Journal of Ubiquitous Systems & Pervasive Networks
Volume 1, No. 1 (2010) pp. 1-8

Ambient Things on the Web

Sujith Samuel Mathew[a]*, Yacine Atif[b], Quan Z. Sheng[a], Zakaria Maamar[c]

[a]*School of Computer Science, University of Adelaide, Adelaide, Australia*
[b]*Faculty of IT, UAE University, Al Ain, UAE*
[c]*College of IT, Zayed University, Dubai, UAE*

Abstract

Recent advances in Radio Frequency IDentification (RFID), wireless sensors and Web services have led to the proliferation of physical things being triggered from the World Wide Web (WWW). The resulting information overflow creates new challenges and at the same time promises a new generation of ubiquitous applications. In this context, managing the large number of things that could be on the Web is exponentially complex. Web application developers require a new architecture for categorizing things so as to facilitate their deployment and management. Failing to do so may result in redundant efforts to put into integrating things in future ubiquitous applications, because of the absence of a common specification of things. In this paper, we propose a classification of things based on their common characteristics and discuss some related properties. Using this classification, an approach is developed to categorize things so that future Web architects can clearly distinguish between different types of things when building context-aware applications on the Web. To illustrate how this classification helps in creating ubiquitous applications, a prototype of an ambient instructional support in a pervasive learning institution is illustrated.

Keywords: *Ambient Environments, Web of Things, RFID, Ontology.*

1. Introduction

Nearly two decades ago, Mark Weiser envisioned "ubiquitous computing", where computing power becomes invisibly integrated into the world around us and accessed through intelligent interfaces. He observed: "*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it*" [21].

Today, we are one step closer to this vision due to recent advances in identification technologies, wireless networks, Web services, and nanotechnology, which make processing power and communication capabilities available in increasingly smaller packages. Indeed, the Internet is evolving into the so-called "Web of Things" (WoT), an environment where everyday objects such as buildings, sidewalks, traffic lights, and commodities are identifiable, readable, recognizable, addressable, and even controllable via the Internet [19].

While the Internet is becoming the platform of choice for connecting physical things, the obvious choice of a universal platform to build applications that use things, is the Web. Tim Berners-Lee commented on WoT: "*It isn't the documents which are actually interesting; it is the things they are about!*" [5]. The contextual scope of WoT exceeds the boundary of today's Web as it is poised to enable physical things to be accessed via a Web browser. This transformation of the Web will enhance people's personal life and enable enterprises to reach new business opportunities through efficient supply chains and improved environment monitoring. With billions of things finding their way into the Web, people will find themselves in ambient environments (i.e., environments that provide seamless communication between people and things).

1.1. Scenario: Ambient Learning Environment

We present a scenario of an ambient learning environment, which illustrates the ever-increasing use of ubiquitous systems in today's social and instructional setups. Consider an ambient learning environment confined within an immersive-learning room of a building. Equipped with an Internet-enabled touch-screen signage, the workspace includes a sound system, a video-conferencing unit and a digital screen (interactive board) as shown in Figure 1. These physical objects are Internet accessible. There are only two Immersive Learning rooms in the building that offer this cyber-physical facility, allocated on a first come first served basis. Typically, a student group member checks the room's availability via the Web or directly checks the schedule on the signage touch-screen. In either case, the room may be reserved. Once the room is reserved, the members of the group get notified online via calendar update requests to join the scheduled session. As a booking of the room is performed, a notification is also sent to the student's supervisor who may join the group at anytime using a Web based interface for the provided videoconference unit. The student IDs are matched with the seats they occupy in the room. As soon as the students join their seats, the workspace is declared busy and attendance is automatically logged. At the same time a videoconferencing invitation is delivered to the supervisor who may be remotely located. A learning session consists of successive or simultaneous annotations on the interactive board from the tablet-like writing pads provided to each participant. The workspace can trace each member's contribution on the shared interactive board. The supervisor may also project some case studies on the interactive board

* Sujith Samuel Mathew. Tel.: +971-50-7937672
E-mail: sujith@adelaide.edu.au

DOI: 10.5383/JUSPN.01.01.001

through the Web, to throw personal notes or illustrations into the topic of discussion.

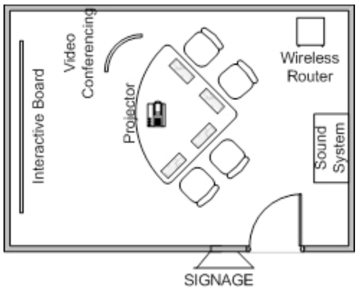


Fig. 1. Immersive Learning Room

1.2. Motivations

With a plethora of things becoming ubiquitous on the Internet, there is a need to model scenarios as depicted in Section 1.1, and plan to handle large number of things in ambient environments. There are quite a few challenges in building a completely integrated system as described in Section 1.1. Accessibility to things via common interfaces currently lacks, which is essential to build applications that exploit their capabilities in a given context. In addition, the heterogeneity of ubiquitous computing systems poses a major problem for system architects with respect to many protocols, component architecture, and data formats. There is no clear understanding of the common characteristics of things or processes for controlling them or querying them for tracking needs. Considering the fact that things can be dynamic in space and time, managing the presence of things is quite an interesting area of study. Also, understanding the relationship between things in an ambient environment presents a significant challenge.

