CEPIS UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://cepis.org/upgrade>

Publisher
CEPIS UPGRADE is published by CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>), in cooperation with the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática,* <http://www.ati.es/>) and its journal Novática

CEPIS UPGRADE monographs are published jointly with Novática, that published them in Spanish (full version printed; summary, abstracts and some articles online

CEPIS UPGRADE was created in October 2000 by CEPIS and was first published by Novática and INFORMATIK/INFORMATIQUE, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>)

CEPIS UPGRADE is the anchor point for UPENET (UPGRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:

- inforewiew, magazine from the Serbian CEPIS society JISA
- Informatica, journal from the Slovenian CEPIS society SDI
- Informatik-Spektrum, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- ITNOW, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- Mondo Digitale, digital journal from the Italian CEPIS society AICA
- Novática, journal from the Spanish CEPIS society ATI
- OCG Journal, journal from the Austrian CEPIS society OCG
- Pliroforiki, journal from the Cyprus CEPIS society CCS
- Tölvumál, journal from the Icelandic CEPIS society ISIP

Cover page designed by Concha Arias-Pérez
"Devourer of Fantasy" / © ATI 2011
Layout Design: François Louis Nicolet
Composition: Jorge Llácer-Gil de Ramales

Editorial correspondence: Llorenç Pagés-Casas <pages@ati.es>
Advertising correspondence: <info@cepis.org>

Subscriptions
If you wish to subscribe to CEPIS UPGRADE please send an email to info@cepis.org with 'Subscribe to UPGRADE' as the subject of the email or follow the link 'Subscribe to UPGRADE' at <http://www.cepis.org/upgrade>

ISSN 1684-5285

Monograph of next issue (April 2011)
"Software Engineering for e-Learning Projects"
(The full schedule of **CEPIS UP**GRADE is available at our website)

CEPIS UPGRADE
The European Journal for the Informatics Professional
http://cepis.org/upgrade
Vol. XII, issue No. 1, February 2011

Monograph
Internet of Things
(published jointly with Novática*)
Guest Editors: *Germán Montoro-Manrique, Pablo Haya-Coll, and Dirk Schnelle-Walka*

UPENET (UPGRADE European NETwork)

CEPIS NEWS

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

Model Driven Development for the Internet of Things

Vicente Pelechano-Ferragud, Joan-Josep Fons-Cors, and Pau Giner-Blasco

The Internet of Things vision is about reducing the gap between the physical and the digital world to make daily activities more fluent. By providing a digital identity to real-world objects, Information Systems can handle them in an automatic way. This enables physical objects to participate actively in business processes. In such systems, the technological heterogeneity in Auto-ID and the fast-changing requirements of business processes hinders their construction, maintenance and evolution. This paper shows how the Model Driven Development can help to systematize the development of business process-supporting systems that integrate physical elements.

Keywords: Auto Identification (Auto-ID), Business Processes, Implicit Interaction, Internet of Things, Mobile Workflows, Model Driven Development, Models.

Authors

Vicente Pelechano-Ferragud is an Associate Professor in the *Centro de Investigación en Métodos de Producción de Software* at *Universidad Politécnica de Valencia*, Spain, and leads the technical supervision of the MOSKitt open source CASE tool, <http://www.moskitt.org>. His research interests include model-driven development, Web Engineering, mobile and ubiquitous computing, and business-process modelling. He has a PhD in Computer Science from the *Universidad Politécnica de Valencia*. <pele@pros.upv.es>.

Joan-Josep Fons-Cors is an Assistant Professor in the *Centro de Investigación en Métodos de Producción de Software* at *Universidad Politécnica de Valencia*, Spain. His research interests include model-driven development, Web engineering, ubiquitous and autonomic computing, UI development and product line engineering. He has a PhD in Computer Science from the *Universidad Politécnica de Valencia*. <jjfons@pros.upv.es>.

Pau Giner-Blasco is an IT consultant at Everis. He is a member of the *Centro de Investigación en Métodos de Producción de Software* at *Universidad Politécnica de Valencia*, Spain. His research interests include business process modelling, the Internet of Things, and mobile computing. He has a PhD in Computer Science from the *Universidad Politécnica de Valencia*. <pginer@pros.upv.es>.

1 Introduction

The introduction of Information Technologies creates a digital world where information can be automatically processed, improving the Information System efficiency. However, computers have a limited vision of the real world they are managing. Thus, there is still a challenge in automating the linkage between digital and physical worlds.

Nowadays, Information Systems that deal with real-world objects (such as baggage pieces in an airport or products in a supermarket) are normally informed by humans. We consider this use of humans as information carriers to be inefficient and error-prone. The gap between the physical and the digital world commonly results in mishandled luggage or long queues at the supermarket.

Internet of Things vision [1] is about reducing this gap to make daily activities more fluent. By providing a digital identity to real-world objects, Information Systems can handle them in an automatic way. This enables physical objects to participate actively in business processes by reducing the gap between physical and virtual worlds [2]. In addition, the wide availability of mobile devices with advanced capabilities allows users to access the information and services where they need them.

Although developing such systems is feasible, the technological heterogeneity in Auto Identification (Auto-ID) and the fast-changing requirements of business processes hinders their construction, maintenance and evolution. Therefore, there is a need to move from ad-hoc solutions to sound development methods in order to assure the quality of the final product. Model Driven Engineering techniques [3] can help the developers to provide development principles for constructing such systems.

The main goal of this paper is to show how Model Driven Development can help to systematize the development of business process-supporting systems that integrate physical elements. The development process defined is focused on the particular requirements of the linkage between physical and virtual worlds. For the system specification, a modelling language is defined to cope with the particular requirements of the Internet of Things domain. Then, a software solution is obtained from this specification by following a set of systematic steps. This solution is supported by an architecture specifically designed to cope with the Internet of Things requirements and to survive to technological evolution.

Specifically, our target applications are physical mobile workflows, which are business processes that take advantage of the capabilities of mobile devices for the identification of physical elements. The high heterogeneity in identification technologies, the fragmentation in mobile platforms and the fast-changing nature of business processes, make it hard to develop such systems in a sound manner. In this paper, modelling techniques are applied in order to develop such systems from a higher abstraction level.

The remainder of the paper is structured as follows. Sec-

> Computers have a limited vision of the real world they are managing

tion 2 presents the main ideas for integrating business processes with the Internet of Things principles. Section 3 discusses taking into account a modelling strategy to help defining systems by avoiding problems regarding technical details. Section 4 presents the main ideas for designing a method to support the development of Internet of Things systems by taking into account a model driven strategy. A tool support for this method is described in Section 5. Finally, Section 6 concludes the paper.

