IoT-aware business processes for logistics: limitations of current approaches

Pedro Ferreira[1], Ricardo Martinho[1], Dulce Domingos[2]

[1] School of Technology and Management, Polytechnic Institute of Leiria, Portugal
[2] Faculty of Science, University of Lisboa, Portugal
{pedro.ferreira, ricardo.martinho}@ipleiria.pt ; dulce@di.fc.ul.pt

Abstract. The Internet of Things (IoT) aims at bridging the gap between real-world business processes and information systems. Supply chain management is one of the major application areas that can benefit from the IoT. When attached to physical items, the IoT technologies such as RFID and sensor networks transform objects of the supply chain into *smart items*. These items have the ability to capture context data and provide information systems with a representation of 'things'. This allows information systems to monitor the supply chain processes through process aware information systems. Smart items can also execute parts of the business processes. In distributed environments, they can exchange data among them and make decisions based on business logic. However, this logic only acts according to pre-planned behaviour. Unpredicted exceptions based on real life events require dynamic process adaption in process definitions and corresponding instances. In this paper we review the main technologies of the IoT associated with automated support of business processes in logistics. We also identify the main limitations in the Business Process Execution Language (BPEL), regarding the support of design and runtime changes in these processes with smart items.

Keywords: The Internet of Things, smart items, logistics, business process, flexibility, BPEL.

1 Introduction

In the last decade, the term Internet of Things (IoT) has been raising interest on the enterprise world, mostly due to a growing web-based service economy [1]. The IoT provides a key role for the future Internet by bridging the gap between the physical world and its representation in information systems.

From an enterprise point of view, manufacturing, supply chain integrity, energy, health and automotive are some of the major application areas of the IoT. Despite the benefits, major technical issues such as internet scalability, identification and addressing, heterogeneity and service paradigms are prominent areas of research in recent years [2].

The supply chain is a network of organisations and business processes for procuring raw materials, transforming them into products and distributing these to customers. There are five major supply chain processes: plan, source, make, deliver and return [3]. Logistics plays an important role in these processes, dealing with the control and planning of all the factors that will have an impact on transporting [3].

Technologies, such as RFID and sensors, provide context data to support decision making at high management level. The introduction of sensors with the ability to execute business logic at the item level, *i.e., smart items* [4], allows local decision-making and therefore reduces centralised processing and the amount of exchanged data. The decomposition of business processes through distributed environments creates a paradigm shift and challenges for business process modelling.

The Business Process Execution Language for Web-Services (WS-BPEL) has emerged as the standard reference to model the behaviour of Executable and Abstract business processes on Web Services [5]. It defines an interoperable integration model, extending the Web Services interaction model and enabling it to support business transactions.

So far it is possible to use information provided by the IoT to support static business processes, *i.e.*, processes defined at design time that do not foresee deviations. However, the use of smart items often requires dynamic business processes that are able to accommodate adaptations, according to changes verified in the execution context or behaviour of smart items.

In this paper, we present the limitations of BPEL to define business processes that support this dynamic behaviour. We also address how this ability can dictate the way business process logic is distributed between smart items and standard process support. We focus on logistics and supply chain-related business processes, which make use of smart items.

The remainder of this paper is organized as follows: section 2 describes the influence of smart items at supply chain; section 3 analyses the impact of business logic at smart items; section 4 discusses the limitations of modelling business processes with BPEL and section 5 concludes this paper.

2 Smart items in logistics

The main purpose of the IoT is to fill in the gap that usually exists between real-world business processes and their representation within information systems. Therefore, technologies such as RFID and wireless sensor networks capture accurate context data. These data can then be used in real-time representations of business processes and involved objects within information systems. For these purposes, the technologies and related devices are commonly referred to as *smart items*.

2.1 Smart items types

Three main technologies are commonly used by smart items in business processes related with logistics. They are *barcodes*, RFID and *sensor networks*. Barcodes are a standard technology for electronic identification of products. A barcode is attached on the product and optically detected by a barcode reader. The reader acknowledges the printed identification and provides acquired data to the information system, which updates the product's state. This solution provides limited information due to line-of-sight requirement. For instance, it is impossible to detect a single item within a closed container of products. The acquisition of product's data during transportation requires a more complex infrastructure. Therefore barcodes are only useful in load and unload processes within the logistics of the supply chain.