An initial step towards engineering the virtual access and control of large number of physical things on a common platform (i.e., the Web) is to identify the types of things and understand the various properties of these things. Classification harnesses the complexity of modeling context-aware scenarios through a common approach and standardizes the interface for a wide range of physical things. A metaphoric illustration of this approach is the capability of database experts to quickly model a wide variety of data because of standardized data classification and structures.

Classification is described as a set of "clusters" into which things can be mapped, in order to standardize processes for creating related infrastructures [13]. Classification ensures consistency and allows future ubiquitous application to interact with things through implicit common interfaces hiding *behind* the inherent intricacies of things. To overcome the universality of things and their intractable complexity, the classification process must be addressed at a high level of abstraction to ensure simplicity and completeness of the proposed representation of things. Without such a classification, ad-hoc and redundant methods of dealing with things will prevail and interoperability will fail..

We focus on the capabilities of real-world things, which we abstract their interfaces on the Web to drive application functionalities in dealing with those things. These things can then be used for various scientific, environmental, business and social needs via common interfaces. Hence, a classification based on capability to characterize things is deemed necessary as a foundational step before defining the architectural possibilities of dealing with the vast span of things to be deployed on the Web. This enables application designers to quickly model a thing, and interoperate with applications that use these things.

1.3. Contributions

As an initial contribution towards the architecture for managing things on the Web, we first identify the underpinning dimensions pertaining to things classification (*Contribution 1*). Then, we propose standard types in the multidimensional space of things (*Contribution 2*). As things connect to the Web, they exhibit certain properties related to their lifespan and dynamic states as well as other social descriptors, which we model as intrinsic attributes of things (*Contribution 3*). Using the hierarchical classification (in Contribution 2) and the schematic attributes (in Contribution 3), we produce an ontology of things on the Web (*Contribution 4*).We reason over the ontology to infer use cases that subsume complex things (*Contribution 5*). We propose an OWL-based implementation of the ontology and SWRL rule-based system to reason over this ontology (*Contribution 6*). Finally, a case study centered on ambient instructional resources in a pervasive learning environment is discussed in the context of the proposed WoT approach presented in this paper (*Contribution 7*).

The research contributions in this paper help in understanding the contextual behavior of things on the Web to assist architects and designers abstract a thing and quickly model it for integrating it into the Web. They promote standard interfaces and operations, which facilitate interoperability and reuse. They also lead to models of composition of things to meet desired deployment scenarios of ubiquitous applications and facilitate the development of templates with lower interoperability barriers that could arise when Web application interact with heterogeneous Web-enabled things.

1.4. Paper Organization

The remainder of this paper is organized as follows. Section 2 provides an overview of research activities towards ubiquitous computing, by integrating things into the Web, as well as the related research activities in this area. Section 3 describes the classification of things in terms of various dimensions and discusses the requirements for integrating things on the Web. Section 4 discusses some important properties of things. Section 5 defines the ontology of things and related reasoning modules and Section 6 presents the system architecture of the proposed WoT approach. Section 7 illustrates the classification of ambient instructional things in a pervasive learning institution. Finally, Section 8 provides some concluding remarks and suggests some future directions of this research.

2. Background and Related Work

A mandatory requirement for accessing things on the Web is to uniquely identify them within a context. Identification technologies such as Barcode, Radio Frequency IDentification (RFID), and Bluetooth allow everyday things to be uniquely identified. There is also a growing trend for using Internet Protocol (IP) addresses to uniquely identify physical things. A contributing factor to the fast and widespread growth of the Internet is the increasing dependence on the Internet as an economical and efficient means of communication. The increasing availability of Internet access points and enhanced infrastructures of whole cities to support wired and wireless Internet connection are fueling this trend [8]. Bodies such as the IP for Smart Objects alliance (IPSO) [1] and the European Future Internet Initiative (EFII) [2] have also accelerated this

trend to connect a variety of physical things into the Internet, with the intention of propagating and managing the wide use of Internet as the common medium for communication. A thing becomes Internet-enabled if it is associated with networking capability (i.e., has an IP address), which uniquely identifies it on the Internet (Figure 2a). Today, devices such as sensors, electric meters, street lights, and access cards are already networked and accessed on the Internet; even IP-connected pacemakers are used to monitor the health of patients [19].
A thing becomes Web-enabled when it is augmented with a Web server (Figure 2b) so that it can expose its functional and non-functional capabilities on the Web through HTTP. Researchers have already successfully embedded tiny Web servers on resource-constrained things (e.g., sensors, smart cards) [10], making Web-enabled things a reality.
Though arguably, there is scope for WS* and REST in the area of Web services, advances in REST based Web service architectures is propagating the abstraction of physical things as services on the Web [7, 12, 20]. This trend gives rise to the possibilities of wrapping things in the physical world as Web services (Figure 2c). Dominique and Vlad [15] successfully demonstrated Web mashups by exposing real world things as RESTful Web services. Their research compares two ways of interfacing real-world devices into the Web by (1) having Web servers embedded in devices and (2) connecting devices to an external proxy Web server, as a gateway.