2 Business Processes and the Internet of Things

Integrating real-world objects in business processes has been successfully demonstrated to reduce media breaks, human errors and delayed information problems [2]. Many benefits are obtained in economic [4] and process improvement terms [5][6]. A better integration of real and virtual worlds not only improves business processes, but also enables new business models [7][8].

However, it is not an easy task to develop systems of this kind. Business processes are constantly changing, which in turn requires the corresponding evolution in the supporting Information System. In addition, systems in the Internet of Things context, involve a great diversity of technologies to bridge physical and digital worlds. This heterogeneity forces the developer to know the details of each technology involved in the system, making these systems difficult to develop and to maintain. From the methodological perspective, there is a need for a systematic development method that can free developers from technological details and that also allows a fast propagation of requirement changes to technological solutions.

This work presents a method that provides a mechanism for defining the desired degree of automation for the physical-virtual linkage of a given business process. In order to systematize the development of such systems, the method is based on the Business Process Management (BPM) initiative principles. BPM is an initiative that promotes the continuous re-engineering of business processes. Since current solutions for BPM are mainly focused on the digital world (i.e. orchestration of digital services), support is lacking for coping with the particularities of the physical-virtual linkage in the different stages of the BPM cycle. This work builds onto existing BPM techniques and extend them to integrate business processes with the physical world at different levels. Existing BPM techniques are complemented with support for capturing the identification requirements, evaluating the user participation in a real environment, and executing the workflow in a software platform.

3 Why a Modelling Approach?

Traditionally, the application of Auto-ID to business processes has mainly been approached from a technological perspective (by developing integration middleware and architectural designs). However, deploying an Auto-ID-enabled system involves a lot more than purchasing the right tags and installing the right readers [9].

The way in which a business goal is achieved depends on the properties of the physical-virtual integration. Certain business models are only feasible with an adequate level of automation in the physical-virtual linkage [7]. For example, using RFID for identifying products in a supermarket allows the checkout process to be automated [10], and it does not require the participation of a cashier in the process. Thus, when modelling a business process it is not possible to determine which tasks are required for handling physical elements (e.g. requiring a cashier to handle them or not) if there is no notion of how they participate in the process. Models are key in our proposal to provide this notion by linking identification requirements to technological requirements in a gradual manner.

Abstraction is one of the fundamental principles of software engineering in order to master complexity [11]. Our approach makes use of modelling techniques in order to promote abstraction in the development of physical mobile workflows. By abstracting technical details, we can describe the physical-virtual connection of a workflow regardless of the particular technology used for the implementation. In the case of physical mobile workflows, modelling techniques are applied to obtain the following benefits:

- **Focus on the process.** Separation of concerns is promoted by our approach in order to allow designers to focus on a specific aspect of the workflow at a time. Business analysts can define the way in which physical elements participate in a business process without considering technology limitations. They can think on the way they want the process tasks to flow, and later, the appropriate identification mechanisms can be chosen to cope with their requirements.
- **Explore the solution space.** The use of models allows the capture of not only a specific solution but also the rationale behind it. In this way, alternative solutions can be re-considered and the design knowledge can be better reused for similar problems. In addition, support for traceability allows to easy identification of the model elements affected when different issues are detected during the system evaluation.

> Internet of Things vision is about reducing the gap to make daily activities more fluent

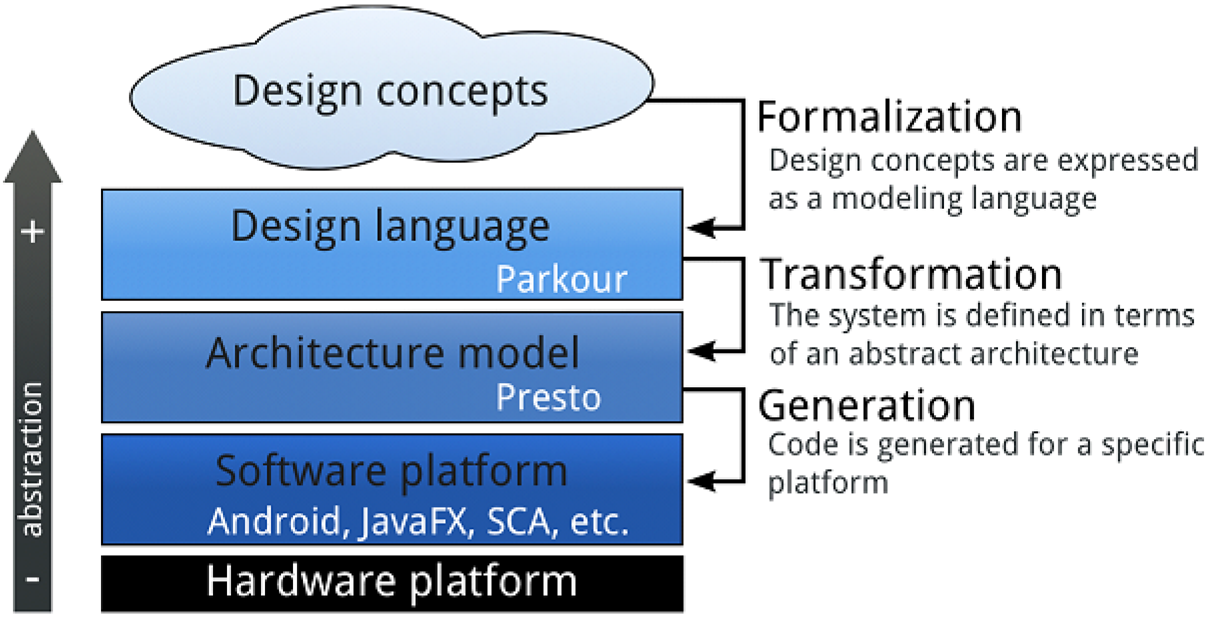

Figure 1: Strategy for covering the Abstraction Gap in the Development of Physical Mobile Workflows.

■ **Support system evolution.** The fast changing nature of business processes, and the technological heterogeneity of identification technologies, suggests that systems in this area must be designed to evolve. By analyzing the knowledge captured in models by our approach, it can be easily determined whether or not a new technology fits better with the requirements of the physical-virtual linkage for a given process.

Our approach involves the manipulation of models in different manners. An overview of the steps involved in our method is provided in the following section.

