Internet of Things to Provide Scalability in Product-Service Systems

Danúbia Espíndola, Nelson Duarte Filho,
Silvia Botelho, Jônata Carvalho
Center for Computational Science
Federal University of Rio Grande
Rio Grande, RS, Brazil
{danubiaes, dmtnldf, silviacb,
jonatatyska}@furg.br

Carlos Eduardo Pereira
School of Electrical Engineering
Federal University of Rio Grande do Sul
Porto Alegre, RS, Brazil
cpereira@ece.ufrgs.br

***Abstract*—In a convergence context between Internet of Things - IoT and Product-Service Systems - PSS, this paper presents a way to integrate concepts and build computing systems to assist PSS. The conceptual basis and a architecture for a middleware to do this convergence was investigated. A prototype of the middleware was built and was ran in a simulated case study in the marine industry. It was concluded that the adopted ideas are feasible and an implementation of the system has viability under performance point of view, in the industry of construction and assembly.**

Keywords; Product-Service Systems; Internet of Things; Human-Computer interaction; Ubiquitous and Pervasive Computing; Marine Industry.

I. INTRODUCTION

Product Service Systems, put simply, are when a firm offers a mix of both products and services, in comparison to the traditional focus on products. PSS, as defined in [1], is "a marketable set of products and services capable of jointly fulfilling a user's needs". PSS can be realized by smart products. In [2], there is also a definition as follows: A PSS is pre-designed system of products, services, supporting infrastructures and necessary networks that is a so-called dematerialized solution to consumer preferences and needs. It has also been defined as a "self-learning" system, one of whose goals is continual improvement.

Researches on Product-Service are not recent. In the computing communities, the first papers date from the 60's [3]. However, terms like servicizing, dematerialization, green industry, functional economy and other concepts aiming at sustainability and competitiveness in production systems are still recent topics of discussion and challenges for the academy [4][5]. Challenges about the implementation and modeling of PSS systems give rise the research area called Service Engineering (SE) [6].

Service Engineering, term introduced by Bullinger in 1995, adopts a technique and systematic approach to the design and development of services using models, methods and tools [7]. One of the major challenges in SE is to deploy service architectures that allow users to incorporate new added-value services to products during the whole product's lifecycle.

Consumers desire new tools to interact with companies offering new services and they want to be able to customize and co-create value. In traditional methods of adding value, value creation is usually a flow from producers to consumers. In co-creation and customization, this distinction disappears once the consumers are engaged in the processes of both defining and creating value [8].

Furthermore, the customization and co-creation are part of a journey of customer's experiences using the products. The initial value is set by producers based on their assumptions on customers' expectations before they purchase the product. As users start interacting with the products and have their personal evaluations resource-based approach to a knowledge-based one, which takes into account users models and perspectives, has to occur.

However, the adoption of this new paradigm for (re)adding value, (re)use and customization, based on the different expectations from distinct agents, require new technologies for its implementation. New Information and Communication Technologies (ICT) infrastructures for acquiring and processing of information, such as smart devices, human-computer interfaces and computational models are required. These infrastructures must describe the relationship between product and service, and manufacturers and consumers, as well as allow the exchanging of knowledge among these agents throughout the product lifecycle. The lack of tools for this purpose is evidenced by the low number of studies in the literature in the Service IT area [7] and, at least so far, it is also due to the difficulty of embedding computing and communication in all elements of PSS.

Besides being useful in PSS context, tools and ICT infrastructures should consider the social issues and the value adding during the whole lifecycle (time domain), in order to lead a general taxonomy that encompasses technology, people and business. Thus, ICT play a fundamental role in supporting the "how to deploy" this transformation from resource-oriented paradigm to knowledge-oriented paradigm. In this sense, new ICT must be considered. For example, pervasive computing (or infiltrating) coupled with mixed reality techniques for human-computer interaction [9][10], might be valuable for discovering new horizons for PSS solutions. The importance of pervasive computing in this context arises from Weiser assertion in 1991: "the most profound technologies are those that disappear…" Also known as ubiquitous computing, the pervasive computing provides an environment highly integrated to user where the perception of being using computers is minimal [11]. It

 ISBN: 978-1-61208-236-3

shows clearly that this characteristic is crucial regarding to this approach, since the usage and knowledge about ICT should not be a required condition from its users.

