Teaching BPM Fundamentals: A Project-Based Hands-On Course for Process-Driven Systems Modeling and Development

Andrea Delgado(✉) and Daniel Calegari

Instituto de Computación, Facultad de Ingeniería, Universidad de la República,
Montevideo 11300, Uruguay
{adelgado,dcalegar}@fing.edu.uy

Abstract. Computer Science curricula usually develop software modeling, design, and implementation skills involving standard languages, best practices, and different languages and architectures. However, they do not often involve the specific modeling, design, and implementation of business processes (BPs) using Business Process Management Systems, which require making BPs explicit (e.g., specified in BPMN 2.0) and driving the system development and execution by such models. A specific theoretical, conceptual, and technological background is needed in this context. This paper presents a project-based, hands-on approach to modeling and developing process-driven systems, which we integrated into our Computer Science curricula. We present the experience and highlight lessons learned about essential elements for students learning.

Keywords: Business Process Management (BPM) · Process-Aware Information Systems (PAIS) · project-based teaching · hands-on approach

1 Introduction

Business Process Management (BPM) [1–3] provides support for organizations to focus their daily operation on their Business Process (BPs). Organizations have increasingly adopted BPM in the last two decades. It is promoted by conceptual and technological support, e.g., the OMG standard Business Process Modeling and Notation (BPMN 2.0) [4], and BPM systems (BPMS) [5] providing support for the complete Business Process lifecycle. BPMS as Process-Aware Information Sytems (PAIS) [6] integrate modules supporting process-driven systems, from modeling, configuration, implementation, enactment, and evaluation to improvement. Although BPMS has some differences in implementation, the functionalities offered are mainly conceptually the same [7,8].

A specific theoretical, conceptual, and technological background on BPs is needed to guide this development. Nevertheless, it is only sometimes included in

J. Köpke et al. (Eds.): BPM 2023, LNBIP 491, pp. 211–221, 2023.
https://doi.org/10.1007/978-3-031-43433-4_14

Computer Science curricula with the essential software systems modeling, design, and implementation. In 2013, we integrated this vision within the undergraduate and postgraduate Computer Science degrees at Universidad de la República in Uruguay. Computer Science is a five-year degree of 450 credits (1 credit equals 15 h of student effort, equivalent to 270 ECTS) organized in semesters. It comprises three basic first years with mandatory courses in mathematics, physics, logic, programming, operating systems, computer architecture, language theory, operations research, numerical methods, computer networks, and databases. The fourth year contains some mandatory courses on software engineering, functional/logic programming, and the final one-year project to graduate. It also offers optional courses covering different areas such as security, advanced computer networks, model-driven engineering, software testing, etc.

This paper presents a project-based, hands-on approach to modeling and developing BPM systems. It consists of a 150 h (6 ECTS credits) optional course of the undergraduate Computer Science degree in the fourth and fifth year, also offered as part of the postgraduate academic Computer Science degree. It is focused on the BPMN 2.0 language for modeling and enacting BPs within an open-source BPMS, following a systematic approach and using best practices for modeling, design, and implementation. We present the course and its evaluation using data from its eight editions. We use the experience to discuss lessons learned, highlighting critical elements for students learning.

The rest of the paper is organized as follows. In Sect. 2, we introduce the course setup, and in Sect. 3, we present our teaching experience. Section 4 discusses results and lessons learned. In Sect. 5, we present related work. Finally, we present the conclusions in Sect. 6.

