A Smart-M3 lab course: approach and design style to support student projects

L. Roffia, A. D'Elia, F. Vergari,
D. Manzaroli, S. Bartolini, G. Zamagni,
T. Salmon Cinotti
Alma Mater Studiorum – Università di Bologna
Via Toffano, 2, Italy
lroffia@arces.unibo.it

Jukka Honkola
Nokia Research Center
Itämerenkatu 11-13, 00180 Helsinki, Finland
jukka.honkola@nokia.com

Abstract

This paper reports on the lesson learned from setting up and running a course on smart spaces based ambient intelligence within a "postgraduate" degree in Information & Computer Science Engineering. The course consists of a lab module where students are asked to propose, develop and demonstrate simple smart environment applications. Specifically, the course is built around Smart-M3, the open source interoperability component of an open innovation platform developed within the European Project SOFIA (2009-11). This paper illustrates the above summarized course, presenting the design style and the application development flow specifically devised for the course, the students evaluation approach taken, the course achievements, the issues that still need to be handled, and it will envisage topics and further extensions.

Index Terms: Information Interoperability, Smart Space, Application Design Style, Smart-M3, RDF, Ontology

I. INTRODUCTION

The course was run well ahead the European Project SOFIA's end [1]. But how could such an unplanned course, based on the preliminary results of an underway research project be so quickly included in a postgraduate degree curriculum? This was possible only thanks to a lucky circumstance: the Graduate Programs Committee of our ICT Department extended by 50% both credits and teaching time assigned to our course, and gave us "carte blanche" on how and what to teach in the additional 30 granted hours. In this circumstance, a choice had to be made: should the current content of the course be smoothly extended, or should we rather go for a rather disruptive education initiative? The latter choice was risky, but it was considered a win-win one, with great potential benefits for both research and education. This was our choice, as, on one side an open source prototype of an interoperability platform for smart environments applications, i.e. Smart-M3 [2, 3], was available and on the other side the hosting course was named "Information Processing Systems", which is a very general identifier widely compatible with the proposed initiative. Many risks were involved in our choice: particularly, no course material (such as textbook and exercises) was available; no application design style still existed; no single person had all the skills required to run the entire course. To deal with these risks, a training team of five researchers was set up, all of them with complementary and course relevant skills. They jointly defined a design style, attended all classes, gave oral presentations and provided direct support to the students. At the beginning of the course the 40 students had been partitioned into groups of five and each group was asked to propose an own project. After the first month each group was asked to formalize the description of its project, using the design style and the application development flow specifically devised for the course. Consequently the students filled out a *project presentation template* that will be described in this paper.

This paper is organized in three main sections: in Section II (*Course Outline*) are presented the course organization, the evaluation criteria and procedure, the course related research framework, and an overview of the course program. In Section III *(Smart-M3*) is presented the smart space management framework that have been adopted to develop the student projects. Section III (*Design style and tools*) reports the methods and tools that have be designed to support student projects. In Section IV (*Conclusion*) conclusion and future developments are drawn.

II. COURSE OUTLINE

The course was implemented through the following steps: first of all, the students were introduced to the *smart environments scenario* by presenting *candidate application domains*. Then the need to *connect physical world elements*, such as places, users, objects and devices, *to the information world* was discussed, together with the *interoperability* requirements. A significant part of the course was focused on the *semantic data representation*. Students were introduced to RDF and OWL and they were guided to the use of the Protégé ontology editor [4]. Moreover, students were introduced to *Smart-M3* (see Section III). At the end, a *design style* was presented as reference model for projects development (see Section IV).

A. Lab setup

The course has been supported by the flexible Moodle e-learning platform [5]. The "*Smart-M3 @UNIBO Moodle*" [6] includes a WiKi which has been used to share student projects and still informal course material, e.g. slides, templates, APIs and links. Each group had its own dedicated workstation (and WiKi page) and, in order to do not interfere with other groups, Smart-M3 has been installed on each single workstation, i.e. 10 interconnected workstations. Each workstation runs Microsoft Windows as host and Ubuntu as guest, i.e. using VirtualBox [7]. The workstation configuration allows also to use a shared networked instance of Smart-M3, i.e. each workstation has two IP address, one for the host and one for the guest. This configuration allows also to develop applications on two different operating systems, i.e. Ubuntu and Microsoft Windows.

B. Student projects

In this subsection, we report the proposed projects on which students have been working on.

