International Journal of Innovative Computing, Information and Control
Volume **8**, Number **12**, December **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 8415–8433

COST EFFECTIVE AND SCALABLE SENSOR NETWORK FOR INTELLIGENT BUILDING MONITORING

Michael Dibley, Haijiang Li, Yacine Rezgui and John Miles

School of Engineering
Cardiff University
Wales, CF24 3AA, United Kingdom
{ dibleymj; lih; rezguiy; MilesJC }@cf.ac.uk

Received July 2011; revised April 2012

Abstract. *Conventional sensor networks for building monitoring lack smart means to exploit collected environmental data to support relevant knowledge generation. Explicit domain knowledge plays the key role for software units to execute relevant tasks autonomously, which in turn can ease the increasing complexity embedded in the contemporary building management process. In many cases, the cost of multi-functional sensor networks is still high and the sensor system deployment still requires special expertise, which hinders their wider adoption for the existing or new buildings. This paper presents a cost-effective hardware design for multi-functional ZigBee sensor unit, which compactly integrates several different types of sensors. Together with a supporting multi-agent software framework, the application described herein can provide real time and smart building monitoring through querying the modelled domain knowledge. The contribution of the developed sensor network lies in its compact assembly, easy deployment, and the intelligence provided by elaborating the explicit building domain knowledge through the use of autonomous software agents. The ongoing testing shows a promising building monitoring network with the original intentions to identify wasted energy consumption in buildings, and further suggest better usage of building spaces.*

Keywords: ZigBee device, Ontology, Multi-agent system, Virtualization, Sensor

1. **Introduction.** Buildings play a key role in supporting daily human activities, and in order to make buildings better fit for their purposes while still behaving in an environmentally friendly way, effective mechanisms are needed to better monitor buildings, understand building operations, and further make timely decisions. Contemporary buildings are becoming much more complex than ever before in terms of their involved components and functionalities, which normally include environmental concerns, such as lighting, heating, power, water and drainage, together with security, fire and communication technologies. To cope with that level of complexity, deploying various types of sensors into buildings is a very effective way to collect real time environmental data in the first place. This, in turn, can be exploited by a building management system (BMS) to assist facility managers in their timely decision-making [1,2]. Conventional sensor networks for building monitoring lack smart means to exploit collected data to support relevant knowledge generation. Explicit domain knowledge plays the key role for software units to execute relevant tasks autonomously, which in turn can ease the increasing complexity embedded in the contemporary building management process [3]. In many cases the cost for multi-functional sensor networks is still high, and the deployment normally requires special experts, which further hinders its wider adoption in existing or new buildings.

Effective management of a deployed sensor network to inform and assist in the decision making process remains an important issue. The underlying software system plays the key role in the process to achieve an efficient and intelligent sensor system. It would be highly desirable to equip sensors with autonomous intelligent software units, which can conduct a certain level of reasoning by following pre-defined or run time rules or goals to negotiate with each other requiring related resources or information [4-6]. There are a number of existing agent based software systems developed for building sensor management [7,8], but only a few [5,9] combine a multi-agent framework with a real time knowledge base query (through ontology models) in the sensor system to provide intelligent software behavior, such as belief-design-intention (BDI) reasoning [10,11], within the context of building management. The ontology rendered agents solution provides an innovative way to automate some parts of the information collecting process, to pursue goals, to learn and build up the explicit knowledge within the relevant application domains, and hence to further simplify the building management tasks [3]. Real time decision making process requires expensive and dedicated supporting computing facilities, while contemporary commercial buildings are usually equipped with a large number of personal computers for routine jobs. Computing virtualization [12] can provide flexible and distributed computing capability to run certain types of computation intensive applications by dynamically sharing and clustering existing computing power [13,14]. The combination of sensor network with virtualization based computing infrastructure can provide a highly economic way to take full use of the computing resources existing within buildings.

In order to address the above mentioned research gaps, this project's main objectives were: (a) to produce compact ZigBee device prototype, to assemble cost-effective multi functional sensor unit, and to make the sensor network easily deployable; (b) to develop a supporting software framework, to provide BDI typed reasoning (supported by explicit domain knowledge captured in a knowledge base) to enable the software units to behave smartly and autonomously. This paper describes the process to realize those objectives. The paper contents are organized as follows. First, a brief review about existing sensor systems within the context of building monitoring is given, with a focus on compact sensor hardware design and intelligent supporting software implementation. This is followed by description of ZigBee device prototype produced in the project. Next, the multi-layer sensor system architecture is explained, including infrastructure layer (directly connected to sensors), virtualization based computing environment (to conduct simulations), and the ontology based smart software agent layer (to provide knowledge based reasoning). Due to the lack of standard testing procedure for ontological reasoning (core element for the system to provide intelligent software behavior) [15] applied in sensor network, a specific purpose oriented verification and evaluation process has been devised to test the sensor hardware design (stability and workability), the agent reasoning aspects (verified against the expected behavior), the integration with the underlying computing infrastructure (virtualized) and entire hardware/software system. Different sensor system testing locations are provided including a domestic flat and an open area (forum) in Cardiff Engineering School, and the test results reveal promising characteristics related to system robustness and intelligence. Finally, the paper provides a discussion of the results and concluding remarks.

2. **Brief Review for Related Work.**

Several existing sensor data communication protocols and technologies could be utilized to design sensor network (applied in FM domain), including serial communications protocol Modbus [16], building automation and control networks protocol BACnet [17]. In contrast to other wireless technologies such as Bluetooth that is used to connect high-volume devices, ZigBee technology is used to connect low-volume sensor components and is very cost-effective and easy to use. Bretolotti et al. [1] introduce a sensor system based on ZigBee technology. A main board is developed as the hosting device, and several purpose based boards ("daughter board") are produced to provide specific functions. Those developed "daughter boards" include temperature and humidity sensor board, accelerometer board, programming board and extra ZigBee-USB adapters used to connect with PC. The idea of "daughterboard" can be further improved, such as to integrate different sensors directly on the same ZigBee host board rather than using many sockets and heads – the new generation sensors are becoming smaller while without decreasing their performances, so it is feasible to build up a multi-purposes ZigBee device to host more functional sensors. Nonetheless, the corresponding sensor software system was not mentioned in [1] due to its reported early stage for development.

Menzel et al. [18] report a wireless sensor system used for energy efficient building operation. The sensor hardware prototype is based on Tyndall prototyping platform [19], and integrates several layers to incorporate a sensor unit, a data processing unit, and communication and power units. Zone controller was used in their testing deployment network to coordinate the communication between sensors. However, no further work has been reported, for example, how to integrate the developed system with the existing BMS (Building Management System); and there is no KR (Knowledge Representation) based "intelligent" components stated in their development to deal with sensor network operation and data collection. Cao et al. [20] developed a wireless sensor and actuator network to compare the two major approaches, centralized control (CC) and distributed control (DC). Two different schemes were specifically designed to compare decisions made by global information and local information. The result suggests DC is more robust considering data loss and has lower computational complexity than that of CC; the DC also has shorter actuation latency under several specific conditions. Therefore, it would be desirable to realize DC approach for a better sensor network.

Kim et al. [5] introduce ontology into wireless sensor networks by developing "service-oriented services". The use of ontology intended to provide certain level intelligence to sensor management – the developed ontology could respond to some natural language enquiry, such as "what is the temperature at current location?" Basing on some existing taxonomy resources, such as SensorML and sensor ontology, OntoSensor [21], three new classes named ServiceProperty, LocationProperty and PhysicalProperty were tried and manually added to form a new specific ontology. The semantic representation of sensor network becomes very important in terms of "precise interpretation is a necessary prerequisite for automatic search, retrieval, and processing of sensor data". Besides Kim, there are some other developments [9,21] combining ontology with sensor systems, which show that sensor systems driven by ontology can improve building management tasks by effectively reducing the complexity. One example is the use of a reasoner to direct behavior according to the explicit domain knowledge (provided by ontology services) where benefits are gained in the easier maintenance and understanding of modelled knowledge. At the current stage, there are still some challenges for the wider practical use of ontology within building management domain, such as (a) the existing of a plethora of taxonomies and ontological resources and constructs in the construction domain, (b) the lack of building-related ontology management and visualization environment, and (c) the lack of an adapted ontology validation framework for the construction domain [3,15].

