Monograph of next issue (April 2011)

"Software Engineering for e-Learning Projects"

(The full schedule of CEPIS UPGRADE is available at our website)
Internet of Things

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.

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-
Section 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 supporting 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 on 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"
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
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 Engineering (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 3: Obtrusiveness Level defined for Each Task in the Smart Library Scenario.](image)

![Figure 4: Different Interactions with Physical Elements orchestrated in a Workflow Model.](image)
The method is based on the Business Process Management (BPM) initiative, that promotes the continuous re-engineering of business processes

There is not a universal interaction technique that is well suited for any situation
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-
This work is part of a comprehensive proposal that includes a framework called Presto.

The ‘Medium’ concept is a technological-independent mechanism for describing identification requirements.
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-
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 an 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.

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 into 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 mobile physical 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

Internet of Things