4 Method Overview

This section presents the development method introduced in this paper. The design stage is the initial stage in our method (see Figure 1). Since we follow a model-driven approach, the specification obtained at design drives the later stages in the development of the system. Thus, the design stage becomes central to the development method. The design method captures, by means of models, the concepts that are relevant in the development of physical mobile workflows.

Modelling techniques describe a system by handling abstractions of the problem space. This allows designs to be expressed in terms of concepts from the application domain instead of concepts from the technical space [12]. By raising the level of abstraction, systems can be developed without taking into account much of the implementation details of the underlying platform. Models are used to organize knowledge about the problem domain in order to

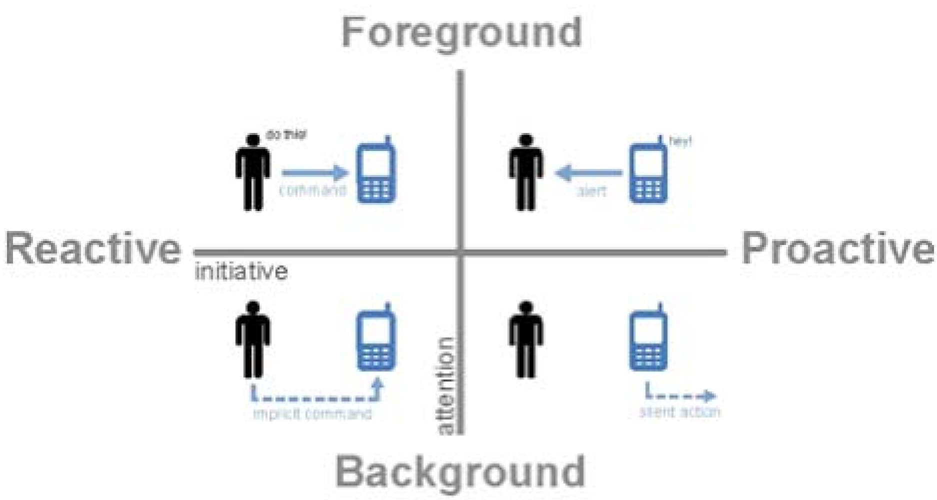

Figure 2: Implicit Interaction Framework used in this Work. Some Mobile Interactions are illustrated for the Regions.

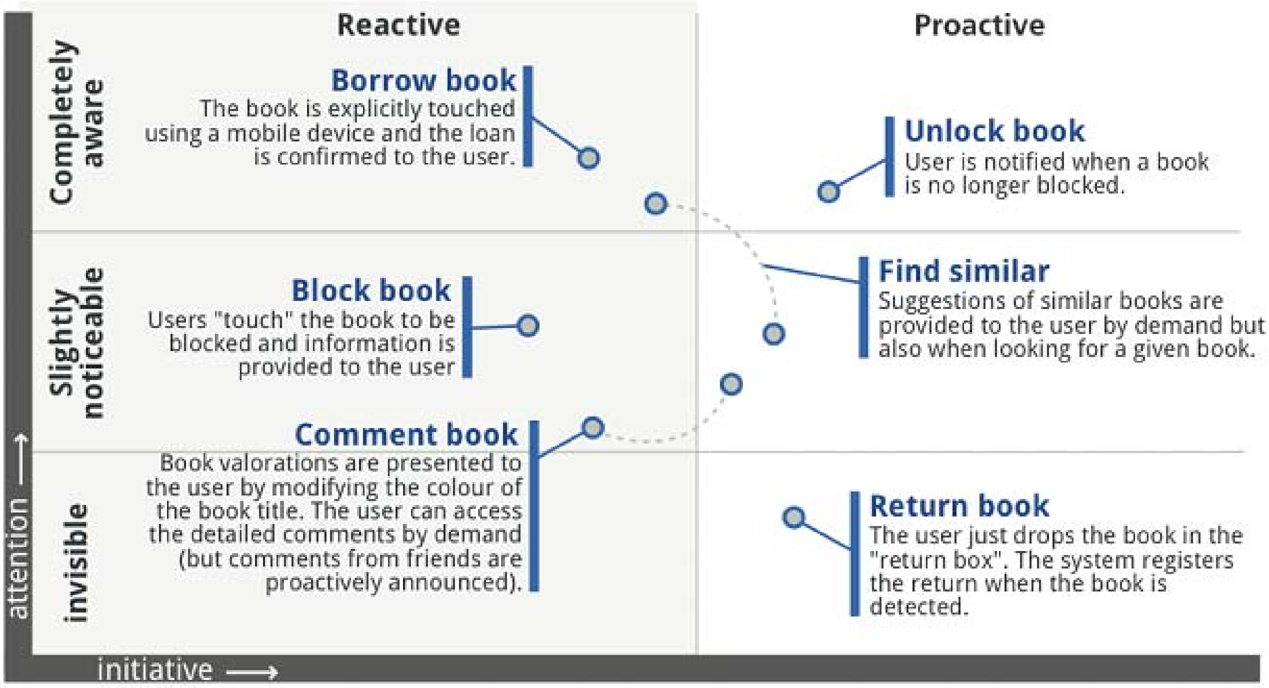

Figure 3: Obtrusiveness Level defined for Each Task in the Smart Library Scenario.

guide the development. Furthermore, when models are machine-processable and precise-enough, they can be used to automate the production of a software system.

Thanks to Model Driven Engineerng (MDE) techniques, it is possible to traverse the gap between the high-level concepts used in design and the technical details of the particular mobile platform that are used for the system implementation.

Figure 1 illustrates how our approach can connect the concepts used at design (e.g. task, obtrusiveness level, physical interaction, etc.) with a particular implementation platform. Our approach introduces two layers to cover the abstraction gap between the design concepts and the software platform:

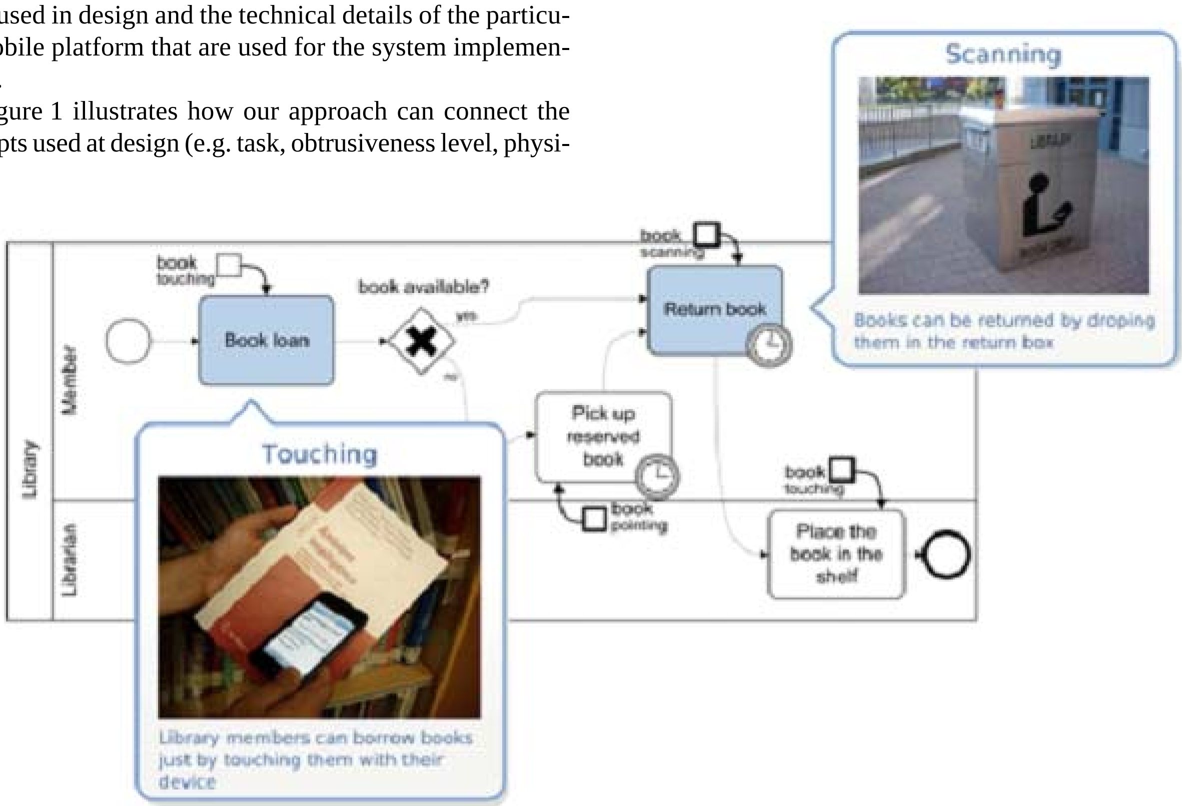

Figure 4: Different Interactions with Physical Elements orchestrated in a Workflow Model.

■ **Parkour.** The design concepts defined are formalized into a modelling language named Parkour. Designers can define the requirements for a workflow by following our design method and use the system specification during the development process. By formalizing design concepts, system descriptions can be processed to validate their consistency and automate some steps in the development. It is possible to transform workflow specifications based on Parkour into a final software solution using MDE techniques.

■ **Presto.** Presto [13] is an architectural framework specifically defined to support applications in the physical mobile workflow domain. Presto is a sustainable software architecture, that is, an architecture that can evolve over time throughout several technological cycles. The architecture elements are defined in a technology-independent fashion and code generation techniques are used to translate the generic architecture into components for each particular mobile platform targeted for development.

We believe that it is important to describe the physical-virtual linkage for a workflow at design time since the way in which a business goal is achieved depends on the properties of the physical-virtual integration. Our approach is based on existing notations for business process modelling such as Business Process Modelling Notation (BPMN). Parkour complements BPMN with the modelling of three aspects that affect the physical-virtual linkage:

■ **Obtrusiveness level for each task.** We identify the obtrusiveness level required for the different tasks of the workflow. Each task can be carried out at a different level of initiative and attention according to the conceptual framework introduced in [14].

■ **Interaction techniques to use.** Users can interact with the objects in their surroundings in different manners. The interaction technique that allows the task to be performed at the adequate obtrusiveness level is selected for each task in the process.

■ **Supporting technologies.** The different mobile devices involved in a workflow must be equipped with technologies that support the interaction technique that was chosen in the previous stage.

More detail on these aspects are provided below. In order to illustrate these aspects we are using the Smart Library case study [13] were different workflows are supported in the context of a Library.

4.1 Obtrusiveness Level for Each Task

The first step in our method is to detect the tasks to support in a workflow and determine at which obtrusiveness level they should be supported. Each task in a workflow can be provided at a different obtrusiveness level. The design method introduced in this work allows designers to specify a workflow by indicating (1) to which extent the different tasks should intrude the users mind and (2) which technologies can be used to support such interactions. Since mobile devices provide rich interaction mechanisms, we are not considering interactions to be either implicit or explicit in a binary fashion but part of a continuum. In this work we use the concept of obtrusiveness level to indicate to which degree an interaction is implicit or not.

> "There is not a universal interaction technique that is well suited for any situation"

> "The method is based on the Business Process Management (BPM) initiative, that promotes the continuous re-engineering of business processes"

In order to specify the obtrusiveness level for a task, we make use of the conceptual framework presented in [14]. This framework defines two dimensions (see Figure 2) to characterize implicit interactions: *initiative* and *attention*. According to the initiative factor, interaction can be reactive (the user initiates the interaction) or proactive (the system takes the initiative). With regard to the attention factor, an interaction can take place at the foreground (the user is fully conscious of the interaction) or at the background of user attention (the user is unaware of the interaction with the system).

We found it very useful to consider *initiative* and *attention* as independent concepts. In the case of physical mobile workflows, automation and user awareness are factors that usually vary independently. For example, an automated task (i.e. proactive in terms of attention) can require the user to be aware of it (i.e. foreground in terms of the attention dimension) or not (i.e. at the background of the user attention) depending on different context factors (such as the user workload).

Figure 3 shows some of the most representative tasks of the Smart Library case study and their obtrusiveness level. The obtrusiveness space in this case was defined by dividing each axis into different parts. The initiative axis in this case is divided into two parts: *reactive* and *proactive*. The attention axis is divided into three segments, which are associated with the following values: *invisible* (there is no way for the user to perceive the interaction), *slightly-noticeable* (usually the user would not perceive it unless he/she makes some effort); and *completely aware* (the user becomes aware of the interaction even if he/she is performing other tasks). Designers can divide each axis into as many parts as they require for describing the obtrusiveness level for the process. This division is later considered when selecting the appropriate interaction mechanisms for each el-

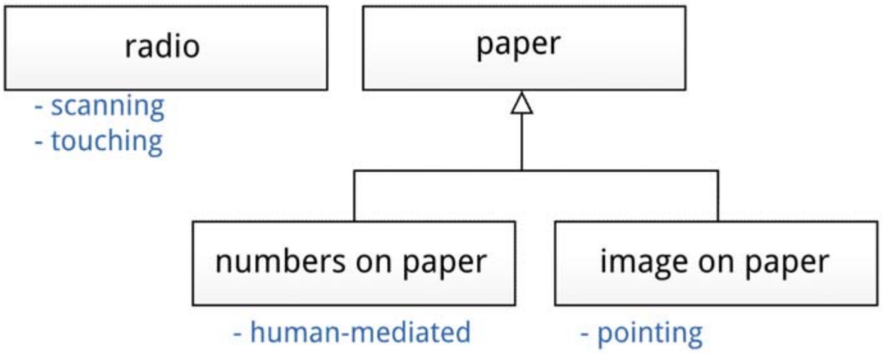

Figure 5: An Example of Different Mediums and Specialization Relationships

ement involved in the process.

