Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 100 (2016) 1280 – 1288

Procedia Computer Science

Conference on ENTERprise Information Systems / International Conference on Project MANagement / Conference on Health and Social Care Information Systems and Technologies, CENTERIS / ProjMAN / HCist 2016, October 5-7, 2016

Using resource reliability in BPMN processes

Dulce Domingos[a,c*], Ana Respício[a,b], Ricardo Martinho[d,e]

[a]*Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, Portugal*
[b]*Centro de Matemática, Aplicações Fundamentais e Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa, Portugal*
[c]*LaSIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal*
[d]*School of Technology and Management, Polytecnic Institute of Leiria, Portugal*
[e]*CINTESIS - Center for Health Technology and Services Research, University of Porto, Portugal*

Abstract

Business Process reliability refers to the probability of a certain process task to be executed during a predefined timeframe. It is one of the Quality of Service aspects of business processes that can be calculated and simulated, in order to support decision making regarding Business Process Management activities such as redesign, execution and improvement of business processes. Nevertheless, existing research focuses only on task reliability, without considering how a certain human or non-human resource assigned to that task can affect its overall reliability. In this paper, we propose the use of the relyBPMN extension to include resource reliability information on business processes modeled with the Business Process Model and Notation (BPMN) language. We illustrate this by providing two use case examples involving sensors and human resources, and including their reliability information to be used on resource assignments. This way, process engineers can conditionally assign resources to a task based on their reliability information and requirements.

Peer-review under responsibility of the organizing committee of CENTERIS 2016

Keywords: business processes, reliability, resources, BPMN

* Corresponding author. Tel.: +351 217 500 524; fax: +351 217 500 084
E-mail address: mddomingos@fc.ul.pt

1877-0509
Peer-review under responsibility of the organizing committee of CENTERIS 2016
doi:10.1016/j.procs.2016.09.243

1. Introduction

Reliability has been a growing research topic in Business Process Management (BPM), as part of a larger group of Quality of Service (QoS) aspects also including performance, cost, and availability [2, 5, 8, 15, 21]. In this context, a business process *task reliability* can be defined as the probability that it operates on users' demand, following a discrete-time model. Cardoso [3] approached reliability and other QoS aspects in the workflow perspective of business processes by using the Stochastic Workflow Reduction (SWR) algorithm. This algorithm relies on reduction rules to iteratively calculate reliability and reduce a business process to a single atomic task. At the end, the reliability of the overall business process is equivalent to that of the single atomic task.

Based on this approach, we have recently proposed the *relyBPMN* language extension [18], that allows for process engineers to include reliability information in Business Process Model and Notation (BPMN) [17] process models. We have also applied this approach and language extension to Internet of Things (IoT)-aware business processes. This enables design and run-time decisions regarding processes that rely, for instance, on sensors, gateways, Information Systems and all the messaging between them [12].

However, these approaches only take into account business process *task* reliability, disregarding the resources perspective and how a certain resource (human kind or others such as sensors) can affect that task's reliability. The resources perspective in this context involves the definition of resource assignments to business process tasks at design time (stating those actors that are supposed to execute on those tasks), and further allocation of resources at runtime (users or roles that can execute the tasks). For instance, considering a "Check Exams" task on a healthcare business process, we can assign it to a certain actor in design time (such as "Physician"), and then allocate it to an available physician in runtime. Nevertheless, this *task*'s reliability can be heavily affected by the skills and other factors of the physician assigned, including its specialty, number of consecutive hours in labor, age and experience [20]. Therefore, we need to consider the reliability of not only control-flow elements of a business process (tasks, sub-processes, gateways) but also of resource-related elements that usually refer to resource information and assignment conditions. Assuming BPMN has become the *de-facto* business process modelling language standard [8], it is also noticeable that its emphasis is on control-flow and data elements, while resource definition and assignment is still rather basic and abstractly specified.

In this paper, we propose the use of BPMN as a means to include reliability resource information on business processes. We also propose to use this information to conditionally assign resources that meet certain business process reliability requirements. We prove this by illustrating two use cases, including sensors and human resources' reliability. We also make use of the *relyBPMN* extension proposed in [18], to define reliability information and requirements on resources. This way, we can calculate the overall reliability of a business process taking into account both control-flow and resource perspectives.

