An Assignment on Information System Modeling
On Teaching Data and Process Integration

Jan Martijn E.M. van der Werf[1] and Artem Polyvyanyy[2]

[1] Department of Information and Computing Science
Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
j.m.e.m.vanderwerf@uu.nl
[2] The University of Melbourne, Parkville, VIC, 3010, Australia
artem.polyvyanyy@unimelb.edu.au

Abstract. An *information system* is an integrated system of components that cooperatively aim to collect, store, manipulate, process, and disseminate data, information, and knowledge, often offered as digital products. A *model* of an existing or envisioned information system is its simplified representation developed to serve a purpose for a target audience. A model may represent various aspects of the system, including the structure of information, data constraints, processes that govern information, and organizational rules. Traditionally, the teaching of information system modeling is carried out in a fragmented way, i.e., modeling of different aspects of information systems is taught separately, often across different subjects. The authors' teaching experience in this area suggests the shortcomings of such fragmented approach, evidenced by the lack of students' ability to exploit the synergy between data and process constraints in the produced models of information systems.
This paper proposes an assignment for undergraduate students which requests to model an information system of an envisioned private teaching institute. The assignment comprises a plethora of requirements grounded in the interplay of data and process constraints, and is accompanied by a tool that supports their explicit representation.

Keywords: Data and process modeling, information system modeling, Computer Science and Information Systems education

1 Introduction

In the information age we live, information systems provide core mechanisms for supporting operational business processes of organizations. Hence, leading Computer Science and Information Systems curricula comprise courses that teach students the art and rigor of designing information systems. Traditionally, modeling of each aspect of an information system, e.g., data and process constraints, is taught separately, often across different subjects. The authors have independently taught the foundations of information systems modeling to undergraduate students at Utrecht University, The Netherlands, and Queensland University of Technology, Australia (for five and seven consecutive semesters, respectively). In this paper, the authors report on identified drawbacks

Postprint, October 2018

of such a fragmented approach to teaching information system modeling, and argue for the need in educating students on data and process integration.

As an example, consider a task of designing a learning management system that keeps track of course offering, and corresponding lecturers and student enrollments. A decision to start by developing a high-quality data model for the proposed scenario may result in a design which requires that every course offering is assigned at least one lecturer. This design may contradict the corresponding business processes that require to assign a lecturer to a course offering only once it reaches the minimum number of student enrollments. Conversely, a decision to introduce a process constraint may limit the number of solutions to the design of the data model in a way that excludes the required solution. Note that even if all the data and process requirements of the desired solution are laid out prior to embarking into modeling, they may lead to a contradiction that does not manifest neither in a data model nor in a process model that satisfies the respective requirements. Thus, an effective approach to modeling an information system should allow a designer to experience the interplay between data and process constraints. Building from this understanding, the paper at hand contributes:

1. An assignment to model an information system of an envisioned private teaching institute;
2. A systematic analysis of challenges experienced by students when solving the assignment in a traditional way, i.e., by tackling modeling of information constraints and business processes of the system separately;
3. A proposal to address the identified challenges by using a new tool capable of representing an interplay between the data and process constraints in an integrated model of an information system.

The remainder of this paper is organized as follows. The next section examines how data and process modeling skills are recognized in the curricula of undergraduate degrees in Information Systems. Section 3 proposes an assignment that aims to teach data and process modeling skills in an integrated way. Section 4 shares our experience, while Section 5 proposes a tool support for designing data and process constraints in an integrated way. The paper closes with conclusions.

2 Teaching Data and Process Modeling in IS Curricula

In 2010, the Association for Information Systems (AIS) and the Association for Computing Machinery (ACM) have released IS 2010, the latest in a series of proposed *model curricula* for undergraduate degrees in Information Systems [15]. IS 2010 provides guidance regarding the core content of a curriculum in Information Systems and suggests possible electives and career tracks.

IS 2010 comprises seven core and several elective courses, among which Data and Information Management (IS 2010.2) and Systems Analysis and Design (IS 2010.6) are recognized to play a central role. Next, we examine these two courses with respect to the proposed learning outcomes and topics that contribute to data and process modeling skills, taking a close look at the skills that are grounded in the interplay of data and process constraints in the designs of information systems.

2.1 Data and Information Management

According to IS 2010, the Data and Information Management (IS 2010.2) course provides students with an introduction to the core concepts in data and information management. Concretely, this course teaches students methods and techniques for identifying organizational information requirements, constructing conceptual models of these requirements, converting the conceptual data models into logical models, e.g., relational data models, verifying the correctness of the models, and implementing the models, e.g., using a Relational Database Management System (DBMS) [11, 14].

Among the 21 suggested learning objectives of this course, we identify three core objectives [3] that specifically target the data modeling skills of a student:

- Use at least one conceptual data modeling technique (such as entity-relationship modeling) to capture the information requirements for an enterprise domain;
- Design high-quality relational databases;
- Understand the concept of database transaction and apply it appropriately to an application context.

The topics of the course that contribute to these skills are conceptual, logical, and physical data models, for example entity-relationship model, relational data model, and data types, respectively. The curriculum suggests that the focus should be on conceptual and logical data modeling skills, while "students should understand the basic nature of the DBA tasks and be able to make intelligent decisions regarding DBMS choice and the acquisition of DBA resources."

