance services.

This scenario is realized using the proposed architecture, where Redland SIB and Java based KPs are used for the semantic middleware [14], and Mule standalone ESB v3.3 [15] installed on an Ubuntu LTS v12.04 are used to implement the SOA technology. Mule is chosen for its simplicity, and because it different web services protocols including Simple Object Access Protocol (SOAP) and Representational State Transfer (REST).

A. *Hardware and Wireless Sensor Network*

The WSN hardware layer at the lower level is realized through a network of ZigBee nodes. The ZigBee nodes can operate stand alone like the ones reporting the environmental status or attached to devices. In addition to the environmental node that should report information on ambient temperature, humidity, and light intensity level, a set of devices is selected to be demonstrated within this scenario in order to demonstrate the value of smart environments. Specifically, ZigBee nodes are attached to AC, refrigerator, multimedia board, and light. The ZigBee network has a single coordinator that is responsible for collecting data from the distributed nodes and sending via serial interface to the device hosting the dedicated KP. User identification is done via RFID module so that system can apply specific actions on the home devices as per a pre-set user preferences. The RFID information is sent via Ethernet connectivity to the device hosting the dedicated KP.

For extendibility on the level of devices, a self-exposure approach for device features is adopted. Each device sends an identification packet upon switch on. By doing so, new devices can be easily plugged into the home network allowing self-configurability of the system.

B. *Smart Home Ontology*

Several ontologies have been developed for smart home applications. For simplicity, and order to avoid any complexity induced by full-scale smart home ontology, we decided to develop our own ontology (See Figure 3). The used ontology is developed with extendibility in mind.

Figure 4 5 illustrate how a new device is defined in the knowledge base. Each device will send an identification packet that defines the device type, id, capabilities, capability parameters, measurements, and data ranges for measures and possible commands. Once received by the dedicated KP, the data is interpreted to the related ontology concepts.

C. *KPs Design*

The composition and distribution of KPs is one of the main considerations to promote interoperability and extendibility in the smart space. A proper design of information exchange mechanisms in the smart space is crucial. Modularity, de-coupling of information producers and consumers, and providing minimal points of integration with hardware and SOA layers can even take extendibility to another level. Figure 6 illustrates how this design issue was handled in our architecture. The Adapter KP is responsible for receiving data from the hardware layer, interpreting the data, and inserting the interpreted semantic information into the smart space. This data includes device identification as well as context information. Inserted information can be then consumed by the Fault Detector KP and the Brain KP. The Fault Detector KP consumes and analyzes the smart space information to produce fault related information and insert it to the smart space. The Brain KP is triggered by the user presence and identification information from the smart space and uses these data to produce new commands to the smart space. The commands will be then consumed by the Driver KP to actuate the devices. In order to expose the semantic information to the service and application layer a Service Adaptation Layer is designed. This is a special KP that will reside on the ESB exposing the required information as low level web services that will be used later to compose higher level services based on application needs.