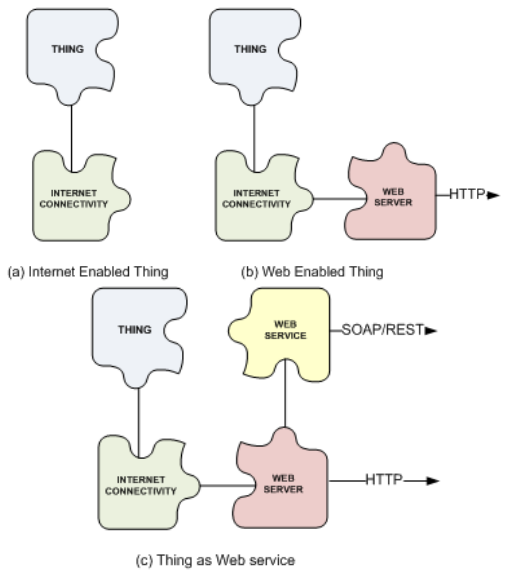


Fig. 2. Connecting things to the Web

At the University of Washington, researchers have created an ecosystem where many things in a building are RFID tagged. These are monitored and accessed through Web-based applications [11]. A study of user experience with the applications in this ecosystem has been conducted. The work shows how the Web is used for accessing real-world things. The Perci Framework [6] enables mobile interaction with real-world objects. The architecture uses Web services for physical mobile interactions (PMI). Tagged physical objects are read by mobile devices in different interactive modes to gather specific information. The framework maps tagged objects onto different service parameters.
There is no significant work done to classify things based on their capabilities or a specification of the various characteristics that would contribute to the architecture of integrating them into the Web. This specification would facilitate the large-scale deployment of things into WoT either as Web resources, providing information or as Web services providing autonomous services. Michael Beigl et al. [4] define smart physical things as things augmented with computing and communication capabilities, which can be accessed by computer applications. Similarly, Friedemann [18] envisions smart things to be able to wirelessly communicate with people and other smart things, with the ability to understand the presence of surrounding objects. Today, these definitions do not formally encompass all things that could be on the Web, for e.g., an RFID tagged chair or a Personal Digital Assistant (PDA), both can be accessed on the Internet. A recent effort in classifying things [17] focuses on application design for industrial hardware. Instruments and tools in industrial scenarios are augmented with sensors, wireless communication capabilities, and display devices, to make them *smart*. These tools are classified as activity, policy, or process-aware objects, based on awareness, representation and interaction capabilities. However, this work is constrained to industrial devices and does not consider the vast majority of objects that could potentially be used to provide useful information. For example, objects that do not have sensing capabilities would not be classified as smart objects. In contrast, we propose a more comprehensive classification model where all objects can be abstracted into the Web.

3. Classification of Things

In this paper, a thing is defined as a tangible physical entity that needs to be controlled or has information to share, on the Web. Considering their scale and variations, things need to be abstracted into standard representations to be integrated into the Web. The abstraction focuses on common characteristics which represent the dimensions of our classification framework. These dimensions provide a high level of abstraction that is generic enough to encompass all things for ubiquitous applications.

3.1. Dimensions for Classifying Things

Things can be classified into four fundamental dimensions that characterize their intrinsic capabilities: *Identity*, *Process*, *Connection,* and *Storage*. We refer to this space of things as the IPCS set.
Identity (I): A thing must be uniquely identifiable with the use of an appropriate identification system. Identification systems like Barcode, RFID, or IP address can be used to locate a thing and access it as a unique resource. A thing could be identified using multiple identification systems (e.g., a thing could have a Bluetooth address and an IP address). Identity is the mandatory and minimal requirement for things to be integrated into the Web.
Processor (P): The processing capability of a thing is a system that has functions which allow a thing to be controlled or managed. This could describe microprocessor at the chip level or an operating system that provides functions to control and manage a device or even a simple interface and that defines functionalities such as start, stop, etc.
Connection (C): The connection interface of a thing is a system enabling interaction with other things. It describes how to read from or write to things. For example, a car stereo with a USB port and Bluetooth connectivity has two connection interfaces. A thing exposed as a Web service provides Web-based Application Program Interfaces (APIs) as connection interfaces for other things to interact with it. Each connection interface is defined for input, output, or both. Properties of

each connection interface (e.g., communication standard, medium and privacy) describe how the connection is to be established.