In sum, this section briefly reviews the cost-effective and compact sensor hardware design and the underlying intelligent software implementation (please refer to [3,15] for more comprehensive review). The objective is to develop an easy deployable sensor network to ultimately deliver better and timely building management support. Based on the proved sensor network design methods and open published ontology development methodologies [4], the ZigBee technology has been adopted to develop a hybrid sensor network which is centrally controlled but with distributed execution capability. The contribution of the developed sensor network lies on its compact form, easy deployment, and the provided intelligence by elaborating the explicit building domain knowledge base through using autonomous software agents. Additionally the framework can conduct near real time decision making processes (e.g., performing building energy simulation tasks) through the virtualized computing facilities.

3. Cost Effective ZigBee Sensor Unit Design.

A number of building related communication protocols exist for the integration of building systems that include sensors, some of which are open standards. However, while the developed software architecture can easily accommodate the connection of suitable adaptors to standard protocols, for the small set of wired network devices connected, no standards conformance was required. Regarding the wireless devices, ZigBee hosted devices were selected. ZigBee is a wireless protocol standard that implements mesh networks specifically for hosting sensors and actuators. Consequently properties of the ZigBee standard are very low power consumption and low data throughput rate, which match the requirements of easy installation, allowing the use of battery power sources. Further specifications are low cost and high reliability and security. Most ZigBee device host offerings are supplied with a firmware stack implementation that offer the user an API at the application layer, providing node network configuration, data send and receive, and message routing, as well as other non network related functionalities.

The designed sensor hardware components provide near real time data to the building monitoring framework. The wired hardware (deployed in an office testing environment running for up to two years [3] with an intention to investigate initial feasibility and test some software algorithms) consists of a number of cheap sensors for motion detection, temperature and humidity measurement and the detection of the open or closed state of door and windows using magnetic proximity switches. In the wired setup, the devices used are not subject to the constraints of low power consumption and narrow voltage range operation as is the case with the wireless platforms, so almost any signal level device, either digital or analogue, can be easily connected to the USB interfaces used. The interface units used are from the National Instrument range, specifically the 6501 and 6009 devices. The devices are supplied with interface software that makes software integration in C# straight forward.

The rationale for parts selection lies on the specific requirements, such as very low current consumption for sensors. The ZigBee device also conforms to that, and was used as it has some processing ability, and wide on board peripherals – some interrupt driven channels, some analogue to digital converts, data channel, etc.; the firmware stack implementation made it possible to meaningfully control with external software. Table 1 shows a working list which includes part number or reference to the actual parts used in sensor system design (including both wired and wireless devices).

The wireless devices used have been specifically developed with the requirements to be easily deployable, having a small footprint – 60mm × 40mm, and battery powered. The platforms utilise a Zigbee host (the Ember/Telegesis ETRX357x product range). Sensors have been selected with the requirement of consuming very low currents. The Zigbee host device has 24 channels which can be configured as either input or output, some can be configured as analogue inputs and one can be configured as an analogue output. A similar set of sensors is mounted on the boards as the wired sensors mentioned above. A fairly comprehensive software interface using the AT device commands supported by the Zigbee host is used. The unit weighs approximately 30g, and two AA batteries used as the power supply. Figure 1 shows (a) an early assembled prototype, (b) milled host platform and (c) the corresponding electrical schematic.

Table 1. Working list for parts used in ZigBee set design

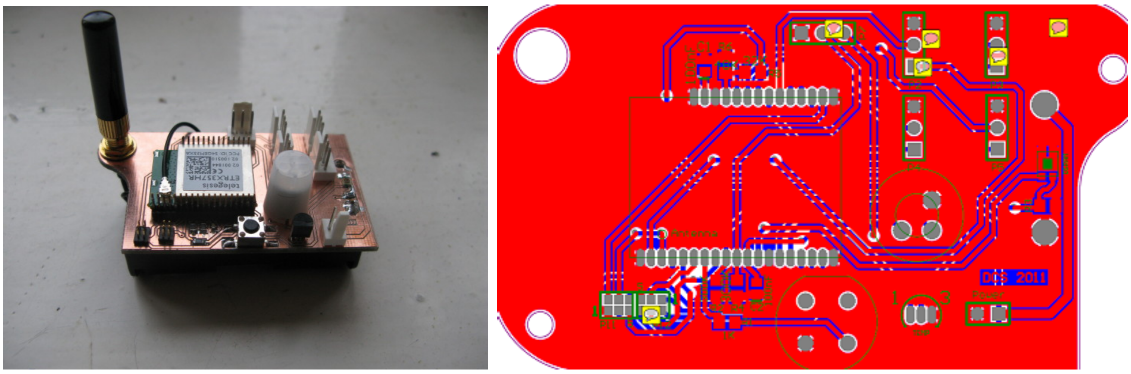
Item	**Part/supplier**
Zigbee module	ETRXn/telegesis
Antenna + connector	Various/telegesis
Osram lux sensor Ambient light sensor w/logoutput, SFH5711	654-9078/RS
Temp sensor Temperature Sensor Analog Serial 2-Wire TMP37FT9Z	709-2772/RS
PIR sensor 5m Spot (truncated cone) (white)	61-1510/Rapid
Battery box 2 *AA 2 X AA BATTERY HOLDER KEYSTONE	18-3683/Rapid
Zigbee module header 1.27mm straight PCB header 40W	254-6312/RS
Zigbee antenna connector	
1.27/1.27mm header 10 way Header 2x10way DIL VERT Pin	681-1193/RS
Reset switch ROUND GREEN KEYBOARD SWITCH/SQUARE YELLOW KEYBD. SWITCH	78-0155 - 78-0265/Rapid
Molex Header 2.5mm WTB, vert, friction ramp, 3w	687-7213/RS
Molex Header 2.5mm WTB, vert, friction ramp, 2w	687-7219/RS
Resistor 32K4, 0805 0.1% 25PPM 0.1W	**1575962**/Farnell
Resistor 48K7, 0805 0.1% 25PPM 0.1W	**1575980**/Farnell
Resistor 0805, 5%, 1K00	**1739229**/Farnell
Capacitor 0603, X7R, 16V, 100NF	**1833863**/Farnell

4. Supporting Software Architecture.

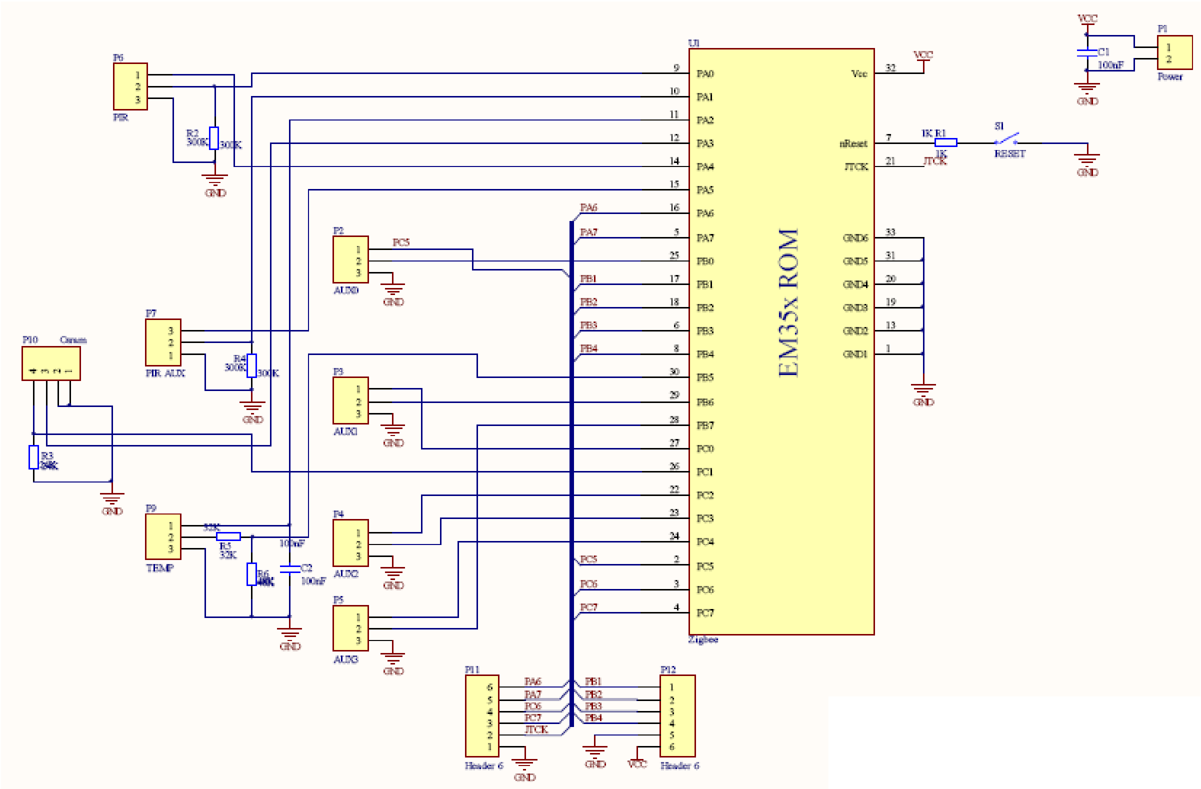
The developed sensor system [15], including hardware and supporting software, has the generic applicability in the scope of monitoring the internal environment of buildings using sensors. It supports building oriented information generation, accumulates and learns knowledge from low level data with semantic elaboration, assisted by a number of ontologies. The multi-agent software architecture is based on the JADEX [22] framework (to realize BDI agents [10]), and the Pellet reasoner [23] delivers reasoning capability to those agents. New agents can be easily integrated either following the existing patterns or introducing new implementations for the pursuit of additional use cases. The agent layer realises goal directed entities that interact with the resources available in the framework rather than simply a data logging facility. The modular infrastructure allows easy integration of any sensors, sensor groups, actuators or devices. A dedicated agent type (sensor node agent type) configures each node dynamically in the network as well as controls and manages connected devices. The agents generate output in text form including the form of OWL [24] statement which is well suited for consumption by other tools. Figure 2 shows the simplified multi-layer architecture.