1. **Smart Multi-robot Environmental Mapping:** Lego Mind Storm® robots move within an environment aiming to create a map of it. The map creation is displayed in real-time on a smart phone. Topics: Space representation; SIB based coordination; *Lego Mind Storm® KP adapter; smart phone*
2. **Smart Traffic Light** (*evaluated*): collision avoidance between two Lego Mind Storm® cars moving on two tracks that have one intersection (simulated solution). Topics: SIB based coordination; *Lego Mind Storm® KP adapter*
3. **Smart Domus:** monitoring power consumption at home and applying power saving policies. Topics: Presence monitor; appliances power monitor; power saving algorithms
4. **Smart Cupboard** (*evaluated*): monitoring products store to notify the user smart phone when quantities go under thresholds. Topics: Alarm generation; barcode products identification; *smart phone*
5. **Smart Speechy:** send vocal commands to devices, e.g. TV, Hi-fi, lights. Topics: JAVA Speech Library; commands & actions
6. **Smart Cart:** creating a shop list at home via Web. At the supermarket the cart identifies the user and it helps her to find products. Topics: User and product RFID identification; *RFID reader KP adapter*

7. **Smart Coffee Maker:** setting coffee preferences and have it ready in your hotel room when you wake up. Topics: Customer preferences; smart devices emulation
8. **Smart Secure Chat:** a secure chat system supporting also room based chats. Topics: Secure login; messages dispatching; messages garbage collection

All the above projects have been described and implemented using the design style and tools presented in Section IV.

C. Course introduction, basic concepts on smart environments and ontologies

As first step, students were introduced to smart environments. The first part of the course has been fundamental also to share a "common language". As shown in Figure 1, we started from presenting the relationships between physical entities and their digital representation, i.e. smart space. Within this framework, students understood the role of sensors, i.e. extracting data from the physical word. With respect to this, some sensing technologies were presented, like RFID, inertial sensors, magnetic sensors and location systems (see *Where?* in Figure 1). After that, we discussed the need to unique identify each single physical entity, i.e. devices, users, objects, places (see *Who?* in Figure 1), within the physical space. As next step, the concept of *data model* was introduced: each physical entity and its relationships should be described in a proper way into the smart space (see *What?* in Figure 1). As introduction to the semantic data model, the Semantic Web project was introduced [8, 9]. A bottom up approach to ontologies was followed. First of all, the *XML* (Extensible Mark-up Language) [10] was presented as a language for representing data and their corresponding metadata as a tree structure. After that, the Resource Description Framework (*RDF*) [11] was discussed and the Universal Resource Identifier (*URI*) [12] was introduced. Moreover, the definition of an RDF *statement* as a triple <*subject, predicate, object*> where the *subject* and the *predicate* are univocally identified through *URI*, while the object may be a *URI* or a *literal*, i.e. a string or a number, was given. Also, it was shown as a set of RDF statements correspond to a *graph*: the *subject* and the *object* are represented by nodes, while each *predicate* is represented by a labelled oriented edge connecting two nodes. Then we moved to the concept of *knowledge base* by presenting the *RDFS* (RDF Schema) [13] and the *OWL* (Ontology Web Language) [14, 15]. As conclusion to the semantic data representation, some examples of OWL ontology modelling were presented using the Protégé ontology editor [4].

