MODELING WITH BPMN AND CHORDA: A TOP-DOWN, DATA-DRIVEN METHODOLOGY AND TOOL

Andrea Catalano, Matteo Magnani and Danilo Montesi
Department of Computer Science, University of Bologna, Mura A. Zamboni 7, 40127 Bologna, Italy

Keywords: BPMN, Methodology, Data, Design tool.

Abstract: In this poster paper we present a methodology, named chorda, for the modeling of business processes with BPMN. Our methodology focuses on the peculiar features of this notation: its ability to illustrate different levels of abstraction, its support for both orchestration and choreography, and the representation of data flows. In particular, this last feature has been extended to allow a better mapping of real processes, where data often plays a fundamental role. To evaluate and tune the methodology, we have developed a tool supporting it.

1 INTRODUCTION

BPMN is the OMG standard notation for the representation of business processes (BPMN, 2009; Wohed et al., 2006; White and Miers, 2008). Even if BPMN constructs are very intuitive, business process diagrams can be difficult to design without any associated methodology — much like building a miniature ship model with glue and screws, but without instructions. In this poster paper we introduce a methodology, named chorda, for the translation of informal process descriptions into BPMN diagrams.

Why do we need a specific methodology for BPMN? This lies in the very nature of this language, which enables the representation of three important aspects of business processes, making it a unique modeling tool. First, we can represent a *choreography* of processes, i.e., how different processes interact with each other to fulfill a common objective. Second, we can represent the *orchestration* of a process, i.e., its internal organization into sequences of activities. Third, BPMN allows the representation of the same information at different levels of detail, using sub-processes — this being fundamental to provide different views of the same process to people with different roles, like top managers and technical staff.

Basically, BPMN allows the representation of complex scenarios because it can include many different aspects into a single diagram: *choreography*, *orchestration*, and *data*, at several different *levels of abstraction* (BPMN, 2009; Barros et al., 2006). Therefore, the main idea behind our approach is that the initial requirements can be split into different classes, that can be specifically addressed during well separated and thus simplified modeling steps. As we have illustrated in Figure 1, after a typical pre-processing of the available informal requirements aimed at removing ambiguities and producing a dictionary with all definitions and synonyms, we split them into atomic statements referring to one of the following aspects: (D) data, (I) interactions between different participants, and (L) local work of a single participant. At this point, each class can be processed independently from the others.

To the best of our knowledge, this is the first presentation of a methodology for BPMN modeling. Obviously, it is based on best practices taken from existing data and software modeling methodologies like the IBM Rational Unified Process (www-01.ibm.com/software/awdtools/rup), the IDEF methods (http://www.idef.com), data modeling using ER diagrams and Object Process Modeling (http://www.objectprocess.org). The POEM (Process Oriented Enterprise Modelling) methodology uses BPMN as one of several basic diagram types. Although we are not aware of existing presentations of this methodology, still under development, it seems to have a wider scope than our proposal, covering several additional aspects of an enterprise, while we present specific results regarding the BPMN notation.

1.1 Extended Notation and Design Tool

Before introducing the methodology, it is worth mentioning the extensions for the representation of data

Catalano A., Magnani M. and Montesi D. (2009).
MODELING WITH BPMN AND CHORDA: A TOP-DOWN, DATA-DRIVEN METHODOLOGY AND TOOL.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 389-392
DOI: 10.5220/0002003603890392
Copyright © SciTePress

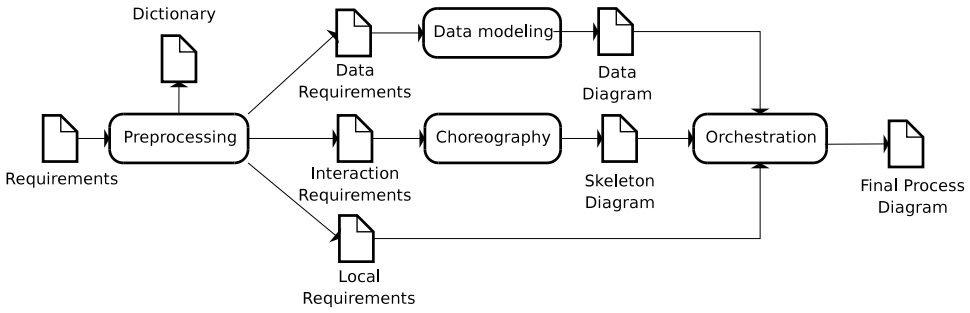

Figure 1: A diagram summarizing the proposed methodology.