4.1. Infrastructure layer (directly related to sensor hardware).

The infrastructure layer comprises of interface software for a range of deployed sensors and devices as well as management computers responsible for data logging and supporting services such as registration. The wired sensor and device interfaces are realized with a number of executables reading digital or analogue data from USB or RS232 ports and configured with an .xml file. On starting up, each interface locates and registers its attached sensors and devices with a sensor node executable and periodically, or according to pre-configured



(a) an early assembled prototype (b) milled host platform before population/assembly



(c) sensor unit schematic

FIGURE 1. ZigBee sensor unit hardware design

criteria, updates the sensor node with the contents of the local buffer and further supporting information. Similarly the wireless network interfaces register its hosted devices with the sensor node. Currently all sensor interfaces and sensor nodes are implemented with C# running on Windows platforms, and class based communication is realized using the Microsoft .Net Remoting framework. Sensors on any platform running Java though can be integrated by using .Net Remoting's customizable protocol and Java's RMI/IIOP [25]. Interface software that decodes standard building management system protocols such as BACnet [17] could be integrated with sensors that form part of those systems – these interfaces will similarly register available system devices with a sensor node agent type. The type of sensors currently connected includes temperature, motion detection (PIR), proximity switches on doors and windows, humidity, ambient light and irradiance sensors. Actuators are supported but currently are only used to control sensor power. The classes

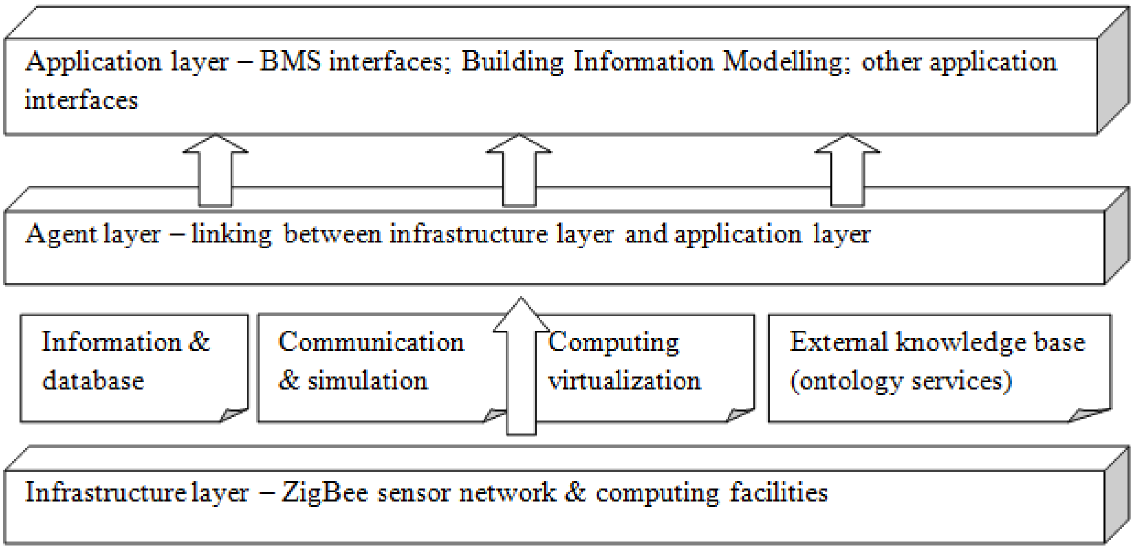


Figure 2. Simplified system architecture

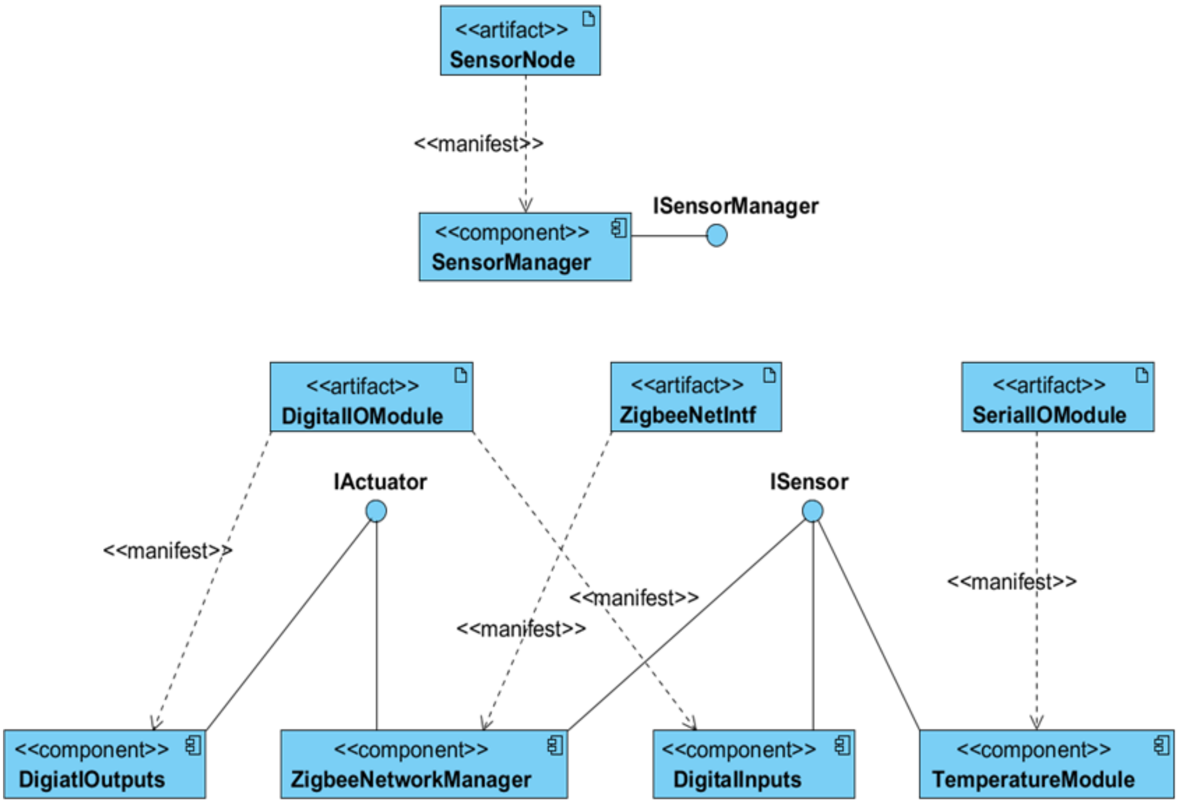


Figure 3. High level interfaces and components in the infrastructure

capturing sensor history which realise persistence were generated using the NHibernate [26] object relation mapping framework so as benefits from database performance enhancement delivered by those libraries. The main high level interfaces and components of the infrastructure are shown in Figure 3.

The wireless interface module allows sensor data sampling, actuator control, network control and wireless node management and configuration. The Zigbee host device ETRX3-57x hosts a range of sensors and communicates to a controller device which is in turn connected to a host PC either over USB or Ethernet. The Zigbee Net Interface module implements a number of node behaviors that can be assigned to each node. The behavior manages the issuing of sequences of commands and maintains the long term state of remote nodes. Read and write commands to carry out a wide range of operations are implemented as classes. Those commands are built from the Telegesis AT command set . The commands are issued to the network manager which manages the handshaking over the wireless network to the target device. Extensive use is made of a REGEX compliant parser [3] for the interpretation of commands. The AT commands typically consists of reading and writing to the target node registers to complete actions realizing the reading of data, invoking a node action or the setting of configurations. The microcontroller has resources such as timers and interrupts which the commands use to achieve a range of actions. Nearly all commands provided by the interface are asynchronous to render simpler client implementation. Figure 4 shows a wireless sensor deployment supporting a range of sensors on an ETRX357x platform.

