Towards an IoT Ecosystem

Flavia C. Delicato[1], Paulo F. Pires[1], Thais Batista[2], Everton Cavalcante[2], Bruno Costa[1], Thomaz Barros[1]

[1] Department of Computer Science, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
[2] Department of Informatics and Applied Mathematics, Federal University of Rio Grande do Norte, Natal, Brazil

{fdelicato, paulo.f.pires, thaisbatista, evertonranielly, brunocosta.dsn, thomazavila}@gmail.com

ABSTRACT

In the near future, it will be possible that every object on Earth can be identifiable and addressable. Such objects will be able to be monitored and monitor their physical environments, and of executing actions on such environments in benefit of human users. Moreover, these so-called smart objects will be endowed with wireless communication capabilities. By being uniquely addressed, wireless endowed and through the use of existing protocols and standardized formats, smart things can be integrated in the Internet and accessed as any other Web resource. In this context, the Internet of Things (IoT) emerges as a paradigm in which smart things actively collaborate among them and with other physical and virtual objects available in the Web, providing value-added information and functionalities for users. The IoT paradigm has recently showed its potential of considerably impacting the daily lives of human beings mainly due to the use and interaction of physical devices in several domains, including complex systems composed of other systems. In this paper we discuss the IoT paradigm from the perspective of Systems-of-Systems and present EcoDiF, a IoT platform that integrates heterogeneous devices to provide real-time data control, visualization, processing, and storage. In EcoDiF, devices, information, users and applications are integrated to create an IoT ecosystem in which new ideas and products can be developed in an organic way.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems – *Distributed applications.* D.2.11 [**Software Engineering**]: Software Architectures – *Domain-specific architectures.*

General Terms

Design, Management, Standardization.

Keywords

Systems-of-Systems, Internet of Things, middleware.

1. INTRODUCTION

In the near future, it will be possible that every single object on Earth can be identifiable, addressable and accessed through the Internet.

Such objects will be endowed with wireless communication capabilities and able of monitoring and acting upon their physical surroundings. These so-called "smart things" integrated in the Internet will actively collaborate among them and with other physical and virtual resources available in the Web, providing value-added information and functionalities for end users. Such emergent scenario constitutes the Internet of Things (IoT) paradigm [2]. The IoT has been discussed in literature for some time, with slightly different definitions. In this paper, we considered the following definition, proposed in the CASAGRAS project [1]: "A global network infrastructure, linking physical and virtual objects through the exploitation of data capture and communication capabilities. This infrastructure includes existing and evolving Internet and network developments. It will offer specific object-identification, sensor and connection capability as the basis for the development of independent cooperative services and applications. These will be characterized by a high degree of autonomous data capture, event transfer, network connectivity and interoperability." Among other features, the IoT characterizes by a high degree of heterogeneity (considering hardware and software elements), a very dynamic environment, and a massive amount of information being exchanged. Besides, data produced by wireless devices such as sensors are prone to inaccuracies related to environmental noises and the precision of the sensing units. Therefore, IoT applications need to deal with a high degree of uncertainty.

An important class of applications envisioned for the IoT will be developed in an opportunistic and ad hoc fashion by (potentially) lay users, on top of smart things, by using, for instance the approach of web mashup [7]. Such applications will rely on a subset of a collection of Web accessible services and data streams provided by devices made available by third parties, intermittently connected to the network. The interaction model among the different services of an application needs to be more relaxed than in traditional Web service compositions. First, the intermittent behavior of devices makes the interactions with them unpredictable. Second, the lifecycle of data streams in the IoT environment is also unpredictable since a device owner can decide to withdrawn his/her device from the system at any time. Third, the dependencies among application services are not fully known at design time since the IoT environment is characterized to be a pool of services opportunistically offered and composed in an organic way.

The aforementioned features make IoT to be characterized as a System-of-Systems (SoS). There is no consensus on the SoS definition, but according to Maier and Rechtin [15] an SoS can be seen as a composition of systems in which its constituent systems are individually discovered, selected, and composed possibly at run-time to build a more complex system. The constituent systems are managed (at least in part) for their own purposes rather than the purposes of the whole and maintain a continuing operational existence independent of the collaborative system. The resulting composed system (the SoS) is more complex and offers more

SESoS 2013 Montpellier, France

functionality and performance than simply the sum of its constituent systems.