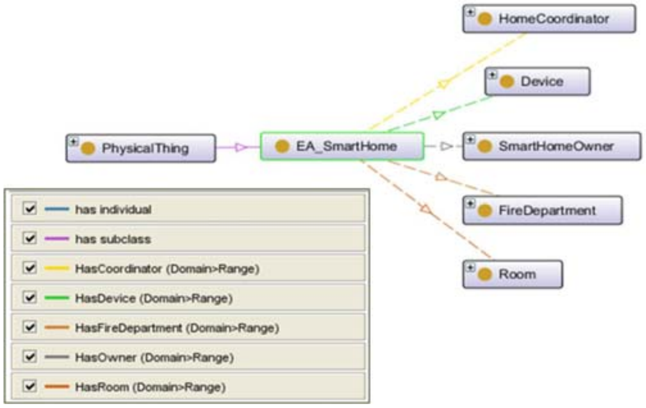


Figure 3. Concepts Related to Smart Home

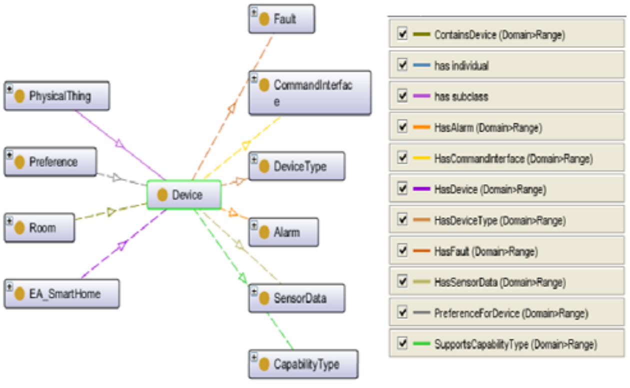


Figure 4. Device Ontological Definition

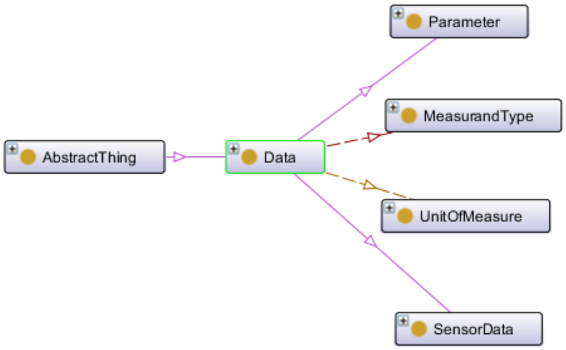


Figure 5. Types of Device Data

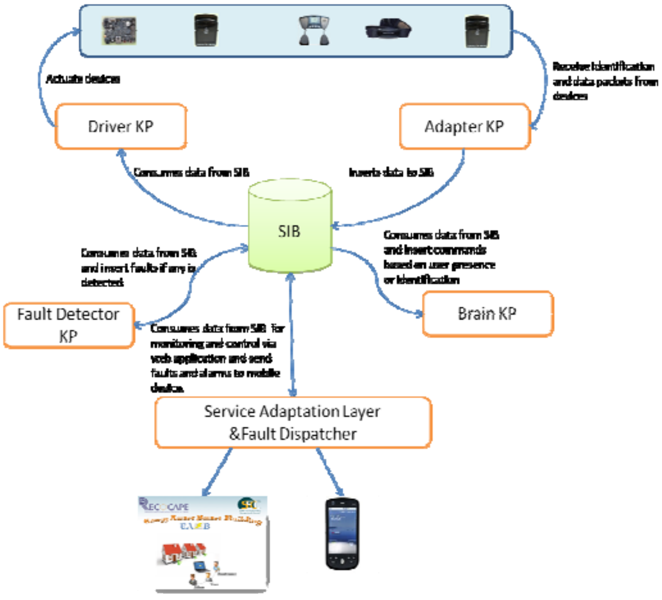


Figure 6. System Data Flow and Component Interaction

D. Service and Application Layer

At the higher level, services and user applications exist. Web and Android applications are designed hiding the lower details of the architecture and presenting their smart functionality through compositions of several services deployed on the ESB. This guarantees a complete isolation as the application can use a service to get ambient temperature in a smart home regardless of the underlying semantic middleware. Dynamic service composition techniques can even allow creating a new service at run time to extend the application features. By deploying the services in the ESB, our system can seamlessly integrate with external applications or services regardless of the communication protocol, message structure and implementation technology they are using. Furthermore, utilizing this service layer makes applications platform independent with applications can run on any user platform, without need to do this exhausting application porting task needed in traditional systems.

V. DISCUSSION ON INTEROPERABILITY AND EXTENDIBILITY

This section discusses how the proposed architecture enables interoperability and extendibility via the demonstration of how a new device can be seamlessly plugged into the smart home scenario discussed above.