Two learning objectives of the IS 2010.2 course may be interpreted as such that suggest an interplay between the data and process modeling skills:

- Apply information requirements specification processes in the broader systems analysis and design context;
- Link to each other the results of data/information modeling and process modeling.

None of the proposed course topics explicitly contributes to the integration of data and process modeling skills of a student. One may argue that such skills are implicit in the topic of "Using a database management system from an application development environment". Still, this topics advocates for a compartmented approach to data and process modeling. At the same time the curriculum acknowledges that "information requirements specification processes must be firmly linked to the organizational systems analysis and design processes".

2.2 Systems Analysis and Design

The curriculum suggests that the Systems Analysis and Design (IS 2010.6) course should contribute to 13 learning objectives, among which only two implicitly target process modeling skills, namely:

[3] Note that several other proposed learning objectives can be seen as refinements of the core ones, e.g., the objective of "Design a relational database so that it is at least in 3NF" can be seen as a refinement of "Design high-quality relational databases".

– Use at least one specific methodology for analyzing a business situation (a problem or opportunity), modeling it using a formal technique, and specifying requirements for a system that enables a productive change in a way the business is conducted.
– Within the context of the methodologies they learn, write clear and concise business requirements documents and convert them into technical specifications.

We identify that the topics of the course that can contribute to these objectives are Business Process Management and analysis of business requirements. The curriculum contains an elective course entitled Business Process Management [1, 2, 8], which refines the learning objectives that address process modeling skills. The main focus of this elective course is on understanding and designing of business processes, which manifests in four learning outcomes (out of 11):

– Model business processes;
– Understand different approaches to business process modeling and improvement;
– Use basic business process modeling tools;
– Simulate simple business processes and use simulation results in business process analysis.

Two proposed learning objectives of the IS 2010.6 course address the integration of data and process modeling skills, namely:

– Use contemporary CASE tools for the use in process and data modeling.
– Design high-level logical system characteristics (user interface design, design of data and information requirements).

However, again, similar to IS 2010.2, none of the proposed topics of IS 2010.6, or those of the elective Business Process Management course, explicitly contributes to the integration of data and process modeling skills of a student.

3 Assignment: Supporting the Private Teaching Institute

An effective assignment to modeling an information system should allow students to experience the interplay between data and processes. The assignment should have a sufficiently challenging and realistic case description, while being manageable in size.

3.1 Learning Objectives

As a first step, we crafted the learning objectives, following the IS 2010 guidelines, and the Bloom Taxonomy [4]. As the assignment focuses on learning to apply techniques, we assume that once the assignment starts, students already have an initial understanding of data modeling e.g. with ERM [6], and process modeling, e.g., with Petri nets [13] and BPMN [8]. In other words, we assume students to start at level 2 (comprehension) of the Bloom Taxonomy. The learning objectives of the assignment cover the next levels, being application, analysis, synthesis and evaluation. After the assignment, the students should be able to:

– Model and analyze process and information requirements using formal techniques;

- Critically assess models and make well-informed design decisions to solve real world problems related to information systems;
- Write clear and concise requirements and convert these into technical specifications using formal techniques;
- Manage the complexity of contemporary and future information systems and the domains in which these systems are used; and
- Use contemporary off-the-shelf components to integrate models into an information system.

Experience from a previous assignment [10], where students had to design and build an information system for an online shop, showed that students had difficulties in understanding the underlying problems of the domain. Therefore, the context of this assignment should be geared to the students' perception of their environment. For this purpose, we designed a case around a fictive educational institute, the Private Teaching Institute (PTI). Several requirements have been left implicit, or are even underspecified to allow students to reflect and perform a proper context analysis. In this way, students can use their own experience to better understand the situation.

3.2 The Case: The Private Teaching Institute

The Private Teaching Institute (PTI) offers education tracks. Each education track consists of several mandatory courses, and some optional courses. PTI consists of a small team per track, the track management, and a small student administration for all tracks together. To deliver the courses, PTI has a pool of lecturers who are qualified to deliver several courses. Everybody is entitled to enroll for a track. As soon as somebody registered themselves, and they are accepted by the track management, they become a student of that track. Students enrolled have to create an educational plan, consisting of the courses they want to follow. This plan has to be approved by the appropriate track management, and filed by the administration.

As soon as the plan is approved, students may register for courses. Once there are sufficient registrations for a course, the management creates a tender and sends it out to the lecturers who are qualified to give that course. After the response offers by the lecturers, the management selects the best offer and appoints the corresponding lecturer for that course. Every course at PTI consists of several lectures, either in a classical class room setting or on-line, practical assignments, and one or more exams, depending on the wishes of the appointed lecturer. Once the student meets all criteria set by the lecturer, i.e., passing a sufficient number of assignments and exams, the student receives a certificate of passing. In all cases, the result is filed by the administration.

Once a student passed all the courses agreed upon in the educational plan, the student is eligible to receive a diploma for that track. The track management verifies the course certificates and the plan, after which the management can award the diploma. Students can choose for a formal ceremony, or to receive their diploma by post.

PTI wants a process-aware information system that supports them in their primary processes, to ease the administrative burden.

