

Business Process Modeling Languages and Tools

James McCutcheon

Murdoch University, Australia

Nik Thompson

Murdoch University, Australia

INTRODUCTION

The formal discipline of Business Process Modeling (BPM) is relatively new compared to the well-established understanding of business activities and formal repeatable work processes. There is some evidence of informal BPM within organizations during the latter half of the 20th century, but concerted attempts to formalize the activities and products of BPM did not occur until early in the 21st century (Aguilar-Savén, 2004). This article will explore the different tools and techniques that have been standardized and adopted by industry and some of the software packages available to assist in the execution and management of these activities. Where appropriate, commentary will be added to offer explanations of market trends and provide insights into decision making and common usage behaviours. Finally, the article will present a discussion of future directions in this domain, followed by some concluding statements.

BACKGROUND

A business process may be defined as a set of interrelated tasks, roles, and resources working in concert to achieve a business objective or goal (Dumas, La Rosa, Mendling, Reijers, 2013). Business Process Modeling is the activity and discipline of transferring tacit and explicit business knowledge and experience into formalized descriptions, specifications, and diagrams that focus on process structure and interaction, rather than technical details (Weske, 2007). These models are expressed as meta-models and are combined with a notation language to consistently describe similar business constructs (Weske, 2007).

BPM has two primary uses: systems development and business process management (Havey, 2009). As the terms Business Process Modeling or Business Process Management may share an acronym there is potential for confusion surrounding the two concepts. This is further confounded by the fact that some modeling tools are contained within Business Process Management software suites. For the purposes of this article, the abbreviation BPM will refer strictly to the topic of Business Process Modeling.

Business Process Management, and by extension BPM, has evolved from the analysis of Workflow Management practices. Workflow Management was the subject of many papers in the 90's (van der Aalst, ter Hofstede & Weske, 2003) and was primarily focused on the automation of structured processes (Weske, 2007). Accordingly, Workflow Management struggled to model behavior, which while difficult, is essential for effectively and accurately modeling business processes (Smith, 2003). This influenced BPM pioneers to focus on bridging the gap between Workflow Management systems and business processes (Weske, 2007).

The primary advantages of BPM are clarity and discovery. BPM identifies, defines, and enumerates steps, inputs, outputs, requisite resources, and expertise so that a business process can be understood and replicated by third parties or in software for automation (Recker, 2006). While a thorough understanding of the process being modeled is desirable before embarking on formalization, a more incremental approach is realistic as the modeling process itself will help to identify shortcomings, misunderstanding and ambiguities in the current documentation and the modeler's process discovery. Dumas et al. (2013) call BPM a "boundary spanning" field that allows stakeholders from a breadth of disciplines to communicate with a shared language.

DOI: 10.4018/978-1-4666-5888-2.ch694

BPM may also be practiced informally and this is likely in organizations that have not been introduced to formal techniques but have independently identified the advantages of BPM activities. Formalized languages, techniques and notations, however, hold the benefit of being standardized: they are understandable by many individuals in the industry, not just those inducted into the ad hoc standard of a single organization, and minimize potential for misinterpretation (van der Aalst, et al., 2003). They have also undergone rigorous analysis and testing by various standards bodies, such as the Object Management Group (OMG).

PROMINENT MODELING LANGUAGES

There are a number of modeling languages that have been produced and standardized to simply and accurately represent business processes. Unified Modeling Language (UML) and Business Process Model and Notation (BPMN; previously called Business Process Modeling Notation) are the two most widely used languages used to define and describe processes in a structured manner (Portela, 2012, Geambasu, 2012). The Activity Diagram is considered to be the most suited of the UML components for BPM, especially with enhancements introduced in UML 2.0 (Geambasu, 2012). UML and BPMN are reported to be used by 18% and 72% of surveyed IT professionals respectively (Harmon & Wolf, 2011). In the same study, the 559 self-selected respondents identified Event-Driven Process Chain (EPC, 8%), Business Process Execution Language (BPEL, 6%), and XML Process Definition Language (XPDL, 4%) as minor languages.