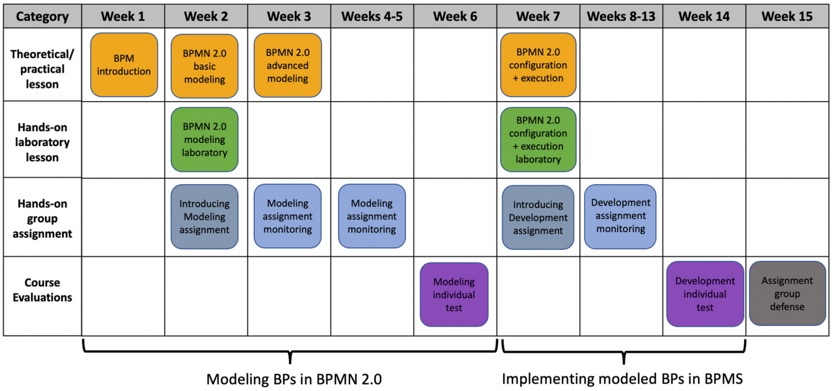
2 BPM Course Setup

The BPM course is taught in one semester, i.e., all activities are distributed within 15 weeks consisting of classes, laboratories, assignments, and evaluations. We defined some general learning outcomes based on Bloom's taxonomy:

- Acquire basic concepts, techniques, and methodologies to support the BPs lifecycle [**Knowledge**].
- Acquire knowledge on BPs modeling using the standard language BPMN 2.0 and best practices [**Comprehension**].
- Generate experience in BPs modeling and implementation using an open source BPMS platform [**Application**].
- Know and experiment with different BPMS platforms that support the BPs lifecycle with different approaches [**Analysis**].

As depicted in the course schedule of Fig. 1, we divide the course into two main parts with four activity groups: theoretical/practical lessons, hands-on laboratory, hands-on group assignments, and course evaluations. Example materials (Spanish) are here[1]. Weekly lessons and evaluations (each block) are 2-hours

[1] Materials: https://www.fing.edu.uy/owncloud/index.php/s/tALdL4dY976v08r.



Category	Week 1	Week 2	Week 3	Weeks 4-5	Week 6	Week 7	Weeks 8-13	Week 14	Week 15
Theoretical/ practical lesson	BPM introduction	BPMN 2.0 basic modeling	BPMN 2.0 advanced modeling			BPMN 2.0 configuration + execution			
Hands-on laboratory lesson		BPMN 2.0 modeling laboratory				BPMN 2.0 configuration + execution laboratory			
Hands-on group assignment		Introducing Modeling assignment	Modeling assignment monitoring	Modeling assignment monitoring		Introducing Development assignment	Development assignment monitoring		
Course Evaluations					Modeling individual test			Development individual test	Assignment group defense

Fig. 1. BPM workshop course typical schedule with lessons categories

activities. The first part (5 weeks plus evaluation) focuses on BPMN 2.0 modeling and best practices. The second part (7 weeks plus evaluation) focuses on implementing the previously modeled process in an open-source BPMS. There is also a general assessment in the form of a group defense of the hands-on work.

Theoretical/practical lessons Since the focus of the course is mostly practical, we introduce key concepts in four weeks of theory/practical lessons for students to get the foundations of BPM knowledge and to perform modeling and development exercises, including:

BPM introduction: we define key concepts of BPM, BP lifecycle (modeling, design, configuration, enactment, evaluation, improvement), BP model, types of BP (collaborative, choreography, orchestration), BP cases, and BPMS.

BPMN 2.0 basic modeling: we introduce the BPMN 2.0 standard and its key elements: activities (tasks and types, sub-processes) and markers (loop, multi-instance, etc.), gateways (AND, XOR, OR), events (start, intermediate, end) and types (message, time, etc.), swimlanes (pools, lanes).

BPMN 2.0 advanced modeling: we introduce and discuss BP modeling best practices: seven process modeling guidelines (7PMG) [9], workflow patterns [10], and process re-design heuristics [11].

BPMN 2.0 configuration & execution: we present key concepts for BPs implementation based on BPMN 2.0 models, which involve designing and developing a software layer to support the BP model execution or high-level elements, depending on the developers-oriented or not focus of the BPMS.

In the practical lessons we delve into most common modeling errors (e.g. granularity of activities), BPMN 2.0 constructs, application of workflow patterns and 7PMG, with a general discussion and key concepts to take away.

