

Classifying Smart Objects using Capabilities

Marco E. Pérez Hernández, Stephan Reiff-Marganiec

Department of Computer Science

University of Leicester, UK

Email: {meph1,srm13}@le.ac.uk

Abstract—The Internet Of Things has emerged, providing an umbrella for the increasing number of heterogeneous Smart Objects that are becoming part of our daily activities. In this scenario, classification approaches are useful to understand differences and identify opportunities of generalization and common solutions, especially when different disciplines are coming together and bringing their individual terms and concepts. We propose a novel model for classifying Smart Objects using capabilities. This five-level model, inspired in the Capability Maturity Models, aims to be simple and inclusive, separating objects with basic capabilities from those with complex ones. In addition, examples of objects for each level are provided as validation of the proposal. The model is useful to identify requirements that Smart Objects have to cover externally as they cannot themselves support them and thus it allows for clear understanding of the external support system (or Middleware) into which the smart object is embedded.

I. INTRODUCTION

The evident diversity of technologies, concepts and models forming the Smart Computing area, brings the challenge of bridging and fitting the achievements in multiple disciplines into a shared vision. Part of this vision includes the existence of a pervasive system where people, smart objects (SOs) and the environment interact dynamically and naturally through multiple global networks: The Internet of Things (IoT). Undoubtedly it has already started, but still handling the heterogeneity of these SOs is one of the big open challenges. Multiple efforts are being focused on allowing these objects to be able to work together regardless of their evident differences, with the research community working actively on interoperability at different levels, such as hardware, software, or protocols.

However, heterogeneity also comes from the origin of the IoT paradigm: it is a conjunction of multiple fields such as ubiquitous, mobile and social computing as well as networking and many others. Each of these fields brings its own existing terminology, concepts and assumptions about the smart objects to the table. As a consequence research work in the IoT field is being somewhat segmented by these previous disciplines, complicating the efforts towards a common understanding as various results and insights have to be "translated" to relevant terminology and concepts. It becomes critical to ensure common understanding of the terms as well as awareness of differences and similarities of different kinds of objects. This would help to identify common scenarios and solutions that can give a basis for further generalizations or customization when required. Classification approaches are crucial to identify these common characteristics and give a sense of order and homogeneity within existing diversity. Note that we will consider the notion of 'smart object' (SO) in more detail later.

Several classification frameworks for SOs have been proposed recently. These efforts have been a useful start point giving some criteria to distinguish some objects from others. However, these approaches are conceived from a single perspective and often make it difficult to apply the classifications in other scenarios than the ones proposed by the respective authors. Uncommon terminology, diverse understanding of the SO, the scope in which it is applied and the complexity to place objects in categories are the main causes.

We propose a classification model based on the capabilities of smart objects. It aims to offer a simple and inclusive way of classifying SOs. We use the concept of SO as basis, but will clarify similarities and differences with other related concepts. In addition, we distinguish between core and optional capabilities, with the latter ones organized in four dimensions according to what they address: internal factors, environment factors, human factors, and engineering factors. Although, most of the capabilities have been largely identified by several authors previously, we add to the state of the art by arranging them together in order to define a model that enables a clear classification following an evolutionary path.

The main contributions of our work are: (1) classification of SOs based on an evolutionary path and including capabilities discussed in the literature so far, (2) clarification of the relations and differences of main terms referred to SO and (3) a review of existing work addressing classification of SOs.

This paper is organized as follows: First, the main works addressing definition and classification of SOs are reviewed. Second, findings towards a common terminology and our definition for SO are presented. Next, the identified SO capabilities are organized and clearly defined. Then, our model and the levels of the evolutionary path are explained and exemplified. Finally, a discussion and conclusions are presented.

II. SURVEY OF EXISTING WORK

Work has been published on concepts, platforms, Middleware and applications of SO. Although the basics lie in Mark Weiser's ideas from 1988 [1], the recent emergence of the IoT paradigm [2] has created a current demand for such work. For this paper, work providing insights on classification and definition of SO are of most interest. We will first explain our methodology and then turn our attention to the relevant literature considering terms and existing classification models.

A. Methodology

1) *Scope Definition*: The literature review was to identify existing classification approaches as well as the definition and characteristics of SOs. As we identified a wide variety of terms

Scoping Questions	What classification models exist for smart objects? What definitions exist for smart object? What are the existing terms to refer to smart objects? What are the differences between related terms? What are the features of a smart object? What are the capabilities of a smart object?
Terms	smart OR Intelligent OR intelligent OR ubiquitous OR pervasive OR digital AND object OR device OR product OR thing OR spime OR blogject)) AND (classification OR categorization OR typology OR taxonomy OR hierarchy OR definition OR term OR notion OR conceptualization OR features OR characteristics OR attributes OR capabilities) AND internet of things OR iot OR web of things OR ubiquitous OR pervasive OR future internet
Sources	IEEE Xplore - ACM Digital Library - ISI Web of Science - Science Direct - Springer Link - Scopus
Quality Assessment	A) Is the work part of the Internet of Things or related fields? Y/N* B) Does the work directly address the problem of smart objects classification? Y/P/N* C) Does the work directly address the problem of smart objects definition? Y/P/N* D) Is a classification model for smart objects proposed? Y/P/N* E) Is a definition for the term "smart object" proposed? Y/P/N* F) Is the work related to Internet of Things or derived fields? Y/P/N* G) Are the characteristics of the smart objects identified? Yes: Y/P/N* *Y:Yes=1,P:Partial=0.5,N:No=0

Fig. 1. Literature Review Protocol

being used for what essentially are 'things' the survey included the terms shown in Fig. 1. The most commonly used terms are discussed in more detail in section II-B. Considering the lack of agreement in definition and characteristics, we adopted 'smart object' as the basic term which we define in section III.