In this paper we present some motivational scenarios for the IoT, and then we briefly describe requirements and examples of existent IoT platforms, detailing EcoDiF, one of such platforms, that aims to integrate heterogeneous devices for providing real-time data control, visualization, processing, and storage. In EcoDiF, devices, information, users and applications are integrated in order to create an IoT ecosystem in which new ideas and products can be developed and executed in an organic way.

2. MOTIVATIONAL SCENARIOS

IoT systems are able to transmit real-time information about natural processes (e.g. temperature, wind, vibration of structures, rain, water level of rivers, etc.) and such information can be used in an integrated way with information and decision making systems for different purposes, e.g. environmental protection, natural resources management, structure health monitoring, biodiversity conservation, and prevention of environmental disasters. Similarly, geophysical and climatic events such as earthquakes, volcanic eruptions, floods, etc. can be foreseen or even avoided through an IoT-based system-of-systems. The combination of remote sensing for monitoring with alert and action mechanisms can compose a complex system to notify authorities and citizens about an imminent disaster or take the proper actions in order to control or avoid such event. Hence, the use of complex IoT-based systems enables not only the automation of monitoring processes, but may prevent global-scale catastrophes that may damage cities and affect the life of the people. Another interesting real-world scenario that requires the seamless integration of heterogeneous devices is related to the monitoring of urban infrastructures. The increasing mobility of the people has brought a high cost to the societies as a consequence of the increase of traffic jams, also responsible for increasing in the pollution level, the stress and anxiety in drivers and users. In the context of smart cities, the traffic jam problem regarding public and private transport in the urban conglomerates may be drastically minimized through the real-time monitoring of public infrastructure information. Several urban data can be collected and disseminated through communication infrastructures that require integrated, heterogeneous, smart wireless communication ways and that may encompass a wide range of networked devices, such as sensors embedded in vehicles or installed on the streets and roads, as well as several citizens' mobile devices (e.g. smartphones, tablets, laptops, etc.) [1]. However, when a GPS system (location information provided by specific location devices or even cellphones) is used to draw a route, it often uses only a map as information source and does not consider other aspects such as road flooding, precipitation, traffic, etc. In order to use the collected data in the urban traffic control and monitoring context, a complex system can combine data provided by several systems and integrated devices (not limited to sensors, but including any kind of device) so that the generated information can support users in their decision-making processes and then generate added-value information. For instance, when a flooding or a tendency to traffic jam (based on the mean speed of the vehicles) is detected, systems that make use of such combined data can identify and distribute alternative routes to the user in an immediate and proactive way. Furthermore, the users themselves may have an active role in this context by feeding such systems with data provided by them.

By being able to monitoring and managing the physical environment IoT technologies have been pointed out as means to achieve the sustainable development, for instance when applied to smart building applications [16]. The aforementioned real-world scenarios are just few examples that clearly characterize an *IoT ecosystem* as a typical system-of-systems (SoS).

3. Requirements of IoT Platforms

One way of managing the complexity of building SoS like IoT systems is to provide a middleware platform that offers a set of services that fulfill the common non-functional requirements of such systems [9][10]:

1. **Scale**. When performing a task over a set of millions of devices, it is practically impossible to automatically coordinate all devices due to constraints such as time, memory, processing power and energy consumption;

2. High degree of hardware and software **heterogeneity**. Therefore, mechanisms to manage the interoperability between heterogeneous devices in several application domains must be provided.

3. **Uncertainties**. A major characteristic of a communication infrastructure in IoT environments is that its topology is unknown and dynamic. Moreover, the devices location may not be precisely determined. Consequently, applications often depend on services and data provided by devices that may not be reachable at the time of their use. Therefore, mechanisms to provide dynamic discovery of devices, adaptation and context-awareness of applications are needed. Furthermore, device metadata as well as sensor readings may be inaccurate or incomplete, thus requiring, for instance, mechanisms to provide (meta)-data reconciliation and inference.

4. **Conflict resolution**. Conflicts arise, for example, when multiple applications attempt acting on the same device in opposite ways or when trying to apply mutually incompatible changes over the environment. Such requirement demands mechanisms to specify and manage policies and relationships among different applications needs.

