Process-mining enabled feedback: “tell me what I did wrong” vs. “tell me how to do it right”

ABSTRACT

Fast advancement of technology has led to an increased interest for using information technology to provide feedback based on *learning behavior* observations. This work outlines a novel approach for analyzing behavioral learner data through the application of *process mining* techniques specifically targeting a *complex problem solving process*. We realize this in the context of one particular learning case, namely, domain modeling. This work extends our previous research on process-mining analysis of domain modeling behavior of novices by elaborating with new insights from a replication study enhanced with an extra observation on how novices verify/validate models. The findings include a set of typical *modeling and validation patterns* that can be used to improve teaching guidance for domain modeling courses. From a scientific viewpoint, the results contribute to improving our knowledge on the cognitive aspects of problem-solving behavior of novices in the area of domain modeling, specifically regarding *process-oriented feedback* as opposed to traditional *outcome feedback* (is a solution correct? Why (not)?) usually applied in this type of courses. Ultimately, the outcomes of the work can be inspirational outside of the area of domain modeling as learning event data is becoming readily available through virtual learning environments and other information systems.

Keywords: teaching/learning modeling, domain modeling, conceptual modeling, process-oriented feedback, modeling patterns, information systems education, process mining, learning analytics.

1. INTRODUCTION

In the modern ICT-driven world the quality of information systems is critical but even more importantly, the value of information systems is largely dependent on their alignment with business strategy and operations. Therefore, in enterprise information systems engineering, enterprise modeling (also sometimes coined as business modeling) is an important step next to (or even prior to) developing an application and technological architecture (TOGAF, 2015).

Typically, an enterprise model addresses different dimensions of the enterprise such as the “what”, “how”, “who”, “when”, “where” and “why” (Bernaert, Poels, Snoeck, & De Backer, 2014; Zachman, 1987). These aspects are captured through different kinds of models such as goal models (capturing the “why”), organizational charts (capturing the “who”), conceptual data models capturing business objects and their relations (“what”), and business processes models (capturing the “how”, related to “who” and “what”). Each of these models captures a particular “view” of the enterprise covering one or more aspects and abstracting away other aspects. While working with different views is a powerful mechanism to master complexity, obviously, those views are not totally independent, but need to present an integrated view of an enterprise and should therefore be consistent with each other. Enterprise modeling is therefore a complex problem solving process, and teaching it requires a careful scaffolding of learning tasks, addressing the modeling of individual aspects first, but ultimately addressing the capability of developing an integrated model consisting of several mutually consistent views.

While different aspects of the enterprise can be described through natural language, it is common practice to capture the different views through the creation of formal (graphical) models because models enable quality control at a level impossible to reach with requirements formulated in natural language (Sikora, Bastian, & Pohl, 2011). Models are represented as diagrams, using specific modeling languages, such as BPMN for business process diagrams, and UML for class diagrams. A particular feature of UML is that this language is based on the principles of object orientation, hence allowing capturing part of the “how” aspects related to behavior of business objects. When using UML for describing business objects, their relationships and their behavior, one therefore speaks about “domain modeling”. As an example, a sales company can be described through a collection of business processes drawn in BPMN, capturing the workflows associated to registering new orders, issuing invoices, shipping goods to customers, handling complaints and so on. An object-oriented domain model will be described through several sub-views. The UML class diagram (the *structural view*) will capture business objects (such as customer, order, invoice, product), and their associations (e.g. each order belongs to exactly one customer; a customer can have zero to many outstanding orders). UML statecharts capture the potential states a business objects can be in (a customer may be blacklisted or not; an order may be invoiced, paid, shipped, ...; a product may be available, out-of-stock, ...) and which actions are allowed in particular states or not (e.g. adding products to an already paid order is not allowed), and how actions may cause transition to a next state (e.g. “pay” causes the transition to the state “paid”). Finally, an interaction model will capture the required coordination across business objects. For example, adding a product to

an order requires checking the state of an order (is the order still modifiable ?), the state of the product (is the product available ?) and potentially even the state of the customer (is the customer not blacklisted ?). If adding the product is permitted, then both the order, and the stock level of the product need to be updated. The statecharts and interaction model together form the **behavioral view**.