3.3 Phases and Deliverables

The information system should be designed and implemented, while ensuring that all deliverables remain consistent. The assignment identifies two phases: the specification phase, and the implementation phase. Instead of following the traditional waterfall approach, the phases run concurrently, and the deliverables of the two phases should be synchronized regularly. Having small cycles assist in keeping the problem at hand manageable, and also allows the teaching staff to provide the students with early feedback.

During the first phase, the students have to analyze the assignment, and identify the involved stakeholders and their interactions with the to-be-designed information system. For this analysis, students may apply different techniques. Some students prefer to create use cases [5], other students perform a PACT analysis [3]. A PACT analysis studies the People involved, their Activities, the Context in which these activities are performed, and the main Technologies used to support these.

Once the context of the assignment has been analyzed to gain a better understanding of the environment, the students have to derive the information requirements and build a specification. Part of the specification is a data model in ERM notation. Many choices have been left implicit in the case description, such as the number of courses a track consists of, whether courses are mandatory for the complete institute, or only for tracks, etc. Students have to discover these choices, and make and document their design decisions. To model the flow of information, the different processes in the case have to be identified, analyzed and modeled using Petri nets. The resulting models should be analyzed for correctness using formal approaches, such as weak termination (i.e., absence of deadlocks and livelocks) and boundedness. Additionally, the different models created should be consistent, and validated with the context analysis, i.e., the use cases and scenarios created initially should be supported by the models.

The context description, information model and process models together with their analyses are captured in the *Specification Document* that the students have to deliver. The resulting document should be concise, clear and contain all important requirements of the case.

Once an initial version of the specification document, containing one or two processes, is being created, the implementation phase starts. The goal of the implementation phase is to use packaged solutions, rather than implement a system from scratch. The assignment relies on the Business Process Management Suite (BPMS) ProcessMaker [4], which has both an open source edition, as well as a commercial cloud service. For the implementation of the information system, each process designed in the specification document should be converted into a BPMN model, together with the forms and triggers for each activities. As the complete information system comprises several processes, the data model has to be implemented, and the forms and activities of the different processes should manipulate the data model. This phase results in two deliverables: the *Implementation Guide*, and the *implementation* itself.

As in real life, processes may be altered, updated or completely revised during the implementation. Therefore, during the different phases, the specification document and implementation guide need to be updated together, ensuring that the revised models remain correct, and the documentation consistent.

[4] http://www.processmaker.com/

Table 1. Grading schema for the assignment

Specification document	Points	Implementation guide	Points
Context analysis	15	Quality BPMN models	25
Data model	10	Model descriptions	5
Quality process models	30	Gateway logic	5
Documentation of models	15	Forms per activity	10
Verification and validation	25	Reflection	10
Layout	5	Layout	5
		Implemented functionality	25
		Demonstration	15
Total	100	**Total**	100

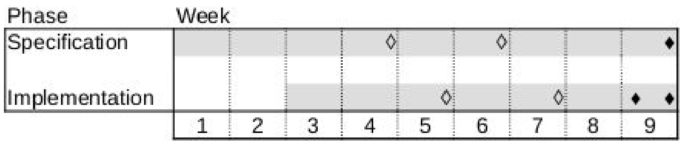


Fig. 1. Gantt chart of the assignment. The open diamonds are feedback moments, the filled diamonds are official deadlines, including a demonstration.

For grading, the schema shown in Table 1 is used. The schema addresses the different learning objectives. For feedback and grading a rubric based on this schema is used [5]. Part of the implementation phase is a demonstration of the system to the teaching staff, simulating the role of a stakeholder at PTI.

4 First Experiences with the Assignment

Last year, the assignment has been executed for the first time during the Information Systems course at Utrecht University, with about 170 first year Information Science Bachelor students. Although the group is quite large, we decided to have the students to create pairs, instead of larger groups. In this way, students are able to cooperate, and discuss design options, at the same time preventing free riders.

The course is taught in the final block of the year, and runs over a period of 10 weeks. As a 7,5 EC credit course [6], students are expected to work 20 hours per week on the subject, including lectures on process modeling and analysis. In total, each student is expected to dedicate in total 100 hours to the assignment. Each phase had two intermediary deadlines for feedback, and a final deadline at the end of the period (see Fig. 1). The demonstrations were in the same week as the final deadline.

Process Identification During the first feedback moment, we noticed that many students found it challenging to discover the different processes in the assignment. Many groups had problems in dividing the case description into smaller, manageable components. Several authors acknowledge the difficulty of discovering the processes in an

[5] The rubric can be found at http://www.architecturemining.org/publications/WerfP18a.pdf
[6] https://ec.europa.eu/education/resources/european-credit-transfer-accumulation-system_en

organisation (cf. [8]), and point e.g. at categories of Processes according to Porter, to assist in this activity. However, as these categories are tailored towards businesses, students found it difficult to apply them on a different context.

Some students delivered a single large model that covered all facets of the institute. For example, the student's enrollment and the tender process for lecturers were combined in a single process. They failed to recognize that by combining these two processes, the complete tender process had to be repeated for each student enrollment. A possible cause is that BPMN leaves the notion of a case implicit. As a consequence, students do not notice that halfway the process the case changes from the "student following a course instance", to "the course instance for which a lecturer needs to be selected". By providing feedback after the first round on how to read the case description, and by posing questions like "what is the subject of this process?" explicitly in the feedback, students understood the notion of cases and processes much better.