Storage (S): Storage is a system that describes the type and amount of information that a thing retains. This capability enables the thing to record states and values. A thing could have multiple storage types and corresponding properties. Each storage type has properties such as name, storage type and capacity that describe its use.

The context in which a thing is used may vary based on the application it is a part of. Specifying the characteristics of things makes it easy to abstract things for various application contexts on the Web. For example, in an asset management application, a personal computer (PC) would only need to have a unique identity (e.g. RFID tag) to indicate its presence. In a network management application, other characteristics such as connection interface (network ports, IP address) and storage (RAM, HDD) need also to be considered.

3.2. Types of Things

In the IPCS space of things, classes of things could subsume some commonalities to encapsulate things under certain categories for developing ubiquitous applications. This is also essential to preserve interoperability among the various applications on the WoT. A thing is categorized based on its projection on the IPCS dimensions with the Identity (I) being necessarily set to a *non-null* value. Depending on the instances of the other dimensions, a thing can be classified as core, primitive, complex, or smart.

A *Core* thing has the bare capability of being uniquely identified within a given context. Examples of such things would be pallets, medicine bottles, shoes, which can be identified uniquely on the Web using an identification system like RFID or Barcode.

A *Primitive* thing adds to its unique identity value, an additional value representing an instance of anyone of the other three dimensions described in Section 3.1.A primitive thing can further be categorized along the following three subclasses (Table 1):

- *Fuzzy:* A fuzzy thing is uniquely identified and process information. Fuzzy things have pre-defined operations but do not have means for other things to connect to it or store information (e.g., washing machine, microwave oven).
- *Plug:* A plug is uniquely identified and has a connection interface. It connects to other things but do not have processing or storage capabilities (e.g., speakers, headphones).
- *Fat:* A fat thing is uniquely identified and has storage capability but does not have processing capability or connection interface, e.g., CD, DVD, etc.

Table 1. Classification of Primitive Things

Name	I	P	C	S
Fuzzy	X	X		
Plug	X		X	
Fat	X			X

A *Complex* thing has a unique identity and combines values of any two of the other three dimensions described in Section 3.1. The combination is shown in Table 2. Two types are mentioned here.

- *Social*: A social thing is uniquely identified, has processing capability and connection interface, but does not have storage capabilities (e.g., remote control, landline phones).
- *Sticky*: A sticky thing is uniquely identified, has a connection interface and storage capability, but no processing capability (e.g., USB Stick, RFID Tag).

Table 2. Classification of Complex Things

Name	I	P	C	S
Social	X	X	X	
Sticky	X		X	X

A *Smart* thing, combines values of the three dimensions described in Section 3.1. It is uniquely identified, has processing capability, a connection interface and storage capability (e.g., PDA, Personal Computer).

3.3. Things on the Web

A thing is Web-enabled when it is connected to a Web server. To connect a Web-enabled thing on the Internet, it must be IP-enabled. The thing must have an operating system to provide management functions (network, I/O, storage) and storage space is required to store Web resources such as HTML or XML.

These basic requirements clearly indicate that for a thing to be Web-enabled it must satisfy all four of the IPCS characteristics (i.e., only smart things can be Web-enabled). If any of the four capabilities (IPCS) is missing, a thing must be augmented to make it smart and hence accessible on the Web. Composing different things creates the possibility for mashing up a smart thing.

Is it possible to Web-enable a primitive thing or a complex thing that is not smart? A primitive thing can be accessed on the Web if it has a "Web-enabled smart friend" i.e., a thing connected to a peripheral system (friendly thing, Section 4.2) with a Web server which acts as a proxy [15]. For example, in our scenario (Section 1.1), equipments in the immersive room are RFID tagged. A Web-enabled RFID reader (smart thing) in the room can track the presence of these equipments as well as participants (using their RFID tagged IDs) and project the information to a website (for attendance tracking for example). This method allows the information of a primitive thing or complex thing that is not smart to be accessed on the Web.

4. Properties of Things on the Web

For successfully modeling things on the Web, it is also important to understand their properties. Considering the dimensions and the types of things described in this paper, some of the generic properties that need to be considered are discussed in this section.

4.1. Lifespan

The lifespan of things indicates the state of a thing and its transitions during its existence (Figure 3). A thing takes on "Active" state after it is created or when it is enabled from a "Renewed" state. It can expire and those expired things must be handled to ensure a manageable and scalable environment (i.e., expired things can be removed in time and new things can be easily added in the environment).

Determining the lifespan of things helps recycle or dispose those things that have expired. For example, an RFID tagged student ID card (primitive thing) could be recycled or destroyed after a student graduates, following our previous motivational scenario. This approach applies as well to other contexts, such as a train ticket or an airline boarding pass. The lifespan of the student ID starts with the "journey" a student commences as she/he joins the institution and it moves into an "Expired" state when the "journey" finishes. Based on policy, it can either move into a "Dead" state or be formatted and move into a "Renewed" state ready to be used again.