4.2. **Virtual machine based computing environment.** The aim of creating a VM based computing environment to support smart building management is to fully utilize any existing computing resources within the same building where the sensor system is deployed; and at the same time to have access to the remotely located high power computing resources, including clusters, GRID computing or Cloud computing facilities. The developed computing system actually evaluated three different kinds of application scenarios (1) using non-dedicated and loosely controlled computing resources in buildings; (2) using relatively high specification dedicated and closely integrated computing resources in buildings; (3) using remotely located high power computing resources to support the relevant building decision making processes. The virtual computing environment works

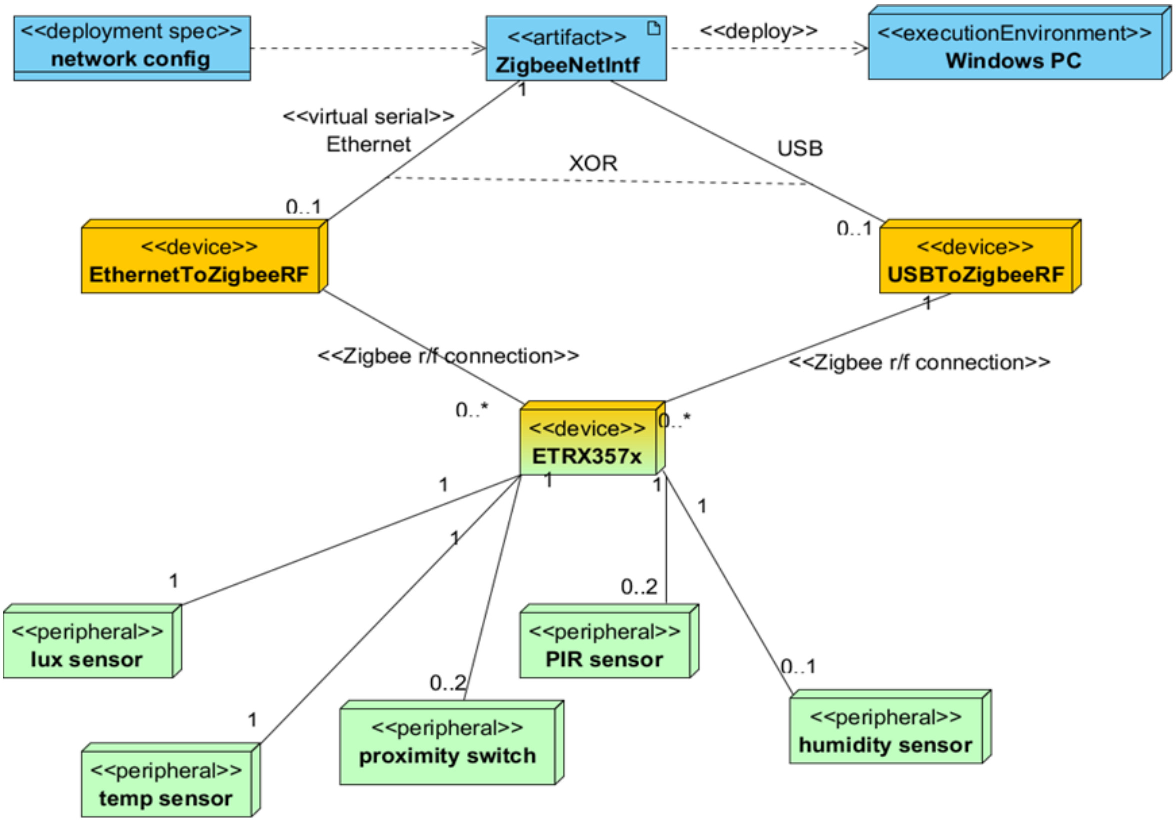


FIGURE 4. A wireless sensor deployment diagram

alongside with the physical computing system in the backend without affecting those normal computing tasks conducted within the same building, that provides a highly cost effective way of utilizing existing computing resources to provide much higher integrated computing power while at the same time without actually buying more physical computers and hence spending more electricity to achieve the comparable computing power.

Depending on the hardware specifications of the hosting physical computers, the feasible deployment of VM technology could easily double (or even more) the physical computing power for extra use. Different VM implementation technologies, including VMWare, VirtualBox, Hyper-V, KVM, etc., have been investigated and Fedora KVM platform is currently selected due to its good performance on the selected hardware platform, easy setting up and good integration with underlying operation system. Figure 5 below shows the connection between VM based computing environment and the deployed sensor system. Whenever computing requirements are needed (initiated from the sensor based building management system), the corresponding VM resources will be invoked to take actions and return with the calculated results. Depending on the details of the computing tasks, the computing execution could be serial/single run, serial/multi-runs, or in parallel way for highly computation intensive tasks.

Based on the currently integrated 8 physical PCs (each has a quad-core CPU and 8 Gigabit memory), using Fedora KVM platform, together with NAT/VPN/Port forwarding, 24 virtual machines (3 extra virtual machines for each physical computer) have been built within the same local area network and serves as the enclosed core computing services with different communication ports opened (through local firewall) for Internet. The specifications for virtual machines are generally the same −2G memory and 60G hard drive, 1 CPU core for each, and each virtual machine can provide several dedicated software services steadily. Altogether the entire virtual computing layer is able to provide enough computing services for normal building management related simulation tasks. The real time VM migrating character makes it possible to provide continuous and robust purpose-focused services dynamically. Together with the developed computing agents, the computing environment becomes a highly flexible, running as required, service oriented and intelligently controlled computing hardware platform.

4.3. **Ontology supported agent layer.** The underlying software utilises the JADE framework that supports multi agent system (MAS) implementations [27] together with an add-in known as JADEX [22] for the realisation of belief-desire-intention (BDI) [28] based agents. JADE provides support for agent infrastructure incorporating FIPA messaging,

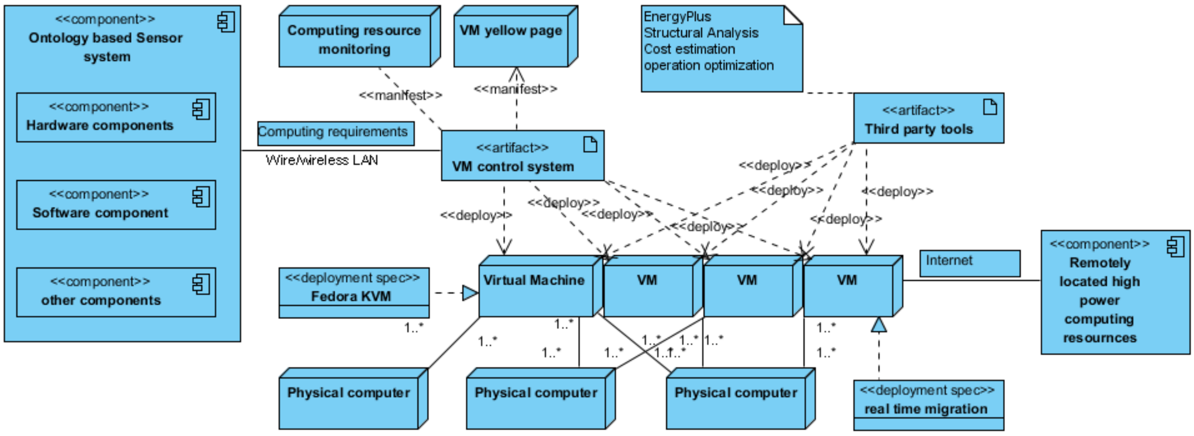


FIGURE 5. Virtual machines based computing environment

agent hosting, lifecycle control, and other infrastructure services such as agent location. Library supporting FIPA-agent communication language (ACL) conformant messaging is provided for message construction and transport but no semantics are forced. Additionally the agents' internal architecture is left open by JADE, allowing flexibility in the application. In the BDI model, agents are utilised but the framework is not constrained to that so custom agent architecture could be used instead. In order to facilitate some simplified (limited) manipulation of semantic information in agents' belief bases, and for inclusion as content of some messages, knowledge in ontologies has been converted to Java class instances. The Jastor project libraries is such a provision, and has been used to generate Java Beans class representations from Protégé [29] authored ontologies. Finally, JADE provides a basic FIPA-SL support library for easy composition and parsing of SL expressions in Java. In sum, the agent implementation shows the following characteristics:

- Supports a variety of sub domains where the entities specialise in that scope.
- Agents build on information that they seek affecting further inference.
- Agents are able to learn from past experience and improve their behaviour, not simply repeating or retrying earlier action that failed before.
- In the MAS as a whole, equilibriums are established that balance collective performance and resource use. For example, the sensor node agent has the goal to minimise resource utilisation while providing adequate support for clients to pursue their goals. With multiple requests the provider is able to reorganise resource provision to deliver benefits from 'economy of scale'.