This paper is organized as follows: in the next section we review related work. Section 3 presents our two use cases involving sensors and human resources. Section 4 illustrates how we can use the relyBPMN extension and resource definition and assignment in BPMN to include resource reliability information and conditions in a business process. Finally, Section 5 concludes the paper and discusses future work.

2. Related work

According to Cardoso [3], reliability of a workflow task is the probability that the components operate on users' demand, following a discrete-time model. The failure rate of a task is the ratio number of unsuccessful executions to scheduled executions. This way, the reliability of a task A, denoted by $R(A)$, is $R(A) = 1 - failureRate(A)$, the opposite of the failure rate.

We can find in the literature some proposals whose aim is to improve the BPMN expressiveness to provide business process with quality of service (QoS) information, such as performance, cost, availability, and, in particular, reliability.

In the context of Internet of Things (IoT)-aware business processes, Meyer et al. identify the need for modeling the certainty of the information that devices provide (from 0 to 100%) and the availability/potential fault of devices [16]. They exemplify how BPMN processes can include this information by using annotations. Following a similar approach, Chiu and Wang suggest that IoT-specific domain tasks can include information about availability and fault tolerance rates [5]. Still considering IoT-aware business processes requirements, Caracas and Bernauer [2] consider that due to energy efficiency constraints, BPMN processes should include information about the protocols used for communications between participants. They use envelope component properties of communication flows to define the transmission mode, such as broadcast or unicast, and the communication protocol, such as IEEE 802.15.4 or TCP/IP. Sungur et al. also propose to use annotations (i.e. performance annotations) to define performance goals and to deal with potential performance conflicts, such as shorter response time, reliable communication, and lower power consumption [21].

Instead of using annotations, in [15], Meyer et al. propose a BPMN extension that provides the BPMN `Task` element with two additional elements: the `IQMCalculated` and the `IQMManual`. The `IQMCalculated` element stores the IoT Information Quality Metric (IQM), calculated from the Unified Service Description Language (USDL) service description, while the `IQMManual` element stores the minimum required value. They calculate these metrics by using time, space, reliability and traceability quality of information parameters available in the USDL service description. Considering that quality of information cannot be assessed alone due to its interdependency with access cost, Martinho and Domingos [11] include two additional elements to the BPMN `Task` element, the `CostMCalculated` and the `CostMManual` elements, to store the cost metric calculated from the USDL service description and the maximum required value, respectively.

Going a step further, in [18], the authors propose a BPMN extension to provide business process models with reliability information and use it to calculate the overall process reliability. Process reliability is calculated by using the Stochastic Workflow Reduction (SWR) algorithm that Cardoso proposed in [3], which applies a set of reduction rules to process blocks, iteratively, until only one task remains.

However, these works only consider task reliability, leaving out the way reliability of resources is reflected in the task reliability. The work we present in this paper provides business process models with resource reliability information and use it to handle task as well as process reliability requirements.

3. Use cases

In this section we present two types of use cases. The first one focuses on the use of sensors as resources, while the second one on human resources.

3.1. Use case with sensors as resources

With the advent of the IoT, business processes use sensors more and more. Sensors capture environment context information, which can be used to optimize and adapt business processes to run-time events and context changes. In addition, due to the increasing computational capabilities, sensors are also used to execute parts of the business logic. For instance, in the makeSense project [4, 21, 22], the authors use sensors and actuators to manage a building ventilation system. Presence sensors detect whether the room is occupied, and when the value of carbon dioxide the sensors read is above a pre-defined threshold, the actuators increase the ventilation. In the IoT-A project, the authors use sensors and actuators to monitor, in a store, perishable goods such as orchids, whose quality decreases when the temperature rises [16, 14]. They use sensors to measure the temperature and actuators to update the price specified on electronic shelf labels. In smart cities, sensors are used for instance to control traffic and to optimize garbage collection and irrigation of green spaces [10]. Without being exhaustive, we mention the healthcare domain and, in particular, ambient-assisted living (AAL) systems, where sensors provide information for continuous health monitoring and actuators can perform automatic care procedures (for instance, using pumps to auto inject insulin when blood sugar values increase over a certain value) [12].