The proposed architecture provides a systematic and easy approach to add new devices and services and to create various applications that can seamlessly interoperate with the existing environment. Clearly, no modifications are needed when a new instance of an existing device type is added to the system, as in such a case, simply the added instance will be plugged and operated in the system. Thus, the question is how the proposed architecture will seamlessly accommodate the addition of a new type of devices and their corresponding services? We will investigate this question by demonstrating the steps needed to allow the user to add a new service (mobile application) that allow her to monitor the energy consumption profile, and receive notifications for energy misuse based on a set of pre-defined preferences for energy control.

To realize this new service, the system, our smart home scenario needs to be extended with a new device, example, a smart meter to allow for the reading of the energy consumption and report these data to the system. Using our system, the sought smart meter will interoperate with the rest of the devices (already plugged and operating) by simply sending its identification packet and extending the ontology with the new concepts of the smart meters domain. In addition to extending the physical space, the application layer needs to be extended by adding a new set of services to the service pool. The proposed architecture enables this in a straightforward way as the use of ESB makes it easy to create this service from scratch or the consumption a ready-made services available already from other vendors. In the following, we summarize the steps needed to incorporate the smart meter into our system.

- Update the data set to be used in the device identification and data packets. As a result of having a generic packet format (Figure 7) the smart meter can be interoperated with the system by adding a new device type, types and ranges of the measurements read by the smart meter, and commands that the smart meter can receive (if any).
- Extend the ontology with a new device type, MeasurandType, and units. As shown in Figures 8 and 9. This can be achieved by simply adding a new set of individuals to the ontology without changing the main graph.
- For the KPs, we need to modify the data interpreter (message receiver) in the `Adapter` KP as well as the command dispatcher in the `Driver` KP. The message receiver and command dispatcher are two java modules independent of the semantic middleware. The two modules are only dependent on the used ontology. This implies that in case the Smart-M3 interoperability platform is changed, the same modules can be reused.

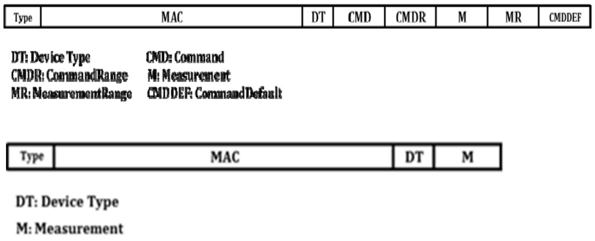
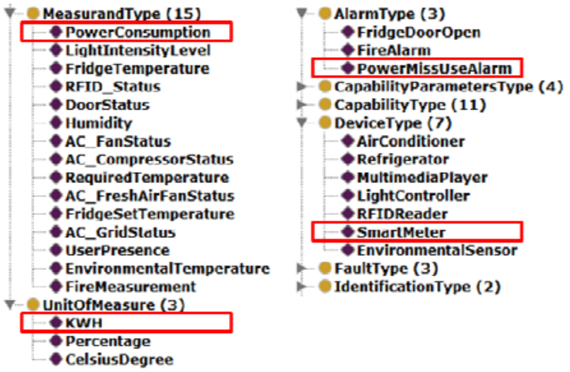


Figure 7. Top: Identification packet format (Type = ‘I’), Bottom: Data packet format (Type = ‘D’)



- MeasurandType (15)
  - PowerConsumption
  - LightIntensityLevel
  - FridgeTemperature
  - RFID_Status
  - DoorStatus
  - Humidity
  - AC_FanStatus
  - AC_CompressorStatus
  - RequiredTemperature
  - AC_FreshAirFanStatus
  - FridgeSetTemperature
  - AC_GridStatus
  - UserPresence
  - EnvironmentalTemperature
  - FireMeasurement
- UnitOfMeasure (3)
  - KWH
  - Percentage
  - CelsiusDegree
- AlarmType (3)
  - FridgeDoorOpen
  - FireAlarm
  - PowerMissUseAlarm
- CapabilityParametersType (4)
- CapabilityType (11)
- DeviceType (7)
  - AirConditioner
  - Refrigerator
  - MultimediaPlayer
  - LightController
  - RFIDReader
  - SmartMeter
  - EnvironmentalSensor
- FaultType (3)
- IdentificationType (2)