Many of the developments here can be reused in further framework provision. The system adds a resource lease implementation that forms the basis of the resource negotiation and provision facility. In the scope of the BDI abstraction, the framework adds the notion of commitments that help to add stability and adherence to strategy. Table 2 describes in high level terms, the responsibilities of the agent types that could be re-used.

Table 2. Re-usable agent types and their high level responsibilities

Agent type	**Main responsibilities**
Zone Agent	Elaborates its (semantic) beliefs through the pursuit of goals, combining and building on its knowledge. Requests resources (sensor leases) to work toward goals as needed. Collaborates with neighbours. Some gaols may be impossible to complete without collaboration or improved efficiency may be gained. The agent may also use collaboration for verification of conclusions.
Sensor Node Agent	Controls the (finite) provision of resources, reconfigures devices dynamically. May refuse or substitute resource provision. Communications between the infrastructure implemented in C# and the Java implemented sensor node agent is achieved with a customised .NET remoting channel employing the IIOP protocol.
Other Agents	Interface to tools, interface to users, location lookup service ('yellow pages') to assist collaboration and brokering.

Several ontologies to support agent behaviour were developed and then extended for specialised agent activity [3]. For the extension of system functionality, much of the ontology content can be reused to support further agent behaviour; all the ontologies are expressed in OWL (Web Ontology Language,), and typically use the full expressivity proved by that language. Several hierarchical ontologies supporting agent behaviour are listed below and demonstrated in Figure 6.

- A building ontology capturing the building geometry and assembly. The origin of the taxonomy was the IFCs . Theories of topology and mereology have been integrated.
- Sensors ontology describes the sensors, the phenomenon that they capture, sensor capability and platform configurability. This is influenced by OntoSensor, which in turn is based on schemas in the SensorML modelling language.
- General purpose ontology, SUMO that captures domain independent concepts. Although some central concepts inherit from SUMO entities, a large proportion of the provision is not used but remains available if required.

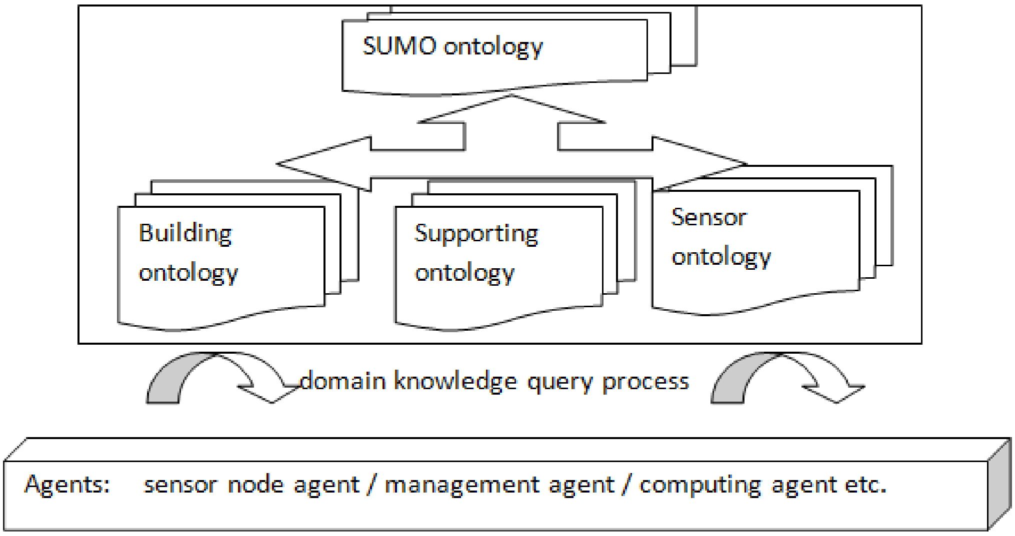


Figure 6. The developed ontologies and the relationships

The main purpose for developing a hierarchical ontology library [15] in this project is to find the feasible route to input certain intelligence into FM domain practice, and this was also required by an industry sponsor for this project. The original expectation was to fully utilize most of the existing resources, including full set of SUMO and IFC conversions. Due to the constrains mentioned above, and the limited time and broad range conducted in this research (numerous topics and related tools have been explored in order to achieve practical while advanced functionalities required by industry sponsor), the research did not target at building up an entire generic ontology for publishing purpose, instead the focus has been laid on the purposed oriented to make the developed ontology manageable, easy maintainable and meanwhile extensible.

5. **Sensor System Evaluation.** In the developed sensor network, the core element delivering expected intelligence lies on the purpose-oriented ontology development and software agent reasoning. As mentioned above and concluded in [15], at present there is a lack of effective ontology validation framework that can be used to test the "intelligence" provided by the developed framework. Therefore, a specific framework evaluation procedure has been devised, including sensor network deployment, sensor nodes communication/negotiation, and agent reasoning (to elaborate the domain knowledge). Figure 7

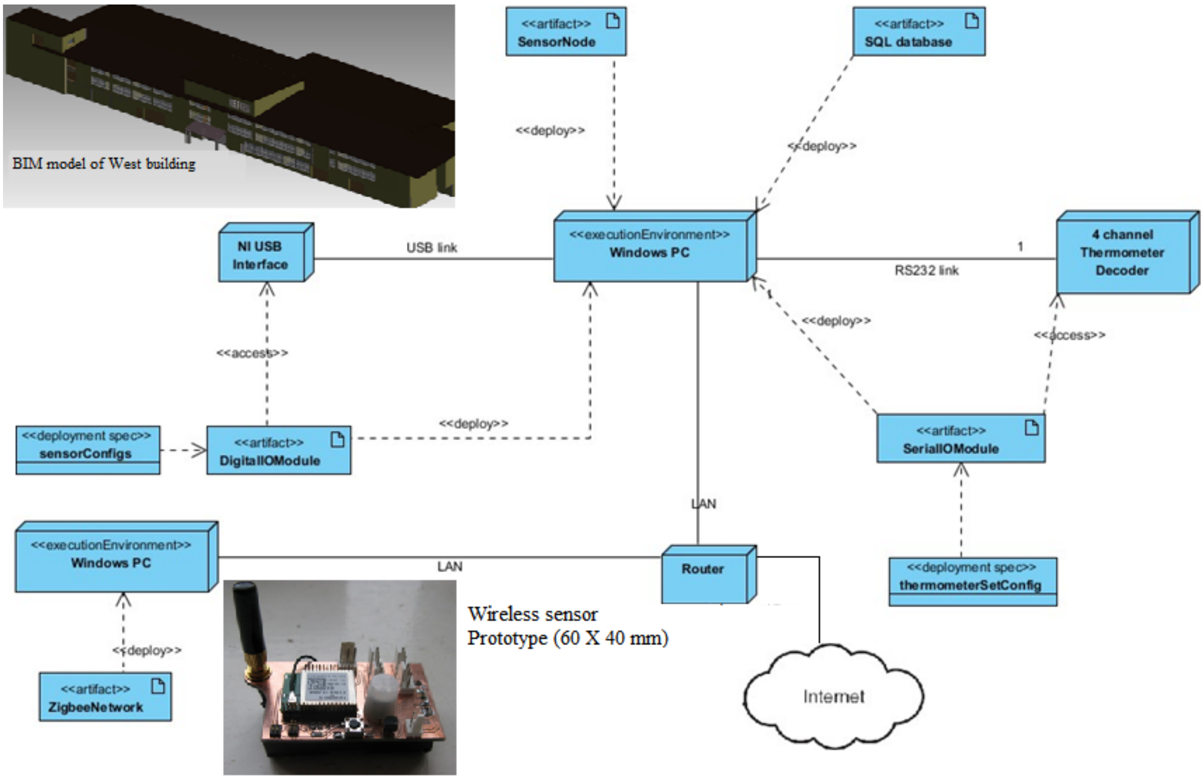


Figure 7. A sensor system testing deployment

shows a sensor system testing deployment including two PCs, one running a sensor node with which devices register, and the second running a wireless host, which also registers its devices with the sensor node on the first.