Radio Frequency Identification (RFID) devices [6] can be identified through radio-based frequency handling technologies. Unlike barcodes, they do not require line-of-sight to be identified. Location tracking is available including products in transportation, depending on RFID readers deployed. They also have the ability to acquire products sensor data (such as temperature) and provide it to the information system. These sensing capabilities are usually very limited [7]. Accordingly to their behaviour, barcodes and RFID can be referred to as *passive* devices [8].

Wireless sensor networks are the most promising approach for logistics processes. The sensor nodes are electronic devices with embedded sensing and computing systems that collaborate within a network. In addition, they are extremely small and can be specifically designed to meet the requirements of the transported products. Unlike RFID, sensor networks can execute parts of the processes from an information system directly on the items. Products become embedded logistics information systems [7]. Sensor networks cover identification, tracing, location tracking, monitoring and real-time responsiveness. For instance, CoBIs [9] presents a sensor network that covers all these aspects.

2.2 Logistic functions, information systems and smart items

Logistic functions can be associated with a set of features commonly supported by IS and smart items' technologies, as illustrated in Table 1. The basic logistics functions are to transport "*the right goods and the right quantity and right quality at the right time to the right place for the right price*" [7]. To address each of these functions, information systems must have specific features, such as *identification*, *tracing*, *location tracking*, *monitoring*, *real-time responsiveness* and *optimisation*. Product identification informs the system about the *right goods*. Tracing allows the system to detect when items are lost. Therefore, it guarantees the *right quantity*. The *right place* is monitored by the information system through location tracking. It keeps track of the transport itself. Monitoring the product's state ensures the product's *right quality*. With all these data within the information system, the overall logistics process can be observed with detail. Therefore, responsiveness to unforeseen events and other actions can be achieved at the *right time*. In addition, these data provide the basis for optimisation affecting the product's *right price*. Smart items play an important role in supporting all of these features. Moreover, the kind of support provided to these features can be directly correlated with the types of smart items referred in previous section.

Table 1. Logistic functions and information systems features to implement them. It also displays the smart items capabilities towards logistic functions.

Functions	Features	Barcode	RFID	Sensor Networks
right goods	Identification	Full	Full	Full
right amount	Tracing	Partial	Full	Full
right place	Location Tracking		Full	Full
right quality	Monitoring		Partial	Full
right time	Real-time responsiveness			Full
right price	Optimisation			Full

2.3 The role of smart items in supply chain integrity

The information generated by smart items allows the monitoring and control of products, along the entire logistics process. For instance, tracking their location can be used to detect if the associated products have been detoured from a pre-planned route. Tracking their state can be used to realise if the products' condition has changed and whether they are still useful or not. This denotes three types of integrity that can be compromised: (1) product integrity, (2) components integrity, (3) route integrity.

For instance, a product's physical integrity can be monitored through sensors that keep status during its life-cycle within the logistics process. As an example, we can consider that a product starts the process tagged with the status *closed*. If this status changes during the process (to *opened* or *tampered*), the product's integrity might have been compromised. For perishable products, sensors with the ability to measure temperature can be used to monitor the product's condition. For example, a truck loaded with fruit starts the process tagged with the status *good*. If the temperature rises above a pre-planned threshold value, the fruit's quality may be affected [8].

Regarding transportation routes integrity, these can be monitored through technologies that provide products location. Transporting these products requires pre-planned routes. However, there may be detours from the original route due to environment changes. For example, when dealing with hazardous products, some restrictions can apply to available routes and they can even be unauthorized. Unpredicted events such as traffic, weather conditions or road blocks might compromise the products route integrity during transportation.

Integrity of components requires the most complex monitoring. It consists on controlling every component of the product during its production and transportation. It ensures that the product keeps its intended use and does not break established rules regarding legal issues or environmental compliance along the logistics process.

In addition, breaking one of these integrity types can result in also affecting the other remaining ones. For instance, the fruit truck may have to detour due to a product integrity breach. Conversely, the product's integrity can be affected due to a forced detour.

3 Business process logic within smart items

Smart items provide new opportunities and challenges in the system design and electronics integration. Based on their potential and collaboration with external services, they are able to do more than providing real-time data. They also process data and make decisions based on it, including exchanging data among smart items that do not depend on a centralised model. In this section, we refer to the software architectures commonly used to address these issues. We also present relevant factors that affect the amount of business logic that is distributed between a central system and cooperative smart items.

3.1 Architectural evolution

The Internet of Things is a concept constantly evolving mostly due to its young existence. It first appeared with the use of RFID and evolved through related technologies such as sensor networks and smart embedded devices. The introduction of smart items at the supply chain logistics processes requires constant optimisation and innovation, in order to enhance enterprises' competitiveness and quality of service.