Hands-on laboratory The theoretical/practical lessons are supported by two hands-on laboratory lessons using BPMS platforms, allowing students to fix ideas and get involved with the tools firsthand. For the modeling laboratory, we introduce BPMS modeling modules from Activiti[2], Camunda[3], Bonita[4] and Bizagi[5], being the first three open source and multi-platform, and the last one freeware for windows. The first two exemplify a developer-oriented approach, and the last two have a non-developer-oriented focus. For the configuration and execution laboratory, we mainly work with Activiti BPMS since it is the BPMS platform we use for the hands-on group assignments work, due to its developer-oriented approach with Java, easy deployment and execution in a web server. We introduce the implementation in Activiti which involves developing: (a) a Java layer for the BP model with classes invoked by the process engine at runtime, and (b) User task forms to be associated with the user tasks and be presented to the user at runtime. For each task type, we present implementation examples, e.g. service tasks invoking web services with a WSDL-based generated client.

Hands-on group assignments For the group assignment, which is the primary learning element of the course, we follow a project-based hands-on approach, in which groups of 3 or 4 students work together on a project we provide for modeling and implementing a process-driven system in Activiti BPMS. Each year we select two to three real BPs, providing students with a simplified version to work with. The course is online in our Moodle EVA[6] platform, and we use Gitlab[7] as code repository for each group. The group assignment consists of two parts: the first focuses on modeling the BP using BPMN 2.0, and the second on implementing and enacting the BP model in the Activity BPMS platform. We expect students to apply the theoretical/practical knowledge showing the achievement of the defined learning objectives. We include several technological requirements regarding integration aspects: invoking a Web Service from a service task, managing PDF documents with an Electronic Document Management System (EDMS), and use of the Activiti REST API to query process execution.

Course evaluations Regarding the course evaluations, we propose two individual tests, one for each part of the hands-on assignments: one when the BP modeling part ends and the second when ending the development part. Also, the group assignment has a defense at the end of the course. In the individual tests, we ask a few theoretical questions and provide a practical exercise of modeling/development to be solved, similar to the ones we solved in the practical lessons. In the defense, students execute the BP system in Activiti BPMS following a guiding script we provide, with selected data for executing different scenarios, qualifying specific elements of the BP modeling and implementation.

[2] https://www.activiti.org/.
[3] https://camunda.com/.
[4] https://www.bonitasoft.com/.
[5] https://www.bizagi.com/.
[6] https://eva.fing.edu.uy/course/view.php?id=423.
[7] https://gitlab.fing.edu.uy/groups/tbpm.

Table 1. Domains and BPs selected for the group assignments

Year	BPs source	Selected BPs
2013–14	University	Teaching position call, Teaching position renewal
2015–16	University	Academic mobility, Agreement management
2017–18	Logistics	Material request, Order dispatch
2019	e-Government	Social benefits allocation, Born alive registration
2021	e-Health	COVID-19 testing, COVID-19 contacts traceability