Teaching modeling skills to novice business analysts is a challenging task considering that system analysis is by nature an inexact skill and the tacit knowledge the experts gain over time is difficult to transfer to novices. As stated by (Schenk, Vitalari, & Davis, 1998), “in their early careers junior system analysts produce incomplete, inaccurate, ambiguous, and/or incorrect information requirements”. Translating these requirements into correct models consisting of different partial views that need to be consistent with each other, adds yet another layer of difficulty.

The adoption of UML as a modeling language has become very prominent since the introduction of the Model Driven Architecture (MDA) framework and the Model Driven Engineering (MDE) approach to software development (Bézivin, 2006). MDA and MDE recognize that models are the foundation of software system development by focusing on automated code generation from models and hence shifting the focus of software quality assurance from system implementation (software testing) towards system modeling (model verification and validation).

Despite the dominance of UML there is a certain degree of difficulty in understanding a system represented by means of UML diagrams (Bavota et al., 2011; Otero & Dolado, 2004; Siau & Cao, 2001; Siau, Erickson, & Lee, 2005). Previous research has indicated several reasons: among which (1) the level of structural complexity of UML exceeding the limits of human working memory (cognitive load) in terms of the ability for effective information processing (Cruz-Lemus, Genero, & Piattini, 2008; Cruz-Lemus, Maes, Genero, Poels, & Piattini, 2010; Erickson & Siau, 2007; Wilmont, Hengeveld, Barendsen, & Hoppenbrouwers, 2013); and (2) lack of comprehension methodologies (Erickson & Siau, 2007) and, in particular, its impreciseness about the combination of interactive, structural and behavioral aspects together in a single model (Gustas, 2010). Furthermore, (3) it is not easy to find relevant subsets suitable for a modeling goal. The (4) “noisiness” of UML with variety of concepts can result in models that use the misused language concepts in a way not intended for the modeling domain (Buckl, Matthes, & Schweda, 2010). Finally, (5) there is a lack of validation guidance and tool support for model testing (Shanks, Tansley, & Weber, 2003).

In a teaching context, model comprehension difficulties are additionally associated with the insufficient level of experience, i.e. domain knowledge, of novices and as a result their limited cognitive resources to identify relevant triggers for model verification (Bradley, Paul, & Seeman, 2006; Damassa & Sitko, 2010; Schenk et al., 1998). According to complexity analysis by (Siau & Cao, 2001) UML class diagram ranks the highest in complexity among the structural diagrams followed by statecharts among the dynamic diagrams (Carbone & Santucci, 2002; Cruz-Lemus, Genero, Manso, Morasca, & Piattini, 2009; Cruz-Lemus, Genero, Morasca, & Piattini, 2007; Genero, Miranda, & Piattini, 2003) because of their high cognitive and structural complexity (Cruz-Lemus et al., 2008; Cruz-Lemus et al., 2010).

The work presented in this paper relies on the teaching of enterprise modeling, according to the method MERODE¹, an Enterprise Information Systems engineering methodology that has been developed by the Management Informatics research group at the faculty of Business and Economics, KU Leuven (Snoeck, 2014). In MERODE an enterprise model consists of a collection of business process models (making use of BPMN as modeling language) and an object oriented domain model (making use of the UML). To alleviate some of the above-mentioned problems related to UML, MERODE uses simplified versions of the UML class diagram and statecharts. In a Delphi² study by (Erickson & Siau, 2007) identifying the kernel of “essential” UML (i.e. diagrams that are highly used) class diagram and statecharts are found to have the highest usability ranks by practitioners and educators from software industry and academic field with the relative importance rate of 100 %. Furthermore these are also among the top used diagrams present in the context of educational material such as books, tools, courses and tutorials (with percentages of 100% (class diagram) and over 96% (statecharts) (Reggio, Leotta, Ricca, & Clerissi, 2013) also being in the subset of diagrams that provide a support for conceptual modeling goals (Embley & Thalheim, 2012). Furthermore, to maintain the high abstraction level required for enterprise modeling, the interaction model has been replaced by a CRUD-table, a technique borrowed from Information Engineering (Martin, 1990). As a result, a MERODE domain model uses three simplified sub-views: an existence dependency graph (EDG) describing business objects and their associations through a simplified UML class diagram, 2. finite state machines (FSMs), a simplified form of UML statecharts, to capture the individual behavior of business objects, and 3. a CRUD-table to capture business object interactions. . For the given example, the different models would look like in [Figure 1](#). The method is supported by a