Architectures used to accommodate interactions between smart items and information systems have also been evolving at a similar pace.

For instance, client-server architectures still play a major role regarding these interactions. Nevertheless, Service Oriented Architectures (SOA) is becoming the preferred approach regarding interactions of information systems with more powerful smart items. Moreover, this is indicated as the dominant architectural approach for these kinds of devices in the future [12].

The integration of smart items into business processes through SOA allows information systems to interact with physical objects and create *the Internet of Services* (IoS). This integration is possible by running instances of web services on these devices. Such architectural change provides an outnumbered set of opportunities and challenges in achieving efficient collaboration between the services and centralised information systems. In order to overcome these challenge, middleware approaches have been a reliable solution to integrate back end applications and services offered by the devices, service-mediators and gateways [12].

The SIRENA (Service Infrastructure for Real-time Embedded Networked Applications) [10] project was developed to leverage SOA architectures to seamlessly connect embedded devices within different domains. This project presents proof-of-concepts that illustrate the feasibility and benefits of embedding web services at devices. However, these pioneer efforts lacked attention to issues such as device supervision, device life cycle management or maintaining the status of discovered devices. SIRENA was also used as a foundation for SODA [11] and SOCRADES [12] projects. The purpose of SODA was to create a comprehensive, scalable, easy to deploy, service-oriented ecosystem built on top of foundations laid by SIRENA. This project led to significant reduction of time to market for innovative services. The purpose of SOCRADES was to develop a design, execution and management platform, exploiting the SOA paradigm at device and application level. The

SOCRADES middleware is an architecture that provides web service enabled devices for business integration with information systems such as ERPs for the manufacturing area.

3.2 Delegating business logic to smart items

As described in section 2, there are different types of smart items according to their behaviour and characteristics. Therefore, some authors fit smart items in two different groups: *passive* and *active*. Passive technologies such as RFID and barcodes can identify products at transhipment points. Semi-passive RFID data loggers allow temperature recording at affordable costs. Active technologies such as wireless sensor networks can communicate among all participants in a supply chain's logistic process (freight, containers, warehouses and vehicles).

Böse and Windt presented a catalogue of thirteen criteria to characterise logistic systems regarding autonomy [13]. The location of the decision making is considered the most important criterion concerning autonomous control. Despite of having an essential role on monitoring the process, smart items have been mostly used as information providers instead of participants in decision making or business process planning. The idea of delegating some business logic to smart items shifts the decision making from centralised, server-based solutions to a network of distributed processing devices. This creates an autonomous cooperation within the logistics business processes. Each smart item has its own piece of software, which can autonomously search for a partial solution when dealing with process-related issues. In transportation scenarios, this software collects information, makes decisions and negotiates with other entities to fulfil their goal. For instance, a truck loaded with several pallets of fruit can have each one equipped with a smart item. These can monitor a physical dimension such as temperature, which in turn will dictate the truck route in order to deliver all products at the minimal costs [8].

In controlled transportation scenarios, *i.e.*, not subjected to unforeseen circumstances, everything is determined before the process begins. Therefore, there is no need for delegating new behaviour to the smart item level. However, changes in traffic, new incoming orders, lack of communications with the central system or any other kind of unforeseen events might require a detour to a pre-planned route. To support these unexpected scenarios, it is necessary to use smart items with embedded intelligence enough to provide for dynamic planning.

This approach may require a shift of the business logic and associated control from the central system level to the smart item level. From this point of view, decisions are made in real-time by smart items on the field using their interaction abilities and intelligence, without human direct intervention. Therefore, in order to keep the system running, the implementation of software to be embedded into the smart item must provide robustness, flexibility, privacy, low communication costs and low computation time.

Supply chain management systems equipped with these smart items must have flexibility to react immediately to sudden changes. For instance, if a road block happens in a transportation scenario, the best alternative route must be searched immediately. This route must be in accordance with the logistic functions and keep supply chain integrity (referred above in section 2). The need of fast responsiveness requires low computation time.

However, a thorough search for optimal route in a complex scenario could take too much time depending on the smart items processing capabilities. If a communication failure occurs in the network, system should be robust enough to continue its work. Internal planning strategies and other sensitive data must be kept confidential. For instance, if a route change is necessary, the delivery time for several customers will most likely change. Despite of being aware of this change, each customer must not have access to other customer's changes [2].