To enable a combination of pervasive computing with the necessity of embedding ICT on all devices of PSS, the Internet of Things (IoT) has emerged as a possible and potential solution. The IoT is a novel paradigm that makes possible the pervasive presence around us of a variety of things or objects. Some devices, such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, mobile phones, and so on, permit, through unique addressing schemes, the interaction and cooperation between things (objects) to reach common goals [12].

Such technological elements, in which information can be collected, processed and accessed from "things", allow the computational modeling of previously disconnected systems. For example, situations, activities and processes that were present only in the social domain can be virtualized in digital frameworks by shaping and aggregating the circulating information among their agents.

In a convergence context between IoT and PSS, this paper presents a way to integrate the concepts and build computing systems to assist PSSs. The conceptual basis and a architecture for a middleware to do this convergence was investigated. A prototype of the middleware was build and was ran in a simulated case study.

The prototype addresses a simulated case study in the marine industry. This choice was made because in this kind of industry the use of information and communication technologies are still scarce and authors have access to a large shipyard that is being installed near of its University laboratories.

In what follows, it is presented the conceptual issues of a model and an architecture of a middleware which aims to be feasible and scalable. A prototype of the middleware is described and finally a simulated case study for validation of the approaches is presented.

II. CONCEPTUAL DESIGN

The conceptual model that follows should help readers to identify the potential use of IoT in PSS as a solution for development of services in SE.

One of the initial ideas with emergence of PSS systems was to extend the product lifecycle through the addition of services such as maintenance and upgrading [3]. Currently, the focus has changed from selling products to selling functions and services, which could even be downloaded and upgraded remotely – at customers location – allowing a customization of the purchased goods. Within this new context, new research topics related to this product "servicification" have emerged. These methodologies should provide means to evaluate the economic and environmental impacts using the product "servicification". Service-oriented approaches have been considered as a potential candidate for handling this problem in PSS studies and SOA (Service-oriented architectures) have been often adapted for use in PSS, in which the service provider is the manufacturer and the customer is the receiver.

Considering the elements involved in PSS and following the definition that a service is an activity [13], for the service development and deployment a provider module and a receiver module are required. The provider module must, as the name already implies, provide the service based on product information. This module is implemented in software and can use a set of programming techniques. Among the possible techniques that can be used, intelligent agents with support to expert systems for composition and delivery of services appear as an interesting option. Fig. 1 depicts an high level conceptual design and aims to provide an overview of modules involved in IoT and PSS in order to allow a better understanding of the issues that must be addressed.

Another PSS element that needs to be considered for the composition and delivery of services is the product lifecycle data. Information about products' features and operational status is taken into account during the creation of services. Product data can be static and dynamic. Static data are generally about product characteristics and are usually related to the BoL (Begin of Life) lifecycle. These data do not change as the time goes by and can be acquired from CAD design models of the product. Dynamic data are data that change in time and correspond to use and end stage product lifecycle (MoL – Middle of Life and EoL – End of Life).

Dynamic data can be either acquired at periodic time intervals or when an event happens. To generate knowledge from these data an intermediate processing stage is necessary. For example, data related to the product's energy consumption can be calculated periodically based on the temporal acquisition. This information can be used to generate a monitoring service of energy consumption costs for a product or a system. An example of dynamic data which is event-based is the occurrence of a failure or error, where the user has to be warned about the need of maintenance or replacement.

Finally, for the services provision is necessary to understand and to visualize the different service types. According to [14] the PSS can be classified into five types:

- Support services – are services to help the product design. Here, the term Services can be a combination of product/service, only product or only service;
- Sale services – provide explanation about usage, training and maintenance and are offered during the product sale as an additional value through the product functions;
- Use-based services – are services such as leasing, renting, pooling or sharing;
- Maintenance services – are services based on lifecycle data of product to extend her life or to avoid breakdowns in the operation;
- EoL services – are services to support recycling, reuse and refurbishing services.

So, an IoT interface must be defined to support these service types. This interface must be far beyond the identification of objects on the network. Its use intends that the system becomes as smart as possible without user