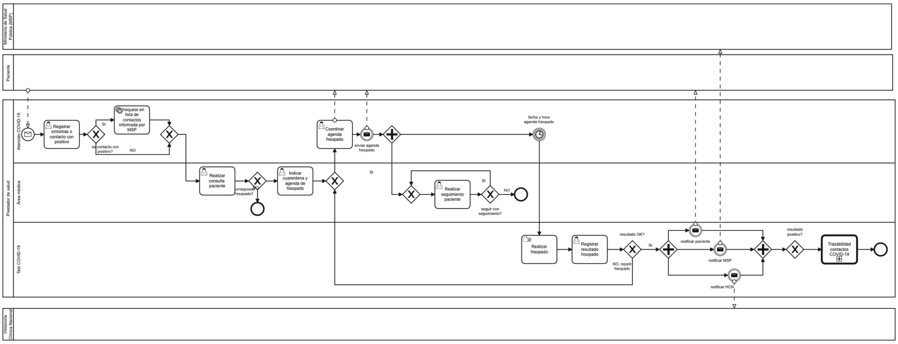

Ministerio de Salud Pública (MSP)
Paciente
Prestador de salud
Atención COVID-19
Área médica
Test COVID-19
Histororia Clínica Nacional
Registrar síntomas o contacto con positivo
SI
Chequear en lista de contactos informada por MSP
es contacto con positivo?
NO
Realizar consulta paciente
corresponde hisopado?
NO
Indicar cuarentena y agenda de hisopado
Coordinar agenda hisopado
enviar agenda hisopado
SI
fecha y hora agenda hisopado
Realizar seguimiento paciente
SI
NO
seguir con seguimiento?
Realizar hisopado
Registrar resultado hisopado
resultado OK?
SI
NO, repetir hisopado
notificar paciente
notificar MSP
notificar HCN
resultado positivo?
Trazabilidad contactos COVID-19

Fig. 2. “COVID-19 testing” e-Health BP excerpt

3 BPM Course Experience

We have taught the course from 2013 to 2019 every year in an in-person format, with hands-on laboratories in computer rooms at the Engineering School; in 2020 and 2022, it was not taught, and in 2021, it was entirely in a virtual format. As mentioned before, we select real BPs for each edition to use as a basis for the project-based group assignment. These BPs come from research projects or collaboration projects, mostly with our country's public organizations, even internal projects from our university that we have carried out. Table 1 presents the BPs used for each year's edition. Figure 2 shows the ”COVID-19 testing” BP as an example of their complexity. Other examples in the provided material.

Students face challenges in the assignments associated with our defined learning objectives. Since the course is taught in a Computer Science degree, they have an important background in programming (four prior specific courses and hands-on workshops and laboratories in different programming languages). However, they have only a couple of courses with modeling aspects, mainly with UML and ER models, but none in BPs or BPMN, so it could be expected for them to find more difficult the modeling part. During the course evaluation before 2019, many students expressed that the models were too large and complex. Thus, for the 2019 and 2021 editions, we simplified the control flow and path complexity of the selected BPs, maintaining key elements to learn.

Table 2. Modeling challenging concepts for students

Challenge	Model performed
Control flow collaboration	Message flows as part of the control flow
Control flow orchestration	Missing end events, throw and capture events misused
Participants and roles	Definition of organizational participants/roles as lanes
Granularity of activities	Activities too coarse grained defined
Activity names	Activities with no action names
Type of gateways	Mixed gateway types for diverging and converging flows
Use of gateways	Same gateway for converging and diverging flows
Use of XOR gateways	Missing conditions and corresponding paths

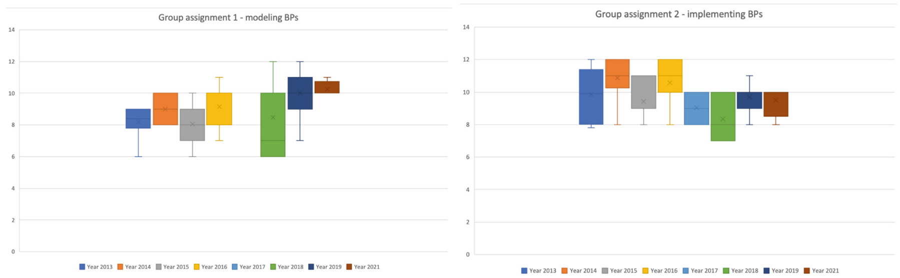
Modeling BPs From the modeling assignment and individual test, we found several modeling elements that are the most challenging for students to understand, as shown in Table 2. Although we provide exercises to illustrate key concepts, including models with errors such as improper use of elements, control flow with inconsistencies, detection of workflow patterns, and modeling best practices, these elements still appear in their solutions. We also found that transitioning from the complete BPMN 2.0 model, e.g., the collaborative model including several pools for different participants, to the single model of each orchestration implemented in the BPMS platform is also challenging.