BPMN processes can easily integrate sensor information and functionalities by using `task services` through web services [23]. However, when modelers want not only to use sensor information but also to define sensors' behavior as part of the BPMN business process, they use additional BPMN `pools`, as illustrated in the example of Fig. 1. This figure presents a simplified BPMN business process model of a ventilation system [4, 21, 22]. The `sensors` pool defines the behavior of the sensor node (also named mote†), which includes two sensors (the presence sensor and the carbon dioxide sensor). The `actuators` pool manages the ventilation system, according to the information sensors provide. The reliability of the process depends on the reliability of the sensors and actuators, which are the resources the system uses to get environment information and to control the ventilation system. Even if, within this use case, we cannot change resources at run time, this information is indispensable to design the system architecture according to its reliability requirements.

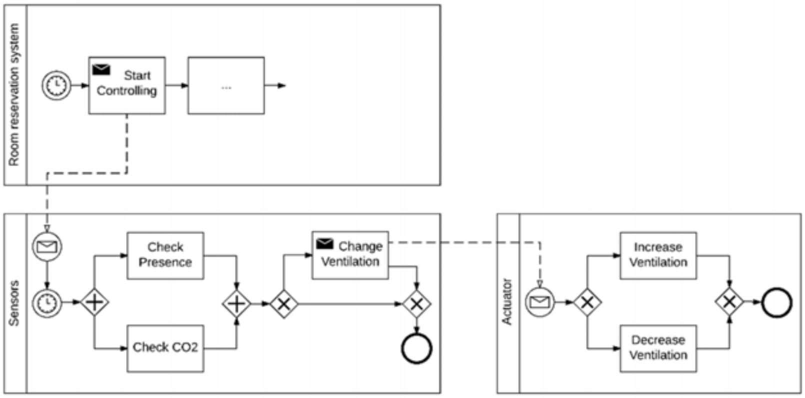

Fig. 1. BPMN process model for the ventilation process use case (adapted from [4, 21, 22])

3.2. Use case with human resources

As in other domains, in the last years we can observe an increasingly use of BMPN to model and execute healthcare processes, which can be administrative as well as clinical processes [19]. In addition, we can find in the literature some proposals that aim at overcoming BPMN limitations when used for clinical processes. For instance, Braun et al. extend BPMN with additional domain specific objects for modeling clinical processes [1], while in [13, 7], the authors extend BPMN to provide flexibility, and, in particular, controlled flexibility, an unavoidable requirement of this kind of processes. However, besides this, the healthcare is also a rich domain where clinical processes cannot bypass reliability requirements. Indeed, reliability of equipment [6] and humans [20] has to be reflected in clinical processes. For instance, in the study presented in [20], the authors evaluate the reliability of physicians in the interpretation of cranial computed tomography (CT) scans for determining eligibility for thrombolytic therapy in acute stroke. The group of physicians in the study included 38 emergency physicians, 29 neurologists, and 36 general radiologists, each of whom was asked to interpret 15 CT scans randomly chosen from a pool of scans that demonstrated intracerebral hemorrhage, acute infarction, intracerebral calcifications, old cerebral infarctions, and normal findings. The results revealed that on average physicians could accurately interpret 77% of scans. Nevertheless, physicians did not uniformly achieve accuracy identifying the level of intracerebral hemorrhage. Complete accuracy in detecting hemorrhage was achieved by 17% of emergency physicians, 40% of neurologists and 52% of radiologists.

Fig. 2 presents an extract of a simplified clinical process observed in emergency care. In this simplified process, the reliability of the "Check Exams" task positively correlates with the reliability of the overall process. In addition, the reliability of this task should reflect the reliability of the physician who performs it. If needed, to improve reliability, more than one physician can perform the task, as modelled in the example with the multi-instance task.