2) *Classification Criteria*: Essentially objects differ from each other by their capabilities. Since a taxonomy that considers multiple features is likely to be useful in several fields [3], the goal was to identify, as complete as possible, a list of SO capabilities. This was initially populated with capabilities identified in previous works discussed in section II. Subsequently, fundamental questions such as What? Where? and How? were applied to the context in order to identify additional capabilities. Capabilities are described in section IV.

3) *Defining Classes*: Our classes form a layered hierarchy inspired in the Capability Maturity Model (CMM[®]) [4]. To test our model we identified concrete examples of objects that can be part of each proposed level. Examples are presented with the description of each level in section V.

B. Terms for Smart Objects in the Literature

Considering that IoT is a conjunction of multiple fields, the wide range of terms used for core concepts is not surprising. In the literature we find indistinct use of "Smart Object", "Smart Thing", "Intelligent Product" and "Ubiquitous objects", among others. Additionally, some authors propose their own original terms that seem to refer to the same or a very similar entity.

A breach in the denomination is evident from the initial proposals. An *intelligent product* was defined, from a manufacturing perspective, as a commercial product with five characteristics: unique identity, communication ability, storage of self-data, a deployed language and decision-making capabilities [5]. Similarly, *smart devices* – including PDAs and mobile phones – were defined as physical objects with computing resources that are able to communicate with each other and with users [6]. Later on, *U-things* are described as physical things with digital abilities, as opposed to merely virtual *e-things*. The U-thing concept is a generalization of *u-objects* -i.e. everyday objects —, u-spaces and u-systems [7].

Other coined terms include *Smart-Its*, *Spime* and *Blogject*. *Smart-Its* is an abbreviation of Smart Artefacts, i.e. objects that maintain their appearance and functionality, but are able to compute in the background [8]. *Spimes* refer to a more abstract entity, namely space-and-time-tracked objects that are dynamic and palpable although part of an intangible system [9]. Emphasizing the role of objects as part of a social web, Bleecker [10] distances from "sci-fi" Sterling's view and proposes *Blogjects* as objects that converse and exhibit location tracking, experience- holding, and the ability to foment action.

Yet, additional interpretations have brought wider definitions. In [11] an augmented entity is the composition of virtual and physical entities. In [12], the concept of *smart product* includes not only physical objects but also software and services. They are placed in a smart environment and gifted with self-organization and interaction abilities that provide simplicity and openness. Recently, Gutierrez et al. [13] attempted to unify this denomination. They proposed a meta model with the *smart thing* as an abstraction of both smart product and intelligent product. However, they did not identify differences between these terms neither relations with other variants such as SO.

Smart object is probably the most common term. It is defined from a technical perspective, as "an item equipped with a sensor or actuator, a tiny microprocessor, communication device and power source" [14]. Similar definitions can be found in [15], [16] and [17], among other works.

C. Classification Models

Based on the existing concepts and definitions, several efforts have been made to develop a classification model. Classifications vary according to the set of attributes considered, nevertheless, two approaches can be distinguished: **Nesting oriented** approaches propose a basic type and additional features lead to more complex classes; while **non-nesting oriented** approaches assign attributes which do not reflect increased capabilities over a basic type.

Nesting oriented approaches

One of the first attempts at classifying SOs was defined by EPC Global[®] [18]. It included a four-level model to classify RFID Tags. Tag functionality is described as responsibilities in each class, e.g. support to additional data beyond EPC code. Other responsibilities are factors such as user memory or networking capabilities. Higher tag classes have more responsibilities than lower ones, however there is no consideration about how to deal with an increase of responsibilities for the features identified. Obvious drawbacks are: (1) focus on RFID tags and we have a much wider variety of SOs available, not only based on these tags; and (2) the proposal lacks uniform and consistent criteria to differentiate one class from another.

The three-dimensional model proposed in [19] enables classification based on level, location and aggregation level of the intelligence. There are three levels of intelligence: information handling, problem notification and decision making. Here, a product in the highest level is the one able to manage its own life. Location can be through a network or at the object. Aggregation levels are proposed to differentiate atomic intelligent products, namely intelligent items, and compounded products (for example, intelligent containers). The main focus

is on internal behaviour of the product while interaction with users and other products are barely considered: users are receptors for information from the object and composition relations are only considered between objects.

Classifying by design principles, three types of objects are identified: activity-aware, policy-aware and process-aware [16]. The latter type, which is the most complex, extends the features defined for the previous type which in turn extends the ones of the simplest type. Objects are classified considering their awareness – mainly related to the context –, how objects are programmed – representation – and the interaction offered – from none to active guidance. This typology is enclosed in an organizational context, where processes and policies are followed by the workers which is not realistic for the IoT. Additionally, RFID tagged objects are not treated as SOs. However for the IoT, we consider that several classes of RFID devices can enhance everyday objects to provide some level of 'smartness', and hence RFID-tagged objects should be considered as part of a SO classification.