Other groups divided the assignment in many small processes, such as "do assignment", which comprised two activities: the student creating an assignment, and a lecturer grading the assignment. Although in essence this is not wrong, the finer the granularity of the processes identified, the more challenging it is to understand the interplay of the different processes. For example, is a student allowed to receive a grade if one of the assignment processes is still running? Having a too fine-grained solution simplifies modeling and analyzing the separate models, but complicates the overall design of the information system.

In the end, most student groups delivered an information system that implemented two to four business processes. These processes capture different aspects of the information system, from enrolling in an educational track, following a course instance, the lecturer tendering process, and obtaining the diploma. Some students combined the enrollment and obtaining the diploma, i.e., the process a student follows in an educational track. Others combined the students following a course instance process with the lecturer tendering process, by taking the course instance as a case, rather than a student following a course instance.

Process Modeling Although having Petri nets as the primary modeling notation helps students in making the state, and thus the case, explicit, it turned out to be difficult for students to give proper meaning to tokens and places. Tokens resembling a single object, such as a lecturer or a student were often found at a first round. However, combining different notions, like "a token in this place resembles a student that is following a course" turns out to be more difficult than initially anticipated. After the first round of feedback, students were taught the concept of place invariants. This increased the students' understanding of the idea of tokens and places resembling combinations of elements, rather than just being single elements representing the state of the net.

As in a previous course on information modeling, students learned to design forms to populate their data model, several groups created "screen-based" processes. Each activity represented a screen a user would see in the system, and the process flow depicted the possible orders in which the screens would be displayed. Discussing their solution after the first feedback round, revealed that these student groups had similar problems in understanding the notion of a case.

Another challenge many students faced is the level of abstraction in activities. For example, several groups produced process models with small activities like “fill in address”, “fill in telephone number”, and “select education track”, rather than having a larger activity “enroll for education track”, leaving the details of what data is needed for an enrollment to a later stage in the process. These small activities appeared either in a large parallel construct, or were modeled consecutively, in a fixed order.

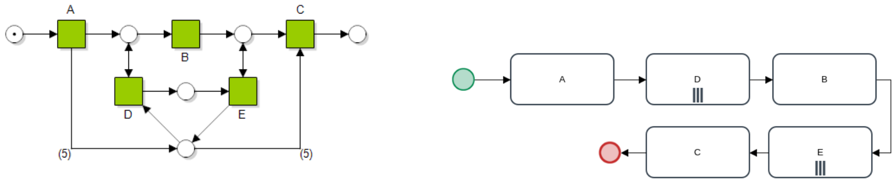
In the final deliverable, all student groups delivered process models with each containing ten to twenty activities. Each activity had a clear form and roles assigned. The interplay between the different processes was expressed both in Petri nets, and implemented using triggers on the activities, and by connecting the data model to the different activities in the process models.

Process Analysis During the lectures of the course, many different analysis techniques, such as reachability and invariant calculus are discussed. Relating these abstract properties, like liveness, boundedness and place invariants to properties turns out to be a good exercise in understanding why these properties help in improving their solutions.

The students had to analyze their solution in different dimensions. The first dimension is intra-process versus inter-process. Within a single process, all properties are relatively easy to verify, especially if their solution contains many small processes. The challenge is in analyzing the interplay between different processes. For example, dependencies may exist, like in the example of the small assignment process: who is allowed to start this process, and when? Similarly, to model a check whether a course instance has sufficient students enrolled, can be challenging if each student enrolls in a separate process instance.

A second dimension is verification within the models versus validation with the context. Verification of the models, i.e., checking whether the models satisfy properties like liveness, boundedness and weak termination, was performed by all students. Validation, i.e., checking whether the models are appropriate for the problem at hand turns out to be more difficult. Most students delivered initially reports containing many, large user stories, but no analysis whether their solution can actually replay the scenarios they described earlier in the same document.

Implementation Another challenge remains in transforming the formal process models designed with Petri nets into BPMN models that are executable by Business Process Management Suites (BPMSs) like ProcessMaker. On the one hand, the formal semantics of Petri nets allow the students to simulate and analyze their processes, and test their dependencies by composing all models into a large Petri net. On the other hand, a BPMS requires the model to be divided into small processes, in which the state is left implicit. In addition, several constructs are needed in Petri nets to keep models analyzable, e.g. the amount of lecturers available to teach a course. In BPMN specialized constructs exist, such as parallel repetition via multi-instance activities, that are designed to solve such situations, as an example shows in Fig. 2. This requires the students to be creative in their solutions on how to move from a formal specification into a technical implementation, while showing that their ideas remain consistent with the specification.



(a) (b)

Fig. 2. Situation modeled in Petri nets (a) for which the multi-instance activity in BPMN (b) gives a more natural solution.

Balancing Data and Processes An important observation we made during the assignment is how subtle the connection between processes and data is. Although these subjects are being taught in different courses, these go hand in hand in an integrated information system.