† For the sake of clarity, we distinguish sensor nodes (as named motes) from sensors. A mote is a sensor network node with capabilities for processing, sensing the environment and communicating with other connected nodes in the network. Motes include sensors, the components capable of sensing [9].

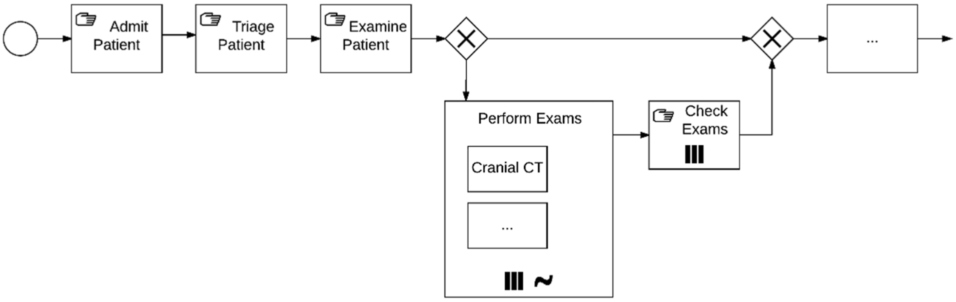

Fig. 2. An extract of a simplified clinical process

4. Resource reliability information in BPMN business processes

We illustrate in this section how we can use BPMN to include reliability information in resource definitions and assignments, taking into account the two use cases explained in the previous sections.

4.1. Resources in BPMN

The BPMN 2.0 specification does not include the definition of organizational models and resources. The `Resource` class is used to specify resources that activities and processes can reference. Resources can be human resources as well as other type of resources, such as sensors. They can have a set of `ResourceParameters`, which can be used in queries at runtime for resource assignment.

Resource assignment is supported through the `ResourceRole` element and can be specified using expressions or parameterized resources, by using the `ResourceAssignmentExpression` or the `resourceParameterBinding` elements respectively. The BPMN class diagram for assigning resources is illustrated in Fig. 3.

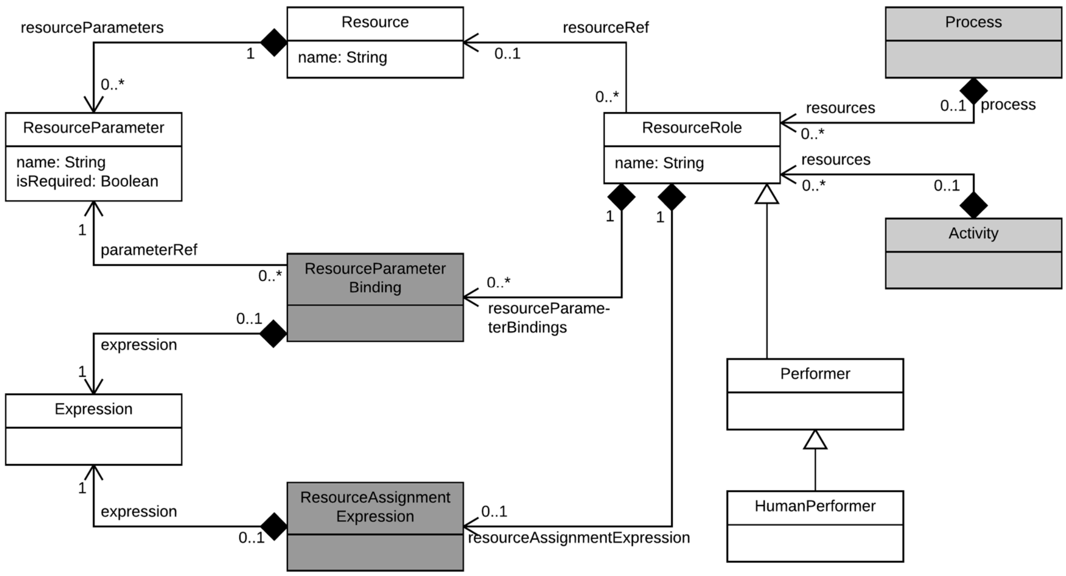

Fig. 3. UML class diagram for assigning Resources in BPMN [17]

4.2. Reliability in BPMN

In [18], we define the `relyBPMN` BPMN extension to include reliability information in business process models and use it to calculate the overall process reliability.