Implementing BPs From the development assignment and individual test, we found that although most students are well-skilled in Java modeling and development and database definition and management, for information systems development, the process-driven execution in the BPMS platform is a challenging concept to understand. Also, modeling and designing the Java layer to support the process model execution by applying design and architectural patterns to define classes (entities) and associated listeners for execution is not straightforward. Another challenge regarding implementing the BP model is the definition and management of the organizational data model to support the application data not registered in the process engine database. This data model directly relates to the class definition (entities) for the Java layer to manage objects within the process engine execution, as needed by activities, gateways, and other elements within the process control flow.

Course grades The final grade for approving the course comprises two group assignments: BPs modeling and BPs implementation (development) in a BPMS, and two individual tests, one of BPs modeling and another of BPs development. The final grade values obtained by students in all editions are between 7 to 12, with 3 being the minimum grade for approval (60% of the work correctly done).

In Fig. 3, we show the distribution of grades for each assignment in all course editions. The lowest value for the modeling assignment in all editions is 6; for the implementation one, it is 7, and the maximum is 12 for both. In the first four editions (2013 to 2016), the modeling assignment presents lower degrees, both maximum and median, than the development assignment. We did not rank the first assignment in 2017, which included a single delivery. In 2018 there were the worst results in both assignments, mainly related to the complexity and domain of the selected BPs, which were based on real warehouse logistics.

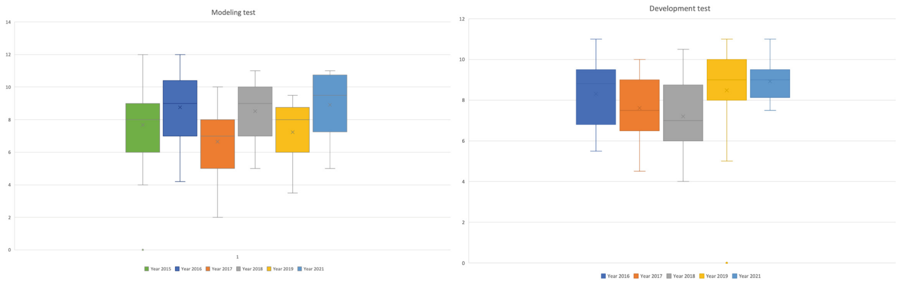
For the last two editions (2019 to 2021), grades are mainly in the same range for both assignments, slightly higher for modeling, being the highest grades for modeling from all editions. We think this results from reducing the complexity of the BPs in these editions, which provided more time to discuss specific modeling aspects reinforcing students' learning. However, this did not impact as expected in the development grades, which remained mainly within the same range in the last four editions, slightly higher in the 2019 edition. In the 2021 edition, the virtual format could have also played a role in this.



(a) Grades of the Group assignment 1

(b) Grades of the Group assignment 2

Fig. 3. Grades of Group assignments 1 and 2 for all editions



(a) Grades of the Modeling test

(b) Grades of the Development test

Fig. 4. Grades of Modeling and Development test for all editions

Regarding the development assignment and defense with execution in Activiti BPMS, although most groups can execute the scenarios we defined with different datasets and execution paths, some fail for various reasons, including design and technology decisions that groups made independently.

Regarding the modeling and development individual tests, we use them as another control for students within the group assignments work; we introduce the modeling test in the 2015 edition and the development one in the 2016 edition. In Fig. 4, we present the grades for the modeling and development tests from the 2015 edition. It can be seen that although grade ranges are more homogeneous than for the group assignments, minimums are worse in some cases, with values including 2 to 5 for the modeling test and 4 to 6 for the development test. It shows that the learning is not balanced in some groups, and some students are unaware of the modeling and development learning concepts.