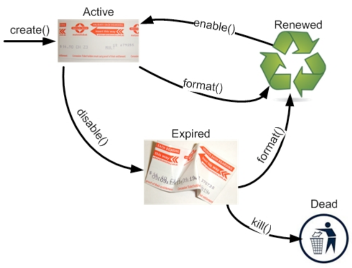


Fig. 3. Lifespan of things

4.2. Friends

Foreseeing a world where everything is connected on the Internet, a thing is bound to interact with other things. In order to make the WoT an environment where things interact with each other, it is necessary to maintain history of interactions. Hence, a thing has friends and the relationship is defined by properties of earlier interactions. The stored details of interactions with friends would ease decisions when interacting in the future (e.g., to choose a friend for a particular service). This raises the requirement for creating a trusted circle of friends and the need to maintain relevant information of these relationships.

4.3. Ownership

Ownership will define the possession rights of a thing. This property dictates the rights to make changes and provide or revoke different types of access to proprietary things on the Web.

4.4. Shareability

This property allows the owner of a thing to provide exclusive access to others for its use. As suggested by Guinard et al. [14], exposing the services using the existing social networks allows the users to be in their comfort zone and there would be no need to build a new database of trusted users. This raises questions about privacy, damage, and misuse when dealing with real world things. Even if things are shared to known members of a community, misuse or damage could occur because access to physical objects is given through a virtual environment (the Web). To mitigate any risk, it is important to have a protocol in place where things accessed by a party (person or system) must agree to legal terms for its use.

4.5. Searchability

Things available on the Web are accessed as URLs. Billions of things on the Web require the use of an easy and quick way of finding them. The chances of seeing Google adding "Things" as one of their search functionality, like Scholar or Images, is not far-off. As things become Web resources, Web crawlers will have more to do in filtering and normalizing URLs.

4.6. Accessibility

Access to a thing is either made via public, private, or protected modes. A thing is public when it is available for open access across the Web (e.g., sensors in parking slots). Private things have access restricted to predefined users restricting the access of things within a community of trusted users. For example, in our scenario, adjusting the camera of the video conferencing equipment is restricted to occupants of the immersive room and their supervisor (who may join the room via the Web). Protected things have private access but with access filtered or limited within a trusted group. For instance, turning on/off the video conferencing equipment is restricted to the supervisor while student in the room can only adjust the camera position.

5. Ontology for Things on the Web

In order to share the proposed model of things, an *ontology scheme* for the WoT is proposed in this section. This standard framework facilitates the development, interaction, and integration of future Web-based ubiquitous applications among software architects using a common specification. It also enables reuse and extends the possibilities of standardizing the architectural framework for developing applications for the WoT.

The ontology provides a formal representation of the physical domain knowledge to propel further the development of intelligent ubiquitous applications via a common knowledge representation of things that architects can refer to. The range of physical devices that could be made accessible is increasingly heterogeneous and ubiquitous. The ontology specification aims at subsuming this diversity to hide inherent disparities into the surroundings, leaving only ontology-specified interfaces as perceivable access points to services and content of things without restrictions in time or location.

There is a need to link the discovery and description of ambient things with domain knowledge representations in order to facilitate a ubiquitous experience. Although this domain may be further specified to some application contexts such as learning technology, multimedia, health environments, we provide a top-level specification of such ontology to map things' descriptions. This approach maps things onto OWL instances via UPnP-like resource discovery protocol. The discovery process subsequently leads to description and capability representations of discovered things: a user or agent could thus autonomously receive content or operate on things that are in reach. Although descriptions of things may exist and may even be in use via the Web, the actual formats of such descriptions may be ad-hoc and application-dependent. The ontology provides a common description format that can be interpreted by Web services that may be typically involved in a delivery chain of ubiquitous services. A further specialization of the ontology facilitates adaptation to particular contexts that may be more relevant to the context at hand. However, the top-level specification addressed in this section enables a general common semantic for future Web applications to develop ubiquitous experiences that require minimum user intervention.

5.1. Things Description

Things are described through a platform-neutral XML template to advertise its services. This template offers a clear separation between the properties and its services. The XML-based description template provides rudimentary information of a thing such as name, type, ID, service, and modality which can be extended.

We use an ontology model for formalizing things on the Web to allow automated programs (e.g., software agents) understand things' modalities and services. Rule-based reasoning methods are then employed to adapt things' class to a thing instance depending on the context (such as user's preferred approach to interact with a given thing) and the application at hand. A snapshot of the ontology is shown below, in view of the previous description of things on the Web. It features the `<type>` tag which instances the type of things as described in Section 3.2, followed by the IPCS categorization of a thing. The details of the identification system are described within the `<ID>` tag. The `<Process>` describes a list of modalities that defines the use of the thing. The `<Connection>` describes a list of connections that enable the interaction with the thing and finally the `<Storage>` describes the different storage options.

```
<thing>
<type>  … </type>
<name> … </name>
<ID> … </ID>
<InterfaceList>
        <Process>
                <operationList>
                <operation> … </operation>
                <operation> … </operation>
                </operationList>
        </Process>
        <Connection>
                <connectionList>
                <connection> … </connection>
                <connection> … </connection>
                </connectionList>
        </Connection>
        <Storage>
                <storageList>
                        <storage> … </storage>
                        <storage> … </storage>
                </storageList>
        </Storage>
<InterfaceList>
<thing>
```