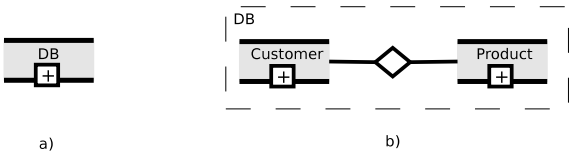

Figure 2: A store (a), which has been expanded in (b) to show it at a lower abstraction level.

that we have implemented in our tool. In our opinion data manipulation is usually the most important aspect and objective of a business process (Sadiq et al., 2004; Russell et al., 2005). In addition, in the BPMN specification data objects have been defined as a predefined artifact of BPMN (together with comments and associations), because the BPMN specification itself declares that in some diagrams they represent *the most important information to be modeled*. However, the current version of BPMN has been mainly developed to represent activities and events, and at the moment the only visual construct used to represent data is the *data flow*, a small rectangle with a folded corner that can be added on top of control flows.

In our tool, we have considered two main additional data representation capabilities. The first consists in the possibility of associating a data object to an XML document or XML Schema, to provide a way to represent complex and structured objects inside a diagram. The second is a new visual construct, the *store*, that has been taken from Data Flow Diagrams (DFD) and can be used to show the full life cycle of data: where it is stored, when it is retrieved to be manipulated and where it is stored again after their manipulation (Gane and Sarson, 1977; Yourdon, 2006). Also stores can be structured, to provide different abstraction levels consistently with the philosophy of BPMN, and to represent expanded stores we use an ER-like notation (Chen, 1976). The visual construct used to represent a store is illustrated in Figure 2.

The design tool developed to support our methodology is a plug-in for the Microsoft Visio application (`http://office.microsoft.com/visio`). The tool extends the POEM stencils (`http://bpmnpop.sourceforge.net`), and can be used to design BPMN diagrams, to annotate them with additional attributes (like the cost of activities) and to generate their XPDL representations (Magnani and Montesi, 2007; XPDL, 2005). A beta time-limited version can be downloaded at `http://www.easybpmn.com`.

2 THE METHODOLOGY

2.1 Preliminaries

We shall assume to start from a single text file describing the requirements. This file can be processed as usual, by clarifying unclear sentences, identifying synonyms, replacing them with consistent terms, and building a technical dictionary. Then, we may identify all the **data** objects mentioned in the requirements, and all the **participants**.

At this point, we can split the requirements into small atomic statements, and assign each statement to one of the three following classes:

- **Data** requirement (the statement concerns only data).
- **Interaction** requirement (the statement refers to two participants).
- **Local** requirement (the statement refers to one single participant).

2.2 Data Modeling

We can now start modeling data requirements (D). Data is a primary component of real business processes and will thus drive all the modeling activities. In fact, a process is basically a sequence of activities aimed at modifying some data or objects, and the production of new data is the way in which business

processes generate value — for example, many business processes are used to transform raw materials into final products. As BPMN offers a limited support to the modeling of data, being more activity/event-oriented, these requirements could be addressed by existing data modeling approaches and attached to the diagrams as complementary documentation. However, as data is so important in many processes and we do not want to lose the connection between the data and the tasks manipulating it, in our tool we use the aformentioned extended version of BPMN, obtained by including some features of ER and Data Flow Diagrams into the notation. After this modeling step, we will have identified a list of all the data/objects referenced in the requirements — the next step will be the definition of how they are exchanged among different participants.

2.3 Interaction Modeling

Data flows will then be used to generate a so called *skeleton diagram* representing how data is exchanged between the participants to produce the final products of the process. Basically, during this step we focus on choreography, i.e., we identify all the participants and their interactions (I). Each participant is represented using a BPMN pool, and we draw a message flow between two of them for each requirement. In this way, after having identified all interaction requirements we can automatically build a skeleton diagram, as we have exemplified in Figure 3(a): each interaction between participants A and B corresponds to a `Send Data` activity in A, a `Receive Data` activity in B, and a `Process Data` sub-process in B, indicating that the received data will be later manipulated — this will be expanded during the next step of the methodology.

Choreography has been studied extensively both in academia (in the area of process algebras) and industry (with the proposal of standard languages): in this paper we do not deal with the representation or verification of choreographies, but with the formalization of existing informal descriptions of a choreography — automated verification tools will obviously be of great utility to check the designed diagram, but this is an orthogonal problem with respect to the scope of this paper.