In a central based approach, objects in the logistics process are simply information providers. Therefore they only execute atomic activities defined in a business process running on a central system [12]. Smart items with embedded intelligence handle incoming data, observe and evaluate surrounding conditions and make decisions based on acquired information. However, these depend on the objects decision freedom within the process and, consequently, their ability for process dynamic changes [8].

4 Dynamic changes to business processes with BPEL

Changing business processes dynamically involves altering the process's control flow, data or resource perspectives at runtime. Examples include adding, skipping, updating or deleting an activity, changing the data objects associated with an activity, or even altering its role-assignment. However, these changes must assure the correctness (syntax) of process definitions and process instances, and consistency among concurrently executed process instances [14]. Therefore, *flexibility* has been an issue concerning the business process management and workflow research areas.

4.1 Process flexibility types

After several case studies and years of research, consensus was obtained concerning the flexibility required to deal with exceptions. Eder and Liebhart [17] grouped exceptions into two groups: predicted and unpredicted. Predicted exceptions represent the unusual but foreseen behaviour of a process. These exceptions can be modelled in the process definition as alternative paths to normal behaviour. The unpredicted exceptions represent the unforeseen behaviour of a real world business process regarding the process definition. To address these unpredicted exceptions, systems need to update the process definition and the corresponding process instances.

In a sequence of also recent contributions, Schonenberg et al. present a taxonomy of process flexibility [15]. Four distinct types to process flexibility are identified, each having its own application area. We enumerate them below, referring a simple transportation process scenario for each one of them:

- *Design*: for handling anticipated changes in the operating environment, where supporting strategies can be defined at design-time;

- *Deviation*: for handling occasional unforeseen behaviour, where differences with the expected behaviour are minimal;
- *Underspecification*: for handling anticipated changes in the operating environment, where strategies cannot be defined at design-time, because the final strategy is not known in advance or is not generally applicable;
- *Change*: either for handling occasional unforeseen behaviour, where differences require process adaptations, or for handling permanent unforeseen behaviour.

Each of the flexibility types operates in different ways. Figure 1 provides an illustration of the distinction between each of the flexibility types in isolation, in terms of the time that specific flexibility options need to be configured - at design time, as part of the process definition or at runtime via facilities in the process execution environment. It also shows the anticipated completeness of the process definition for each flexibility type.

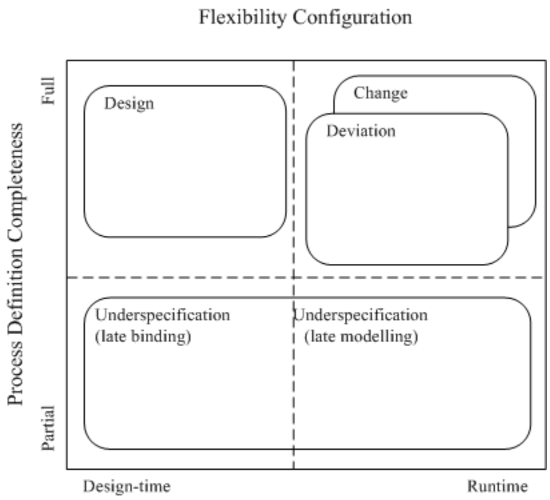

Fig. 1. Taxonomy of process flexibility according to Schonenberg et al. (adapted from [15]).

4.2 BPEL limitations to process flexibility

So far we have described the types of smart items and how they can benefit business processes in logistics. We have also observed the delegation of business logic to smart items due to architectural evolution. This evolution also allows the decomposition of business processes through distributed networks instead of central based solutions. However, none of these approaches supports flexibility in process that also includes smart items. This means that these business processes do not foresee either predicted or unpredicted changes that may force updates in the business logic running on both central system and smart items.

As referred above, WS-BPEL has emerged as a standard reference language for modelling and executing business processes. A WS-BPEL process definition includes partner links that define the relationships with other business partners, declarations of process data, handlers for various purposes and the activities. Basic activities only perform their intended purpose, such as receiving a message from a partner, or manipulating data. Structured activities can contain other activities and define the control flow business logic (see [5] for further details).

We foresee its application also in business processes for logistics that use smart items. However, and as we already referred, this kind of processes can be subjected to many predicted and unpredicted changes. For this matter, WS-BPEL has some limitations, and we will classify them according with the flexibility types illustrated in Figure 1.