 ISBN: 978-1-61208-236-3

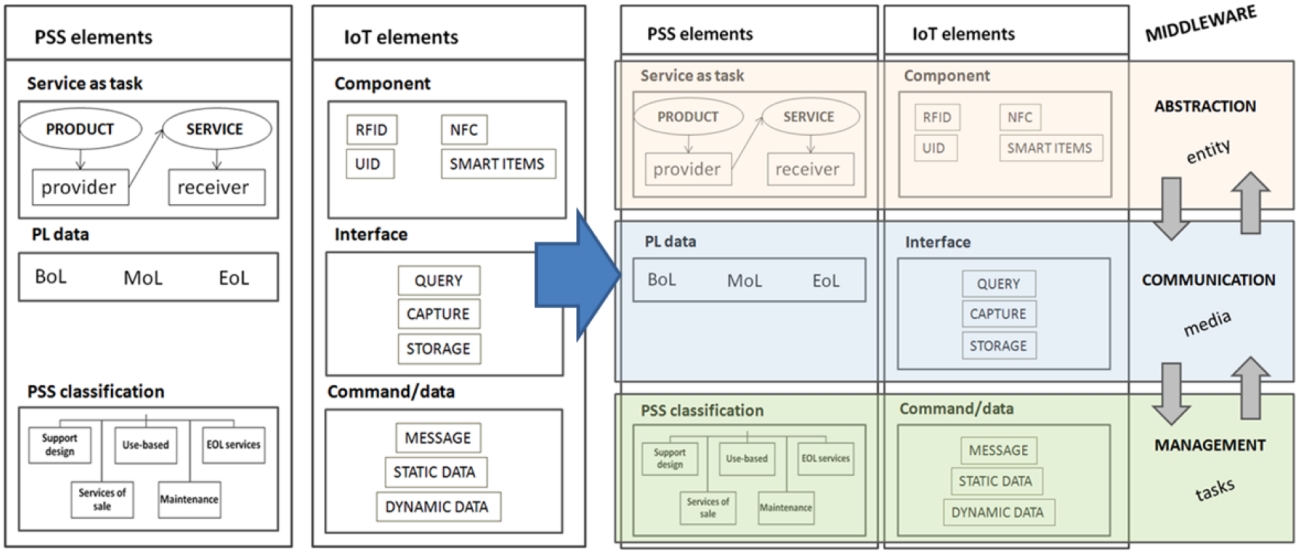

Figure 1. Elements of areas (left). The conceptual design (right).

intervention. That is, by having a pervasive computing system that makes users interact minimally with the devices.

It should allow devices to communicate with each other in order to generate knowledge for system operators and end users. The use of embedded components in products allows the automatic acquisition of information without any intervention from operators/customers. The idea is that devices like RFID, smart items and so on, do automatic data acquisition and store these data in repositories to support the creating of intelligent environments. The repositories should organize the informations in an intelligible way for that pervasive systems can supply services based on these data.

The basic elements for implementing the IoT paradigm are: the embedded components, the communication interface and the command/data module. The embedded components are devices such as RFID, UID, NFC or Smart items used for identification and data transmission to the processing module. Data transmission requires a communication channel. Issues about protocols, processing, energy, distance and storage should be considered when the embedded component is chosen. These components embedded in products constitute the pervasive computing that becomes the environment intelligent.

The other element of IoT paradigm is the information that is transmitted/exchanged by IoT components. This information can be data or commands. Aspects about information management in devices without processing capability should be considered. In addition, acquisition commands, such as those of command/data module of IoT must guarantee the processing of static and dynamic data from product lifecycle.

It is worth mentioning that the IoT paradigm is recent and the basic structure of representation of this technology still differs considerably [15] [16] [17] [18]. Researches that explore solutions for standardization are required in order to achieve a consensus. Although the paradigm is under construction and it is still new, the IoT is a result from multiple visions about the Internet of the future, what began to be discussed around 2005 with things like RFID and ubiquitous computing [19] [20] [21].

It is still important to consider the basics aspects about components like RFID, NFC, UID to be used for applying the IoT paradigm. For example, the spread of RFID usage was mainly due to their low cost and small size. However issues of energy (batteries and harvested), processing and communications (asymmetric and peer-to-peer) have motivated researches on networks like WSN (wireless sensor networks) and RSN (RFID sensor networks) and their applications on IoT. A detailed study comparing these technologies is presented by Atzori [16]. The main characteristics of RFIDs are: they usually do not have processing power, the communication is asymmetric, the communication range is around 10m, the energy is harvested, the useful life time is indefinite and the ISO 18000 standard is used.

III. A MIDDLEWARE FOR THE USE OF IOT IN PSS SYSTEMS

Based on the following premises: (i) re-use, re-customization and adding value are key elements for implementation of PSS, (ii) issues of social domain (environmental, social and economical aspects) hardly integrated PSS models; (i) and (ii) are associated with the acquisition, processing and use of large amounts of information distributed in time and space between different players through the lifecycle. A middleware to use the Internet of Things as a technological support to provide the effective service deployment is proposed. The IoT can provide a self-configuring network of wireless sensors, which can interconnect all space-time information. Re-use,

 ISBN: 978-1-61208-236-3

customization, social information from players and things, such as location, status and preferences can be tracked and updated in real time.