During analysis we decided the appropriate obtrusiveness level for certain tasks. In the example, *borrowing books* is performed in an explicit manner by users. Users initiate the interaction (reactive) and they are informed about the *loan* (completely aware). The *return of books* task is performed in a completely unobtrusive manner, the user leaves the book in the *return box* and the system initiates the *return process* without notifying the user. Other tasks such as *finding similar books* are supported at different levels of obtrusiveness. In the example *suggestions* can be accessed by users or suggested by the system when the user is located near one of the related books.

The divisions defined in the obtrusiveness space allow designers to classify the different interaction techniques available. In this way, designers can later chose the interaction technique that best fits with the requirements captured. The following subsection provides more details on physical interaction techniques.

4.2 Interaction Techniques

Users can interact with a physical element in different manners. For example, users can access the services that are associated with an element either by pointing to the element, touching the element, or by scanning nearby elements with their mobile device. These are only some examples of the interaction techniques that have been defined for the interaction between users and their surroundings in the literature [15][16][17][18]. However, there is not a universal interaction technique that is well suited for any situation. We make use of the implicit interaction conceptual framework presented in Section 4.1 to determine the obtrusiveness level that can be supported for each interaction technique. For example, the *touching* interaction technique can be used for interactions that take place at the *reactive* region in terms of initiative, since the user initiates the interaction by explicitly touching a physical element with the mobile device.

Considering the obtrusiveness level required by each task we can determine the possible interaction techniques to use. For example, *touching* seems appropriate for users to borrow books and scanning can be used to silently detect the return of the books. The next step in our method is to specify how these different interactions fit in the workflow. BPMN is used for this purpose.

Figure 4 shows a BPMN diagram for the example of the *loan process* in a library. A BPMN diagram is divided into different lanes, each of which represents a participant in the process (e.g. *librarians* and *library members*). Each participant is in charge of performing several tasks (rounded squares) following a given flow (see arrows in the diagram). Finally, some control elements called gateways (diamonds in the figure) are used to synchronize and parallelize the workflow. We have extended the standard BPMN nota-

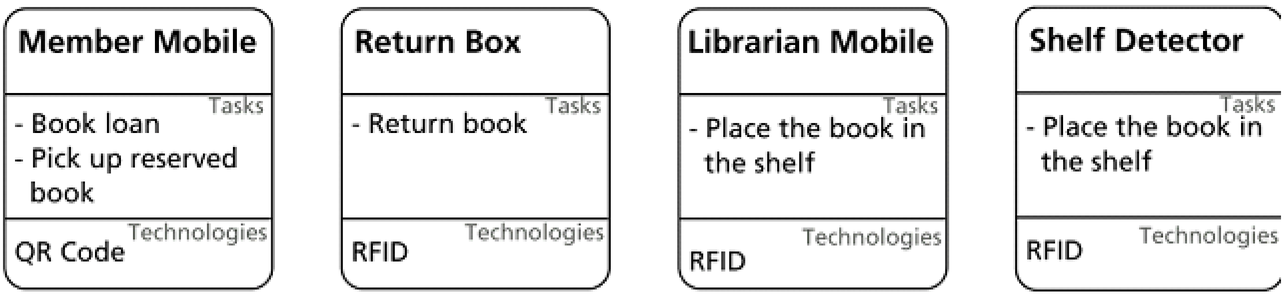

Figure 6: Auto-ID Services involved in the Smart Library

> “This work is part of a comprehensive proposal that includes a framework called Presto”

tion [19] since it lacked support for describing the participation of physical elements in a workflow. Figure 4 shows the interaction techniques that are used in each task from the workflow. For illustration purposes, the figure also includes an image that represents a possible implementation of each interaction technique in the context of a library.

Designers can find in the literature guidelines to determine the pros and cons of each technique [16][20]. For example, the *scanning* interaction technique has been selected for the return of books in the example since it allows books to be detected automatically with minimal user intervention. However, the interaction technique selected also depends on the identification technology used for the physical element. For example, *touching* cannot be applied if the physical element is only identified by means of a barcode. Next section describes the process of selecting the most adequate technology to support a given interaction technique.

4.3 Technology Selection

Many technologies have existed for long time to bridge the gap between the physical and the virtual worlds [21]. Thus, physical elements can be identified in a myriad of different ways (e.g. using a sequence of bars on paper, or a radio wave emitted by a RFID tag). The selection of the most appropriate one depends on many different factors. For example, RFID may not be appropriate for the users of the example library if we want them to autonomously borrow books with their device since RFID-reading capabilities are not present on the average mobile device whereas as far as we know cameras, to capture images, are more often present. Nevertheless, for taking such decisions it is necessary to understand the impact that each technology has in the process. For example, we can wonder if there is another technology that can be used to replace RFID and supports our workflow as it was described (and in the case that changes are required, easily identify them).

In order to organize the knowledge about the different identification technologies we define the medium concept. Mediums are defined in this work as physical supports for identifiers. Designers can define mediums to represent a genre of technologies in an abstract manner. For example, designers can define the *paper* medium (see Figure 5) in order to describe in an abstract manner a set of identification technologies that have some requirements in common: (1) they are cost-effective by using identifiers that are cheap to produce and that require simple devices for their capture; and (2) the technologies require direct line-of-sight since identifiers are recognized optically. At this point of the design we are not interested on the specific technology used (barcodes, visual markers, Quick Response (QR) Codes, etc.) but in a broad category of identification technologies with common properties. In particular, we consider which interaction techniques they support.