Background and hours Around 60% of students work in the software industry, at least 20 h a week and several 40 h or more. They are asked to dedicate 10 h a week to the course, including 2–4 hours for classes and the rest to work on the exercises and assignments. In students' evaluation of the course, they mostly agreed that the hours dedicated are consistent with the required ones. Some extra hours are dedicated to setup the environment for Activiti BPMS, databases, and Java development since they are unfamiliar with the process-driven system perspective. Also, changes in the Activity versions prevent some technical requirements from being fulfilled in the last editions.

4 Discussion and Lessons Learned

From the results of teaching the course from 2013 to 2021, we obtained insights on the challenges Computer Science students face when modeling and developing a process-driven system using a BPMS platform such as Activiti.

Students do not have a background in process-driven systems and modeling languages for processes, but mostly programming in different languages and modeling with UML and ER models. Although we provided students with several familiar domain BP modeling exercises to practice identifying BP elements, become familiar with BPMN 2.0 constructs, and apply best practices, modeling results were not as good as expected, making mistakes as those described in Table 2. However, by reducing the BP's complexity and focusing with them on solving specific parts of the model, results improve.

BP modeling lesson learned 1:
Modeling BPs is challenging for Computer Science students with no process-driven background. Providing a process to model that requires the correct use of key modeling concepts and constructs but with reduced complexity helps them focus on solving a set of specific key challenges.

BP modeling lesson learned 2:
Providing students with the BPMN 2.0 language constructs and reinforcing BP modeling with best practices and workflow patterns does not directly impact the expected results for modeling and development. Students still make basic mistakes which we believe they can improve with practice.

Most students are skilled in Java modeling and development, database definition, and management. Moreover, they have experience in collaborative software development projects for traditional information systems. However, the process-driven execution in the BPMS platform is a challenging concept. Also, the design of the Java layer to support the process model execution and the definition and management of the organizational data model are not straightforward.

BP development lesson learned 1:
Although most students could be well-skilled in programming, they are unfamiliar with process-driven systems, and understanding the process engine and the inversion of control is challenging. Project-based hands-on assignments provide them with a natural and valuable experience developing these systems.

BP development lesson learned 2:
Guides on defining organizational data models to support application data managed by the BP model apart from the process engine database are needed. Understanding how to connect organizational data with the Java layer within the process model execution by the BPMS platform is not straightforward.

The experience yields information that allows some preliminary conclusions to be drawn. However, a more in-depth validation is necessary to transform them into recommendations. For this, we need to reinforce the evaluation of some aspects. In terms of modeling, we could compare the modeling results of groups that use good practices with groups that do not, as well as address domains with different levels of complexity. We could also get an actual client to evaluate the understanding of the models made by the client beyond the modeling errors that we, as teachers, identify. Regarding development, we could determine the use of other platforms that have evolved better, such as Camunda or Flowable.

5 Related Work

There is significant work on process modeling concerning best practices [9], workflow patterns [10], and process re-design heuristics [11], traditionally assessed in theoretical and practical contexts. Most works focus on teaching and learning such modeling concepts. In [12], the authors present an e-learning approach for process modeling education based on requirements derived from related work on modeling and e-learning. In [13], the authors focus on identifying difficulties of teaching/learning state-oriented modeling based on reflections from teachers and learners and a small-scale survey. In [14], the authors describe an introductory BPM course lectured in a college of business administration focused on process modeling and simulation. In [15], the authors present an undergraduate and postgraduate course in which students are asked to conceptualize, analyze, and articulate real-life process scenarios for PAIS design. There are also works about teaching a more general BPM perspective, not only focused on process modeling. In [16], the authors present a disruptive strategy based on role-play using Second Life to introduce students to the properties of ERP systems and simultaneously introduce tools for virtual team collaboration. In [17], the authors report on an analysis of the current BPM offerings of Australian universities. In [18], the authors describe their experience in teaching BPM as a Massive Open Online Course (MOOC) with a comprehensive coverage of the BPM lifecycle.