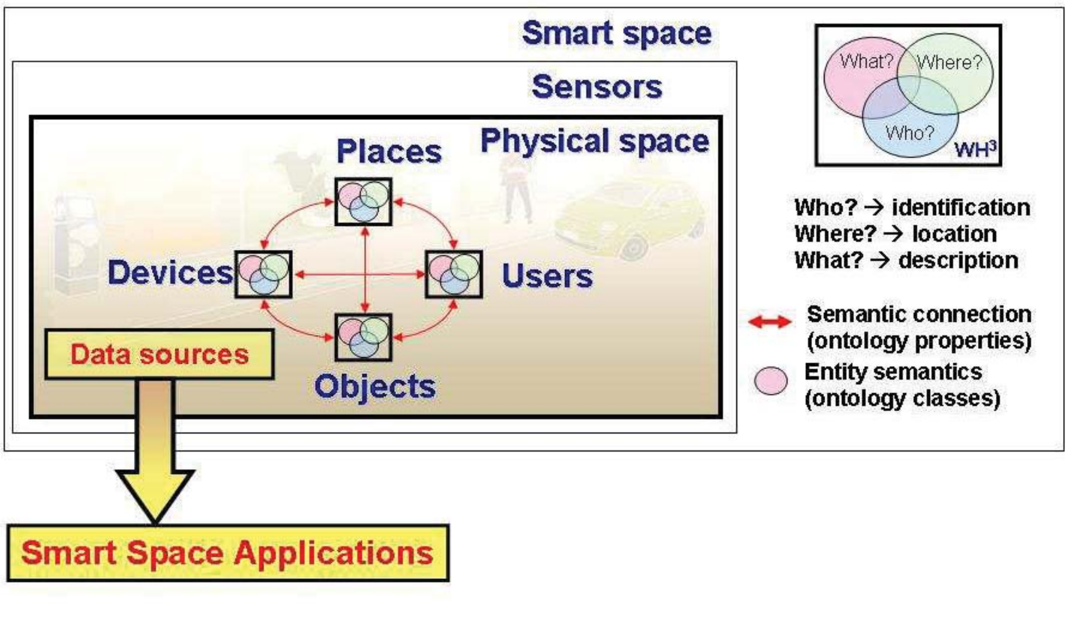

Figure 1. Connecting the physical word to the information word

The course introduction ended presenting the role of a smart space as a "*semantic interface*" between the physical and the digital world (see Figure 1) and introducing Smart-M3 as the reference smart space of the course (see Section III for more details on Smart-M3).

E. Evaluation criteria and procedure

The formal examination requirement is to evaluate the team participants individually, and the *evaluation criteria* - specified during the course - were identified as follows:

- **Level of understanding of the big picture** and of the role in the Information Society of **information interoperability**, **context aware computing** and **pervasive computing**
- **Level of understanding of the course goals** and required engineering practices
- **Motivation** of the proposed project
- **Individual contribution** to the same

To this end the *examination procedure* was defined as follows:

- Each project team was evaluated in a **single comprehensive session** partitioned into **five sequential and interactive presentations**, each one assigned to an individual student:
- The **first student** motivates and illustrates the project, supported by a **Power Point presentation** previously prepared, which includes the *scenario* and the *KP templates* (see Section IV)
- The **second student** discusses the project **ontology**, motivating the choices and describing the ontology definition process (see Section IV)
- The **third student** runs the **demo**
- The **fourth student** describes the **design process**, the **tools** used and the design itself
- The **fifth student** describes the **code**

The session has been very interactive and during each session useful feedback was gained to improve next year course and possibly the platform and the design process themselves. In the end, these examination sessions turn out to be interesting, fun and a potential opportunity to be run in large classrooms to demonstrate their value to a larger public. So far we did not have any public to the examination sessions and we evaluated just 2 groups on 8 (see Section II.B).

III. SMART-M3

Smart-M3 is an open source implementation of an RDF [16] based information interoperability system [2, 17]. The system consists of Knowledge Processors (KPs) connected to one or more Smart Spaces. The KPs share information by modifying information in a Smart Space by means of insert, remove and update operations. The KPs can also query for information using supported query languages, and set up persistent queries, subscriptions, which provide notifications to KPs when the query results change. In addition, there are operations for KPs to join and leave a Smart Space, which can be used to implement access control.

A domain model of the system is presented in Figure 2 describing the concepts of M3 Smart Spaces, and Table 1 shows an overview of the operations available to modify and access the information.

The Smart Space is realized by one or more[1] Semantic Information Brokers (SIBs) which together form a combined search extent of information. The KPs connect to the SIB by using Smart Space Access Protocol (SSAP). SSAP describes the protocol messages needed to perform

[1] The current implementation does not support multiple SIBs

the operations described in Table 1, and can be used over multiple connectivity mechanisms, such as TCP/IP.

Table 1. Basic Smart Space operations

Operation name	Description of operation
Join	Joins a KP to a named M3 space if the credentials match what is required.
Leave	Leaves a named M3 space. No more operations may be performed after a leave until a join.
Insert	Atomically inserts a graph into a Smart Space.
Remove	Atomically removes a graph from a Smart Space.
Update	Atomically updates a graph in a Smart Space. This is an atomic combination of Delete and Insert operations, where Delete is performed first.
Query	Queries for information in the Smart Space using one of the supported query mechanisms.
Subscribe	Sets up a subscription (persistent query) in Smart Space. The KP is notified when the subscription results change.
Unsubscribe	Cancel a subscription.