The medium concept is a technological-independent mechanism for describing identification requirements. This concept is useful for guiding the technology selection process. Designers can organize mediums into a hierarchy in order to better capture their commonalities and variability. In the example, the *paper* medium is specialized in two mediums which are considered as sub-types. These subtypes distinguish between an identifier that is expressed by means of numbers from one that is expressed by means of an image. In the example in Figure 5, image on paper and numbers on paper are paper-based mediums but the interaction mechanisms they support are different. The image on paper medium supports the *pointing* mechanism, whereas the numbers on paper medium *requires users* to perform the identification and type of the associated identifier (i.e. user-mediated interaction).

With this analysis we can determine the impact of the technology selection in the process. For example, if RFID (based on the *radio* medium) were considered not appropriate for its use by the library members due to the lack of RFID-capable mobiles widely available in the market, we can explore the consequences of using another kind of identification technology such as a *paper*-based one. In this case, using a *visual marker* (based on the image on paper medium) would require to use the *pointing* interaction technique instead of the *touching* one. The final decision depends on the designer but our approach allows the consequences of these decisions for the workflow to be tracked. Figure 6 shows the identification technology selected for different tasks of the Smart Library (at the bottom of each figure).

This study of the commonalities and variability of medium properties is also useful for supporting process evolution. When business processes are re-engineered new technologies can be considered for their support. By classifying the new technologies according to the existing medium hierarchy, it can be determined whether these new technologies are really adding value to the process or not.

> “The ‘Medium’ concept is a technological-independent mechanism for describing identification requirements”

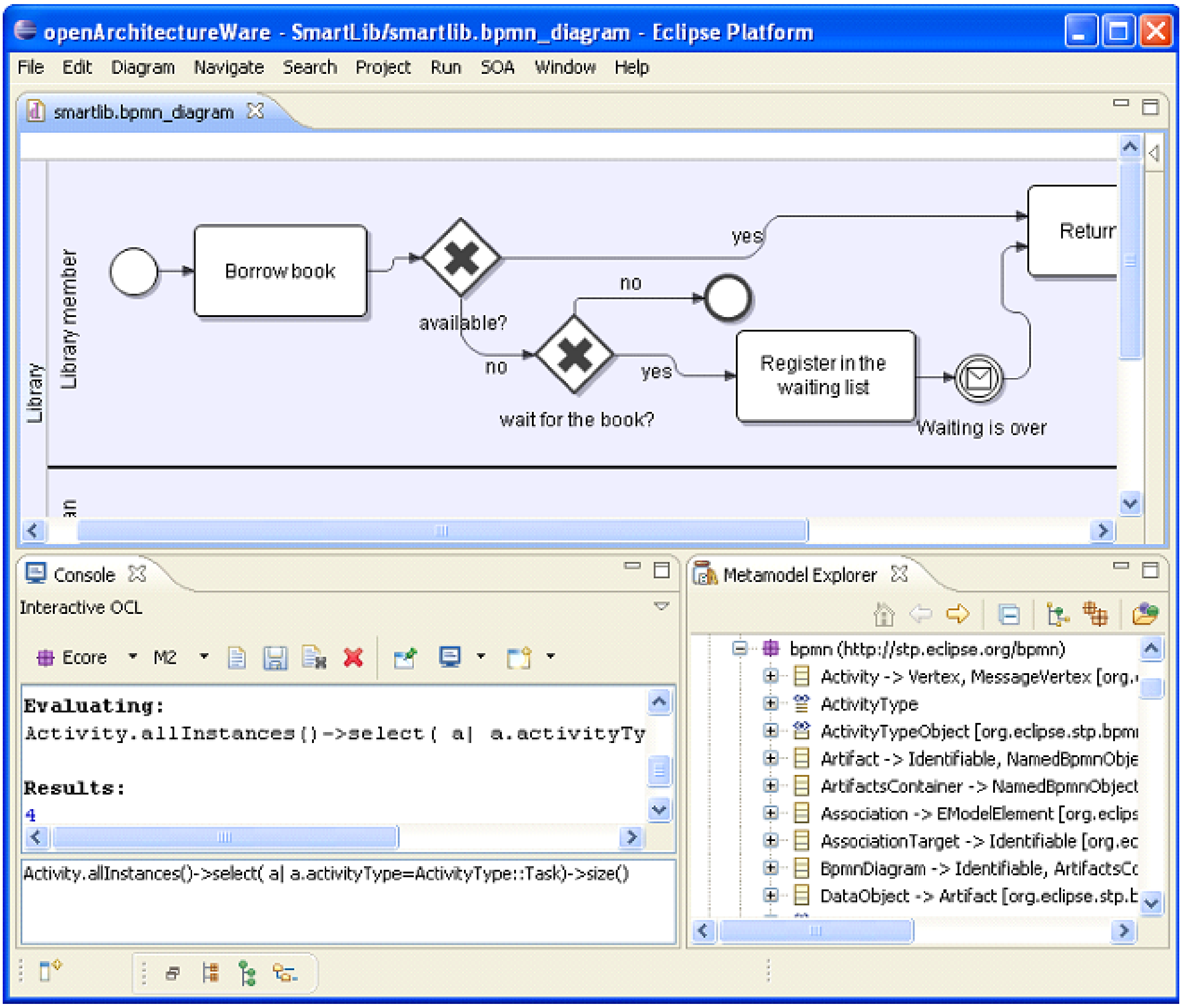

Figure 7: Eclipse-based Tools for editing and querying Business Process Models.

4.4 Deployment Distribution

Functionality in support of a mobile business process is normally distributed across different computing resources. In order to support the integration of physical elements in the different tasks from a business process, the identification functionality should be organized. This involves defining the setting of the different resources for the system. We use the concept of *deployment unit* to encapsulate the functionality required for the support of different tasks by means of a set of technologies that is deployed in a particular device.

To provide an abstract view of the Smart Library setting, the different deployment units are specified in our approach. Figure 6 represents a diagram for the Auto-ID related deployment units for the Smart Library case study.

The following properties are defined for each deployment unit; the **task** that it supports, the **physical elements** that are involved and the **technologies** that are used. The *Member Mobile* deployment unit represents a set of software components that support the *book loan* and *pick up reserved book* tasks by making use of QR Code technology for their completion. The software solution for this deployment unit is accessed by the library members from their mobile devices. The *Return Box* deployment unit is in charge of automatically detecting the *returned books* by means of RFID. Thus, each return box requires one or several RFID antennas capable of detecting its content. The *Librarian Mobile* deployment unit is accessed by the librarians from their RFID-equipped mobile devices in order to transfer the *returned books* from the return box to their shelf. The *Shelf Detector* deployment unit is also supporting the *place the book in the shelf* task. In this case, it detects whether a book is placed in a wrong shelf.