The generic iterative software/hardware developing process has been utilized to gradually tune the system to better fit for its developing purpose. Several ongoing tests are being conducted, including a domestic flat, a large meeting area (forum) in engineering school, Cardiff University. Figure 8 shows an excerpted IFC model focusing on Forum area.

The rationale for each testing environment differed. The first, the domestic flat, was used for initial development and was a location where hardware could be temporarily fitted if necessary without concern for appearance. The second, a more realistic and large scale deployment, the university set of offices and rooms, was used for later stage evaluation. The objectives of the second deployment were to provide a more complex and challenging testing environment (in terms of building geometry, sensor deployment and space usage), make further iterations in development to improve performance, check flexibility and evaluate robustness. The deployment was kept to a realistic level avoiding an artificially high density of sensors. The exact (fine) positioning of sensors and wireless host platforms is not critical but regarding general positioning, adequate provision is made to allow for all the testing scenarios required, and the details are given in the sections below. Most motion sensors are used in multiple simultaneous roles, especially where there are adjacent zones with associated agents.

The greatest variation in zone agent type behaviour arises due to the evaluation of occupancy goals, ranging from asserting that a door or opening has not been used to occupancy detection and counting. The deployments therefore aim to test a range of building geometries for that purpose, as well as to test some of the other zone agent type capabilities. In the larger deployment, a number of different types of room were selected,

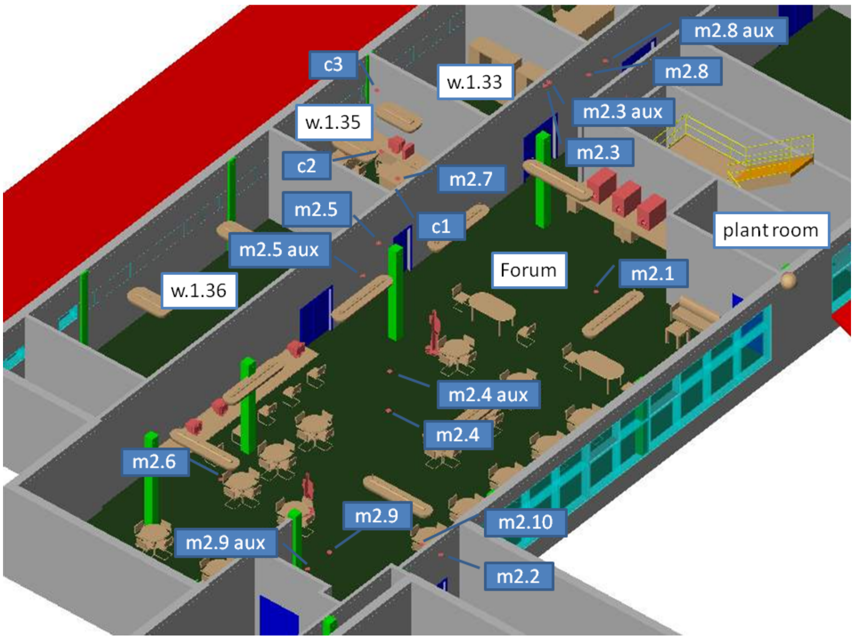


FIGURE 8. An excerpted IFC model focusing on Forum area [15]

from those with simple geometry, to more challenging spaces such as the ‘Forum’ room. The latter has numerous and different types of openings and occupancy patterns. For all cases, the wireless devices host ambient light, two motion sensors, and temperature sensors, additionally some platforms host additional proximity sensors mounted on doors. The wireless platform has, however, been designed for easy expansion in that spare channels available and the provision for easy electrical connection has been made. Table 3 shows the hardware deployment details.

A bottom up ‘glass box’ approach where detailed knowledge of the implementation is used to derive test plans from execution paths was the approach generally taken in preliminary unit and early integration testing. The smallest test units are those software entities that are defined by class boundaries. These units, together with assemblies of units including a common façade, could be tested without too much overhead to write test harness code to create input and realistic contexts. Where the creation of test harnesses was not considered worthwhile, and especially if combinations of units could readily be debugged after integration, small assemblies were tested and evaluated together, typically by ‘hard coding’ some of the supporting units to give predictable responses.

The unit and early integration tests identified implementation errors and indicated the performance characteristics of the units and unit assemblies. Integration testing in general gave more insight into the system as a whole and made a larger contribution to the later stages of the software development lifecycle. The integration tests covered functionality including general start up and initialisation of the infrastructure and agents, location functions, registration (yellow pages and other inter-agent), message exchange, and ontology interaction (updating and querying) with the ‘in memory’ knowledge bases. As well as testing the infrastructure and agent layer software, the testing necessarily incorporated the supporting artefacts including the ontologies and the IFC models. Tests were configured for typical and worst case loading in order to confirm adequate system performance.

Table 3. The hardware deployment at the university site

Sensor unit id	Attached sensors	Capability and rationale
m2.8	Spot pir, temp, lux, aux spot pir	Observes a virtual opening from corridor into Forum, person counting capable and environment monitoring
m2.3	Magnetic door switch, temp	Activity monitoring of the door between Forum and w.1.33, assumed no internal access to w.1.33 so allows the Forum agent to still perform person counting under some established conditions thus demonstrating practical flexibility
m2.5	Spot pir, temp, lux, aux general purpose pir	Environment monitoring, participate in zero (person) occupancy detection with the general purpose aux pir, and can participate in tracking (2 nodes)
m2.4	Spot pir, temp, lux, aux general purpose pir	As above but has central location so enhances the role of zero occupancy detection
m2.1	Spot pir, temp, lux, aux wide angle pir	Environment monitoring, and participation in zero (person) occupancy detection with the wide angle aux device which is centrally located
m2.6	Spot pir, temp, lux, aux wide angle pir	As above
m2.9	Spot pir, temp, lux, aux spot pir	Observes double doors giving access into Forum which are often propped open, person counting capable and environment monitoring
m2.10	Spot pir, temp, lux	Observes double doors giving access into Forum which are often propped open, in conjunction with 2.2 can perform person counting. Additionally environment monitoring. Very easily deployed configuration without aux wired devices
m2.2	Spot pir, temp, lux	As above
m2.0	Spot pir, lux	Observes spring loaded door. Person counting capable in conjunction with magnetic switch in c1. Also ambient light monitoring
c1	Magnetic door switch, temp	Detects opening of door between Forum and w.1.35. Also temperature monitoring
c2	General purpose pir, temp	In w.1.35 for environment monitoring, person occupancy determination and can participate in zero (person) occupancy detection with the general purpose pir. Also temperature monitoring
c3	General purpose pir, temp	As above

In particular, integration testing identified where performance was inadequate. Workarounds took the form of adjusting timeouts or buffering information. The latter implementations either involved buffering of inferred knowledge from the ontologies, or buffering of external events while the reasoner executes. Integration tests also gave insight into realistic deployment contexts (testing with various IFC models), which is particularly relevant to the choice of KB inference used. Significant reduction in reasoning time was achieved as expected by using the less expressive, e.g., reflexive RDFS inference, for simple sub queries, but its scope of application was severely constrained. All the library functions implemented allow the passing of an ontology model that is appropriate to the context, so the appropriate models were reconfigured where necessary. Typically agents retain in their belief base a handle to several ontology models with various attached reasoners.

The sensors ontology alone is primarily used by the sensor node agent for dynamically configuring the ZigBee nodes and attached sensors in the wireless networks, as well as generally locating resources in order to supply clients with requested data. Constructs to place the sensors (or other entities) into a context are the scope of the buildings ontology, and for modelling that context the sensors ontology is imported into the buildings ontology. A small excerpt of the sensors ontology is shown in Figure 9. The figure captures a subset of the asserted model, i.e., no inferences are shown, the purpose is to show an overall picture for sensor ontology constructs (please refer to [15] for more details).