To give an example, most students create a data model in which a course instance always has a lecturer (a one-to-many relation), has one exam and one assignment. However, in the process of running a course instance, the track management first decides that a course instance, for which students already could subscribe, will start, and only then decide to start a tender for which lecturers can apply. Hence, although the course instance already exists, no lecturer is assigned to it. Consequently, the data model is violated, as the one-to-many relationship is not valid, whereas adding a lecturer while creating a course instance violates the process model. This results in a deadlock caused by the integration of the two models. Although the example seems trivial, it turns out that many such integration issues occur in the assignment.

The interplay between processes and data is very difficult to analyze and discover at design time, and is mostly found only while testing the information system, which is already difficult and challenging in itself. This debugging and "bug hunting", as some students named it, is a very time-consuming and frustrating process, as it is scattered over the different forms, triggers and database handling in all processes.

Overall Perception All student groups delivered an integrated information system that supported most functionality. The specification document and implementation guide typically were consistent. Reduction rules [13] combined with reachability graphs were the most used analysis tool to verify the models, and several groups used place invariants to show that their resources, such as lecturers, courses and students remained constant in the system.

Afterwards, the course was evaluated by the students (n=41) using closed questions on a 1-5 likert scale. Students pointed out that the lectures were well usable for the assignment (85% scored ≥ 4), and that they learned "a great deal" (83% scored ≥ 4). Although labor intensive, the students valued the early feedback rounds and stated that the feedback helped improving their results (73% scored ≥ 4). In the open feedback questions, students posed that the used system has its problems and peculiarities. This made it often difficult to understand what went wrong, and how this could be mitigated. However, the students valued the freedom the assignment provides, ensuring that everybody has a different solution, enabling them to discuss alternatives among each other.

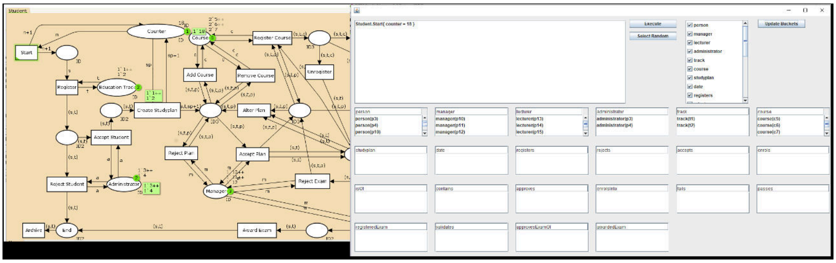


Fig. 3. ISModeler. The tool combines CPN Tools with a theorem prover for the data model.

5 Next Steps

Based on the results of the first run of the assignment, we found that integrating data and processes is experienced as challenging by the students. For many practitioners, experience plays an important role in knowing how to adapt processes and data, and when. In some cases it is better to alter the data model, in other cases the process model. This requires experience, and practice.

In our view, integrating processes and data is given too little attention in current curricula. The assignment shows that students find it very difficult to analyze the specification on deadlocks caused by the integration of data and process models. To our knowledge, hardly any analysis technique taught in textbooks is grounded in both data and processes. At the same time, we see that courses on Data and Information Management (IS2010.2) focus on information requirements and data modeling. Processes are acknowledged, but play a very small role in the IS 2010 guideline. Similarly, process modeling courses, like the elective on BPM, focus on processes, but tend to ignore that these processes manipulate (structured) data.

A course on information system modeling should not only focus on these two aspects, but also show the synergy between the two modeling paradigms. We therefore developed the tool *ISModeler* that makes this synergy explicit [16]. It combines a process model in the form of a Petri net in which tokens carry identifiers [10, 12], a data model, and a transition specification that defines how each transition manipulates the data model through transactions. The tool builds upon CPN tools [17], and a theorem prover to validate the transactions on populations of the data model. In *ISModeler*, a transition is enabled if it is both enabled in the Petri net, and the transaction yields a valid population. Fig. 3 shows a screenshot of the system. In the top part of the window, the enabled transitions are shown, whereas the bottom part depicts the population of the data model, by listing per entity type and relationship the elements it contains. In this way, we envision that students will better understand the synergy between data and processes, and thus design and build better integrated information systems. The tool is planned to be put into action in next year's edition to evaluate its effectiveness.

6 Conclusions

In this paper, we propose an assignment that allows students to experience the design and implementation of an information system using a BPMS. The proposed assignment combines data and process modeling, forcing students to design and analyze their solution using formal techniques, and translate their solution into an information system.

Running the assignment for the first time shows that the assignment helps students to experience design issues that arise while studying the case description. Students discovered that abstract properties used in verification can be linked to actual properties in the case description, and assist them in improving their solution.

However, the run also shows that students find it difficult to understand the synergy between data and processes. Although in scientific literature several approaches exist that allow to model this (cf. [7, 9, 12]), experiences with the assignment show that these have not yet been embedded sufficiently in our education curricula.