² Attempt to reach a reliable consensus by incorporating opposing views from experts in specialized areas

tool JMermaid³. While this tool offers some built-in features for basic consistency checks (Snoeck, Michiels, & Dedene, 2003), the modeler needs to actively cross-validate the different views to ensure an integrated perspective on the system-to-be.

modeling. So far, no research was found that observes a modeling process involving conceptual data modeling, object oriented domain modeling, or modeling tasks comprising multiple diagrams that represent different views of a single system.

Self-regulation is an important concept in learning theory which is based on the cognitive ability to monitor one's own learning process and reflect on what and how one learns by pausing to evaluate the progress and using the outcome of that evaluation to direct one's learning behavior (self-improvement) (Nathan & Sawyer, 2014; Zimmerman, 1990). The concept of self-regulative learning is closely intertwined with research on self-oriented feedback and is found to be strongly correlated with learning outcomes of novices. Yet most learners need help doing effective monitoring and self-regulation and they require feedback to verify their self-evaluation (Nathan & Sawyer, 2014). Usually feedback is not available during modeling activities but is given after a task has been completed, in the form of **outcome feedback** – a minimal form of external guidance (is a solution correct? Why (not)?) (Asterhan, Schwarz, & Cohen-Eliyahu, 2014). Fast advancement of technology has generated increased interest for previously non-feasible approaches for exploiting information technology to provide feedback for enhancing student engagement and learning outcomes by looking into how learning processes (learning behavior) is captured in information systems. Process mining (van der Aalst et al., 2009) is a field of research situated at the intersection of the fields of data mining and business process management which has already built a reputation of being capable of analyzing rich data trails and activity streams in various contexts (De Weerd, Schupp, Vanderloock, & Baesens, 2013). Over the last decade, process mining has attracted a vast amount of researchers who developed tools, techniques, and methodologies to analyze business processes, thereby not relying on, but rather going well beyond, the application of traditional statistical or data mining techniques, by scrutinizing the underlying execution data captured by information systems. This work investigates the perspectives of process-oriented feedback that can be set up by adopting a process-mining analysis view on learning processes. We realize this approach in the context of domain modeling, more specifically targeting feedback needs based on the question of “how novices learn modeling in the context of object-oriented domain modeling and how different learning styles can be associated with their learning outcomes?”.

Our previous work uses process mining techniques to identify modeling patterns of novices that can be associated with worse/better learning outcomes (Sedrakyan, Snoeck, & De Weerd, 2014). This work extends our previous work with a (1) replication study that is enhanced with (2) an extra observation for self-regulative activities such as testing and validation of models by novices within a modeling process. We further elaborate the modeling process patterns of novices as a basis for **process-oriented feedback**. We used a controlled learning environment (JMermaid) to collect data on the learners' modeling activities over one semester period of time. Two empirical studies were conducted spanning two academic years. Domain modeling event data of 36 cases (event logs of students' group works), 28.455 events in total have been subject to a three-dimension analysis using process mining techniques: 1. data analysis at **different abstraction levels** (activities grouped for different model views, fine-grained analysis zooming into each view in isolation), 2. **contrast analysis** to identify differences based on **modeling performance** (best vs. worst scoring groups), 3. **time trend analysis** by making distinction between “early” and “late” sessions. From a process mining perspective, we could make use of two prominent techniques: (1) process map discovery using Disco⁴ and (2) dotted chart analysis in ProM (van der Aalst et al., 2009). The results largely confirm the findings of our previous study showing that certain modeling patterns can indeed be associated with better/worse learning outcomes in terms of reaching a satisfactory model quality (Sedrakyan, Snoeck, & De Weerd, 2014). These patterns can be exploited to enhance teaching guidance for conceptual modeling courses with **process-oriented feedback**.