An example of the above XML template used to describe a Projector as a thing to be accessed on the Web, as follows:

```
<thing>
<type> Complex </type>
<name> Projector A123 </name>
<ID> 192.20.242.7 </ID>
<InterfaceList>
<Process>
<operationList>
        <operation> PowerOn </operation>
        <operation> PowerOff </operation>
</operationList>
</Process>
<Connection>
<connectionList>
        <connection>Bluetooth</connection>
        <connection>WLAN</connection>
</connectionList>
</Connection>
<Storage>
<storageList>
        <storage> NULL </storage>
</storageList>
</Storage>
</InterfaceList>
</thing>
```

5.2. Hierarchy of Things

The role of the proposed ontology in this paper is to provide a unique vocabulary and description logics based on modeling things for rudimentary reasoning. The ontology consists of several modules, which are accessed as separate Web resources with specific URIs. These modules cover for example the shared architectural knowledge layers, services that can operate on things, etc. In this paper, we focus on architectural specification which relates to basic properties of things. Industry-standard discovery mechanisms and capabilities descriptions respectively, are adopted in this paper. Based on OWL, the proposed ontological specification shown in Figures 4 - 6, exhibit the categorization of things.

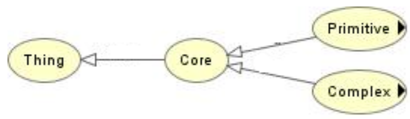


Figure 4: Concept of Core thing

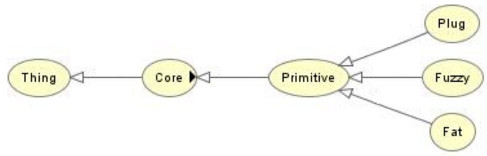


Figure 5: Concept of Primitive thing

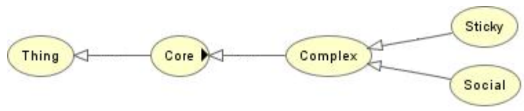


Figure 6: Concept of Complex thing

5.3. Composite things

We use SWRL [22] to reason over the proposed ontology to infer Composite things. W3C suggests the use of SWRL as an extension of OWL and RuleML, which can be expressed in OWL rules. In doing so, new types of things could extend the existing ones to best map satisfy specific application requirements. For example, Composite type of things is identified using the following rules:

isPartOf (?y, ?x1) AND isPartOf (?y, ?x2)
isComplex (?x1) AND isComplex(?x2) → isComposite (?y)

A Composite thing inferred by the above rules, is the aggregation of different things (primitive or complex). For example, in our scenario (Section 1.1) each of the two immersive rooms in the building is a composite thing made up of different individual things like seats, interactive board and overhead speaker. If the individual things that form the composition cohesively have all four characteristics (IPCS), then the composite thing is a smart thing. We illustrate this further in the presented prototype (Section 7). Mashups of physical things can be abstracted on the Web, dynamically composing and assembling things for a particular application, where the capabilities of the participating things are utilized to creating a synaptic Web device.

6. System Architecture

The combination of the ontological and the rule based framework discussed in the previous section of this paper leads to a knowledge based structure of things on the Web. The main benefit of this structure is its semantic power in conceptualizing knowledge about things on the Web. The ontology of the related knowledge base acts also as a directory services for ubiquitous context-aware applications. These applications acquire and process information about surrounding environments based on implicitly derived information about

ambient things. According to information fusion approaches, which may combine data sources from multiple things, more complex contextual states can be derived as a basis for triggering or offering new services. Moreover, ubiquitous environments can be very dynamic where things (and thus related services) are likely to dynamically join or leave the environment.
As shown in Figure 7, the knowledge base server is the main element of the system architecture, which mandates that things on the Web are all registered with the knowledge base server. Higher-level situation models components may also be registered with the knowledge base server as well. Following their registration, the knowledge base server acts as a directory of services for applications that need to interact with things on the Web. Or simply, it provides information about a things' attributes such as identification and operational status. The knowledge base system architecture has three main components, the Smart Space Manager that acts on or probes ambient physical things and the Knowledge Base Agent that discovers services and maintains their profiles. These services can reside in different locations and can be offered by alternative service providers based on a service level agreement. The Knowledge Base Server also acts as a directory service as mentioned earlier. The Smart Space Manager is a middleware which provides a gateway to things on the Web for building ubiquitous applications.