Figure 8. Adding a new device type

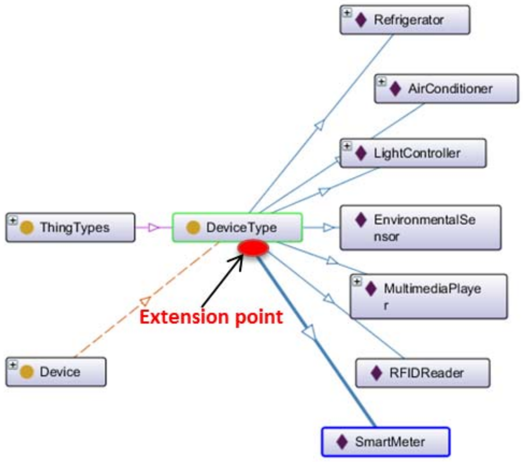


Figure 9. Adding a new device type

- Finally, we will expose the new features as services and deploy it to the ESB upon which new application features are built for both web and mobile applications. In this example two services are added. The first is to enable the user to remotely monitor power consumption, and the second is to notify the user of unexpected load patterns.

As illustrated above, deploying the proposed integration architecture can support system extendibility and interoperability by providing a modular design with reduced points of modification in the existing smart system.

Device interoperability and extendibility are achieved by allowing each device to self-expose its features and capabilities through generic data packets to be interpreted into semantic information according to the defined ontology via the dedicated KP to feed the smart space. Using ESB as a deploymrnt environment for numourous services of different technologies eases the creation of new services and permits extending the application layer to new domains.

Moreover, the use of SOA/ESB supports *modularity* of services and applications by nature due to the loose coupling of services. Added to this, the proper design of data flow and component interaction in the semantic space is crucial. The used approach in the smart home scenario separates components that feed the smart space with information from those that consume the information. Moreover, data consumers that act on devices are de-coupled from data consumers exposing semantic information to the SOA layer.

VI. CONCLUSION

An effective design of smart homes (or environments, for this matter) needs to deal with various practical challenges in order to enable real value to the uses. Among these challenges are the *interoperability and extendibility* that arise due to the need for heterogeneous devices to be integrated and interoperate to deliver the required business value of the system. This paper proposes and demonstrated, via the real implementation of a simple smart home scenario, the use of the known semantic Smart-M3 and the well-adopted SOA/ESB technologies to develop an architecture that can enable interoperability and extendibility at both devices/information and services/applications layers. Smart-M3 supports interoperability at the information layer by utilizing semantic repository at its core, where information are stored in a standard representation based on domain specific ontology. Smart-M3 also is an enabler for system extendibility, with any components can be easily involved into the system under the condition that it uses same ontology. SOA/ESB provides the required support to realize interoperability at the service level, where applications running on any devices or based on any platforms or operating systems can access system features by the aid of these loosely coupled, platform independent services deployed in the service layer. It can extend the system functionality through composition of these services to support more complex scenarios and through deployment of new set of system services. With the ESB added, not only devices using the same semantic middleware will interoperate, but also those with different semantic middleware can also be integrated, which allows legacy systems to coexist and operate with new ones.

ACKNOWLEDGMENT

This work supported by the RECOCAPE project funded by the EU Commission under call FP7-INCO-2011-6.

REFERENCES

[1] C. A. N. Guerra, and F. S. C. da Silva, “A Middleware for Smart Environments,” Proc. of AISB Symposium on Intelligent Agents and Services for Smart Environments, 1-4 April 2008, vol. 8, pp. 22-24.

[2] J. Krumm, Ubiquitous Computing Fundamentals, Taylor and Francis Group, LLC, 2010.

[3] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniqus and Systems," IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, vol. 37, No. 6, November 2007, pp. 1067-1080.

[4] A. Flammini, P. Ferrari, D. Marioli, E. Sisinni, and A. Taroni, "Wired and wireless sensor networks for industrial applications," Microelectronics J.l, vol. 40, Issue 9, Sept 2009, pp. 1322-1336.