The remainder of the paper is structured as follows. Section 2 gives an overview of related work and the research contribution. Section 3 describes the experimental setup and research method followed by the data analysis in section 4 and subsequently reports on and discusses the results in section 5. Section 6 presents the findings as a set of modeling and validation process patterns to be used as guidelines to improve teaching guidance. Finally, section 7 concludes the work by discussing the contributions and limitations of the work and proposes some future research directions.

2. RELATED WORK

Studies on *learning quality improvements* are closely related with feedback research (Barber, Bagsby, Grawitch, & Buerck, 2011; Butler & Winne, 1995). While feedback is usually given after a modeling task has been completed, referred to as *outcome feedback*, indicating whether or not results are correct (Butler & Winne, 1995), in feedback research the effectiveness of more informative types of *feedback that guide the process of cognitive activities* is

⁴ Disco is a commercial tool developed by Fluxicon: <http://fluxicon.com/disco/>

highlighted (Butler & Winne, 1995; Narciss, 2008; Shute, 2008). To better understand such feedback needs cognitive problem-solving processes can be observed. Previous research differs from the research presented here in terms of methods used for observation and/or in terms of the kind of model that is targeted by the problem solving task.

In previous empirical studies cognitive processes have been analyzed by means of either pure statistical techniques or in combination with multi-modal emotion tracking techniques such as verbal reporting, eye movement tracking, etc. (Mason, Pluchino, & Tornatora, 2015). A few publications can be found on the application of process mining techniques in an educational context (Mukala, Buijs, Leemans, & van der Aalst, 2015). Some studies focus on observing collaborative learning activities using change logs of shared documents by different students (Southavilay, Yacef, & Calvo, 2010). (Trčka & Pechenizkiy, 2009; Trčka, Pechenizkiy, & van der Aalst, 2010) apply process mining to curricula choices of students to provide recommendations based on learner historical behavioral data and program/course relationships information such as prerequisites and follow up dependencies in curriculum (Should I take now courses A, B, C or C, D ? Am I on track according to my performance path analysis with regard to recommended paths for a specific program ? What are the recommended choices in the curriculum for me ?, etc.). Another example of application of process mining to educational data includes the work of (Pechenizkiy, Trčka, Vasilyeva, van der Aalst, & De Bra, 2009) observing the effects of sequence of answers (specified vs. random sequence) in multiple choice question tests with respect to assessment outcomes. With respect to these studies we focus on observing a **complex problem solving process**, namely the domain modeling as part of enterprise modeling and the cognitive learning process such as **self-regulation** within the task completion process. A bit closer to our study is the work of (Bannert, Reimann, & Sonnenberg, 2014), who use process mining techniques to measure learner motivation by observing self-regulated activities (such as testing and monitoring) with respect to learning outcomes. One difference from (Bannert et al., 2014) is that rather than limiting observations to the effects of such regulative activities on learning outcomes in this study we target the broader context of a process, i.e. can a specific sequence of activities matter?. We investigate the *process-oriented feedback* mechanisms that can be derived by exploiting *modeling patterns* indicative for learning outcomes. This work extends our previous work on identifications of modeling process patterns of novices (Sedrakyan, Snoeck, & De Weerd, 2014) with a replication study that in addition includes an extra observation for self-regulative activities of novices in a modeling process such as testing of models. In particular, we expand our observations to include validation behavior of novices when using simulation of a model to test and verify a semantic conformance of their model with the requirements. These new insights allow to exploit modeling process patterns of novices to provide recommendations on process-oriented teaching guidance.