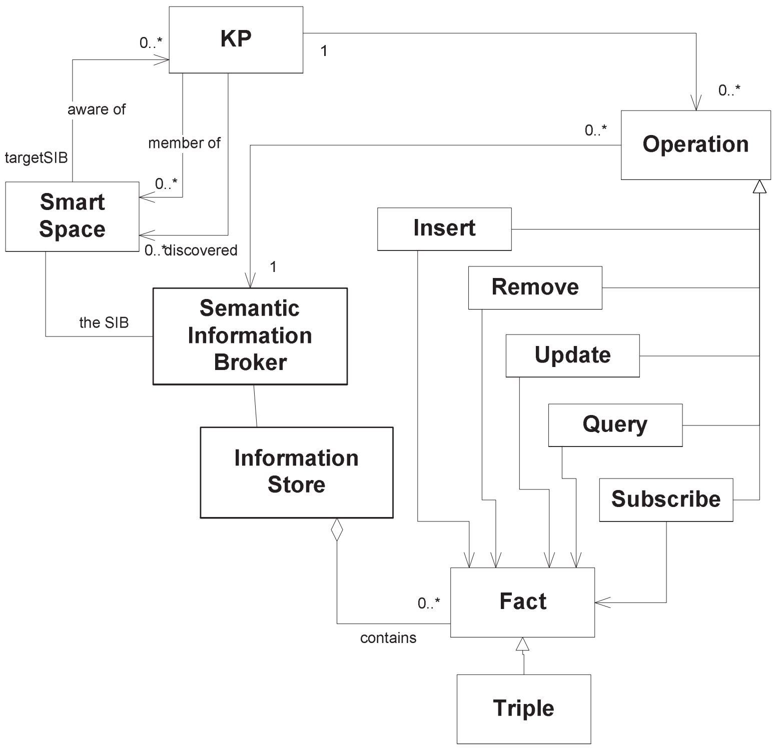

Figure 2. Domain Model of M3 Smart Space

Smart-M3 has implemented SSAP protocol KP side front-ends (KP Interface, KPI) for three different programming environments: GLib/C, Qt/C++ and Python. In addition, there are independent KP side SSAP implementations for Java [18], ANSI C [19] and C#.

In addition to the SSAP libraries, there are ontology code generators for GLib/C, Python, and ANSI C [20] which aim to ease the development work by automatically generating class definitions from ontology, allowing developers to program using familiar domain concepts instead of using SSAP operations and RDF directly. The Smart-M3 implementation architecture is described in Figure 3.

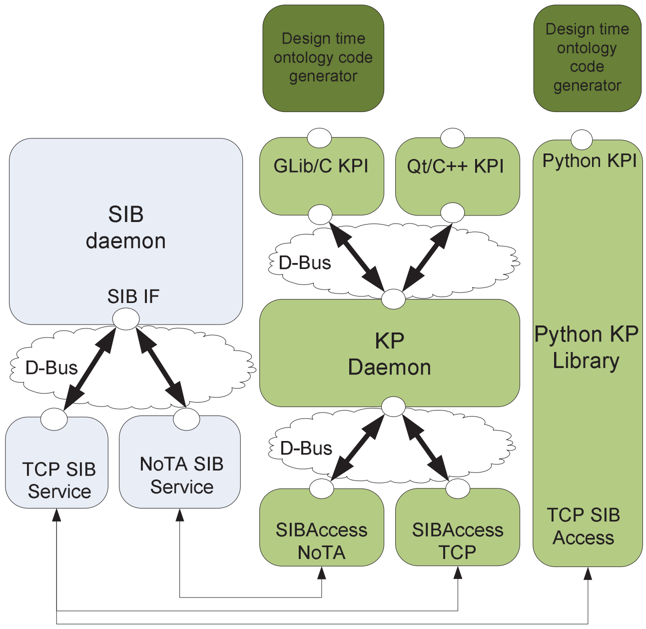

Figure 3. Smart-M3 Implementation Architecture

In the course, the SIB and Python parts were utilized, as well as the Java and C# KPI. Ontology code generators were not used in the course. Students have been free to choose on which programming language(s) develop their projects.

IV. Design style and tools

This section reports on the method that has been design to support student projects. It is composed by **four main parts**: a *scenario templat*e, a *Power Point presentation*, a *KP taxonomy* and the *KP templates*. All of them need to be implemented to properly describe a Smart Space Application (SSA). In the following subsections, each part of the method, i.e. *design style*, is presented with reference to a simple SSA example.

A. Scenario template

This template consists of **three tables** and it provides a *scenario description* (see Table 4) along with the list of the involved actors, i.e. *users* and *artifacts* (see respectively Table 2, Table 3).

Table 2. Users table. It reports end users and their roles. Every user has an unique identifier

F_ID	USER_ID	Type	Role
2.1	U1	General user	People moving around
2.2	U2	Monitoring operator	People monitoring the environment

Table 3. Artifacts table. It reports devices with the relative description and the associated owner

F_ID	A_ID	USER_ID	Type	Description
3.1	A1	U1	Nomadic device	Smartphone with RFID reader
3.2	A2	U2	Monitoring system	Centralized monitoring system with room level granularity

Table 4. Scenario description. It reports the executive summary by giving a short narravite description of the scenario

F_ID	FIELD_NAME	Value	Notes Do not fill this column
1.1	Short name	E_LAB_1	*Insert an acronym to be used as a reference in other templates (Max 8 Chars)*
1.2	Title	RFID Based User Localization	*A title for the scenario*
1.3	Short narrative description	A user, carrying on a nomadic device equipped with an RFID reader, moves within two environments. At the entrance of each environment it is placed an RFID tag. By reading a TAG, the user notifies the system in which environment is. The scenario includes the following entities: • 2 environments • 1 person • a nomadic device equipped with an RFID reader	*Use cases are short stories of envisioned real life situations that describe a specific user group in a specific contextual setting interacting with specific devices to fulfil specific tasks and intentions.*
1.4	Reference person	Guido Zamagni gzamagni@arces.unibo.it	*Insert name and contact of the team leader and team_ID.*
1.5	Date	14/05/2010	*Insert date of scenario creation*