6 Conclusions

We have presented an undergraduate and postgraduate project-based hands-on course for process-driven systems modeling and development we have taught in a Computer Science degree including: course setup, experience, critical results for the 2013 to 2021 editions concerning the learning objectives, and lessons learned.

Among the challenges for students learning, we have identified that modeling BPs in BPMN 2.0 with best practices is the most challenging task of the course for them. Also, BPs implementation in a BPMS platform poses a critical challenge, mostly how the process engine works, the inversion of control from traditional systems (i.e., work list), and modeling and designing the Java layer and data model to support the process. We have taken a few actions to ease students learning for the challenges identified, from which some have already impacted in a positive way. Further evaluation is needed to deepen the results.

References

1. Weske, M.: BPM - Concepts, Languages, Architectures, 3rd Ed. Springer (2019). https://doi.org/10.1007/978-3-662-59432-2
2. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of BPM, 2nd Ed. Springer (2018). https://doi.org/10.1007/978-3-662-56509-4
3. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business process management: a survey. In: van der Aalst, W.M.P., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-44895-0_1
4. OMG: Business Process Model and Notation (BPMN) vol 2.0 (2011)
5. Chang, J.: BPM Systems: Strategy and Implementation. CRC Press (2016)
6. Dumas, M., van der Aalst, W., Hofstede, A.: Process-Aware Information Systems: Bridging People and Software Through Process Technology. Wiley (2005)
7. Delgado, A., Calegari, D., Milanese, P., Falcon, R., García, E.: A systematic approach for evaluating BPM systems: case studies on open source and proprietary tools. In: Damiani, E., Frati, F., Riehle, D., Wasserman, A.I. (eds.) OSS 2015. IAICT, vol. 451, pp. 81–90. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-17837-0_8
8. Delgado, A., Calegari, D.: Systematic evaluation of business process management systems: a comprehensive approach. CLEI Electron. J. **21**, 1–19 (2018)
9. Mendling, J., Reijers, H., van der Aalst, W.: Seven process modeling guidelines (7PMG). Inf. Softw. Technol. **52**(2), 127–136 (2010)
10. Russell, N., Aalst, W.M.V.D., Hofstede, A.H.M.T.: Workflow Patterns: The Definitive Guide. MIT Press (2016)
11. Reijers, H., Liman Mansar, S.: Best practices in BP redesign: overview and qualitative evaluation of successful redesign heuristics. Omega **33**(4), 283–306 (2005)

12. Neubauer, M.: E-Learning support for business process modeling: linking modeling language concepts to general modeling concepts and vice versa. In: Stary, C. (ed.) S-BPM ONE 2012. LNBIP, vol. 104, pp. 62–76. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29133-3_5
13. Koutsopoulos, G., Bider, I.: Teaching and learning state-oriented business process modeling. experience report. In: Reinhartz-Berger, I., Gulden, J., Nurcan, S., Guédria, W., Bera, P. (eds.) BPMDS/EMMSAD -2017. LNBIP, vol. 287, pp. 171–185. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59466-8_11
14. Saraswat, S.P., Anderson, D.M., Chircu, A.M.: Teaching bpm with simulation in graduate business programs: an integrative approach. J. Inf. Syst. Educ. **25**(3), 221–232 (2014)
15. Recker, J., Rosemann, M.: Teaching business process modelling: experiences and recommendations. Commun. Assoc. Inf. Syst. **25**, 32 (2009)
16. Rudra, A., Jaeger, B., Aitken, A., Chang, V., Helgheim, B.: Virtual team role play using second life for teaching business process concepts. In: 2011 44th Hawaii International Conference on System Sciences, pp. 1–8 (2011)
17. Marjanovic, O., Bandara, W.: The current state of BPM education in Australia: teaching and research challenges. In: zur Muehlen, M., Su, J. (eds.) BPM 2010. LNBIP, vol. 66, pp. 775–789. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20511-8_69
18. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Teaching BPM as MOOC. BPTrends (2015)