In terms of type of model, prior studies on observing the process of modeling at conceptual level focused mainly on business process models (Claes et al., 2013; Claes et al., 2012; Pinggera et al., 2013; Pinggera et al., 2014; Recker, Safrudin, & Rosemann, 2010). (Recker et al., 2010) examine how novice analysts design business process models when only having basic design tools such as paper and pencils available, and little to no understanding of formalized modeling approaches. The authors aim at revealing formats (such as text labels and graphical forms) that are well-suited to describe business processes in terms of articulating important design elements (business rules, states, events, activities, temporal and geospatial information) in a process model by statistically correlating the properties with the quality dimensions of the design.

(Pinggera et al., 2013) propose a qualitative model for analyzing business process modeling processes which includes modeler-specific (e.g. expertise, domain knowledge, working memory capacity and the modeler's personality) and task-specific factors (task's inherent complexity or the presentation of the task to the modeler that determine the cognitive load) by also dividing a modeling process into the phases of comprehension (thinking), modeling (materializing a solution) and reconciliation (reorganizing a model, e.g. renaming, changing a layout). The authors also propose methods on how these factors can be quantitatively measured (e.g. identifying approximated measurables such as number of iterations of activities over time, breaks in terms of time when the modeler does not work on a model, etc.). Using data mining techniques (such as clustering) the authors further investigate how these factors can affect modeling styles by also correlating them with model quality dimensions. The authors identify different clusters based on effective use of cognitive resources, increased cognitive load, perceived difficulty, etc. and behavioral patterns such as planning time it takes to start modeling (e.g. some start modeling right away while others invest some time in understanding before they start modeling), or efficiency of use of modeling constructs (e.g. while some create model elements in efficient manner, others require several attempts). Based on cluster analysis the authors identify patterns of modeling such as (1) an "efficient modeling style" characterized by a limited time needed to think about the modeling task, and a fast rate of adding elements to the model; (2) a "layout-driven modeling style" which involves much time in creating a comprehensible layout while being less efficient in creating the model; and (3) an "intermediate modeling style" that is neither particularly efficient nor invests particularly into model layout. (Pinggera et al., 2014) aim at further understanding of the reasons behind these patterns by extending

the research with observations of thinking processes captured by “thinking aloud” method (asking modelers to verbalize their thoughts during modeling) and of modeler's mental effort captured as eye movement data for visual information perception. The patterns are further detailed with modeler-specific and task-specific factors, e.g. authors find that goal-oriented modeling approaches yield better model quality vs. unstructured way of modeling. (Claes et al., 2012) introduces a way for visualizing a modeling process called PPMChart which is based on an improved version of one particular process mining technique called dotted chart. The technique makes use of a chart diagram to visualize a modeling process as a sequence of process properties represented as dots that allows observations of process characteristics such as modeling speed, modeling pauses, amount of modeling elements, amount of modeling operations, etc. Using PPMChart (Claes et al., 2013) investigated whether (1) a modeler's structured modeling style, (2) the frequency of moving existing objects over the modeling canvas, and (3) the overall modeling speed is in any way connected to the ease with which the resulting process model can be understood. The authors find that aspects of a modeler's style can be operationalized and quantified, providing means to distinguish between more and less effective approaches to create a process model. In particular they conclude that structured and goal-oriented modeling approaches are more attractive than diverting one's attention across different parts of a model at the same time.

With respect to previous research on the process of modeling our research differs in several ways. First, we observe the process of object-oriented domain modeling which -compared to business process modeling- adds an additional layer of complexity to the modeling task as it requires developing multiple integrated views. Next, besides tracking only the modeling activities, the learning environment supports the logging of simulation activities enabling tracking the validation and thus self-regulated activities. Rather than using statistical, data mining or dotted chart diagramming from process mining we apply process discovery maps aiming at visualization and investigation of sequential aspects of modeling and validation activities, i.e. if a sequence in which learners implemented certain activities can be indicative for learning outcomes. We also observe a modeling process over a longer period of time (a full semester) compared to observation limited to one session. Finally, we examine modeling processes at multiple granularity levels by abstracting observable activities into different process levels such as modeling activities for structural and behavioral aspects, within each diagram, based on the usage of modeling construct/element as well as by looking within session processes and making distinction between early and late sessions of modeling processes.