When defining the different deployment units it is essential the selected technology to be consistent with the other aspects considered during design. That is, the technology selected must use a medium that allows the interac-

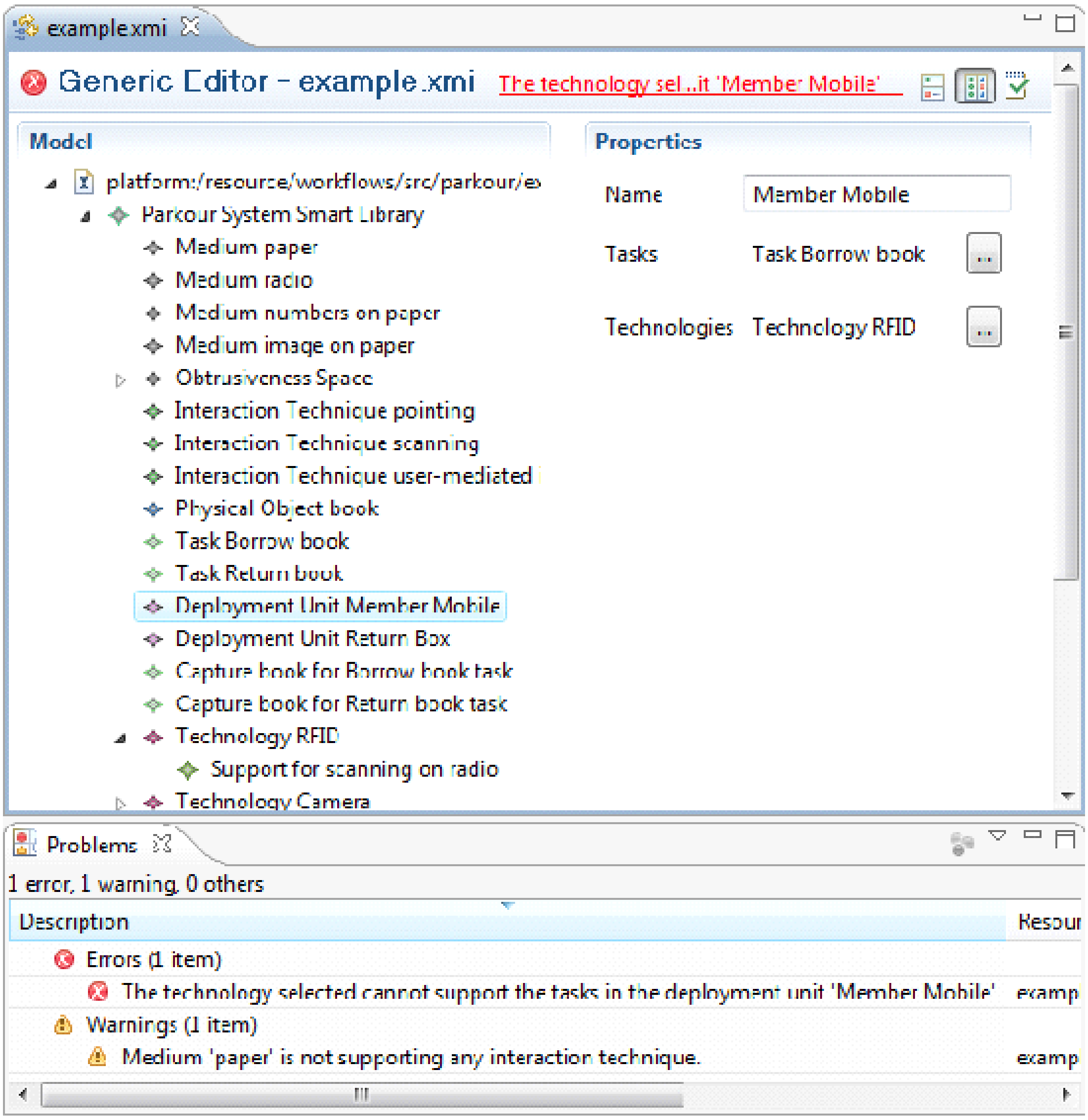

Figure 8: Validation for the Deployment Unit Constraint.

tion technique defined for the task, and this interaction technique supports the obtrusiveness level specified for this task. For example, the *Member Mobile* deployment unit makes use of QR Codes which is based on a image on paper medium that supports the pointing interaction technique since we considered this technique adequate to replace the touching interaction technique initially considered (as discussed in Section 4.3). This kind of validation can be difficult to perform as the workflow model grows. For this reason, tool support has been provided to automate this process.

5 Tool Support

Parkour introduces design concepts to specify different aspects of the physical-virtual linkage for a workflow. Tool support has been provided to model and validate workflows that are designed according to our method.

For the specification of business process models we took the BPMN metamodel defined in the SOA Tools Platform Project (STP) as a basis. The STP metamodel defines the modelling constructs for the BPMN modelling language. The STP metamodel has a very complete support for the BPMN specification covering almost all BPMN shapes, connections and markers except the layouts and appearance of the lanes inside a pool and the group-artifact. The STP project also provides a functional editor for BPMN diagrams (see Figure 7, top) which is integrated with other Eclipse-based modelling tools.

In addition to the editing support, we provided validation capabilities to verify that the description of the physical-virtual linkage in the workflow is consistent. Eclipse Modelling tools are used to formalize the concepts introduced in Parkour and to specify the different constraints on them. For example, the following expression checks whether a specific medium supports a particular interaction tech-

> “We have defined code generation support for the Android platform”

nique by taking into account the medium hierarchy:

```
Boolean
 supportsTechnique (Medium this,
 InteractionTechnique inter):
 this.InteractionTechnique.contains (inter) ||
 (this.parent != null?
 this.parent. supportsTechnique (inter) :
 false);
```

By enforcing the check of different constraints, inconsistencies are automatically detected in workflow specifications. This allows designers to anticipate the detection of problems in the workflow before effort is put intothe implementation and deployment. In this way, designers can foresee the impact of removing, adding or changing a specific identification technology by simply modifying the model. This allows the answering of questions such as *¿ Can RFID be used by library members to borrow books by pointing at them?*

Figure 8 shows a Parkour instance model where the editor has detected some inconsistencies. The editor verifies the constraints each time the model is saved (or by user demand). The errors and warnings detected are integrated into the standard error view provided by Eclipse. In this case, the model contains one error and one warning. The error is produced because the deployment unit is supporting the *borrow book* task by means of RFID while the book was only tagged with a paper-based identifier in the example. Thus, the task cannot be supported with the technology defined.