5. Issues related to the management of **massive data**, **privacy**, and **security** in the IoT context [10].

There are several researches that address the conception and implementation of IoT middleware platforms to address some of the aforementioned requirements. In the next section, we briefly present some infrastructures for IoT in terms of the integration of heterogeneous devices and composition of applications that take advantage of such integration. Most of them are not in a mature state and the proposed solutions are still under development.

4. IoT Platforms

An IoT middleware platform should provide abstractions for the physical devices and services to applications and end-users, thus abstracting away low-level details regarding the highly heterogeneous, dynamic, uncertain, and distributed IoT environment.

An example of IoT middleware platform is EcoDiF (*Web Ecosystem of Physical Devices*), which is being developed in the context of a project funded by the Brazilian agency RNP (The National Education and Research Network). EcoDiF aims to integrate heterogeneous physical devices in order to provide services to support real-time data control, visualization, processing, and storage. As a typical SoS, EcoDiF can be decomposed into its constituent systems. An SOS for IoT typically encompasses: (i) the system composed of smart things, which may be sensors, actuators, or various other types of devices; such system also includes data generated by these objects and the wireless networks necessary to connect them, (ii) the system composed of applications that run on top of the objects, using the data provided by them and other resources and services available on the Web, and (iii) the socio-technical system composed of users and interfaces that allows them to interact opportunistically among themselves and with the interconnected devices and applications. In this context, EcoDiF is a platform that enables the integration of devices, information, users and applications creating an IoT ecosystem in which new ideas and products can be developed in an organic way. Figure 1 shows an overview of EcoDiF illustrating the integration of heterogeneous physical devices and the use of the provided data by different kinds of users.

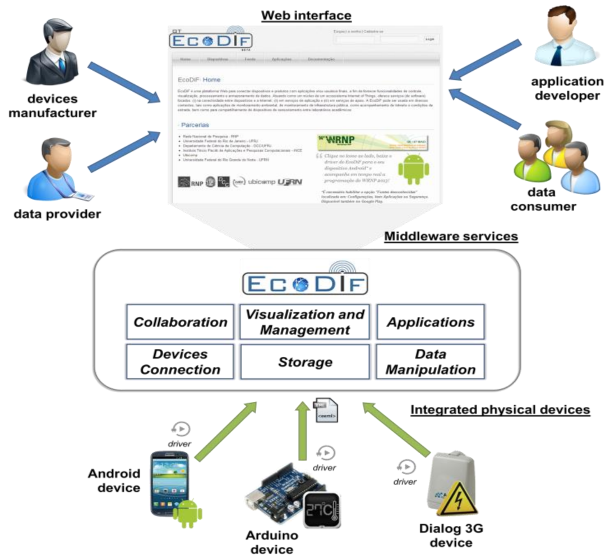

Figure 1. EcoDiF architecture.

EcoDiF envisions four types of stakeholders (Figure 1): (i) *devices manufacturers*, which develop *drivers* to their devices to make them compatible with the EcoDiF open API, as well as *data profiles* that specify the metadata that describes the type of data provided by the their devices; (ii) *data providers*, which are device owners that want to make the data produced by their devices available at the IoT ecosystem through EcoDiF; (iii) *application developers*, which build Web applications or services that use as inputs the data (raw data provided by connected devices or more refined data produced as output of other applications) available at EcoDiF as well data produced by any other Web accessible resource, for their business purposes, and; (iv) *information consumers*, which are users that interact with the EcoDiF ecosystem through its Web or programming interface in order to search and/or use the information (raw or processed data, applications, etc.) available at the IoT ecosystem.

4.1 EcoDiF: Architecture and Implementation

Figure 1 also shows the modules that compose the EcoDiF logical architecture. The *Devices Connection Module* aims to facilitate the connection of physical devices to EcoDiF and, consequently, to the Internet. In this perspective, manufacturers configure their devices according to the EcoDiF's specific API to enable the integration with the platform, as well as users connect their pre-configured devices to perform the operations of the provided API. This connection between the respective device and EcoDiF is enabled by a customized *driver* to the specific device, so that data providers can use such driver for connecting their devices and make available, at EcoDIF, the data obtained from the devices. These data, the so-called *feeds,* are represented using EEML (*Extended Environments Markup Language*) [4], an XML-based language that describe data obtained from devices in a specific context (environment). The acquired data is sent by the device to EcoDiF (through the specific driver) by making an HTTP PUT request following the REST (*REpresentational State Transfer*) [5],[6] architectural style, so that users can manipulate such data at real-time through EcoDiF *Data Manipulation Component*.