B. Project presentation

The project presentation consists of several sections: the *application layout*, the *software architecture*, i.e. in terms of KPs and their relationships, the *hardware-software infrastructure*, i.e. on which device each KP runs, the *run-time data flow*, the *application ontology*, and the *ontology awareness* level of each KP, i.e. which are the instances that each KP is aware of.

As shown in Figure 4, the *application layout* aims to give a general graphical overview of the SSA. As reported in Table 4, the described SSA example is related to the localization of a general user (*U1*) who is carrying a nomadic device equipped with an RFID reader (*A1*). In this example there are two rooms, each of them identified by an RFID tag. The current user position, i.e. Room 1 or Room 2, is monitored by a monitor operator (*U2*) on a remote computer (*A2*).

Figure 4. Application layout

Once the application layout is presented, students are asked to describe the SSA in terms of KPs and their interactions with Smart-M3, i.e. software architecture (see Figure 5).

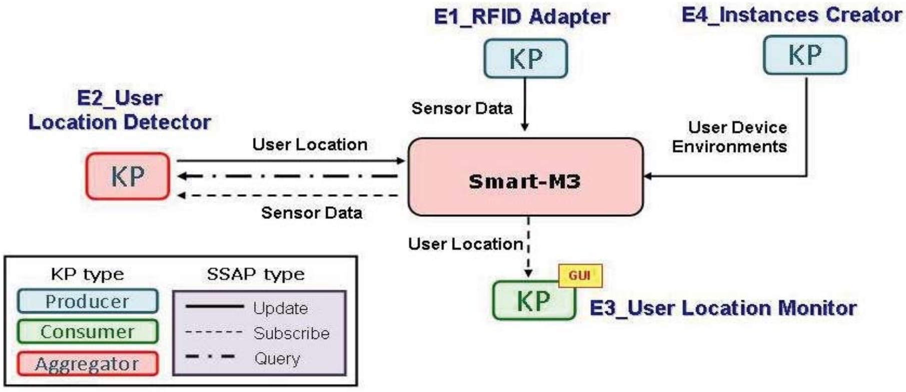

Figure 5. Software architecture

The next step consists in describing on which device each KP runs, in which programming language it has been developed and where Smart-M3 is located, i.e. *hardware-software infrastructure* (see Figure 6).

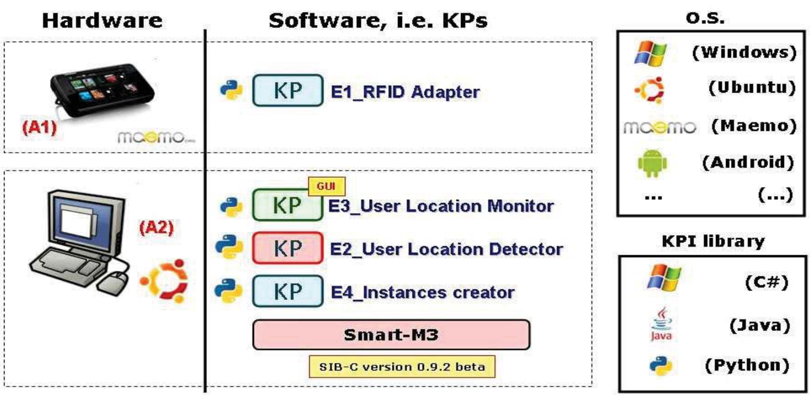

Figure 6. Hardware-software infrastructure

In order to have a general overview of the *data flow at run-time*, i.e. abstracting from the presence of the smart space, another description has been designed: as shown in Figure 7, here are represented the KPs, connected through labelled arrows, i.e. on each arrow is reported the "*concepts flow*" extracted from the software architecture (see Figure 5).

Figure 7. Run-time data flow

The following steps aim to present the data model in terms of the *reference ontology*, the *instances* and the *ontology awareness* of each KP. First of all, the ontology class tree is presented (see Figure 8). As Smart-M3 stores the ontology instances as RDF triples, the main RDF *instances* are represented showing the relevant predicates (see Figure 9). The project presentation ends by presenting the *ontology awareness* for each KP. The focus is showing which are the RDF triples a KP must be aware of. In Figure 10 is shown an example related to the "E2_User Location Detector" KP that acts as aggregator within the SSA example. Note that, as shown in Figure 10, at this step is important to underline if the KP modify some predicates, i.e. *semantic connections*.