6 Conclusions

This paper has presented a design method to specify physical mobile workflows. It has illustrated how to capture the requirements for the physical-virtual linkage in a gradual manner by means of modelling techniques. The use of models helped to centralize the knowledge about the workflow and organize it in a way that is easy to handle by designers (e.g. work with technology independent concepts, detect inconsistencies, etc.).

The design method provided relies on proven techniques and frameworks for business process modelling, implicit interaction design and physical interaction patterns. The tool support provided for Parkour enabled the anticipation of errors in the workflow described. However, the use of the tools required some advanced knowledge in modelling techniques.

This work is part of a comprehensive proposal that includes a framework called Presto and code generators to fully support the development method presented in Figure 1. Presto defines an architecture that fits with the specific needs of physical mobile workflows. This architecture has been defined in a technological independent manner. In this way, components in this architecture can be mapped to different technological platforms. Our approach provides support to automatically obtain an architecture design for a given workflow specification. Code generation is also provided to produce, from a model based in our abstract architecture, the implementation assets that are required for the technological infrastructure in a given technology platform. In particular we have defined code generation support for the Android platform that frees developers from dealing with Android-specific components and focus on implementing the required business logic in Java.

Acknowledgments

This work has been developed with the support of MICINN under the project EVERYWARE (TIN2010-18011).

References

[1] Neil Gershenfeld, Raffi Krikorian. The Internet of Things. Scientific American, 291(4):pp. 46-51, 2004.

[2] Martin Strassner, Thomas Schoch. Todays Impact of Ubiquitous Computing on Business Processes. En Friedemann Mattern, Mahmoud Nagshineh editors, Short Paper Proc. of International Conference on Pervasive Computing, pp. 62-74, 2002.

[3] Jean Marie Favre. Foundations of Model (Driven) (Reverse) Engineering – Episode I: Stories of the Fidus Papyrus and of the Solarus. En Jean Bézivin, Reiko Heckel editors, Post-Proceedings of Dagsthul Seminar on Model Driven Reverse Engineering, Germany, 2004.

[4] Marc Langheinrich, Vlad Coroama, Juergen Bohn, Michael Rohs. As we may live - Real-world implications of ubiquitous computing. Technical report, Swiss Federal Institute of Technology, 2002.

[5] Elgar Fleisch. Business perspectives on Ubiquitous Computing. Technical report, M-Lab Working, Switzerland, 2001.

[6] Uwe Sandner, Jan Marco Leimeister, Helmut Krcmar. Business Potentials of Ubiquitous Computing. En Eva-Maria Kern et al. editors, Managing Development and Application of Digital Technologies, pp. 277-292. Springer, 2006.

[7] Andrew E. Fano, Anatole Gershman. The future of business services in the age of ubiquitous computing. Commun. ACM, 45(12):pp. 83-87, 2002.

[8] Elgar Fleisch, Christian Tellkamp. The Challenge of Identifying value-creating Ubiquitous Computing applications. Proc. of the Workshop on Ubiquitous Commerce, Ubicomp, Seattle, USA, 2003.

[9] Sanjay E. Sarma. Integrating RFID. ACM Queue, 2(7):pp. 50-57, 2004.

[10] George Roussos, Juha Tuominen, Leda Koukara, Olli Seppala, Panos Kourouthanasis, George M. Giaglis,

Jeroen Frissaer. A Case Study in Pervasive Retail. En Marisa-S. Viveros et al. editors, Proc. of the Workshop Mobile Commerce, pp. 90–94. ACM, 2002.
[11] Jeff Kramer. Is abstraction the key to computing? Commun. ACM, 50(4):pp. 36-42, 2007.
[12] Douglas C Schmidt. Model Driven Engineering. Computer. Guest Editor's Introduction., 39(2):pp. 25-31, 2006.
[13] Pau Giner, Carlos Cetina, Joan Fons, Vicente Pelechano. Developing mobile workflow support in the internet of things. IEEE Pervasive Computing, 9(2):pp. 18-26, 2010.
[14] Wendy Ju, Larry Leifer. The Design of Implicit Interactions: Making Interactive Systems Less Obnoxious. Design Issues, 24(3):pp. 72-84, 2008.
[15] Liviu Iftode, Cristian Borcea, Nishkam Ravi, Porlin Kang, Peng Zhou. Smart Phone: An Embedded System for Universal Interactions. Proc. of Future Trends of Distributed Computing Systems, pp. 88-94. IEEE Computer Society, 2004.
[16] Enrico Rukzio, Karin Leichtenstern, Victor Callaghan, Paul Holleis, Albrecht Schmidt, Jeannette Shiaw-Yuan Chin. An Experimental Comparison of Physical Mobile Interaction Techniques: Touching, Pointing and Scanning. En Paul Dourish, Adrian Friday editors, Proc. of International Conference on Ubiquitious Computing, UbiComp, volume 4206 of Lecture Notes in Computer Science, pp. 87-104. Springer, 2006.
[17] Gregor Broll, Markus Haarländer, Massimo Paolucci, Matthias Wagner, Enrico Rukzio, Albrecht Schmidt. Collect&Drop: A Technique for Multi-Tag Interaction with Real World Objects and Information. En Emile H. L. Aarts et al. editors, Proc. of the European Conference on Ambient Intelligence, AmI, volume 5355 of Lecture Notes in Computer Science, pp. 175-191. Springer, 2008.
[18] Rafael Ballagas, Michael Rohs, Jennifer G. Sheridan, Jan Borchers. BYOD: Bring Your Own Device. Proc. of the Workshop on Ubiquitous Display Environments, Ubicomp, Tokyo, Japan, 2004.
[19] Pau Giner, Victoria Torres, Vicente Pelechano. Modeling Mobile Business Processes for the Internet of Things. Proc. of the Workshop on Web Services and SOA, JSWEB, Madrid, Spain, 2009.
[20] Enrico Rukzio, Gregor Broll, Karin Leichtenstern, Albrecht Schmidt. Mobile Interaction with the Real World: An Evaluation and Comparison of Physical Mobile Interaction Techniques. In Bernt Schiele et al. editors, Proc. of the European Conference on Ambient Intelligence, AmI, volume 4794 of Lecture Notes in Computer Science, pp. 1-18. Springer, 2007.
[21] Roy Want, Kenneth P. Fishkin, Anuj Gujar, Beverly L. Harrison. Bridging Physical and Virtual Worlds with Electronic Tags. Proc. of the SIGCHI Conference on Human Factors in Computer Systems, CHI, pp. 370-377, 1999.