As shown in Listing 1, the `relyBPMN` extension has two elements: the `ReliabilityInformation` and the `Probability`. The `requiredReliability` attribute of the first element defines the minimum accepted reliability value for processes or activities, while the `calculatedReliability` attribute is the reliability of the BPMN element, which is calculated with the SWR algorithm for decomposable activities and processes. The `probability` value is used with conditional `SequenceFlow` elements in conditional blocks or loop blocks and defines the probability of the process execution path taking them.

Process reliability is calculated by using the SWR algorithm, which applies a set of reduction rules to the process, iteratively, until only one activity remains. The reliability of the remaining activity corresponds to the reliability of the process. The BPMN blocks for which we apply reduction rules are: sequencial, parallel, conditional, loop, fault tolerant, and network (or subprocess).

Listing 1. The XML Schema of the *relyBPMN* extension defined in [18]

```
<xsd:group name="relyBPMN">
  <xsd:sequence>
    <xsd:element name="ReliabilityInformation" type="tReliabilityInformation"
               minOccurs="0" maxOccurs="1"/>
    <xsd:element name="Probability" type="tProbability" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:group>
<xsd:complexType name="tReliabilityInformation" abstract="false">
  <xsd:attribute name="requiredReliability" type="xsd:float"/>
  <xsd:attribute name="calculatedReliability" type="xsd:float"/>
</xsd:complexType>
<xsd:complexType name="tProbability" abstract="false">
  <xsd:attribute name="value" type="xsd:float"/>
</xsd:complexType>
```

4.3. Reflecting resource reliability in processes

For the first use case, to assign sensors to the tasks considering the required reliability, we define a resource class with three parameters as illustrated in Listing 2. The name parameter identifies the sensor, the function parameter identifies the sensor type (i.e., what it can measure), and the required reliability defines minimum accepted reliability value for the sensor.

Listing 2. XML extract of a sensor resource definition for the BPMN ventilation process, with a `requiredReliability` parameter

```
<resource id="sensor" name="sensor">
  <resourceParameter id="name" name="name" type="string"/>
  <resourceParameter id="function" name="function" type="string"/>
  <resourceParameter id="reliability" name="reliability" type="float"/>
<resource>
```

Considering the *Check Presence* task of the BPMN use case process in Fig.1, we only define the function parameter of the resource, so it can be used to query the resource model to select the adequate (presence) sensor, as defined in Listing 3. By using the *relyBPMN* extension, we define that the minimum required reliability value for the *Check CO2* task is 0.98 (line 12 in Listing 3). This value is, in this case, directly related to the reliability value of the CO2 sensor to be used.

The conditional assignment of this sensor is accomplished using the `resourceParameterBinding` and `formalExpression` XML elements (lines 20 to 24 of Listing 3) and corresponding pairs of attribute/value. The `parameterRef` attribute binds to the kind of sensor information we want to retrieve (the sensor's `reliability`, in this case), and the `formalExpression` is written in XPath, to get the value of the `requiredReliability` extension element from our `relyBPMN` extension. This value can then be used by the Business Process Management System (BPMS) in place to query the sensors that can be assigned to this script task.

Listing 3. XML extract of the ventilation process, including sensor resource assignment expressions for the tasks *Check Presence* and *Check CO2*

```
<scriptTask id="CheckPresence" name="Check Presence">
  <performer id="performer4CheckPresence" name="performer">
    <resourceRef>sensor<resourceRef>
    <resourceParameterBinding parameterRef="function">
      <formalExpression>Presence Sensor<formalExpression>
    </resourceParameterBinding>
  <performer>
<scriptTask>

<scriptTask id="CheckCO2" name="Check CO2">
  <extensionElements>
    <relyBPMN:ReliabilityInformation>
      <relyBPMN:requiredReliability>0.98</relyBPMN:requiredReliability>
    </relyBPMN:ReliabilityInformation>
  </extensionElements>
  <performer id="performer4CheckCO2" name="performer">
    <resourceRef>sensor<resourceRef>
    <resourceParameterBinding parameterRef="function">
      <formalExpression>CO2 Sensor<formalExpression>
    </resourceParameterBinding>
    <resourceParameterBinding parameterRef="reliability">
      <formalExpression>getActivityInstanceAttribute('CheckCO2',
                          'ReliabilityInformation')/requiredReliability
      <formalExpression>
    </resourceParameterBinding>
  <performer>
</scriptTask>
```