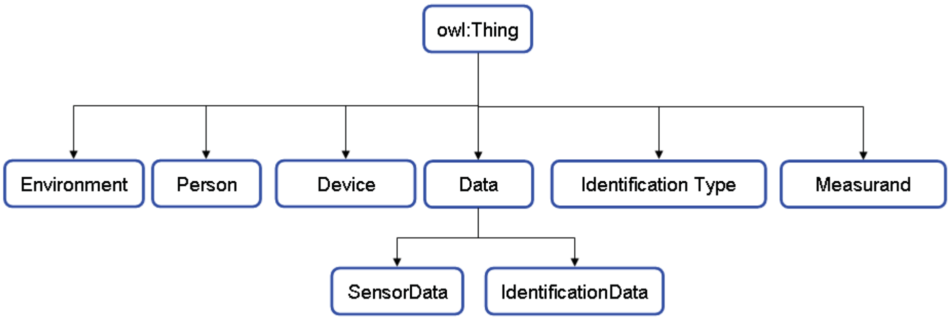

Figure 8. Ontology class tree

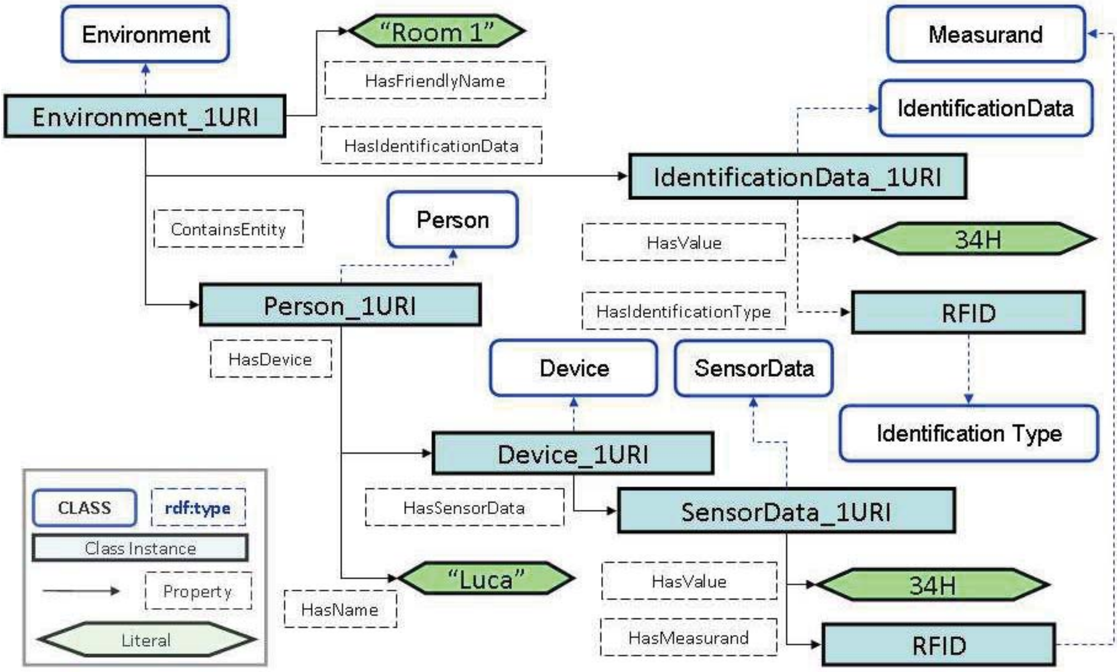

Figure 9. Instances

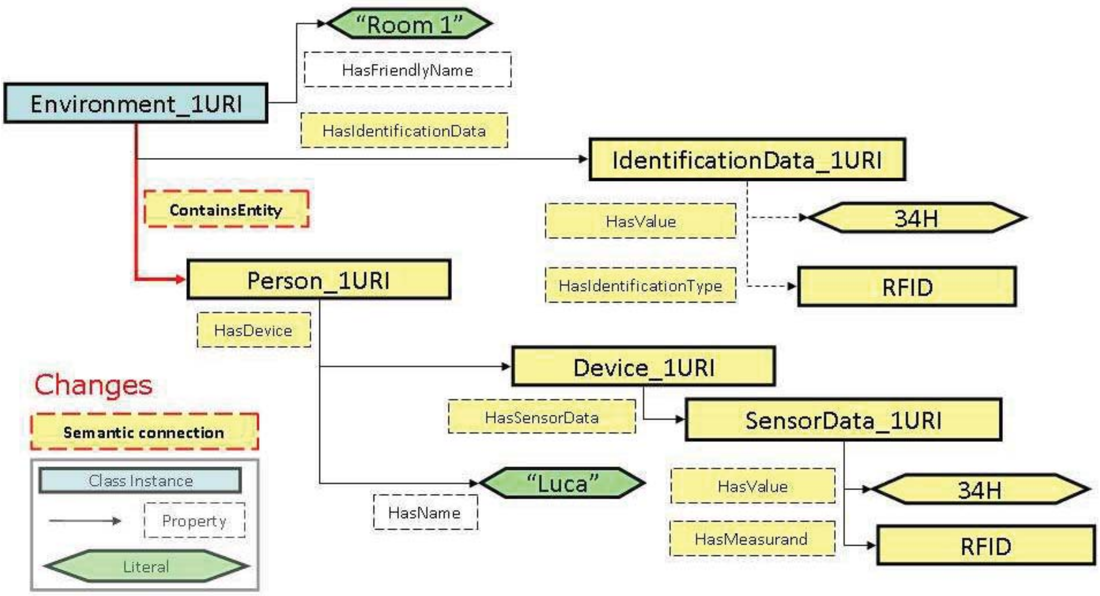

Figure 10. Ontology awareness for the E2_User Location Detector KP

C. KP Taxonomy

The purpose of a KP taxonomy is to facilitate the smooth growth of SSAs and maximize KP reuse by assisting the developer in searching a suitable KP from the KP repository and storing new ones therein. The work of building a KP taxonomy (and also a more general KP ontology) is taken in change within the SOFIA project. As shown in Figure 11, the three KPs of the presented SSA example (see Figure 5) have been classified according to a taxonomy.