Regarding flexibility in *design time*, we can identify the following limitations when dealing with WS-BPEL:

- The definition of alternative flows is possible but limited concerning the number of paths – WS-BPEL allows the handling of predicted exceptions with exception handlers, as well as alternative flows through the use of if/else control structures (flexibility by *design*). However, WS-BPEL fails in allowing for a compact process definition for a larger number of exceptions and alternative paths, which cannot be foreseen or practically defined. In the IoT context, business processes definitions that rely on smart items' collected data may imply a large number of exception handlers or alternative paths;
- All process perspectives (control flow, data types and handlers) must be defined in a static way and *a priori* – WS-BPEL does not allow for flexibility by *underspecification*, meaning that process definitions cannot be partially defined or incomplete, or even dynamically specified (e.g., it is not possible to provide a partner link's name later on when the process as already began to execute);
- The definition of business logic that is to be run in smart items is not possible with WS-BPEL. It would be valuable to access and specify all sub-process definitions that compose a business process model together. This may include definitions of the business process logic to be executed either centrally or on the smart items. WS-BPEL provides extension mechanisms that can be used to define additional language constructs, in order to model the business logic to be loaded into the smart items;
- WS-BPEL does not foresee the distribution of business logic between a central system and smart items, according to smart items properties. We must keep in mind that smart items are electronic devices. Therefore, they have physical properties such as power (batteries) and computation speed that limit their autonomy. When delegating business logic into smart items, processing is required. As more of the business process is delegated, the more processing will be necessary. Therefore, power consumption will slightly increase. On the other hand, the less of the business process is delegated to smart items, the more communication will be required, highly increasing power consumption. In addition, smart items can have different

capabilities. Therefore they can support distinct amounts and types of business process logic;

- WS-BPEL does not allow controlling which, how and by whom parts of a process definition can be changed. This *controlled flexibility* [20] can be useful for our context, specifically for clarifying which parts of a process can be changed, when the process is distributed between the central system and the smart items;

As for flexibility in runtime, the major limitations that we can identify in WS-BPEL are:

- The lack of support for changes of business process instances, due to unpredicted, ad-hoc circumstances. Logistics with smart items are subjected to a plethora of these circumstances, which generate events that must be immediately reflected in changes in the governing business processes.
- The lack of support in migrating instances from old process definitions to new ones, when a redefinition of the business process occurs. Some works have already addressed these challenges in WS-BPEL, including correctness and compliance issues (e.g., see [14], [18]), but out of the context of the IoT. However, it is possible to redefine smart items behaviour in runtime, for example using the Callas language presented in [19].

Moreover, combining these types of flexibility adds specific challenges, also not addressed by WS-BPEL. These include the use of runtime ad-hoc changes and design changes together. Reichert *et al.* allow changes to be propagated to the process instances, which were already subjected to ad-hoc changes [16]. Also, the use of design changes together with underspecification flexibility in runtime (late binding) raise additional challenges regarding correctness and compliance between process definitions and underspecified process running instances.

5 Conclusions

The IoT is a concept raising interest in logistics business processes, mostly due to use of technology commonly referred to as smart items. These items provide accurate context data to information systems, which they use in real-time representations of business processes. Smart items like wireless sensor networks with embedded computing systems can do more than just providing data. They can execute parts of business processes and cover the basic logistics functions.

Central based solutions still play an important role in logistics processes; however distributed solutions are becoming the preferred approach. The introduction of sensors with the ability to execute business logic at the item level allows local decision-making and therefore reduces centralised processing and the amount of exchanged data. However, none of these approaches supports predicted or unpredicted changes that can occur in real world business processes. These changes require business processes to be redefined or process instances to be changed handled dynamically, including changes in the process control flow, data and resources at runtime, such as reprogramming the smart items.

Summing up, we stressed WS-BPEL's limitations on process flexibility and classified them according to the taxonomy types defined in Section 4.1. These limitations shed some light on future topics that we intent to explore on WS-BPEL, namely allowing for the definition and distribution of business process logic between central systems and smart items. Also, we are already addressing some of these challenges through an extension language for WS-BPEL regarding the definition of smart items business logic. For this we are taking advantage on the native extension mechanisms on WS-BPEL, specifically through the `<extensionActivity>` element.

Acknowledgments

This work was supported by FCT through project PATI (PTDC/EIA-EIA/103751/2008) and through LASIGE Multiannual Funding Programme.