Two interesting classifications have been proposed from a social perspective. First, Moawad et al. [20] argue that objects can be classified according to 4 levels of social interaction. Level 0 and 1 objects, just receive or send information respectively, while level 2 can perform both tasks with a specific object and level 3 can do it with any neighbour. More recently, Atzori et al. envision a three-step evolutionary process towards a new type of social objects [21]. Each stage is a category of objects by itself. *Res sapiens*, groups the objects that have been the foundation of the IoT, with certain isolation, but able to interchange information with several systems and existing social networks, depending on human interaction. The next step, *Res agens*, refers to objects that exhibit environment awareness and interaction as well as a pseudo social behaviour with near objects. *Res socialis* is the top of evolution, such objects are able to build and maintain their own social network and use this information to offer services to humans. This model is very interesting and provided inspiration for our work, however as presented by Atzori is mainly focused on the social capabilities of the objects.

Non-nesting oriented approaches

The authors of [22] present a flexible model in which object classes do not inherit features from lower classes. The proposal includes classes formed by objects that combine, at least, any two of five optional characteristics: **(I)**dentify, **(S)**ensing, **(A)**ctuation, **(D)**ecision-making and **(N)**etworking. The disadvantage here is that it requires the identification of exceptions to avoid inconsistencies – e.g. a networked object without an identity –, therefore conducting to the fact that some features are not optional. In addition, they conceive SOs without identity and communication capabilities – e.g. (S)(A)(D) and (S)(A) – in contrast with most of the reviewed literature and what is expected in the IoT vision. However, the characteristics identified in this work are absolutely relevant to classification and we incorporate them in our proposal.

Despite valuable efforts to define a classification model, a consistent, clear, inclusive, and multi-purpose classification, able to provide order in the universe of SOs, is still missing. Such a model should rely on characteristics that allow unambiguous differentiation of objects. In our view, it should build

on SO capabilities, so we will present suitable capabilities. However, we first consider the terms for smart object and define our core concept.

III. COMMON TERMINOLOGY AND OUR DEFINITION OF A SMART OBJECT

Many terms have been proposed in the research arena and we already briefly reviewed these in section II-B. Here we will build on this in an effort to establishing clear differentiation between the terms. The aim is to base our classification model in a scoped term and identify where capabilities can suit and help for categorization purposes. The result is presented in Fig. 2 and the main findings are summarized below:

- Existing entities are enhanced in the IoT domain, by adding digital attributes and essential capabilities. E.g. a smart phone is the enhancement of an already existing device: the phone.
- If smart devices are novelties such as electronic devices, it is a natural improvement process to add more features or capabilities to them. However, adding such capabilities to regular (everyday) objects is what actually constitutes the IoT and implies deeper changes to everyday life. IoT solutions are those which are aimed not only to devices but to regular objects as well.
- When there is no significant or useful differentiation, terms can be used as equivalent. For example, *smart environment* is equivalent to *smart space*.
- The qualifiers *Smart* and *intelligent* are used indistinctly, hence there is no justified differentiation.
- Smart thing includes not only objects but environments containing smart objects.
- Smart products can be seen from multiple views. In a business context, they are specialized SOs with attributes such as price, brand, etc. From an engineering perspective, they are just the result of a manufacturing process and so are equivalent to smart devices, invented with a particular purpose. For simplicity we have adopted the engineering view.

We summarize the main findings in the following definition that we use as basis for the classification model:

Definition 1: A smart object is a physical object with enhanced digital capabilities including, at least, identification, communication, retention and energy-harvesting. Smart objects are derived from non-smart objects and maintain these objects original essence. Smart objects are a type of smart things and include not only devices but regular objects.

IV. SMART OBJECT CAPABILITIES

We propose to establish definition and classification models in the functional nature of the object, i.e. what the objects can actually do, the services they offer. The model is not just based on structural properties but also on exhibited behaviour. This creates a simple separation from traditional objects, which are unable to demonstrate such dynamic nature.

Capabilities are based on the existence of both physical and digital features (i.e. Hardware and Software), but they can be