Whilst respondents were able to select multiple languages, only a third, at most, chose to do so. The important conclusion is that up to 66% of respondents were satisfied with a single language. Even if every respondent who chose more than one answer (at most 33%) used BPMN, that still leaves 39% of all respondents using BPMN alone, more than twice that of UML in second place. The fact that BPMN is the youngest of these languages suggests that an exceptional fitness for purpose has propelled BPMN into a position of dominance in the BPM market.

Business Process Model and Notation

S

BPMN is the newest of the widely used modeling languages, with version 1.0 released in 2004 by the Business Process Management Initiative (BPMI). In 2011, BPMN 2.0 was released (Object Management Group, 2011) after a 2009 beta version release. BPMN shares a lot of similarities with UML Activity Diagrams and non-standardized flow charts, both in function and appearance. This only holds true, however, for very simple processes as BPMN offers a much larger range of symbols compared to Activity Diagrams. Some of this new functionality includes timers, designed to halt the process until a specific temporal event occurs, a richer set of gateways (decisions) that allow for more complex logic (like XOR and AND), and symbols to represent looping tasks or those which will have multiple instances running in parallel.

These additional symbols allow the meta-model to be expanded to allow much more complex process representation over a much more arbitrary scope of industries. Using timers as an example, there is no equivalent in a UML Activity Diagram as in software engineering it is often assumed that processes will push towards completion as soon as possible. Within business operations, however, processes may have real world constraints, such as mandatory waiting periods on contracts.

The improvements implemented in version 2.0 of the standard are dominated by the inclusion of a meta-model allowing serialization to XML. This additional meta-model allows for two new important capabilities. The first is the ability to use XML as an interchange language for porting BPMN diagrams from one tool to another, without a loss of information. Previously this had to be achieved through another language, typically XPDL. The second new capability is the ability to directly execute BPMN models in a process engine, such as Activiti or jBPM. As with using XML for interchange, process execution previously had to be handled by another language, such as BPEL, though this method suffered from a lack of semantics included in BPMN 1.0. The new meta-model adds the required semantic detail to effect useful process execution.

6 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/business-process-modeling-languages-and-tools/112403

Related Content

A System to Match Behaviors and Performance of Learners From User-Object Interactions: Model and Architecture

José Guillermo Hernández-Calderón, Edgard Benítez-Guerrero, José Rafael Rojano-Cáceres and Carmen Mezura-Godoy (2019). *International Journal of Information Technologies and Systems Approach* (pp. 82-103).

www.irma-international.org/article/a-system-to-match-behaviors-and-performance-of-learners-from-user-object-interactions/230306

Studying Information Infrastructures

Petter Nielsen (2012). *Phenomenology, Organizational Politics, and IT Design: The Social Study of Information Systems* (pp. 143-158).

www.irma-international.org/chapter/studying-information-infrastructures/64682

A CSP-Based Approach for Managing the Dynamic Reconfiguration of Software Architecture

Abdelfetah Saadi, Youcef Hammal and Mourad Chabane Oussalah (2021). *International Journal of Information Technologies and Systems Approach* (pp. 156-173).

www.irma-international.org/article/a-csp-based-approach-for-managing-the-dynamic-reconfiguration-of-software-architecture/272764

Improving Dependability of Robotics Systems

Nidhal Mahmud (2018). *Encyclopedia of Information Science and Technology, Fourth Edition* (pp. 6847-6858).

www.irma-international.org/chapter/improving-dependability-of-robotics-systems/184381

FLANN + BHO: A Novel Approach for Handling Nonlinearity in System Identification

Bighnaraj Naik, Janmenjoy Nayak and H.S. Behera (2018). *International Journal of Rough Sets and Data Analysis* (pp. 13-33).

www.irma-international.org/article/flann--bho/190888