The *Visualization and Management Module* provides a Web interface to enable users managing the devices connected to EcoDiF. Through such interface, users can monitor the state and localization of their devices, create alerts and notifications (*triggers*) about the environment, and visualize historical data. The *Collaboration Module* aims to facilitate the collaboration among EcoDiF users, enabling them to perform searches for devices and applications registered at the platform from their respective metadata (type, user, localization, etc.). This module provides a Web portal through which users interact with EcoDiF. The *Storage Module* basically consists of two repositories: (i) a repository for *storing data* using a relational database, and; (ii) a repository for *storing application scripts* in a file system. It is important to highlight that these repositories may use a Cloud Computing infrastructure to store relational data and files, thus ensuring their robustness, reliability, security, availability, and scalability.

Finally, the *Applications Module* is intended to provide a model and environment for programming and executing applications that can make use of the data (feeds) available at EcoDiF and generate new information to be available at the platform. In EcoDiF, these applications are built as Web *mashups* [7], which are ad-hoc Web applications created from the composition of different types of information provided by several sources. For instance, a sensor monitors the temperature in a given location, but the user wishes to combine this information with a map that informs the localization of the collected measures. Therefore, a single mashup application can be developed by composing the temperature and localization information. The EMML (*Enterprise Mashup Markup Language*) [8] XML-based language is adopted as standard language for developing Web mashup applications by integrating data from several sources, as well as from Web services, third-party APIs, and relational databases.

4.2 Other IoT Platforms

OpenIoT [11] project aims to develop an open-source middleware for IoT applications in a *Sensing as a Service* model, which is available at a cloud environment that can be transparently accessed and configured by users. OpenIoT connects sensors with the cloud environment, so that the cloud resources can be used for data processing and management, which are useful functionalities that usually cannot be performed at the IoT infrastructure because the resources are limited. *Cosm* [12] (former *Pachube*) is a Web environment for managing data provided by sensors. It provides an API for directly sending data from the sensors, enables to visualize historical data, and provide mechanisms to trigger events based on the acquired sensing data. In Cosm, data are organized in terms of *feeds*, *datastreams*, and *datapoints*. Feeds represent the environment data with its respective datastreams, which represent the data provided by the sensors, whereas datapoints represent a specific value of a datastream at a given time. In addition, the data available at Cosm can be retrieved in the JSON, XML (more specifically, EEML), and CSV formats. The *Hydra* project [13] (now *LinkSensor*) proposes a service-oriented middleware for embedded systems that supports the development of applications composed of heterogeneous physical devices. Such middleware provides interfaces based on Web services for controlling any kind of physical device and enables developers to incorporate physical devices into their applications.

Finally, RestThing [14] is a REST-based Web service infrastructure intended to abstract away the heterogeneity of physical devices and provide a way to integrate devices embedded in Web applications. RestThing platform enables developers to create REST-based applications by combining physical and Web resources so that the devices and information are both represented as resources and manipulated through a uniform RESTful interface, thus enabling accessing them by using the conventional operations defined in the HTTP protocol. In addition, the accessed information can be retrieved in JSON, XML and CSV formats.

The IoT platforms presented in this paper have features in common, mainly in terms of the integration and interoperability among heterogeneous physical devices. However, only EcoDiF enables building applications by using the mashups technology. In this sense, EcoDiF is the first initiative in the IoT context that is tackling the complexity associated with the dimension of IoT SoS related to handling the programing of loosely coupled, opportunistic applications whose components interact and depend upon each other in a unpredictable way.

5. Discussion

Despite of already properly dealing with several requirements of the IoT environment, such as high heterogeneity and dynamism, current IoT platforms have several limitations, and lack mainly to tackle aspects of SoS. Among the limitations, we can identify:

1. Lack of mechanisms to deal with the uncertainty of sensor data and meta-data, and with the existence of conflicting application requirements;
2. Need for new architectural approaches for building applications that take into account the complexity of the interaction patterns among application components, often not known at design time, and that can lead to unforeseen or undesired behaviors. In particular, the application developer must have the support of tools to assess, before the deployment of the application in the integrated environment, the possible effects of its execution on other systems currently active.