References

1. W. M. P. van der Aalst and C. Stahl. *Modeling Business Processes - A Petri Net-oriented approach*. MIT Press, 2011.
2. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods and Systems*. Academic Service, Schoonhoven, 1997.
3. D. Benyon. *Designing Interactive Systems: A Comprehensive Guide to HCI, UX and Interaction Design*. Pearson, 3rd edition, 2014.
4. B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl. Taxonomy of educational objectives: The classification of educational goals. In *Handbook I: Cognitive domain*. David McKay Company, 1956.
5. G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 2005.
6. P.P. Chen. The entity-relationship model: Towards a unified view of data. *ACM Transactions on Database Systems*, 1:9–36, January 1976.
7. A. Deutsch, R. Hull, F. Patrizi, and V. Vianu. Automatic verification of data-centric business processes. In *ICDT '09*, pages 252–267. ACM, 2009.
8. M. Dumas, M. La Rosa, J. Mendling, and H. Reijers. *Fundamentals of Business Process Management*. Springer, 2018.
9. K.M. van Hee. *Information System Engineering - A Formal Approach*. Cambridge University Press, 1994.
10. K.M. van Hee, J. Keiren, R.D.J. Post, N. Sidorova, and J.M.E.M. van der Werf. Designing Case Handling Systems. *ToPNoC*, 5100, 2008.
11. D. M. Kroenke and D. J. Auer. *Database Concepts: Seventh Edition, Global Edition*. Pearson, 2015.
12. M. Montali and A. Rivkin. DB-Nets: On the marriage of colored Petri nets and relational databases. In *Petri Nets*, volume 10470 of *LNCS*, pages 91–118. Springer, 2017.
13. T. Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.
14. A. Silberschatz, H. Korth, and S. Sudarshan. *Database System Concepts*. McGraw Hill, 5th edition, 2006.
15. H. Topi, K. M. Kaiser, J. C. Sipior, J. S. Valacich, J. F. Nunamaker, Jr., G. J. de Vreede, and R. Wright. Curriculum guidelines for undergraduate degree programs in Information Systems. Technical report, ACM, 2010.
16. J. M. E. M. van der Werf, A. Polyvyanyy, S. J. Overbeek, and R. A. C. M. Brouwers. On a synergy between data and processes. Technical Report UU-CS-2018-004, Utrecht University, 2018.
17. M. Westergaard and L.M. Kristensen. The access/CPN framework: A tool for interacting with the CPN-tools simulator. In *Petri Nets*, volume 5606 of *LNCS*. Springer, 2009.

Teaching Notes: Specification and Design of an Information System

Assignment 2016/2017

Jan Martijn E.M. van der Werf[1] and Artem Polyvyanyy[2]

[1] Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
j.m.e.m.vanderwerf@uu.nl
[2] The University of Melbourne, Parkville, VIC, 3010, Australia
artem.polyvyanyy@unimelb.edu.au

1 Introduction

During this course, you will design and implement your own information system for the Private Teaching Institute (PTI). An information system is always a synergy between data and processes. The data aspects describe the structure of information, whereas the process aspect focuses on the information streams within an organisation. The data aspect has been covered in different courses, e.g., in the course Data Modeling (INFOB1DM) for information sciences, and Databases (INFODB) for computing sciences. In this course, we will focus on the process aspect. However, as any information system manipulates data, you will have to create and implement a data model.

The assignment consists of two phases. In the first phase, you will work **in groups of two students** to design and analyse a specification for an information system to support PTI. In this phase, you will apply different techniques covered during the lectures.

The second phase focuses on the actual implementation of the information system, and is an **individual assignment**. At the end of the course, you have designed, and built a completely working information system. For this, we rely on the Business Process Management System (BPMS) **ProcessMaker**. A BPMS allows the quick realisation of real information systems. In ProcessMaker, you will design the different processes in the Business Process Modelling Notation (BPMN), add your data model, and implement forms for the different activities. Last, some logic (in PHP) is required to connect data and processes.

As you will notice, the case is underspecified. It will be your task to carefully consider the different options, and make deliberate decisions, which you have to document as well. Therefore, part 1 is a group assignment, whereas phase 2 is an individual assignment. The idea behind this approach is that you can model in groups, discussing different options and solutions, whereas in the implementation you can make a difference. It is allowed to cooperate in phase 2, but each has to demonstrate the system individually. Additionally, you have to write a section on how your system differs from your group partner.

Good luck with the assignment!

2 The Case

The Private Teaching Institute (PTI) offers education tracks. Every educational track consists of several mandatory courses, and some optional courses. PTI consists of a small team per track, the track management, and a small student administration for all tracks together. To deliver the courses, PTI has a pool of lecturers who are qualified to deliver several courses. Everybody is entitled to enrol for a track. As soon as somebody registered themselves, and they are accepted by the track management, they become a student of that track. Students enrolled have to create an educational plan, consisting of the courses they want to follow. This plan has to be approved by the appropriate track management, and filed by the administration.

As soon as the plan is approved, students may register for courses. As soon as there are sufficient registrations for a course, the management creates a tender and sends it out to the lecturers who are qualified to give that course. After the response offer by the lecturers, the management selects the best offer and appoints the corresponding lecturer for that course. Every course at PTI consists of several lectures, either in a classical class room setting or on-line, practical assignments, and one or more exams, depending on the wishes of the appointed lecturer. Once the student meets all criteria set by the lecturer, i.e., passing a sufficient number of assignments and exams, the student receives from the administration a certificate of passing. In all cases, the result is filed by the administration.