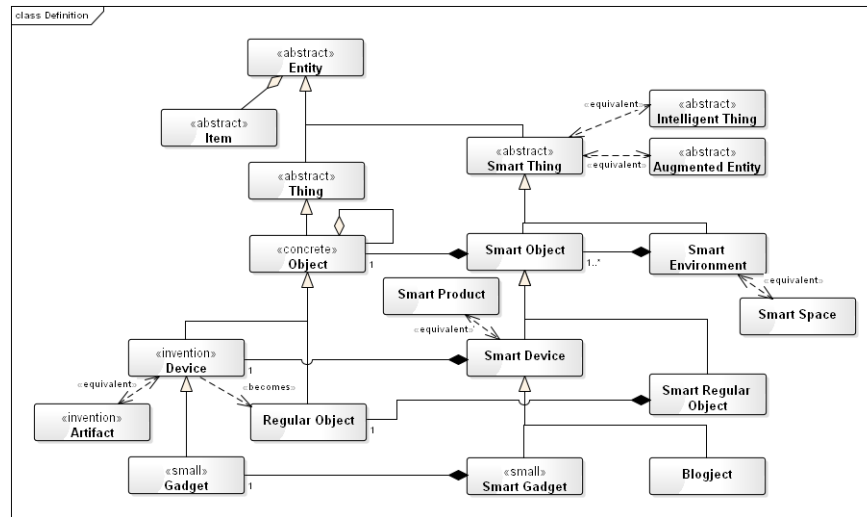


Fig. 2. Ontology of terms related to Smart Object

achieved with different configurations e.g. smart objects can harvest energy from radio waves or from an *ad hoc* battery, similarly networking capabilities can be achieved either with wireless or wired interfaces. Capabilities can be **Hardware-driven**, i.e. these are achieved, mainly, by modifying the hardware structure of the SO, Software changes are insufficient to reach it; or **Software-driven**, i.e. SOs develop these capabilities, mostly, by deploying software routines, models, components or services with minimal (if any) changes in Hardware structure. Capabilities can also be **simple** i.e. atomic that do not require any other capability to exist; or these can be **derived** or complex i.e. these rely on the existence on other atomic or complex capabilities.

We believe capability to be a simple concept, that is not only easy to perceive but also avoids dealing with detailed parameters for each specific property. As a general concept it is intended to allow classification suitable for and from viewpoints of multiple perspectives and domains. The proposed conceptualization of SO based on capabilities is presented in Fig. 3. The nesting oriented model is composed from groups of capabilities that are arranged in a radial setting assembling core and optional capabilities distributed around this core. A fully qualified SO will demonstrate all the capabilities defined.

The capabilities discussed next are based on the extensive literature study conducted for building this model and include features that are suitable for all views of SOs.

A. Core Capabilities

Core capabilities constitute the most fundamental nature of a SO, sine qua non an object cannot be considered smart. These capabilities are atomic, the simplest abilities a SO can have. They are based on three of the characteristics identified in [5] for intelligent products, but we extended each one and included a fourth characteristic:

Digital Identification enables information access and object presence in a digital context. This capability requires the existence of an unique and immutable identity but, beyond, it refers to the ability of the object to identify itself to other objects, systems and even humans.

Retention refers to the ability of an object to store information about itself or the environment, minimally its identity. It relies on the existence of a local or remote memory that in more complex cases can be a large repository.

Communication is essential to interchange information with other objects or users. In the simplest SOs it is a basic point-to-point mechanism with numerous restrictions – e.g. any object with a RFID/NFC (passive) tags. At this basic level of communication, objects usually require others to access Internet.

Energy-harvesting: As dynamic entities, SOs require energy to carry out the processes and tasks they are intended for. This is the the ability to gather the demanded energy either from external sources or by generating it autonomously. Usually, the complexity of the SO tasks is proportional to the energy consumption. Therefore, the more energy the objects can get, the more complex the capabilities they can have.

B. Optional Capabilities

While a complete SO would exhibit all the capabilities described below, currently, existing constraints related to the object's purpose, lifetime, design, or just technical or financial matters might be sufficient reason to develop just some of the optional capabilities. Optional capabilities are derived.

From these optional capabilities, there are two particular **Hardware-Driven** capabilities that are the base for most of the other optional capabilities:

Processing refers to the ability of executing fixed or adjustable instructions and tasks and compute in the background as the object meets its purpose. It relies on the existence of, at least, an attached processing unit such as an embedded controller or system-on-chip (SoC), although some objects can leverage resources by taking advantage of distributed or cloud processing. This capability can broadly vary from one object to other, considering multiple hardware architectures and configurations. Generally, the processing capabilities are required to develop further capabilities. **Dependencies:** Energy-harvesting.

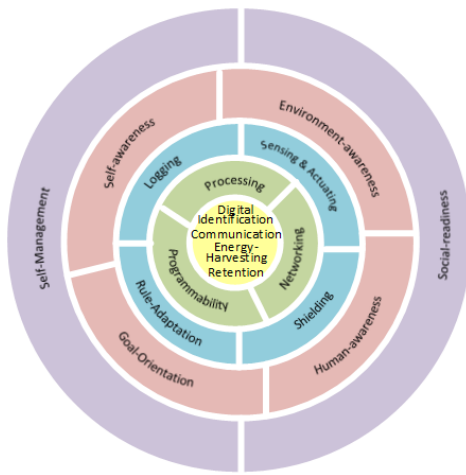


Fig. 3. Smart Objects Capabilities

Networking is an evolution of the communication capabilities with the same central purpose, but involving more complex functions. It implies existence of network adapters, support to a protocol stack – like network layer protocols as in the OSI Model –, the ability to join a variety of system and object networks and support to multiple communication patterns as introduced by [14] (one-to-one, one-to-many and many-to-one). SOs can have intra-networking and/or inter-networking capabilities, for the latter case, it is also required to ensure protection of the object in open environments. **Dependencies:** Processing.