So, an IoT-based middleware is presented to enable the implementation of the PSS paradigm throughout the product/service lifecycle. It connects services and objects/things tracked by RFIDs, sensor and actuator networks, mobile phones, etc, virtual elements and human-computer advanced interfaces. The middleware considers the following aspects in order to provide general solutions:

- Architecture – event-driven and bottom-up organization with elements of the semantic web, aiming to handle exceptions and issues non-deterministics associated with the design, manufacturing and information obtained from the customers;
- Intelligence - the network should be non-deterministic and open. The real or virtual entities should be self-organized or intelligent, capable of acting in context dependent manner;
- Time issues - the system includes several parallel and simultaneous real time events. Time will no longer be used like a common dimension and linear, but will depend on each product-service. The middleware should be based on parallel and real time processing;
- Complex system - the middleware supports the complexity associated with the large number of different connections and interactions between players and its ability to integrate new actors during all lifecycle;
- Space considerations - the middleware provides the location of acquired information. An new definition for space-time patterns is essential.

In summary, the proposed middleware provides conditions for acquisition, processing, visualization and usage of different information that comes from the various players involved in the lifecycle of each product-service and enables tracking, sensing, data acquisition and data query, even in real time of "things" (people and all kinds of devices). This way, the middleware will support the construction of applications targeted to PSSs, whose requirements of the social domain are not neglected during the technical modeling of product-service relationship.

The proposed middleware aims to provide solutions for three key aspects to deployment and operation of PSS: abstraction (acquisition and processing data), communication (among different devices) and management (providing service as task), each one represented by a module layer of the middleware.

The first step of an IoT implementation in PSS systems is to embed devices like RFID, NFC, etc, on the products and distribute them in the environment. The integrations between devices and product will represent entities. These abstractions are necessary to represent elements in an object-oriented model.

After having represented the entities, the next step is to define the communication between them. Each entity must be able to identify new entities entering the environment and be able to use a communication media to transfer data of product and environment for the management module (better explained bellow). For this, standardization of messages is necessary in order to provide communication between different entities. The communication module will allow the data acquisition and through the communication interface will transmit data about the product lifecycle. The communication interface implements functions like query, capture and storage of information that comes from BoL (Begin of Life), MoL (Middle of Life) and EoL (End of Life) cycles. Figure 2 represents the layers of middleware.

Another component of the middleware is the management layer. The management layer is responsible for activites like information treatment, service composition and application interface. The management issues must provide means to handle static and dynamic data and trigger commands to other layers. The data acquired about product lifecycle should be processed to allow the developing of different types of service and present them in application interfaces. The application interfaces will present the services in web pages.

The three layers of the proposed middleware encompass the five aspects presented above and ensure an scalable open solution. The architecture aspects are provided by the conceptual structure of the middleware. The intelligence and the complexity of system are considered by the management layer, where the intelligence is implemented by the information treatment module. The space considerations are addressed by the abstraction layer. i.e., by embedding tags in products and using IoT components one can identify the presence (location) of the entities in the environment. The time aspects are solved by the communication layer.

To implement the proposed model of the middleware an abstraction was adopted. In the abstraction every entity has a communication interface and when an entity communicates with other, in context of a service without user interaction, a smart environment emerges. The set of smart environments is called Hyper-environment (H-ENV). The next section presents the H-ENV architecture based on entities, tasks and media to describe the middleware implementation done.

IV. THE H-ENV ARCHITECTURE AND MIDDLEWARE IMPLEMENTATION

Based on the technological environment depicted in the last section, the following components co-exist:

- Elements, referenced in this work as real elements, such as animate beings or inanimate objects present throughout the lifecycle of the product/service (staff, customers, tools, equipment, consumers, etc.);
- Ubiquitous and pervasive technologies that allow even non-existent or inanimate elements in the environment, to become devices - things capable of presenting interactive behaviors, perceive, make decisions and act in pervasive environments (e.g., RFID technologies, smart devices, sensor networks, PDAs, among others associated with equipments and devices);
- Communication and interface technologies that enable the integration of real elements, virtual elements and