Listing 4 shows the BPMN implementation for the clinical process illustrated in Fig. 2. Here, a resource of type physician is in place from lines 1 to 5, including the `name`, `specialty` and `reliability` parameters. Then, this information is used in runtime to allocate a physician to the "Check Exams" task, where a `requiredReliability` value is included (line 10), as well as a `formalExpression` referring to the physician's specialty (lines 15 and 16). Moreover, there is another expression in lines 20, 21 and 22 that conditionally determines the end of this multiple instance task, based on the previous reliability requirement. This means that the number of physicians required to check the patient exams depends on the reliability value of each physician allocated in runtime to each instance of this task. Each of these reliability values are then used to calculate the overall task reliability value, and to verify its conformance with the required reliability one.

Listing 4. XML extract of the clinical process, including human resource assignment expressions for the task *Check Exams*

```
1  <resource id="physician" name="physician">
2    <resourceParameter id="name" name="name" type="string"/>
3    <resourceParameter id="specialty" name="specialty" type="string"/>
4    <resourceParameter id="reliability" name="reliability" type="float"/>
5  <resource>
6
7  <manualTask id="CheckExams" name="Check Exams">
8    <extensionElements>
9      <relyBPMN:ReliabilityInformation>
10     <relyBPMN:requiredReliability>0.85</relyBPMN:requiredReliability>
11     </relyBPMN:ReliabilityInformation>
12   </extensionElements>
13   <humanPerformer id="humanPerformer" name="Physician">
14     <resourceRef>physician</resourceRef>
15     <resourceParameterBinding parameterRef="specialty">
```

```
      <formalExpression>"emergency" or "neurologist" or "radiologist"</formalExpression>
    </resourceParameterBinding>
  </humanPerformer>
  <multiInstanceLoopCharacteristics isSequential="true">
    <completionCondition> getActivityInstanceAttribute('CheckExams',
   'ReliabilityInformation')/requiredReliability <= getActivityInstanceAttribute('CheckExams',
'ReliabilityInformation')/calculatedReliability</completionCondition>
  </multiInstanceLoopCharacteristics>
</manualTask>
```

5. Conclusions and future work

Reliability in business processes can also be affected by the intrinsic reliability of the resources used to perform its tasks. In this paper, we propose the use of BPMN and the *relyBPMN* extension to include reliability information regarding the resources perspective of business processes. We present examples regarding two use cases involving sensors and humans as resources that must be defined and assigned to tasks. We also show how their information can be used to calculate the reliability of a business process task.

From the unspecified and XML (code)-intensive way BPMN foresees to include this kind of reliability information, we are aiming our future work on the development of a corresponding graphical notation for BPMN. This will allow for business process engineers to focus on the definition of reliability requirements using graphical notations (for both control-flow and resource perspective process elements), rather than having to express these requirements using XML.

Acknowledgements

This work is partially supported by National Funding from FCT - Fundação para a Ciência e a Tecnologia, under the projects PTDC/EEI-ESS/5863/2014, UID/MAT/04561/2013 and UID/CEC/00408/2013.