The remaining optional capabilities are organized in four dimensions according to the most related factor they are addressing: (a) Internal Factors, (b) Environment Factors, (c) Human Factors and (d) Engineering Factors. Each of these dimensions can be fulfilled at a fairly basic level or at quite advanced levels (and a spectrum in-between).

a) Internal Factors Capabilities allow objects to discover their own features, status, possible changes and issues. Including also the capability to adapt and use information to manage the object's own life cycle and trigger healing mechanisms.

Logging refers to an object's ability of registering events about itself or the environment. In order to log these events, the SO has attached storage or connects to remote repositories through its networking capability. **Dependencies:** Processing

Self-Awareness refers to the capability of an object to know its own status and structure as well as any change on it and its history. **Dependencies:** Logging, Sensing & Actuating.

Self-management goes beyond self-awareness and includes the development of abilities to use the information gathered in order to manage the object's own life cycle including behaviour, resources, services, response to incidents, problems, maintenance and self-repair. It also includes the ability to learn from experience in order to improve operation and use of resources. **Dependencies:** Self-Awareness, Goal-Oriented.

b) Environment Factors Capabilities are focused in objects to obtain and improve knowledge as well as discover and manage the environment – both physical and digital – in which it is

placed. This includes awareness of nearby things, establishing different relations and inducing desired behaviours.

Sensing & Actuating. According to the purpose, it is common that SOs have one or multiple sensors gathering live information from the environment (e.g. home, human body, etc.) or the objects own structure. Although sometimes it is seen as a fundamental characteristic, with the existence on multiple on-line data repositories being permanently updated, SOs can still be smart without actually sensing. Actuating refers to the ability of provoking a change either on the environment or on other objects. Usually actuating and sensing are linked, although many objects can have either one or the other. **Dependencies:** Processing, Programmability.

Environment-Awareness is part of the context-awareness. It refers in particular to the ability of gathering information from the environment and the surrounding objects in order to improve the user-experience for example by adjusting the object's behaviour. This ability goes beyond having a few sensors; it involves knowledge of environment conditions (e.g. temperature, noise, etc.), locations [23] (e.g. relative and absolute), the present infrastructure and platform, and available services and objects, among others. **Dependencies:** Sensing & Actuating.

Social-Readiness is related to services that enable the object to exhibit social behaviour, joining object social networks, establish several kinds of relations with others, generating and interchanging information in order to meet its purpose and improving services and functionalities offered to the user. Objects are able to relay on and generate social interactions with each other. **Dependencies:** Environment-Awareness, Goal-Oriented, Self-awarenesses.

c) Human Factors Capabilities Although fulfilling the user expectations is the purpose for what the object was built and hence is a driver in any group of capabilities, this dimension focusses on features and services that improve the interaction with the human end-user, for example through offering added-value services. In addition, this dimension considers one of the main concerns of people using SOs: trust.

Shielding comprises the services an object offers to preserve the critical characteristics of the information it deals with. These characteristics include availability, accuracy, authenticity, confidentiality – and privacy –, integrity, utility and possession as described in [24]. Regardless of the security mechanisms in networks and platforms with which the object interacts, the object itself is able to provide protection against any threat and thus enhancing trust in users. **Dependencies:** Processing, Programmability.

Human-awareness is also part of the general context awareness, but it is focused on the services related to gather information from the humans that interact with the object. It includes habits, emotional state, social interaction, spontaneous activity, among others [23]. Since this information is sensitive, shielding becomes a pre-requisite to this capability. It also includes services to improve interaction with human users such as friendly and customized user interfaces. SOs offering direct interfaces to human users without requiring others, are more complex than those which do not. **Dependencies:** Shielding, Processing, Programmability.

d) Engineering Factors Capabilities are related to how the object behaviour is obtained, i.e. how engineers interact with the object in order to induce the desired functionality. In some cases operations are based in detailed pre-programmed instructions, in others, objects have partial control of some activities and in others, advanced objects control every aspect of operation simplifying SO creation and management.

Programmability refers to the ability of objects to be programmed. Programming can be fixed (single-time) or dynamic (many-times, upgradable) and can use one or several models (structured, object-oriented, aspect-oriented, etc.). It is highly related to the representation design dimension as proposed in [16]. Objects with this capability require detailed instructions from a programmer to be able to accomplish their tasks. *Dependencies:* Processing.

Rule-adaptation is the ability of the object to modify its operation based on a predefined set of rules in reaction to data sensed from the environment. Engineers can define degrees of freedom for the object and when the conditions are met, the object launches a set of pre-programmed tasks. *Dependencies:* Programmability.

Goal-Oriented is an object's ability to act based on defined high-level objectives. Engineers set the objectives and the object is then able to reason and generate the best plan to achieve the objectives. These plans can be dynamically generated based on atomic tasks and considering design restrictions and policies. This behaviour gives more autonomy to the object and enable the existence of other derived capabilities. *Dependencies:* Programmability.

V. CLASSIFICATION MODEL

We will now present our classification model. It is inspired by the CMM[®] models [4] and based on the capabilities defined in section IV. As discussed earlier, we see that SOs are evolutions from everyday objects or enhancements ('new versions') of man made objects. In the light of this, we perceive the fundamental idea of CMM[®] as relevant, as objects can be evolved and matured by adding new capabilities. The ultimate 'destination' of a SO along the evolutionary path is a an entity that exhibits all capabilities. Our approach is inspired by the three-step evolutionary scheme proposed by Atzori in [21], although we extend beyond the social capabilities. The identified levels are summarized in Fig. 4 and explained now.