 ISBN: 978-1-61208-236-3

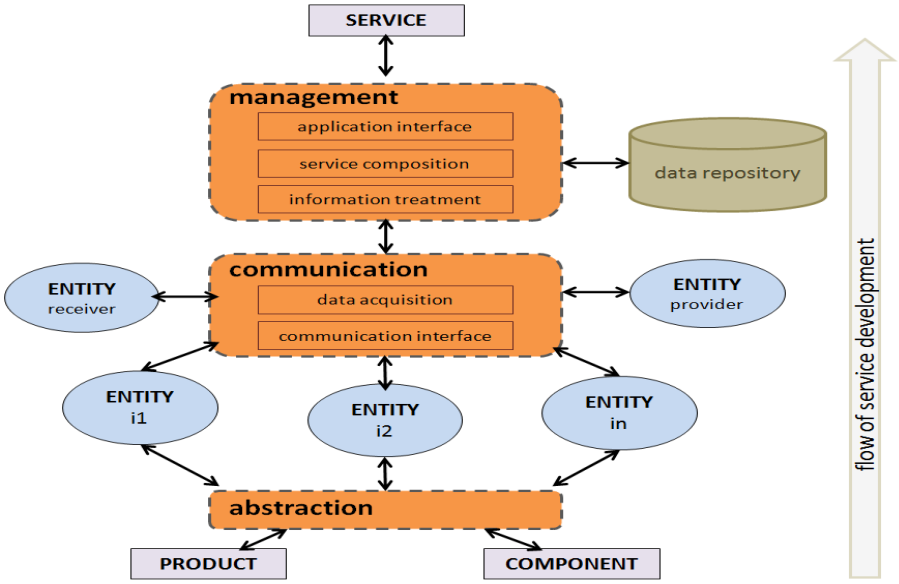

Figure 2. Middleware layers.

devices (such as internet protocols and technologies: wireless technologies - Bluetooth, etc.).

A possible conjecture about the architecture definition allows the construction of mixed environments, called hyper-environments, composed of real elements, avatars and things.

The hyper-environment must enable context-sensitiveness resulting from technological integration of acquisition, processing and use of large amounts of information, distributed in time and space between different players through the PSS lifecycle. Thus, in order to better characterize the different levels of organization that can arise in a hyper-environment, it is necessary to present some concepts that will be used throughout this section.

Hyper-environments elements. Real elements, avatars and things arise throughout the PSS lifecycle. These elements and their interrelationships compose the basic components of hyper-environments, as follows.

Media. Any medium, together with their properties, able to carry information, this way enabling communication.

Entity. Real elements, things or avatars associated with products/services that have technology able to assure their identifications to be perceived by others.

Tasks. Data processing activities performed by entities motivated by consumer's intention or for some specific purpose.

Organizing the Environment. Throughout the product lifecycle it can be established to exchange information from different sources and different spatial and temporal levels of abstraction.

Basically, the following levels of organization emerge:

Smart Environment. Is the basic unit. The entities aim to achieve common goals (intelligence). The tasks associated with the different entities are available on the hyper-environment. For example, one can cite a virtual-pervasive smart environment associated with the maintenance avatars, equipment repair - things, remote technicians, etc.)

Interface. The interface is responsible for the availability of the tasks inherent to a smart environment to extern communication. Smart peers also centralize the information of each smart environment.

Hyper-environment. Set of interfaces (or smart peers), merging different smart environments (real and/or pervasive and/or virtual) that intentionally and sporadically are related to execute specific goals.

An hyper-environment is responsible to combine informations in space-time throughout the PSS lifecycle. These information can be used to develop new services. Different actors involved with the services and products can exchange information during lifecycle.

To present the performance of the prototype of middleware that will be showed in the next section, some components that constitute the hyper-environment must be presented here. So, the main interfaces and classes implemented in Java are described below.

Entity class, with the following methods:

- **Start**: has the address parameter or a list of addresses of other entities in order to notify the entry of a new entity in the hyper-environment;
- **GetEntityList**: requests the list of all the entities present in the hyper-environment;
- **GetServiceList**: requests the list of tasks available in the hyper-environment;

 ISBN: 978-1-61208-236-3

- **CreateTask**: allows that the entity provides services to the hyper-environment, receiving as parameters: name, description and input signals.

Task class. The Task class has basically two methods, namely (i) StartTask which receives name, description and task list of input data, and (ii) TaskHandler that receives a function which transforms information received by task**.**

Smart Environment Class. This class allows the creation of an intelligent environment. When associated with a hyper-environment it is treated as a different entity represented by a smart peer.