References

1. ISTAG Working Group Report on "Web-based Service Industry", February 2008, ftp://ftp.cordis.europa.eu/pub/ist/docs/web-based-service-industry-istag_en.pdf
2. Haller, S., Karnouskos, S., Schroth, C.: The Internet of Things in an Enterprise Context. In Future Internet — FIS 2008: First Future Internet Symposium, FIS 2008 Vienna, Austria, September 29-30, 2008 Revised Selected Papers, pages 14–28, Berlin, Heidelberg, 2009. Springer-Verlag.
3. Laudon, K.C., Laudon, J. P.: Management Information Systems. New Jersey: Pearson Prentice Hall, 385--389 (2006).
4. Uckelmann D.: A Definition Approach to Smart Logistics. S. Balandin et al. (Eds.): Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN 2008), number 5174 in LNCS, pages 273-284, Springer-Verlag, 2008
5. OASIS. Web Services Business Process Execution Language (WS-BPEL), Version 2.0. Technical report, Organization for the Advancement of Structured Information Strandards, 2007.
6. Finkenzeller, K.: RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification (2003)
7. Decker, C., Berchtold, M., Chaves, L., Beigl, M., Roehr, D., Reidel, T., Beuster, M., Herzog, T., Herzig, D.: Cost-Benefit Model for Smart Items in the Supply Chain. In C. Floerkemeier and et Al., editors, Proceedings of The Internet of Things. First International Conference (IOT 2008), number 4952 in LNCS, pages 155-172. Springer-Verlag, 2008.
8. Jedermann, R., Lang, W.: The benefits of embedded intelligence - tasks and applications for ubiquitous computing in logistics. In C. Floerkemeier and et Al., editors, Proceedings of The Internet of Things. First International Conference (IOT 2008), number 4952 in LNCS, pages 105–122. Springer-Verlag, 2008.
9. Decker, C.,Reidel, T., Beigl, M., sa de Souza, L.M., Spiess, P., Mueller, J., Haller, S.: Collaborative Business Items. In 3rd International Conference on Intelligent Environments (2007)
10. Bohn, H., Bobek, A., Golatowski, F.: SIRENA - Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains, icniconsmcl, pp.43, International Conference on Networking, International Conference on

Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06), 2006
11. Deugd, S., Carroll, R., Kelly, K., Millett, B., Ricker, J.: SODA: Service Oriented Device Architecture, IEEE Pervasive Computing, vol. 5, no. 3, pp. 94-96, c3, July-Sept. 2006
12. Souza, L., Spiess, P., Guinard, D., Köhler, M. Karnouskos, S., Savio, D.: SOCRADES: A Web Service based Shop Floor Integration Infrastructure. In C. Floerkemeier and et Al., editors, Proceedings of The Internet of Things. First International Conference (IOT 2008), number 4952 in LNCS, pages 50–67. Springer-Verlag, 2008.
13. Böse, F., Windt, K.: Catalogue of Criteria for Autonomous Control. In: Hülsmann, M., Windt, K. (eds.) Understanding Autonomous Cooperation and Control in Logistics – The Impact on Management, Information and Communication and Material Flow, pp. 57–72. Springer, Berlin (2007)
14. Reichert, M., Rinderle, S.: On design principles for realizing adaptive service flows with bpel. In Proceedings of International Conference on Conceptual Modeling (EMISA 2006), pages 133–146. Lectures Notes in Informatics, 2006
15. Schonenberg, H.; Mans, R.; Russell, N.; Mulyar, N. & van der Aalst, W. M. P. Towards a Taxonomy of Process Flexibility Proceedings of the Forum held at the 20th Conference on Advanced Information Systems Engineering (CAiSE'08), 2008
16. Reichert, M., Hensinger, C., & Dadam, P. (1998). Supporting Adaptive Workflows in Advanced Application Environments. In Proceedings of the EDBT Workshop on Workflow Management Systems. Valencia, Spain.
17. Eder, J., Liebhart, W.: The workflow activity model WAMO. In: Proceedings of the 3rd International Conference on Cooperative Information Systems (CoopIS'95), Vienna, Austria, May 1995, pp. 87–98
18. Fang, R., Zou, Z. L., Stratan, C., Fong, L., Marston, D., Lam, L., Frank, D.: Dynamic Support for BPEL Process Instance Adaptation. In Proceeding of the 2088 IEE International Conference on Services Computing, 2008, pp. 327–334
19. Martins, F., Lopes, L., Barros, J.: Towards Safe Programming of Wireless Sensor Networks. In Proceedings of Programming Language Approaches to Concurrency and Communication-cEntric Software (PLACES), 2010
20. Martinho, R.; Varajão, J. & Domingos, D.: Modelling and Learning Controlled Flexibility in Software Processes. *International Journal on Knowledge and Learning*, 2009, 5, 423-442