The above mentioned implementations have been tested, and the core functionality is stable although deployment in other contexts should initially verify adequate operation. Some components of the developed sensor system have been tested for up to two years, and other aspects have been developed fairly recently and have been unit tested or deployed on a small scale. Several aspects to address agent performance, particularly during reasoning with ontologies have delivered good results. The tests exploit a number of ontology models with different reasoning semantics, utilising full inference semantics only where required. In essence, the corresponding sensor software system has been developed as background software 'services', there is no need for GUI for its running. The collected interfaces included in Figure 10 are only used to show the system running information (there is no specific hierarchical relationship included) – (a) shows the sensor network communication; (b) shows the real time data collection; (c) shows the run time ontology (knowledge) inputs; (d) shows the extraction process for building geometry information – two test locations are included: university "forum" and domestic flat; (e) shows one application – the intelligent service has been integrated in an existing FM solution to

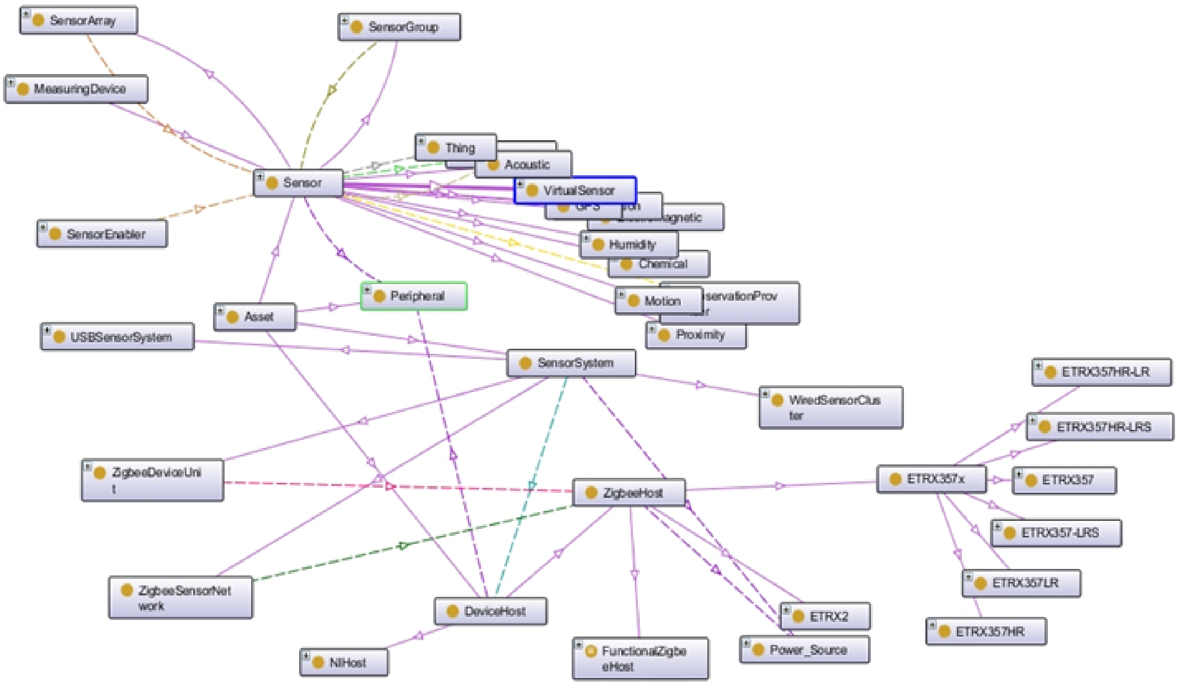
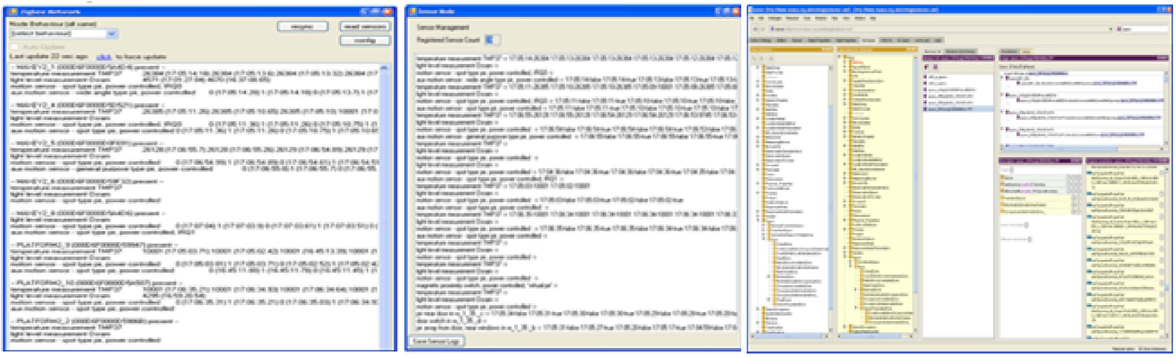
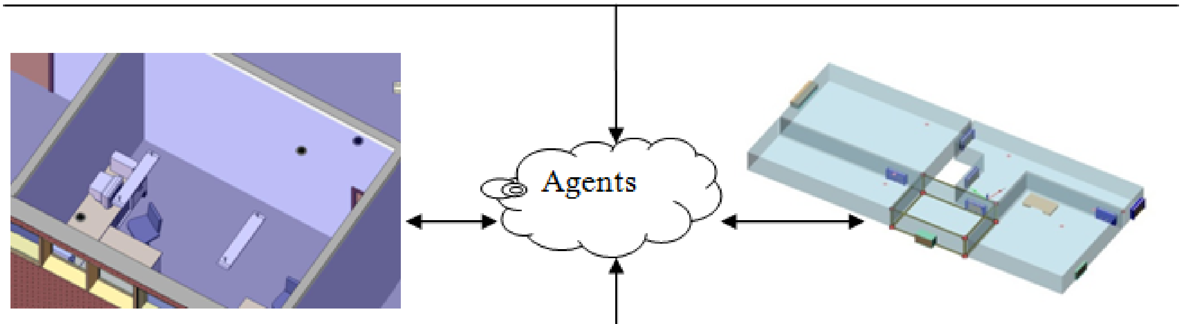


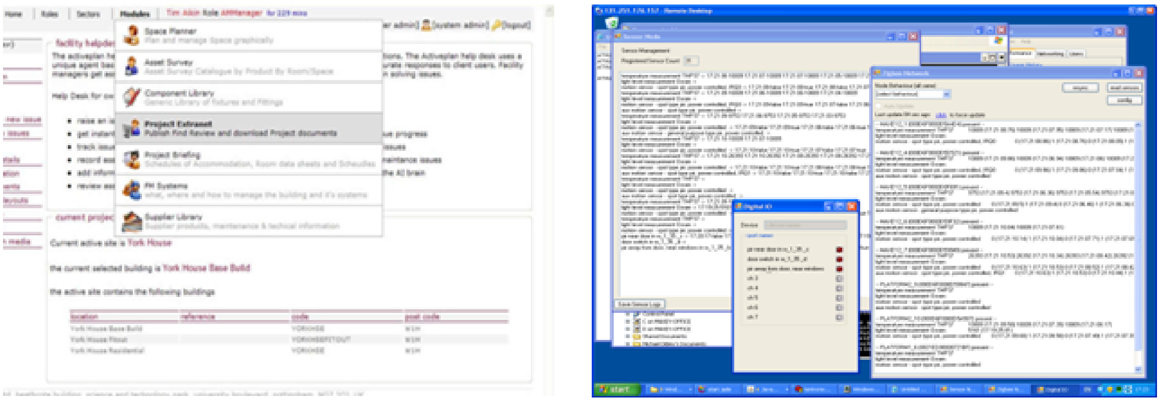
FIGURE 9. Excerpt of the sensors ontology


(a) sensor network communication (b) real time data collection (c) ontology input (ontology model)



(d) sensor test in an office (sensors in black points)

sensor test in a domestic flat (sensors in red points)


(e) Web based Helpdesk BMS

(f) remotely controlled ZigBee network

FIGURE 10. Collected interfaces showing system execution

provide "help desk" function to facility manager regarding the space usage and energy consumption; (f) shows an Internet based interface that is used for remotely monitoring the system running. The data collected from sensors have been stored into a database, and the dataset has been accumulated for more than 2 years.

Figure 10 shows the entire system running procedure, which has been used for framework evaluation. The evaluation process includes two stages: (1) preliminary testing and (2) later framework integration. The testing results coming from step (1) have been continuously fed back to further improve the integration process in step (2). The details of selected tests for the sensor node agent type are shown in Table 4. Testing with the senor node agent in the university deployment handled higher data throughput so that agent was used in order to derive conclusive results for tests.

Table 4. Selected tests for the sensor node agent type

Functionality	High level details	Test case/s – selected illustrative example/s	Result
Locate infrastructure services and maintain connections (infrastructure sensor nodes and Zigbee network interfaces)	Periodically poll IIOP endpoints for new resource provision	Standard operation. Shut down infrastructure elements and check for reconnection after restarting	Working as expected
Advertise resources	advertise resources provided via infrastructure, maintain associated resource provision	DF agent registry	Working as expected
Monitorz infrastructure	Extract new events from the infrastructure. Condition events as appropriate. Notify lease holders of new events and service requests for reading devices 'on demand'	Standard operation	
Listen, action and reply to requests	As above. Requests trigger setting of goals and behaviour to typically retrieve data from the infrastructure, deliberate about and action lease requests	Reception of requests from zone agents	Working as expected
...			