2.4 Local Modeling

Now, we will have a skeleton diagram with all the participants (pools) and all messages exchanged between them, representing the complete (abstract) data paths used to produce the final outcome of the process, be it a document, a product, or any other valuable object. For each exchanged message, we will also have a sub-process (the rectangle with a small *plus* sign represented in Figure 3(a)) hiding the local activities performed by the participant to manipulate the data. Therefore, we can focus on the remaining requirements (L) describing these activities. This modeling step can be performed in a top-down way, following the philosophy behind BPMN which uses abstraction levels as a basic tool to provide different views on the same process. Therefore, L-statements will be structured hierarchically (as trees of requirements, or nested item lists) and modeled recursively. For example, consider the following statements:

1. The CIO shall store a copy of the report into the archive, and
2. prepare an IT plan as follows:
   (a) collect information on all the systems currently used in the company,
   (b) then evaluate their life cycle state (trailing, leading or bleeding edge)

These can be modeled as illustrated in Figure 3(b), where we have also used one of our data modeling extensions: the *store* (archive). Later, the `Prepare IT plan` sub-process can be expanded including the statements describing this activity (items 2.a and 2.b, in this example), and so on recursively.

3 SUMMARY

The methodology proposed in this paper is composed of the following main steps:

1. Pre-process the initial requirements (remove ambiguities, update and refine unclear sentences and generate a dictionary with explanations of the technical terms and indications of synonyms).
2. Split the requirements into elementary statements.
3. Identify the participants and the exchanged data.
4. Separate data (D) statements from activity statements and model the data.
5. Mark each remaining statement as a local (L) activity (orchestration) or an interaction (I) among participants (choreography).
6. Draw the skeleton of the process, modeling interaction activities.
7. Tree-structure local activities, associate them to sub-processes in the skeleton diagram, and model them in a top-down way by increasing the level of detail at each iteration (if necessary).

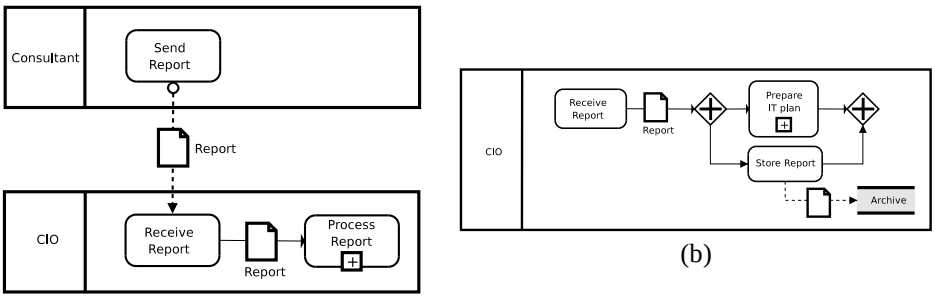

Figure 3: (a) The skeleton diagram corresponding to a statement: *The report shall, as soon as it has been established, be transmitted by the external* consultant *to the* CIO, and (b) the modeling of local (L) requirements, in a top-down fashion.

In this way, part of the modeling activity can be semi-automated, and the definition of the orchestration inside each pool can be performed by focusing on small portions of the initial requirements. In addition, each statement can be easily associated to a specific part of the final diagram, and it can be then verified if the diagram is *complete*, i.e., if it models all the initial requirements.

REFERENCES

Barros, A., Dumas, M., and Oaks, P. (2006). Standards for web service choreography and orchestration: Status and perspectives. In Springer, editor, *Workshop on Web Service Choreography and Orchestration for Business Process Management*, volume 3812 of *LNCS*.

BPMN (2009). Business process modeling notation specification.

Chen, P. (1976). The Entity-Relationship model — toward a unified view of data. *Transactions on Database Systems*, 1(1):9–36.

Gane, C. and Sarson, T. (1977). *Structured Systems Analysis: Tools and Techniques*. IST, Inc.

Magnani, M. and Montesi, D. (2007). BPMN: How much does it cost? an incremental approach. In *Business Process Management (BPM), 5th International Conference*, volume 4714 of *Lecture Notes in Computer Science*, pages 80–87. Springer.

Russell, N., ter Hofstede, A. H. M., Edmond, D., and van der Aalst, W. M. P. (2005). Workflow data patterns: Identification, representation and tool support. In *24th International Conference on Conceptual Modeling*, volume 3716 of *Lecture Notes in Computer Science*, pages 353–368. Springer.

Sadiq, S. W., Orlowska, M. E., Sadiq, W., and Foulger, C. (2004). Data flow and validation in workflow modelling. In *Fifteenth Australasian Database Conference*, volume 27 of *CRPIT*, pages 207–214.

White, S. A. and Miers, D. (2008). *BPMN Modeling and Reference Guide*. Future Strategies Inc.

Wohed, P., van der Aalst, W. M. P., Dumas, M., ter Hofstede, A. H. M., and Russell, N. (2006). On the suitability of BPMN for business process modelling. In *Business Process Management, 4th International Conference*, volume 4102 of *Lecture Notes in Computer Science*, pages 161–176. Springer.

XPDL (2005). Process definition interface – XML process definition language (XPDL) specification.

Yourdon, N. E. (2006). Just enough structured analysis. ch. 9: Dataflow diagrams. www.yourdon.com.