In the adopted peer-to-peer network model the methods were implemented using the basic primitives of the JXTA [22] [23] [24] and the architecture and protocols of the PUCC [25] [26]. The entities and smart environments were implemented by Groups of JXTA peers-to-peers. Another special type called Rendezvous peer was responsible for communication of entities in a network with entities located in external networks. This type of peer is used to implement the definition of smart peer in the H-ENV architecture. The hyper environment is represented by a special group called the World Peer Group, which includes all JXTA peers. They are also used for the discovery mechanisms of JXTA and for the discovery treatment of new entities, tasks, and smart environments.

Due to the use of task concept (services) in the description of the H-ENV architecture, the prototype specification was performed based on Service-Oriented Architecture (SOA) [27]. However, an implementation that enables the use of Event-Driven Architecture (EDA) [28] was also build to provide means for an entity receive warnings through events.

V. VALIDATION

Conceptual issues of a model and an architecture of a middleware to converge computational aspects of PSS and IoT were addressed. To validate the searched ideas there was necessary verify if the middleware is feasible and scalable.

A prototype of the middleware was then build and a simulated case study, in the marine industry was ran. This choice was made because in this kind of industry the use of information and communication technologies are still scarce, and authors have access to a large shipyard that is being installed near the University premises. The major constructs of the shipyard will be eight platforms hulls for use in exploring recent oil reserves discovered in Brazil, the so called pre-salt.

The first phase realised was the design of a hyper environment capable of modeling the production process of the industry in focus, as well as their byproducts. Different environments present on all lifecycle of a ship were then modeled. Some predominantly virtuals (associated with design and monitoring tools), other ubiquitous (installed in the workshops or aboard of ships already in operation), as shown in Table I. In each one of these environments, called smart environments, the data acquisition was done by RFID readers installed in tools, equipments, supplies and workers, by sensors embedded in measuring devices, by camcorders, by cell phones, by PDAs, and by tablets ported by workers, etc. Also, there was computers to access network servers that provide various kinds of services, called smart peers. The servers have databases containing design libraries, equipment descriptions, staff of construction, operation and maintenance of the ships, navigation maps, maintenance charts, among other.

The described elements to acquire, process and store data constitute the hardware of the focused IoT. The software that runs over this hardware was a prototype of the middleware described in the previous sections. So, the prototype was a distributed system.

In order to examine the adequacy of the H-ENV in a scenario like this, tests of computational performance were conduced. The boot-time of the computational processes that represent peers of the H-ENV and the RTT (Round Trip Time) of the messages exchanged by them were mesured. The system's capacity to react on situations of joining and leaving elements in the environments were also evaluated. For example, the reaction time of the system to the inlet or outlet of a worker or a equipment in its premises was measured. The round trip time (RTT) was used to evaluate the capacity of the system to transport informations between their components.

To assess the boot-time, the runtimes of Start method were measured. To obtain the RTT, the runtimes of GetEntityList method were measured. Both methods belong to the Entity Class.

The simulations involved forty smart environments (twenty ubiquitous and twenty virtual). In ten of the ubiquitous environments, it was simulated the existence of a machine with a sensor and an actuator. In each one of ten smart ubiquitous environments a computer monitor was used to show all entities present in the hyper-environment. Finally, in each one of the twenty virtual smart environments it was installed a computer monitor which shows the avatars of ten employees chosen to be tracked. In one of these virtual smart environments a database capable of providing information about the entities present in hyper environment was simulated.

The simulation was performed in two laboratories, each one with 20 desktop computers connected through a 10Mbps wireless LAN (Local Area Network). The labs are connected by a university internet provider via a 1Gbps network. Each desktop computer runs Linux and have an Intel i3 processor, 4GB of main memory and a 500GB HD.

 ISBN: 978-1-61208-236-3

TABLE I. DESCRIPTION OF ENVIRONMENTS THROUGHOUT THE SHIP LIFECYCLE

Description	Predominant environment	Lifecycle stage	Peers
CAD/CAM design, supervision and scheduling project systems	Virtual	BoL	Tools and libraries for projects, equipment, personnel building
Yard plates, panels lines, block assembly lines, paint shops, edification area	Ubiquitous	Bol	RFID, video cameras, tablets and PDAs fitted sheets, trollers and workers
Comissioning, supervision and Control of the areas of deck, engine room, cargo holds	Ubiquitous	MoL	RFID readers, cameras, PDAs and tablet computers installed in equipment, devices, and workers
Dismiss and Conversions (oil tanker into an FPSO for example)	Ubiquitous	EoL	RFID readers, cameras, PDAs and tablet computers installed in equipment, devices, and workers

In each laboratory, there were three cell phones, a temperature sensor and an actuator connected through the serial port of a computer to control a air conditioner equipment.