Once a student passed all the courses agreed upon in the educational plan, the student is eligible to receive a diploma for that track. The track management verifies the course certificates and the plan, after which the management can award the diploma. Students can choose for a formal ceremony to receive the diploma, or it can be send by post.

PTI wants a process-aware information system that supports them in their primary processes, to ease the administrative burden.

3 Assignment

The assignment is to specify and implement an information system that supports PTI. The assignment is divided into two phases. In the first phase, you will design and specify the information system. In the second phase, you will implement the information system itself, using the BPMS *ProcessMaker*.

3.1 Phase 1: Specification

Per group of two students, you will design, analyse a specification of the information system. The specification consists of two parts: a data model, and several process models. For the data model, you can use a notation of choice, such as ER-diagrams, ORM, or UML class diagrams. Please document which notation you use. For the process models, you will have to use Petri nets. Divide the system into several processes, model each process, and show how these models cooperate.

Document your specification. For each model, describe the general idea of the model, the main design decisions, and describe the elements, and their intention. For the data model, this implies describing the entity and relation types, and their cardinalities. For the process models, describe each transition.

Next step of the specification phase is to analyse your design, using techniques explained during the lectures. It is not needed to use all discussed techniques, as long as you can show that:

1. the solution is correct (verification, e.g., boundedness, liveness, soundness, etc.)
2. Your solution is a correct solution for the given situation (validation)

This phase results in a specification document comprising at least the following elements:

End-user analysis What are the main users of the system, and how will these use the system? Use use cases or scenarios to describe these.

Data model What are the main concepts and data elements the system manipulates? How are these related?

Process models What are the different processes, and how are these related?

Analysis Verify and validate your solution: why is the solution a correct solution, and that your solution is correct. Do this by showing how your model supports the described use cases, and verify the models, and their relations.

The goal of this specification document is twofold:

A. to analyse the context of the information system, deriving a proper solution;
B. to guide the implementation of the information system.

3.2 Phase 2: Implementation

Once you have an (initial) version of the specification document, you can start implementing the system yourself. This phase is an *individual assignment*.

As a first step, design BPMN models that implement the process models you designed. As Petri nets and BPMN have different semantics, you have to provide a rationale of why your BPMN models implement each process specified in phase 1. Next, you have to design the forms, gateway logic, and data interactions to create a properly functioning information system.

This phase results in two deliverables: the implementation guide, and the implementation itself. The guide comprises at least the following elements:

1. Workflow models in BPMN;
2. Description of the processes;
3. Database structure;
4. Forms per activity;
5. Logic per gateway;
6. A short reflection;

Although phase 2 is an individual assignment, you are allowed to cooperate with your partner with whom you created the specification document. However, you have to add a reflection, that states how and where your information system differs from you partner.

3.3 Assessment & grading

For the assessment and grading of the assignment, we use the criteria specified in the table below. The final grade for the assignment is determined by dividing the sum of all gained points by 20.

Specification document	**Points**
Context analysis	15
Data model	10
Quality process models	30
Documentation of models	15
Verification and validation	25
Layout	5
Total	100

Implementation guide	**Points**
Quality BPMN models	25
Model descriptions	5
Gateway logic	5
Forms per activity	10
Reflection	10
Layout	5
Implemented functionality	25
Demonstration	15
Total	100

3.4 Some tips

1. Pay attention to edge cases;
2. What happens if people do not abide by what was agreed upon?
3. Write down your assumptions in the rationale, as these shape your solution!
4. Work in small iterations: add a small piece of functionality, verify and validate, and then expand to the next feature;
5. Divide the problem space smartly, and you will conquer the solution!

4 Course Planning

This is a 7,5 EC course, which equals 210 hour. For this course, the expected hours per task are as follows:

- Lectures: 30 hours
- Lab sessions: 30 hours
- Assignment: 100 hours
- Exam (including preparations): 50 hours

The course consists of three phases. The first two phases have lectures three times a week, and are concluded with a written exam. The last phase is concluded with an individual demonstration of your implemented information system. All deadlines are Friday 17:00 CEST. The intended schedule of the course is given in the table below.

week	Monday	Tuesday	Thursday	Deadline
1	Graphs	Labeled Transition Systems	Petri nets	
2	Petri nets I	Petri nets II	BPMN & Workflows	
3	Reachability graphs	BPMSs & ProcessMaker	*Ascension day*	
4	Question & Answer		**Partial exam A**	SD-1
5	*Pentacost*	Bisimulation I	Bisimulation II	IG-1
6	Coverability I	Coverability II	Invariants I	SD-2
7	Invariants II	Process mining I	Process mining II	IG-2
8	Question & Answer		**Partial exam B**	
9	Hands-on ProcessMaker			
10	Demonstrations			SD, IG

(SD = Specification document, IG = Implementation Guide)

A Rubrics For the Specication Document

User and Context Analysis (max 15pt)

Actors	−−	−	0	+	++
Scenarios / use cases	−−	−	0	+	++
Analysis	−−	−	0	+	++

General remarks: The problem analysis shows that you understand the problem of the client: what are the main processes the system should understand? How can these processes be broken down into activities? What are the actors, and how are these supposed to work with the system. This should be written down for each of the processes the system will support.