Another aspect that requires close attention is how IoT systems should be thought. The traditional Software Engineering approach, which focus on breaking down the individual pieces of what is being analyzed, is not suitable to study dynamic and complex systems such as SoS [15]. Approaches such as Systems Thinking [18] are being pointed out as a good alternative to analyze SoSs. Systems Thinking focuses on how the object of study interacts with the other parts of the system of which it is a part. Therefore, Systems Thinking works on the opposite way of traditional modeling by expanding the analysis to take into account larger numbers of interactions as the focus of the analysis progresses. Such way of exploring interactions tends to reveal behaviors that are difficult to identify in traditional approaches, mainly when the system is dynamically complex or is influenced by a great number of different sources (internal or external). Since these are typical IoT system features, it seems promising to explore how approaches such as Systems Thinking, System Engineering [18], and alike can be used to help modeling and management such environments.

6. ACKNOWLEDGMENTS

This work is supported by the Brazilian Education and Research Network (RNP) through the GT-EcoDiF Working Group.

7. REFERENCES

[1] H. Sundmaeker, et al., Vision and challenges for realising the Internet of Things, Cluster of European Research Projects on the Internet of Things - CERP IoT, 2010.

[2] Luigi Atzori, Antonio Iera, Giacomo Morabito (2010) The Internet of Things: A survey. Computer Networks, vol. 54, pp. 2787-2805.

[3] Macedo, D. F. et al. 2012. $(CIA)^2$-ITS: Interconnecting mobile and ubiquitous devices for intelligent transportation systems. In *Proceedings of the 2012 IEEE Int. Conf. on Pervasive Communications Workshops*, USA, 447-450.

[4] Extended Environments Markup Language. Available at: http://www.eeml.org/.

[5] Fielding, R. 2000. Architectural styles and the design of network-based software architectures. PhD Dissertation. University of California - Irvine, USA.

[6] Webber, J., Parastatidis, S., and Robinson, I. 2010. REST in practice: Hypermedia and systems architecture. O'Reilly, Sebastopol, CA, USA.

[7] Guinard, D. et al. 2009. Towards physical mashups in the Web of Things. In *Proceedings of the 6th Int. Conf. on Networked Sensing Systems*. IEEE Computer Society, USA, 1-4.

[8] OMA EMML Documentation. Available at: http://www.openmashup.org/omadocs/v1.0/index.html.

[9] Bandyopadhyay, S. et al. 2011. Role of middleware for Internet of Things: A study. *International Journal of Computer Science & Engineering Survey* 2, 3, 94-105.

[10] Teixeira, T. et al. 2011. Service-oriented middleware for the internet of Things: A perspective. *Proc. of the 4th European Conf. on Towards a Service-Based Internet*. Springer-Verlag, Germany, 220-229.

[11] Soldatos, J et. al. 2012. Covergence of Utility Computing with the Internet of Things. *Proc. of the 6th Int. Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing*, USA, 874-879.

[12] Cosm: IoT platform connecting devices and applications for real-time control and storage. Available at: http://cosm.com/.

[13] The Hydra (LinkSensor) Project. Available at: http://www.hydramiddleware.eu.

[14] Qin, W. et al. 2011. RestThing: A RESTful Web service infrastructure for mashup physical and Web resources. *Proceedings of the 9th IEEE/IFIP Int. Conf. on Embedded and Ubiquitous Computing*, USA, 197-204.

[15] Maier, M.W., and Rechtin E. The Art of Systems Architecting, 2nd Edition. CRC Press, London, 2000.

[16] Schor, L. et al. (2009). "Towards a Zero-Configuration Wireless Sensor Network Architecture for Smart Buildings", Proc. of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, pp 31–36.

[17] Jackson, M. C. (2006). Creative holism: A critical systems approach to complex problem situations. Systems Research and Behavioral Science, 23(5), 647–657.

[18] Petkov, Petkov et.al. (2010) Towards a wider application of the systems approach in information systems and software engineering. In: Paradice, D., (ed.) Emerging Systems Approaches in Information Technologies: Concepts, Theories, and Applications. IGI GLOBAL, pp. 67-85.