The 20 ubiquitous environments were simulated in one laboratory and the 20 virtual environment in the other.

In order to evaluate the scalability of the middleware, it were simulated 1000 entities in the hyper-environment, twenty-five in each one of the smart environments. Table II shows the boot-time to some of these entities, according to their entering order in the system. The average of the boot-time was 52s with a standard deviation of 28s, performed through ten samples.

Regarding the ability to disseminate information, the middleware was evaluated by monitoring the method GetEntityList. The response times of a request sent to all entities present in the hyper-environment was measured. An average of 1050ms RTT in the implementation of the middleware using version 2.6 of the JXTA, running on the JVM version 1.6, using safe pipes of HTML was obtained.

TABLE II. BOOT-TIME OF THE PEERS.

Entity	Startup (seconds)
1	39
2	45
3	41
4	61
...	...
1000	75

Considering that the simulations were consistent with the operational conditions under which the possibility of use of the middleware was glimpsed and considering that the average startup time and RTT of 1,000 peers consumed solely by the JXTA protocols are around 10,000ms and 200ms [29] respectively, the results of about 60,000ms and 1,050ms obtained by the H-ENV implementation were adequate. It must be noted that they encompass the processing time consumed by other operations beyond communication.

With these results it can be concluded that the adopted ideas are feasible and a implementation of the system has viability under performance point of view in the application depicted.

By the way, at this point, we emphasized that the application envisioned in this work was thought to be used in a restricted environment, to be eventually implemented in the premises of the Brazilian petroleum company - PETROBRAS and under its full control. In short, the system prototyped was addressed to be used strictly in the offshore platforms that will be build to the PETROBRAS in the shipyard cited above. In this conditions, the apparent big problem domain with a relatively simple middleware solution is mitigated.

VI. CONCLUSION AND FUTURE WORK

The paper presented the use of IoT to provide a scalable architecture to PSS. The IoT use enables to obtain information throughout the lifecycle of the system, including data about the developers and users of the PSS, through networks of pervasive objects. The use of IoT as an ICT solution can assist the construction of PSS by using

 ISBN: 978-1-61208-236-3

information feedback loops in any lifecycle stage. It means that aspects of any phases: Begin of Life, Middle of Life and End of Life can be automatically considered to influence aspects of any stage, in cycles of any size. These possibilities are mediated by applications that use databases and models to describe the system dynamic and to offer the necessary subsidies for manual or automatic decision-making.

In this way, the contributions of this work intends to start discussions about a conceptual design for the IoT use in PSS systems. So, an approach to identify the main problems to be addressed in the context of IoT-PSS, in order to design a midlleware to be used in various areas of PSS applications was proposed.

Finally, stands out the emerging need for new methodologies, models and tools to improve PSS building for general use. However, the use of IoT is a possibility to obtain informations about all stakeholders of the system in real time, during all its lifecycle, through RFIDs, mobile phones, sensors networks and so on, thus representing a differential to building a PSS.

REFERENCES

[1] C. Van Halen, C. Vezzoli, and R. Wimmer (2005). Methodology for Product Service System Innovation. ISBN 90-232-4143-6.

[2] B. Cope and D. Kalantzis (2001). Print and ElectronicText Convergence. Common Ground. ISBN:1-86335-071-3.

[3] O. Mont (2004). Product-service systems: Panacea or Myth? Doctoral dissertation, The International Institute for Industrial Environmental Economics, University of Lund, Sweden.

[4] E. Heiskanen, M. Jalas, and A. Kärnä (2000), The Dematerialisation Potential of Services and IT: Futures Studies Methods Perspectives, Workshop on Futures Studies in Environmental Management, Turku, Finland.

[5] H. Chen, R. Kazman, and O. Perry, (2010), From Software Architecture Analysis to Service Engineering: An Empirical Study of Methodology Development for Enterprise SOA Implementation, In IEEE Transactions on Services Computing, vol. 3, no. 2, pp. 145-160

[6] C. Rolland, M. Pinheiro, and C. Souveyet, (2010), An Intentional Approach to Service Engineering, In IEEE Transactions on Services Computing, vol. 3, no. 4, pp.292-305.