[5] S. Farahani, ZigBee Wireless Networks and Transceivers, Elsevier Ltd., 2008.

[6] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, Wireless Sensor and Actuator Networks – Technologies, Analysis, and Design, Academic Press, 2008.

[7] L. Roffia, T. S. Cinotti, P. Azzoni, E. Ovaska, V. Nannini, and S. Mattarozzi, “ Smart-M3 and OSGi: The Interoperability Platform,” Computers and Communications (ISCC), 2010 IEEE Symposium, June 2010, pp. 1053-1058.

[8] J. Honkola, H. Laine, R. Brown, and O. Tyrkkö, “Smart-M3 Information Sharing Platform,” Computers and Communications (ISCC), 2010 IEEE Symposium, June 2010, pp. 1-6.

[9] L. Ngo , “Service-oriented architecture for home networks,” TKK T-110.5190 Seminar on Internetworking, 2007, pp. 1-6.

[10] T. Perumal, A. Ramli, C. Y. Leong, S. Mansor, and K. Samsudin, “Interoperability among Heterogeneous Systems in Smart Home Environment,“ SITIS '08. IEEE Int. Conf., Dec. 2008, pp. 177 - 186.

[11] E. Ovaska, L. Dobrica, A. Purhonen, and M. Jaakola, “Technologies for Autonomic Dependable Services Platform: Achievements and Future Challenges,” Software and Data Technologies Comm. in Computer and Information Science, vol. 303, 2013, pp. 199–214.

[12] OSGi™-The Dynamic Module System for Java™, http://www.osgi.org [retrieved: March 2014].

[13] Requirements for SOA Protocols, DIEM – Devices and Interoperability Ecosystem, BA – Building Automation, Aalto University – Media Technology .

[14] Smart-M3 Interoperability Platform, http://sourceforge.net/projects/smart-m3/ [retrieved: March 2014].

[15] Mule ESB, http://www.mulesoft.org/ [retrieved: March 2014].

[16] I. Corredor, J. Iglesias, A. M. Bernardos, and J. R. Casar, "A development methodology to facilitate the integration of Smart Spaces into the Web of Things," 4th Int. Workshop on Sensor Networks and Ambient Intelligence, 23 March 2012, pp. 829 – 834.

[17] V. Luukkala, D. Binnema, M. Börzsei, A. Corongiu, and P. Hyttinen, "Experiences in implementing a cross-domain use case by combining semantic and service level platforms," IEEE Symposium on Computers and Comm. (ISCC), 22-25 June 2010, pp. 1071 – 1076.

[18] J. Rodríguez-Molina, J. Martínez, P. Castillejo, and L. López, "Combining Wireless Sensor Networks and Semantic Middleware for an Internet of Things-Based Sportsman/Woman Monitoring Application," Sensors Jjournal, doi:10.3390/s130201787, 2013, pp. 1787-1835.

[19] P. Castillejo, J. Martínez, L. López, and G. Rubio, "An Internet of Things Approach for Managing Smart Services Provided by Wearable Devices," International Journal of Distributed Sensor Networks, vol. 2013, Article ID 190813, 2013, pp. 1-9.

[20] CHIRON EU project, ARTEMIS, http://www.chiron-project.eu/ [retrieved: March 2014].

[21] M. Kasza, V. Sz˝ucs, V. Bilicki, G. Antal, and A. B´anhalmi, "An Implementation Model for Managing Data and Service Semantics in Systems Integration," The Fifth International Conf. on Advances in Databases, Knowledge, and Data Applications, 2013, pp. 182-189.

[22] G. Pan, L. Zhang, Z. Wu, S. Li, L. T. Yang, M. Lin, Y. Shi, "Pervasive Service Bus: Smart SOA Infrastructure for Ambient Intelligence," IEEE Intelligent Systems, 20 Dec. 2012.

[23] http://www.sofia-community.org/ [retrieved: March 2014].

[24] http://www.secc.org.eg/RECOCAPE/ [retrieved: March 2014]