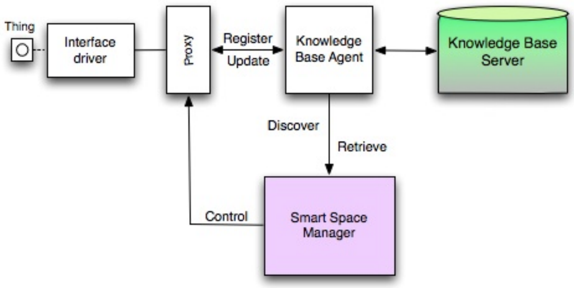


Figure 7: System Architecture

7. Prototype

We illustrate here the classification of things for the design of Web based applications in an ambient educational institute. Various applications for network management, asset management, and ambient learning are planned for this building which is monitored by a Smart Space Manager similar to the one discussed in Section 6. The classification provides different levels of abstraction to model and reason about things in this environment.
Ambient learning in an educational institute aims at matching the profiles of learners with instructional things, to reach a personalized learning experience [3]. Learners in the building are guided to various resources to complete individual learning goals in a closed loop process shown in Figure 8. The Presence module declares intention and location attributes of the learners. These attributes directs learners to learning objects (LOM [1]) which are instructional things in the building. Individual learner attributes such as personal goals, interests and progress are structured using standard profiling approaches (LIP[2]) and learning plans to ensure wider interoperability. This prototype uses the proposed Web based categorization of things to match appropriate learning services within the learner's context captured by the Presence module.

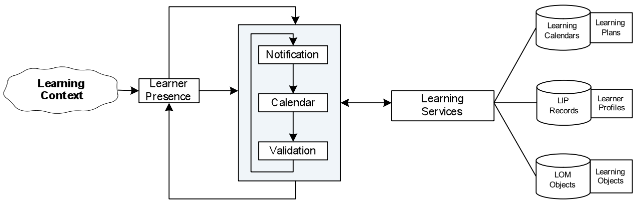
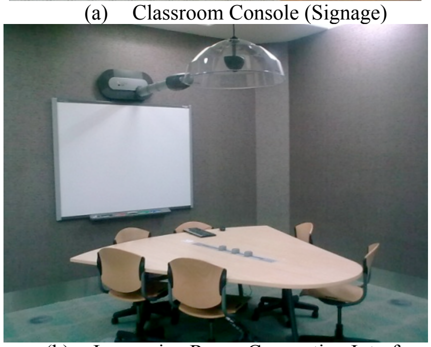


Figure 8: Ambient Learning Process

The applications use various things in the building (e.g., rooms, consoles, A/V equipments) which are accessed via the Smart Space Manager module of our architecture. Next, we illustrate how we abstract classrooms in the building and abstract the things in a classroom.
The building has many rooms of different natures and some are abstracted as learning objects from the application's perspective. We abstract rooms in the building as learning objects. These rooms are potential resources that move a learner forward in her/his learning plan. In this illustration, classrooms are classified as Smart or Social (Section 3.2). Smart classrooms satisfy all IPCS categorization while social classrooms are limited to IPC categories.

(a) Classroom Console (Signage)



(b) Immersive Room Connection Interface

Figure 9: Immersive Room in the Ambient Educational Institute

The Immersive room (Smart thing) has the following features.

- Identity: It is uniquely identified with a room number and it is displayed on the classroom console (touch-screen signage) as shown in Figure 9a.
- Process: The classroom console has a computing system with touch screen interface. It provides operations for assigning faculty to rooms, scheduling courses in rooms, special message display, etc.
- Connection: Classrooms provide interaction via, video conferencing, IP phones and audio equipments as shown in Figure 9b. This enables remote participation in classroom activities.
- Storage: Session details are stored via, video recording capability, which is used to record the sessions that are

[1] http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html
[2] http://www.imsglobal.org/profiles/lipinfo01.html

scheduled in these rooms. The history of captured video recordings is made available to learners.

Classrooms without storage capability are classified as social venues since they satisfy only three (IPC) of the four characteristics. The provision of this classification enables appropriate selection of learning objects (classrooms) for the application architecture illustrated in Figure 8. For example, when learners have to choose to attend parallel learning sessions happening in smart and social venues, the application would give priority to social venues since the smart room session recording can be delivered in a conflict-free time slot.

While designing rooms, we classify various things in the room to use as learning objects. Interactive boards are smart things, chairs are primitive things and projectors are social or smart things. This enables us to have a common classification while defining various things in the building which allows us to abstract and adapt new things for applications in the building.

8. Conclusion

In this paper, we have proposed a generic approach in classifying things for the future Web where everyday objects are readable, recognizable, addressable, and controllable via the Web. The capability based classification of things lays a foundation to integrate different types of things into the WoT. The deployment of Web-enabled things would allow their access, control and remote management on a common, widely accepted and existing platform of the Web, across the boundaries of vendors and manufacturers. Exposing things as Web services would further enable ambient environments where things can be autonomously composed to provide new services.