References

1. Braun, R., Schlieter, H., Burwitz, M., & Esswein, W. (2015). Extending a Business Process Modeling Language for Domain-Specific Adaptation in Healthcare. Wirtschaftsinformatik Proceedings.
2. Caracaş, A., & Bernauer, A. (2011, June). Compiling business process models for sensor networks. In *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)* (pp. 1-8). IEEE.
3. Cardoso, A.J.S. (2002) Quality of service and semantic composition of workflows (Doctoral dissertation). University of Georgia.
4. Casati, F., Daniel, F., Dunkels, A., Karnouskos, S., Montero, P. M., Mottola, L., ... & Tranquillini, S. Invited Poster: makeSense: Easy Programming of Integrated Wireless Sensor Networks. In European Conference on Wireless Sensor Networks EWSN 2011 (p. 53).
5. Chiu, H., & Wang, M. (2013). A Study of IoT-Aware Business Process Modeling. International Journal of Modeling and Optimization, 3(3), 238-244.
6. Dhillon, B. S. (2011). Medical Equipment Reliability: a Review, Analysis Methods and Improvement Strategies. International Journal of Reliability, Quality and Safety Engineering, 18(04), 391-403.
7. Domingos, D., Martinho, R., & Varajão, J. (2016). Controlled Flexibility in Healthcare Processes: A BPMN-Extension Approach. Encyclopedia of E-Health and Telemedicine, p. 521-535.
8. Harmon P, Wolf C. The State of Business Process Management. Business Process Trends; 2008. http://www.bptrends.com/bptrends-surveys/, last access: 27/03/2015.
9. Karray, F., Jmal, M. W., Abid, M., BenSaleh, M. S., & Obeid, A. M. (2014). A review on wireless sensor node architectures. In Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), 2014 9th International Symposium on (pp. 1-8). IEEE.
10. Madakam, S., & Ramachandran, R. Barcelona Smart City: The Heaven on Earth (Internet of Things: Technological God). ZTECOMMUNICATIONS, 3.
11. Martinho, R., & Domingos, D. (2014). Quality of information and access cost of IoT resources in BPMN processes. *Procedia Technology*, *16*, 737-744.
12. Martinho, R., Domingos, D. & Respício, A. (2016). Evaluating the Reliability of Ambient-Assisted Living Business Processes. In Proceedings of the 18th International Conference on Enterprise Information Systems (pp. 528 – 536), Scitepress.
13. Martinho, R., Domingos, D., and Varajão, J. (2015). CF4BPMN: a BPMN extension for controlled flexibility. Procedia Computer Science, 64, 1232-1239.
14. Meyer, S., Ruppen, A., & Magerkurth, C. (2013). Internet of things-aware process modeling: integrating IoT devices as business process resources. In International Conference on Advanced Information Systems Engineering (pp. 84-98). Springer Berlin Heidelberg.

15. Meyer, S., Sperner, K., & Magerkurth, C. (2011). Towards Real World Aware Enterprise Systems-Reflecting the Quality Information of Physical Resources in Services and Processes. In Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on (pp. 843-848). IEEE.
16. Meyer, S., Sperner, K., Magerkurth, C., & Pasquier, J. (2011). Towards modeling real-world aware business processes. In Proceedings of the Second International Workshop on Web of Things (p. 8). ACM.
17. Object Management Group (OMG), Business Process Model and Notation (BPMN) Version 2.0, 2011.
18. Respício, A., & Domingos, D. (2015). Reliability of BPMN Business Processes. Procedia Computer Science, 64, 643-650.
19. Russo, A., & Mecella, M. (2013). On the evolution of process-oriented approaches for healthcare workflows. International Journal of Business Process Integration and Management, 6(3), 224-246.
20. Schriger, D. L., Kalafut, M., Starkman, S., Krueger, M., & Saver, J. L. (1998). Cranial computed tomography interpretation in acute stroke: physician accuracy in determining eligibility for thrombolytic therapy. Jama, 279(16), 1293-1297.
21. Sungur, C. T., Spiess, P., Oertel, N., & Kopp, O. (2013). Extending bpmn for wireless sensor networks. In 2013 IEEE 15th Conference on Business Informatics (pp. 109-116). IEEE.
22. Tranquillini, S., Spieß, P., Daniel, F., Karnouskos, S., Casati, F., Oertel, N., ... & Voigt, T. (2012). Process-based design and integration of wireless sensor network applications. In International Conference on Business Process Management (pp. 134-149). Springer Berlin Heidelberg.
23. Zeng, D., Guo, S., & Cheng, Z. (2011). The web of things: A survey. Journal of Communications, 6(6), 424-438.