[7] G. Pezzotta, S. Cavalieri, and P. Gaiardelli (2009), Product-Service Engineering: State of the Art and Future Directions, In Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing Moscow, Russia, June 3-5, 2009, pp. 1329-1334.

[8] O. Isaksson, T. Larsson and A. Rönnbäck (2009), Development of Product-Service Systems: Challenges and Opportunities for the Manufacturing Firm, In Journal of Engineering Design Special Issue on Product-Service Systems, 2009, pp.329-348.

[9] I. Barakonyi and D. Schmalstieg (2006), Ubiquitous Animated Agents for Augmented Reality, In IEEE/ACM International Symposium on Mixed and Augmented Reality, 2006. ISMAR 2006.

[10] J. York and P. Pendharkar (2004), Human–computer interaction issues for mobile computing in a variable work context, Int. J. Human-Computer Studies 60, pp. 771-797.

[11] U. Hansmann (2003), Pervasive Computing: The Mobile World. Springer. ISBN 3540002189.

[12] L. Atzori, A. Iera, and G. Morabito, (2009), The Internet of Things: A survey, Int. J. Computers Networks, pp.2787-2805.

[13] T. Tomiyama (2001), Service Engineering to Intensify Service Contents in Product Life Cycles, pp. 613, 2nd International Symposium on Environmentally Conscious Design and Inverse Manufacturing (EcoDesign'01), 2001.

[14] O. Mont (2002), Clarifying the concept of product–service system, Journal of Cleaner Production, 2002, pp. 237-245.

[15] D. Uckelmann, H. Mark, and F. Michahelles (2011), an Architectural Approach Towards the Future Internet of Things, D. Uckelmann (Eds.), 2011.

[16] L. Atzori, A. Iera, and G. Morabito (2010), The Internet of Things: A survey, Computer Networks, v. 54, n. 15, 28 Oct. 2010, pp. 2787-2805.

[17] M. Presser and A. Gluhak (2009), The Internet of Things: Connecting the Real World with the Digital World, EURESCOM mess@ge – The Magazine for Telecom Insiders, vol. 2, 2009, http://www.eurescom.eu/message, retrieved: aug, 2012.

[18] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V. Terziyan (2008), Smart semantic middleware for the internet of things, in: Proceedings of the Fifth International Conference on Informatics in Control, Automation and Robotics, Funchal, Madeira, Portugal, May 2008, pp. 169-178.

[19] E. Fleisch and F. Mattern (2005), Das Internet der Dinge: Ubiquitous Computing und RFID in der Praxis: Visionen, Technologien, Anwendungen, Handlungsanleitungen. Springer, Berlin, 2005.

[20] N. Gershenfeld, R. Krikorian, and D. Cohen (2004), The internet of things, Scientific American 291 (4), 2004, pp. 76–81.

[21] L. Srivastava (2006), Pervasive, ambient, ubiquitous: the magic of radio, in: European Commission Conference ‘‘From RFID to the Internet of Things”, Bruxelles, Belgium, March 2006.

[22] L. Gong (2001), Jxta: a network programming environment. Internet Computing, IEEE, 5(3), pp. 88-95.

[23] H. Park, E. Paik, and N. Kim (2009). Architecture of collaboration platform for ubiquitous home devices. Mobile Ubiquitous Computing, Systems, Services and Technologies, International Conference on, pp. 301-304.

[24] L. Barolli and F. Xhafa (2010). Jxta-overlay: A p2p platform for distributed, collaborative and ubiquitous computing. Industrial Electronics, IEEE Transactions on, pp. 2163-2172.

[25] H. Sumino, N. Ishikawa, S. Murakami, T. Kato, and J. Hjelm (2007). Pucc architecture, protocols and applications. In Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE, pp. 788-792.

[26] K. Fukushima, Y. Tanaka, H. Kato, and N. Ishikawa (2010). Home network system for gas appliances using pucc technologies. In Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE, pp. 1-5.

[27] D. Krafzig, K. Banke, and D. Slama (2004). Enterprise SOA: Service-Oriented Architecture Best Practices (The Coad Series). Prentice Hall PTR, Upper Saddle River, NJ, USA.

[28] H. Taylor, A. Yochem, L. Phillips, and F. Martinez (2009). Event-Driven Architecture: How SOA Enables the Real-Time Enterprise. Addison-Wesley Professional, 1st edition.

[29] E. Halepovic and R. Deters, JXTA Performance Study, in Proc. PACRIM 2003, pp. 149-154.

 ISBN: 978-1-61208-236-3