6. **Conclusions.** The research described in the paper delivered: (a) a ZigBee sensor set that compactly manufactures several different types of sensors together, including temperature, humidity, motion and light sensors, while addressing the aforementioned requirement for compact, cost-effective, multi-functional and easy deployable sensor network; (b) an intelligent supporting software framework, which provides BDI typed reasoning (supported by external domain knowledge) to enable the software units to behave smartly and autonomously. This addresses the requirement for intelligent building monitoring. The sensor framework described in this paper is essentially a back-end software service. It provides support for goal based activity rather than simply a data logging facility. The current application is for identifying wasted building operation resources informed by real-time analysis of gathered sensor data. The software agent architecture facilitates easy integration of very flexible new goal seeking entities. Existing hardware provides comprehensive monitoring of the environment but new devices can be added easily at several levels of abstraction, e.g., from low level sensors, moving up to 'feeds' from a building management system, or other types of information such as weather data from 'RSS' (Really Simple Syndication).

The main defining feature of the framework lies on its support of goal directed entities that interact with the resources available in the framework. A further feature is the intelligent management of sensors that allow data delivery using battery powered units that are very easily deployed. Ontology plays a central role here in the management process, and ontology and inference are used to support the agents by continually revaluating the requirements, the current holistic hardware configuration (network) and determining new configuration. Most of the framework has been tested and works stably with the integrated facility management help desk system. The nature of agents allows further functionality to be easily integrated, exploiting other knowledge being built in the system. The modular infrastructure architecture allows easy integration of any sensors, sensor groups, actuators or devices as well.

As for the limitation and future work, more ontology development would be expected to enhance the intelligent behaviour of additional agents, aligned with the goals of those new agents. An area that has so far not yet been modelled relates to how building users interact with the building. It is expected that such knowledge could be readily used by existing agents and significantly benefit agents in making inference to support goal delivery. To improve the ease of software deployment and management the integration of the JADEX platform into a component framework may be desirable. Open source software already exists to integrate JADE with OSGi so while it is regarded as a trivial task, no work in this area as yet has been carried out. The wireless sensor and actuator network is currently functional to a level such that agents configure each node dynamically in the network as well as control and manage connected devices. As described above that is achieved using a standard command set. The framework may benefit from some implementation of custom commands for direct execution by the ZigBee nodes' microcontrollers and further level of detailed modelling in the (sensor) ontology. The benefits would be more compact command sequences and more fine grained control of the node's operation. The agents generate output in text form including the form of OWL statements which is well suited for consumption by other tools.

Regarding the location of sensor host nodes, they currently need to be specified in the (IFC) building model. Triangulation may be possible as signal strength (received signal strength indicator (RSSI)) data can be retrieved from the network controller or routers. Although the effects of building materials would attenuate the signals differently, estimates could be made from calculated paths using the building model. Whether unknowns such as antenna orientation for example would make such an effort impractical remains to be determined. Some ZigBee hosts would still need to be located manually though in order to act as bases for triangulation. So the practical benefit may be negligible. Sensor devices have been adequately modelled as well as some of the aspects of the ETRX357 device (topology of microcontroller and peripherals, etc.) but the ZigBee network has not been extensively modelled. More efficient mesh network configuration (routing) might be possible from inferences about zone topology, building topology and mereology with respect to the locations of sensor nodes. Using an enhanced network model in conjunction with the ZigBee platform model, further improvements in operational efficiency such as reduced wireless traffic and minimum transmit powers could be gained by higher grained configuration adjustment.

Acknowledgment. The authors would like to thank the four anonymous reviewers for their constructive comments that have enhanced the readability and academic standing of this paper; the authors also would like to thank the EPSRC for their Ph.D. sponsorship for this project.

REFERENCES

[1] G. Bertolotti et al., A multi-purpose wireless sensor network based on ZigBee technology, *Sensors and Microsystems*, pp.401-404, 2010.

[2] S. F. Larsen et al., An experience on integrating monitoring and simulation tools in the design of energy-saving buildings, *Energy and Buildings*, vol.40, no.6, pp.987-997, 2008.

[3] M. J. Dibley, H. Li, J. C. Miles and Y. Rezgui, Towards intelligent agent based software for building related decision support, *Advanced Engineering Informatics*, 2010.

[4] M. Compton et al., A survey of the semantic specification of sensors, *Proc. of Semantic Sensor Networks*, p.17, 2009.

[5] J. H. Kim et al., Building a service-oriented ontology for wireless sensor networks, *IEEE*, 2008.

[6] S. Runde, A. Fay and W. O. Wutzke, Knowledge-based requirement-engineering of building automation systems by means of semantic web technologies, *IEEE*, 2009.

[7] K. J. Witt, Enabling sensor webs by utilizing swamo for autonomous operations, *ESTC*, 2008.

[8] A. Rogers, D. D. Corkill and N. R. Jennings, Agent technologies for sensor networks, *IEEE Intelligent Systems*, pp.13-17, 2009.

[9] R. Vaisenberg, S. Mehrotra et al., SEMARTCam scheduler: Semantics driven real-time data collection from indoor camera networks to maximize event detection, *Journal of Real-Time Image Processing*, vol.5, no.4, pp.215-230, 2010.

[10] M. Okaya and T. Takahashi, BDI agent model based evacuation simulation (demonstration), *The 10th International Conference on Autonomous Agents and Multiagent Systems*, vol.3, 2011.

[11] A. Orfila, J. Carbó and A. Ribagorda, Autonomous decision on intrusion detection with trained BDI agents, *Computer Communications*, vol.31, no.9, pp.1803-1813, 2008.

[12] W. Sun et al., Software as a service: An integration perspective, *Service-Oriented Computing – ICSOC 2007*, pp.558-569, 2010.

[13] D. Gupta et al., Difference engine: Harnessing memory redundancy in virtual machines, *Communications of the ACM*, vol.53, no.10, pp.85-93, 2010.

[14] B. Sotomayor, K. Keahey and I. Foster, Combining batch execution and leasing using virtual machines, *ACM*, 2008.

[15] M. J. Dibley, *An Intelligent System for Facility Management*, Ph.D. Thesis, Cardiff University, 2011.

[16] I. Modbus, *Modbus Application Protocol Specification V1. 1a*, North Grafton, Massachusetts (www. modbus. org/specs. php), 2004.

[17] T. J. Park et al., BACnet over ZigBee, A new approach to wireless datalink channel for BACnet, *IEEE*, 2007.

[18] K. Menzel et al., Towards a wireless sensor platform for energy efficient building operation, *Tsinghua Science & Technology*, vol.13, pp.381-386, 2008.

[19] B. O'Flynn et al., A 3D miniaturised programmable transceiver, *Microelectronics International*, vol.22, no.2, pp.8-12, 2005.

[20] X. Cao et al., Building-environment control with wireless sensor and actuator networks: Centralized versus distributed, *IEEE Transactions on Industrial Electronics*, vol.57, no.11, pp.3596-3605, 2010.

[21] D. Russomanno and J. Goodwin, OntoSensor: An ontology for sensor network application development, deployment, and management, *Handbook of Wireless Mesh and Sensor Networking*, McGraw Hill, New York, 2008.

[22] A. Pokahr, L. Braubach and W. Lamersdorf, Jadex: A BDI reasoning engine, *Multi-agent Programming*, pp.149-174, 2005.

[23] E. Sirin et al., Pellet: A practical owl-dl reasoner, *Web Semantics: Science, Services and Agents on the World Wide Web*, vol.5, no.2, pp.51-53, 2007.

[24] D. L. McGuinness and F. Van Harmelen, OWL web ontology language overview, *W3C Recommendation*, vol.10, 2004.

[25] E. Pitt and K. McNiff, *Java.rmi: The Remote Method Invocation Guide*, Addison-Wesley Longman Publishing Co., Inc., 2001.

[26] F. Maulo, NHibernate for .NET, in hibernate, *Red Hat Middleware*, 2006.

[27] J. Ferber, *Multi-agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley, 1999.

[28] M. E. Bratman, D. J. Israel and M. E. Pollack, Plans and resource-bounded practical reasoning, *Computational Intelligence*, vol.4, no.3, pp.349-355, 1988.

[29] M. Musen et al., The Protégé ontology editor and knowledge acquisition system, *Stanford Center for Biomedical Informatics Research*, 2009.