A. Capability Levels

The classification of SOs is based on five levels. Each level requires a number of capabilities as presented in Fig. 3. Capabilities of one layer rely on others from the same or the immediate inner layer. Dependencies are indicated in section IV-B for each derived capability – e.g. environment-awareness in Layer 4, requires sensing and actuating in Layer 3 –. SOs can potentially evolve to any higher level, although some capabilities are Hardware-driven implying a considerable modification of the SO. Objects lacking capabilities of Level 1 are not SOs. Also, an object at a specific level might have some features of a higher level, but these could be considered insufficient to classify it at the higher level.

Level 1 (Essential): Objects on this level are focused on offer a view of them in the digital world, this is

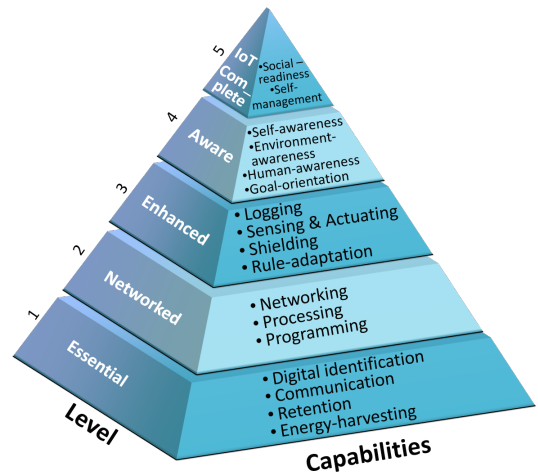


Fig. 4. Smart Object Capability Levels

achieved with the core capabilities they all have, i.e. identification, communication, retention and energy-harvesting. It includes most basic RFID or NFC tagged objects. *Example:* A Smart bag with an embedded RFID tag which is able to identify itself in any place, including airports, train stations, hotels, offices, etc. Depending on the type and capacity of the tag, it provides information about its origin, destination and owner among others. Owners would update travel information and track the location of the bag through specific mobile or web applications. It is not programmable since it only retain static information about its status or owner details. Further classification within this level can be achieved using the RFID classification model [18].

Level 2 (Networked): Objects at this level exhibit primarily processing and programmability capabilities. Although the latter could be fixed and very basic, it is needed to define what to process. Besides these two, SOs could maintain the basic communication mechanism defined in Level 1, however, it is usual that, having processing, also networking capabilities are added. Many objects have a single fixed purpose, similar to embedded systems, while others are more flexible near to multi-purpose personal computers. These capabilities enable objects to connect to the internet, directly or through a gateway, and interface with services and applications based on programmed tasks. DIY objects built over popular prototyping architectures, as arduino, raspberry or beaglebone, start to appear here. *Example:* A smart light bulb offers the possibility of changing the light intensity and color, by using its wireless connection, controlled from any web browser or smart phone application. To keep the bulb simple it is not able to adjust autonomously to its environment (e.g. through ambient light detection) however surrounding things could manage rules (e.g. scheduling) to activate or deactivate the bulb with specific settings.

Level 3 (Enhanced): SOs can either have one, some or all the capabilities of this level. Objects with sensing and actuating capabilities can gather or actuate based on the, usually limited and unrelated, number of sensors and actuators attached. Similarly, logging capabilities involves some tracking of the activities being carried out by the object. Objects can also offer services for protecting the data being gathered and deterring third-parties to appropriate it.

Example: The smart running shoes are able to sense the movement and determine the distance that the user has run. These shoes gather the data from a piezoelectric sensor and log the activity information in an internet repository. Shielding is basic, based on a user profile, but it does not offer full coverage to security and privacy requirements.

Level 4 (Aware): This level presents evolutions of the respective Level 3 capabilities, including numerous sensors and actuators as well as full monitoring of the operation and activities of the components of the SO. As in the previous level, these capabilities can be further developed individually, therefore it is possible to find objects that have self-awareness but not environment-awareness.

Example: A smart bike helmet is able to sense position, light conditions, weather and status of its components. This helmet includes a light that is activated when the light conditions are poor. In addition, if there is an accident and part of the structure of the helmet is damaged, it can request the user to deactivate the emergency notification, or otherwise emergency services are alerted the occurrence of serious accident.

Level 5 (IoT Complete): This is the top evolution in which SOs demonstrate management capability and learn from their social experience and from the environment. They are able to base decision making on reasoning of experienced situations. It includes the use of artificial intelligence approaches such as machine-learning, machine consciousness to generate knowledge and simulate-human-brain behaviour.

Example: A smart car that assists in multiple scenarios by communicating with nearby objects and controlling its own components. One scenario is when it is approaching the destination, the car requests information from deployed sensors (e.g. a light sensor can determine when space is available in a parking facility) in the neighbourhood to identify the best parking places. The car then adjusts the route to go directly to the identified place. In the same way, the car monitors the status of the gas, battery, oil, tires, etc. and adjusts driving (e.g. economic route, power-saving mode, etc.) when a specific threshold is reached. It also identifies the cheapest rates for gas oil and plans the next journey to include a stop in the identified gas station. It also books an appointment to change oil or battery when required. It offers secured user-friendly interfaces and can even talk to the passengers in several languages.