Data model (max 10pt)

Description of the model	−−	−	0	+	++
Concepts and attributes	−−	−	0	+	++
Relations	−−	−	0	+	++
Quality of the models	−−	−	0	+	++

General remarks: The data model should be properly documented: are the document and model conforming? Are all basic concepts there? Lecturers, courses, tracks, grades, diplom, exercises, mandatory per track?, Are the minimallly required relations present? Are the cardinalities correct?

Process models (max 30pt)

Subscription process	−−	−	0	+	++
Curriculum management	−−	−	0	+	++
Course progress	−−	−	0	+	++
Teacher tender process	−−	−	0	+	++
Model interrelationship	−−	−	0	+	++
Quality of the models	−−	−	0	+	++

General remarks: Each of the processes should be modelled using Petri nets. One can use ordinary Petri nets or Workflow-nets, based on the type of process under modeling.

There are a few basic processes that should be supported by the system, which are listed in the feedback form. These models should be able to replay the scenarios written down in the problem analysis. More models are encouraged, and – if correct – result in extra credit points. These models interact: a student cannot register for a course before they are registered at the organisation. A course cannot be finished without a teacher, etc. These dependencies are modelled using interrelationships, e.g. via subnets.

Description models (max 15pt)

Description fits models?	−−	−	0	+	++
Explanation design decisions?	−−	−	0	+	++
Explanation most important transitions / flows?	−−	−	0	+	++

General remarks: A model without description is not a model. Hence, one should describe how the model works, what are the main activities in the process model, what is the intention of the model? What are the main assumptions and design decisions made in the model? This should be denoted, so that others are able to follow your models and reasoning.

Analysis (max 25pt)

Verification	−−	−	0	+	++	
Reachability / Coverability	−−	−	0	+	++	N/A
Place & Transition invariants ?	−−	−	0	+	++	N/A
Soundness / weak termination	−−	−	0	+	++	
Validation	−−	−	0	+	++	

General remarks: One should show that the produced specification is correct. Correctness is done in two ways: verification, i.e., whether the model is correct, and validation, i.e., whether it is the correct model. For the former analysis techniques taught during the course should be used. At least one should discuss soundness of the created models, i.e., weak and proper completion of the models, and use at least either coverability/reachability analysis or place and transition invariants to analyze the correctness. For validation, one should show that each scenario from the problem analysis can be replayed with the proposed solution.

B Rubrics For the Implementation Guide

BPMN models and database (max 25pt+10pt)

Each process has a clear case?	−−	−	0	+	++
Use of BPMN elements / patterns	−−	−	0	+	++
Transformation Petri nets to BPMN	−−	−	0	+	++
Transformation Data model to table structure	−−	−	0	+	++

General remarks: Each process should have a clear case that flows through the system. For example, a student following an education track, from registration up to diplom ceremony, or a student following a case. A clear case helps in thinking in terms of the process to be followed, and often simplifies the model. The transformation from PN to BPMN should be clear, and if there are transformation choices made, these should be explained under Model Description. The BPMN model should make sense.

Description models (max 10pt)

All models described?	−−	−	0	+	++
Transformation / draft decisions documented?	−−	−	0	+	++
Gateway logic	−−	−	0	+	++

General remarks: To what extent are each of the models explained and detailed? Are there any design decisions made? Are these made explicit, together with the assumptions? Each gateway has some logic to operate on, based on variables. Is each gateway-decision detailed and explained?

Description of the activities (max 10pt)

Forms used for the activity	−−	−	0	+	++
Role permissions	−−	−	0	+	++
Triggers	−−	−	0	+	++

General remarks: Each activity can be performed by one or more users or groups. These need to be specified for each activity, together with the form that the user needs to fill in to complete the activity. If the activity is manual, there should be a clear description. In case triggers are used in the activity (e.g. to populate a form, or to save elements from a form to the database, this needs to be documented as well

Reflection (max 10pt): −− − 0 + ++

General remarks: Although the implementation is an individual assignment, it is allowed to cooperate with your group patner, as you both created the same specification document. However, there are many design decisions that influence the implementation. Most likely you will take different strategies. In the reflection, one needs to compare their work with the team partners, and evaluate where and how the implementations differ.

System implementation (max 25pt)

Functionality supported by the system	−−	−	0	+	++
All models realized in ProcessMaker?	−−	−	0	+	++
All models integrated in ProcessMaker?	−−	−	0	+	++

General remarks: Is all functionality implemented in ProcessMaker? A clear indicator is to check in which notation the BPMN models are created, and which elements these model contain. In the end, a fully functioanl system should be delivered. This requires several processes to communicate via a database. How much of the desired functionality (by the assignment text) has been implemented? Similar, how much of the functionality described by the specification document has been implemented?

Demonstration (max 15pt)

Is the system workable for PTI?	−−	−	0	+	++
Did they present the happy flow?	−−	−	0	+	++
Would you buy the system?	−−	−	0	+	++

General remarks: Based on the demonstration, is the system workable for the client? Would you buy the product for your educational institute? It is not so much as how well it is presented, but how much of the important use cases can be demonstrated and are supported by the system?