Our work presented here is a continuing work towards the development of applications for WoT. Future research directions include ontology for providing a formal structure for our classification and creating an architecture for things on the Web. Another important area of research is to understand how to manage large number of things on the Web and resolution of conflict between things. We will also continue to build more applications for the educational institute to further study the validity of our approach.

References

[1] Promoting the use of IP in Networks of Smart Objects. *IPSO Alliance*, http://www.ipso-alliance.org/Documents/IPSO_Briefing.pdf, visited on 2010.07.13.

[2] White Paper on the Future Internet PPPDefinition. *The European Future Internet Initiative*, http://www.future-internet.eu/fileadmin/documents/reports/Future_Internet_2020-Visionary_Panel.pdf, visited on 2010.07.12.

[3] Y. Atif and S. S. Mathew. Ambient Learning Companion. In *Proc. of the International Conference on Education and New Learning Technologies (EduLearn10)*, Barcelona, Spain,2010.

[4] M. Beigl, H.-W. Gellersen, and A. Schmidt. Mediacups: Experience with Design and Use of Computer-Augmented Everyday Artifacts. *Computer Networks*, 35(4):401-409, 2001. http.dx.doi.org/10.1016/S1389-1286(00)00180-8

[5] T. Berners-Lee. The Web of Things. *Special Theme on The Future Web, ERCIM News - the European Research Consortium for Informatics and Mathematics*. Available at http://ercim-news.ercim.eu/en72/keynote, 2008.

[6] G. Broll, E. Rukzio, M. Paolucci, M. Wagner, A. Schmidt, and H. Hussmann. Perci: Pervasive Service Interaction with the Internet of Things. *IEEE Internet Computing*, 13(6):74–81, 2009. http.dx.doi.org/10.1109/MIC.2009.120

[7] F. Leymann, C. Pautasso, and O. Zimmermann. Restful Web Services vs. "Big" Web Services: Making the Right Architectural Decision. In *Proc. of the 17thIntl. World Wide Web Conf. (WWW'08),* Beijing, China,2008.

[8] A. Danigelis. 10 Cities with Widespread Wireless Internet.http://dsc.discovery.com/technology/tech-10/wireless-cities-top.html, visited on 2010.07.12.

[9] A. Dunkels. uIP - TCP/IPConnectivity for Embedded 8-bit Microcontrollers. Visited on 2010.04.27.

[10] S. Duquennoy, G. Grimaud, and J.-J. Vandewalle. The Web of Things: Interconnecting Devices with High Usability and Performance. In *Proc. of the Intl. Conf. on Embedded Software and Systems (ICESS'09)*,HangZhou, Zhejiang, China,2009.

[11] E.Welbourne, L. Battle, G. Cole, K. Gould, K. Rector, S.Raymer, et al. Building the Internet of ThingsUsing RFID: The RFID Ecosystem Experience. *IEEE Internet Computing*, 13(3), 48–55, 2009. http.dx.doi.org/10.1109/MIC.2009.52

[12] R. T Fielding. Architectural Styles and the Design of Network-Based Software Architectures. PhD thesis, University of California, Irvine, 2000.

[13] S. Leigh, S. Geoffery, and C. Bowker. Sorting Things Out: Classification and its Consequences. *MIT Press*, Cambridge, MA, 2000.

[14] D. Guinard, M. Fischer, and V. Trifa. Sharing Using Social Networks in a Composable Web of Things. In *Proc. of the First IEEE Intl. Workshop on the Web of Things (WOT2010)*, Mannheim, Germany,2010.

[15] D. Guinard and V. Trifa. Towards the Web of Things: Web Mashups for Embedded Devices. *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), held with the 18th Intl. World Wide Web Conf. (WWW'09)*, Madrid, Spain, 2009.

[16] G. Han, J. Ni, and D. Choi. A Configurable Tiny TCP/IP Protocol Stack Supporting Mobile Node for Sensor Networks. In *Proc. of the 4th Intl. Conf. on Wireless Communications, Networking and Mobile Computing(WiCOM'08)*,Dalian, China,2008.

[17] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton. Smart Objects as Building Blocks for the Internet of Things. *IEEE Internet Computing*, 14, 44–51, 2009. http.dx.doi.org/10.1109/MIC.2009.143

[18] F. Mattern. From Smart Devices to Smart Everyday Objects. In *Proc. of Smart Objects Conference (SOC'03),* Grenoble, France, 2003.

[19] G. Mulligan. The Internet of Things: Here Now and Coming Soon. *IEEE Internet Computing*, 14(1):35-36, 2010.

[20] V. Stirbu. Towards a RESTful Plug and Play Experience in the Web of Things. In *Proc. of the 2nd Annual IEEE Intl. Conf. on Semantic Computing(ICSC'08)*,CA, USA2008.

[21] M. Weiser. The Computer for the 21st Century. *ScientificAmerican*, 265(3):94–104, 1991. http.dx.doi.org/10.1038/scientificamerican0991-94

[22] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, http://www.w3.org/Submission/SWRL/, visited on 2010.07.12.