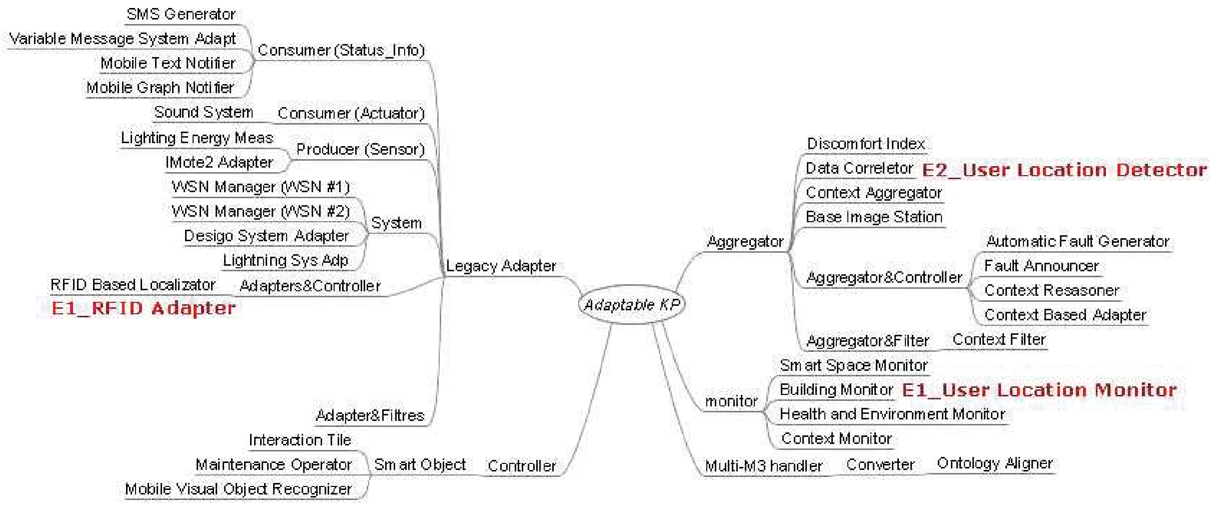

Figure 11. KP taxonomy: an example from a SOFIA public deliverable

D. KP templates

In order to have a quick reference for each KP, a template has been design. The template consists in **two tables**: the first is used to give a *short description* and a *name* to the KP; the second one is used to summarize how the KP interacts with Smart-M3 in terms of *exchanged RDF triples*. The second table includes also some *implementation details*, like the programming language used to implement the KP, if the KP exposes a GUI and the SIB version for which the KP has been developed (see Table 5).