SOs derived from the ones cited in the examples would not be always classified in the stated level since several SO can be derived from the same original object and then classified in different levels. The model proposes an evolution path that SOs follow towards the IoT vision.

B. Evolution through the Levels

Although our model is inspired by CMM[®], the promotion – possible when SO is upgradable – in the established levels does not completely rely on the maturity of the object, but also on the increment of the exhibited capabilities. These can be further developed and evolved, sometimes it implies or requires the existence of new ones. For example, a SO can have a basic communication mechanism, but to develop networking capabilities requires processing. The SO is classified according to capabilities it actually exhibits and the evolution – when possible – takes place if it shows at least one of the capabilities

of that level, as long as, there are no dependencies (See section IV-B) on other capabilities of the level. e.g. Processing is required for Networking but not in the other direction.

In levels 3 and 4 SOs can specialize in any of the corresponding capabilities, and progress only in the group of capabilities that are most developed. For example, one SO can specialize towards a goal-oriented programming approach, but still only offering basic sensing and actuating capabilities (It is not reaching the next level in the environment factors capabilities). In the other case, Objects can develop every defined capabilities towards the IoT complete SO. Note that evolution comes from a common core, then it can be split out within the dimensional capabilities and then it can be joined again in the level-5 capabilities, which rely on the existence on multiple lower-level capabilities.

Autonomy and usefulness are the two boosters of the evolution in the model. The more decisions the SOs are able to make and the more activities they are able to carry out by themselves, the higher SOs are in the evolution.

C. Validation of the model

Validation of the model is based on the described examples that demonstrate both how actual objects can fit into each level and how they are distinguished from other objects at other levels. Commercial references are deliberately avoided, however, many existing products would fit into levels 1 to 3.

VI. DISCUSSION AND CONCLUSIONS

In the coming years it is likely that SOs will come with all capabilities envisioned so far and beyond (e.g. smart phones and similar mobile devices have more capabilities with each generation). However, while such new devices might directly emerge at level 5, multiple objects are already populating the environment and will remain with lower level capabilities. Also, there will be other drivers, such as cost, to continue to deploy low level devices for certain situations. Platforms will need to differentiate objects and propose efficient models to take advantage of the resources and capabilities of each one.

A classification model for SOs introducing five layers of capabilities that SOs develop in the IoT arena was presented. Relevant works in SOs definition and classification were reviewed, as the base for definition of capabilities. The main contributions of our work are: the classification based on an evolutionary path of capabilities, clarification of the relations and differences of main terms used to refer to SOs and a review of existing work on classification of SOs.

This model is intended to help determine what the object is able to do by itself and what requirements can be covered externally by applications, services, platforms or other objects. This model thus allows classification of Middleware solutions, according to the kind of SOs they can work with. The IoT will rely in the aggregated capabilities of the linked objects. While it is possible to provide some of the capabilities in the Middleware or applications that use the devices, we see it as desirable to provide capabilities directly on-device as this increases independence from the context and allows for easier integration of new objects. Such easier integration of objects is part of the open challenges on interoperability and trust needed to be resolved for the IoT vision to become a reality.

Perspective	Possible Uses
Software Engineers	<ul style="list-style-type: none"> To easily identify requirements of an SO to run an application or use an offered service. To determine either if Middleware solutions or platforms are suitable for specific SO or not. Drive adaptation of code based on SO types.
Researchers	<ul style="list-style-type: none"> As a reference framework for organising achievements addressing SOs challenges. As a tool for scoping and inferring assumptions for SOs classes.
End Users & Industry	<ul style="list-style-type: none"> Enable quick understanding, differentiation and identification of SOs product characteristics and differences for users and manufacturers.
Common	<ul style="list-style-type: none"> Provide an unifying language for referring to SOs with similar characteristics.

Fig. 5. Possible uses of the classification model

Currently, we are specially interested in the application of the model in the implementation of Middleware able to adapt services according to SO capabilities. Capabilities can be discovered using a simple SO command or a pre-set property (e.g. mainly in level 1 objects). Calling this command (reading the property) would return SO details including the capability level, driving Middleware rules for using on-object or cloud resources and services. Some of the uses for the model from different perspectives are summarized in Fig. 5.

We are working on a Middleware implementation relying on a multi-layered agent architecture. An agent first is intended to identify the type of SO and then choose suitable plans to achieve SO's goals. These plans include use of a mix of on-object and cloud services. The Middleware then serves as broker to cloud services enhancing on-object capabilities which are identified based on the proposed model.

We followed a top-down approach as we part from the inspiration we had (literature review and CMM[®]) rather from specific objects (bottom-up). A further validation could consist on identifying a sample of products and DIY objects, classify them based on similarities—for example using a clustering analysis—and compare the results against the five-level model.

An area of work that we have not considered, but that in the longer term might be worthwhile to consider is the difference between SOs and robots. Robots exhibit much of the features of level 4 and 5 SOs blurring the line separating the two.