Table 5. The KP template

		KP name
Executive summary		
F_ID	**Field Name**	**Value**
1.1	**Author**	*Author*
1.2	**Description**	*Brief description*
1.3	**Type**	*Producer/consumer/aggregator/ legacy adapter*

		KP name
Information Level and Semantic Interoperability		
2.1	**Ontology Awareness Required**	***Class_1*** *- properties_a* *- properties_b* *- …* *…* ***Class_n*** *- properties_x* *- properties_y* *- …*
2.2	**Information published on the Smart Space**	*List of RDF triples (subject, object, predicate)*
2.3	**Information consumed from the Smart Space**	*List of RDF triples (subject, object, predicate)*
System Level and Portability		
3.1	**Programming language and KP API**	*Python/Java/C#/C*
3.2	**Run time GUI**	*Yes/No*
3.3	**SIB version**	*SIB-C version 0.9.2 beta*

V. Conclusion

As conclusion, a new education approach was experimented. The real value of the course within the addressed curriculum may not be assessed yet. Certainly, a community of Smart-M3 developers was created. All of these students will have to do their six months postgraduate thesis next year. All of them are therefore candidates for research contribution to projects based on SOFIA Interoperability Platform. Therefore we may claim that a fruitful closed loop research → education → research was set up which could turn out to be quite interesting in the coming years. The students background was mostly software, and no hardware lab was available. Therefore, during this first course edition no hardware adapters to connect legacy devices to Smart-M3 could be developed, i.e. no real device was connected to the Semantic Information Broker. Devices and sensors were mostly simulated. This was a course limitation, which could be overcome in the future by admitting to the course not only information engineering but also electrical engineering students. We would like also to make projects on device and service level interoperability, experimenting different solutions like the one proposed by NoTA.

This experience allowed us to create a new team of Smart-M3 trainers that, along with the implement design style, should hopefully give a contribute to research and teaching on smart space and more in general on smart environments.

There are still some needs to be handled and we envisage topics and further extensions for the future. First of all, one of the aims will be to build a multi-domain smart environment by fusing together all the ontologies (and KPs) derived from each single project. This will bring to a “Smart-M3 Lab Ontology” supporting both current and future projects. We would like also to spend more effort on KP taxonomy modelling in order to have a large set of reusable and meaningful KPs. Finding synergies with other courses in the same term will be also a challenge: introducing agent systems technologies, AI algorithms and advanced reasoning mechanisms will be a promising way to enhance the smartness of future SSAs.

Acknowledgment

This paper is part of the dissemination activity carried out within the European Research Project SOFIA (http://sofia-project.org/) coordinated by NOKIA and funded by the European Commission through the ARTEMIS JU, contract n. 100017.
The course presented could not be held without the support from the SOFIA research team from NOKIA which greatly contributed with seminars, material, support and, encouragement.
A special thank goes to the 40 Cesena students who took part to the first run of the course, providing precious feedback for further education innovation.

References

[1] "SOFIA Project," ARTEMIS - VII EU FP, 2009-2012, pp. http://www.sofia-project.eu/.
[2] J. Honkola, L. Hannu, R. Brown, and O. Tyrkko, "Smart-M3 information sharing platform," *IEEE Symposium on Computers and Communications (ISCC)*, pp. 1041-1046, 2010.
[3] "Smart-M3 on Wikipedia," http://en.wikipedia.org/wiki/Smart-M3.
[4] "Protégé Ontology Editor," Stanford University School of Medicine, http://protege.stanford.edu/.
[5] "Moodle," http://moodle.org/.
[6] "Smart-M3 Lab@Unibo," University of Bologna (Cesena) - Italy, http://www.moodle.unibo.it/mod/wiki/view.php?id=10624.
[7] "VirtualBox," http://www.virtualbox.org/.
[8] T. Berns-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, pp. 34-43, 2001.

[9] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data - The Story So Far," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 5, pp. 1-22, 2009.
[10] "Extensible Markup Language (XML)," W3C, http://www.w3.org/XML/.
[11] "Resource Description Framework (RDF)," W3C, http://www.w3.org/RDF/.
[12] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," RFC 3986, http://labs.apache.org/webarch/uri/rfc/rfc3986.html January 2005.
[13] "RDF Vocabulary Description Language 1.0: RDF Schema," http://www.w3.org/TR/rdf-schema/.
[14] "OWL Web Ontology Language Overview ", http://www.w3.org/TR/owl-features/.
[15] I. Horrocks, "Ontologies and the semantic web," *Commun. ACM*, vol. 51, pp. 58-67, 2008.
[16] "Resource Description Framework," Accessed October 7th 2010, http://www.w3.org/RDF/.
[17] I. Oliver and J. Honkola, "Personal Semantic Web Through A Space Based Computing Environment," presented at Second IEEE International Conference on Semantic Computing, Santa Clara, CA, USA, 2008.
[18] "Smart-M3 Java KPI Library," Accessed October 7th 2010, http://sourceforge.net/projects/smartm3-javakpi/.
[19] "KPI low," Accessed October 7th 2010, http://sourceforge.net/projects/kpilow/.
[20] D. Korzun, A. Lomov, P. Vanag, S. Balandin, and J. Honkola, "Generating Modest High-Level Ontology Libraries for Smart-M3," presented at UBICOMM, Florence, Italy, 2010.