For us the most interesting avenue of future work, and the motivation for conducting this classification, is to enhance support for easier integration of objects into systems, reducing the burden on developers that is needed with current Middleware. This is not an easy task, but a capacity oriented framework would allow for easier identification of features and more generic profiles of objects. Another dimension of future work could be the classification of more currently available objects, building up a structured catalogue of SOs.

VII. ACKNOWLEDGMENT

M E Pérez Hernández is supported by the Graduate Teaching Assistantship program of the University of Leicester. Part of this work was conducted while S Reiff-Marganec was on study leave supported by the University of Leicester.

REFERENCES

- [1] M. Weiser. (1996, Mar.) Ubiquitous Computing. [accessed 03-June-2014]. [Online]. Available: <http://www.ubiq.com/hypertext/weiser/UbiHome.html>
- [2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
- [3] R. Sokal, "Numerical Taxonomy," *Scientific American*, vol. 215, no. 6, pp. 106–116, Dec. 1966.
- [4] S. E. Institute, "CMMI for Development, Version 1.3 CMMI-DEV, V1.3," Tech. Rep. November, 2010.
- [5] C. Y. Wong, D. McFarlane, A. Ahmad Zaharudin, and V. Agarwal, "The Intelligent Product Driven Supply Chain," in *IEEE International Conference on systems, man and cybernetics*, vol. 4. IEEE, 2002, p. 6.
- [6] C. Carabelea and O. Boissier, "Multi-agent platforms on smart devices: Dream or reality," in *Proceedings of the Smart Objects Conference (SOC03)*, Grenoble, France. Citeseer, 2003, pp. 126–129.
- [7] J. Ma, "Smart u-Things - Challenging Real World Complexity," in *IPSI Symposium Series*, vol. 19, Japan, Nov. 2005, pp. 146–150.
- [8] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen, "Smart-its friends: A technique for users to easily establish connections between smart artefacts," in *Ubicomp 2001: Ubiquitous Computing*. Springer, 2001, pp. 116–122.
- [9] B. Sterling, *Shaping Things*. Cambridge, Mass.: MIT Press, 2005.
- [10] J. Bleecker, "A Manifesto for Networked Objects Cohabiting with Pigeons, Arphids and Aibos in the Internet of Things," pp. 1–17, 2005.
- [11] M. Bauer, M. Boussard, N. Bui, and F. Carrez, "Project Deliverable D1.5 Final Architectural Reference Model for IoT," IoT-A Project - UniS, Tech. Rep., Jul. 2013, [accessed 16-March-2014]. [Online]. Available: http://www.iiot-a.eu/public/public-documents/d1.5/at_download/file
- [12] M. Mühlhäuser, "Smart Products : An Introduction," in *Constructing Ambient Intelligence Workshops*. Springer, 2008, pp. 158–164.
- [13] C. Gutierrez, J. Garbajosa, J. Diaz, and A. Yague, "Providing a Consensus Definition for the Term "Smart Product"," in *Engineering of Computer Based Systems (ECBS)*. IEEE, Apr. 2013, pp. 203–211.
- [14] J.-P. Vasseur and A. Dunkels, *Interconnecting Smart Objects with IP: The Next Internet*. Burlington, MA: Elsevier-Morgan Kaufmann, 2010.
- [15] F. Mattern, "From smart devices to smart everyday objects," in *Proceedings of smart objects conference*, 2003.
- [16] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, "Smart objects as building blocks for the internet of things," *Internet Computing, IEEE*, vol. 14, no. 1, pp. 44–51, 2010.
- [17] G. Biamino, "A Semantic Model for Socially Aware Objects," *Advances in Internet of Things*, vol. 02, no. 03, pp. 47–55, 2012.
- [18] F. Armenio, H. Barthel, L. Burstein, P. Dietrich, J. Duker, and J. Garrett, "The EPCglobal Architecture Framework," GS1 EPC Global, Tech. Rep., 2007.
- [19] G. G. Meyer, K. Främling, and J. Holmström, "Intelligent Products: A survey," *Computers in Industry*, vol. 60, no. 3, pp. 137–148, Apr. 2009.
- [20] A. Moawad, V. Efthymiou, P. Caire, G. Nain, and Y. Le Traon, "Introducing conviviality as a new paradigm for interactions among IT objects," in *Workshop on AI Problems and Approaches for Intelligent Environments*. CEUR-WS.org, 2012, pp. 3–8.
- [21] L. Atzori, A. Iera, and G. Morabito, "From "smart objects" to "social objects": The next evolutionary step of the internet of things," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 97–105, Jan. 2014.
- [22] T. Sánchez López, D. C. Ranasinghe, B. Patkai, and D. McFarlane, "Taxonomy, technology and applications of smart objects," *Information Systems Frontiers*, 2009, [accessed 04-March-2014]. [Online]. Available: <http://link.springer.com/article/10.1007/s10796-009-9218-4/fulltext.html>
- [23] A. Schmidt, M. Beigl, and H.-W. Gellersen, "There is more to Context than location," *Computers & Graphics*, vol. 23, no. 6, pp. 893–901, Dec. 1999.
- [24] M. E. Whitman and H. J. Mattord, *Principles of Information Security*, 4th ed., C. Technology, Ed. Boston, MA: Cengage Learning, 2011.