3. METHODOLOGY

This work aims at investigating learning behavior patterns that can be associated with learning outcomes of novices in the domain of object-oriented domain modeling. We realize this goal in the context of the course “Architecture and Modeling of Management Information Systems”⁵. The course targets at final year master level students with heterogeneous backgrounds from the Management Information Systems program at KU Leuven. The goal of the course is to familiarize the students with modern methods and techniques of *Object-Oriented Analysis and Design for Enterprise Information Systems* and to let them acquire sufficient skills of developing an enterprise model as basis of an enterprise information system.

In order to achieve our goal we opted for an empirical study. During the semester students are assigned a group project (approximately 5 page specification document based on real-world requirements (Sedrakyan, Snoeck, & De Weerd, 2014)) in which they are faced with a task of constructing a semantically correct conceptual model that reflects the structural and dynamic view of a given domain description. The project has two deadlines: a deadline for submitting an intermediate solution to receive peer feedback, and the final submission deadline by the end of the semester. During the task completion process they make use of the JMermaid modeling environment. The models designed in JMermaid are also executable (transformable into software code). The integrated MDE-based simulation tool (Sedrakyan & Snoeck, 2013c, 2014c) allows generating fully functional prototype applications out of MERODE models. This allows to test a model by testing its corresponding prototype application. The simulation tool is adapted to limited technical expertise of novices allowing producing a prototype with just a single click. In this way, the use of the tool allows limiting the extraneous cognitive load from complicated tools and notations (Crapo, Waisel, Wallace, & Willemain, 2000). The generated prototype application simulates the structure and behavior of a designed model providing a novice with the advantage of prototyping-based testing of models as well as experience-based learning. The generated applications are additionally augmented with a feedback that links the prototype-based testing results to their causes in a model solution, e.g. if an execution of a business event fails a user receives a textual and graphical explanation prompting the parts of a model that cause the error. Empirical studies show that this simulation feedback can significantly improve the learning outcomes (Sedrakyan & Snoeck, 2012, 2013a, 2014a,

⁵ The course's page can be found on <http://onderwijsaanbod.kuleuven.be/syllabi/e/D0171AE.htm>

2015; Sedrakyan, Snoeck, & Poelmans, 2014). Finally, the modeling tool has been expanded with an experimental logging functionality allowing to record data on modeling and prototyping (also referred to as simulation) activities. The described advantages of the MERODE make the JMermaid environment particularly suitable for the goals of this study.

We conducted two studies with participation of 165 students (86 in the original and 79 in the replication study) randomly assigned to 36 groups. Modeling behavior (activities) data have been collected through logging functionality of the modeling environment throughout a semester of study while students were working on their group's project. We used process mining techniques to observe novices modeling process and synthesized modeling patterns of novices that are indicative for learning outcomes. Identification of patterns allows demonstrating the applicability of process mining techniques in observing cognitive learning processes, while also providing insights for process-oriented guidance in teaching conceptual modeling courses.

For the collection and interpretation of data, different concepts relating to the modeling process needed to be operationalized. To assess the quality of the result of a modeling process, we assess the outcome based on its **semantic quality**. Semantic quality refers to the level to which the statements in a model reflect the real world in a valid and complete way (feasible completeness, feasible validity) (Lindland, Sindre, & Solvberg, 1994; Nelson, Poels, Genero, & Piattini, 2012). This semantic quality was evaluated by the teacher.

Since it would not be possible to actually track a human mind in our study, the concept of **modeling effort** was operationalized as the number of modeling activities as approximation of modeler's mental effort. Likewise, since it would not be possible to track the actual process of model validation in our studies, we made use of the simulation activities motivated by the fact that 1. simulation serves as an instrument for model validation and 2. the simulation feature used for this study was empirically proven to have a significant positive effect on novices' capabilities for validating the semantic conformance between a model and requirements statements (Sedrakyan & Snoeck, 2012, 2013a, 2015; Sedrakyan, Snoeck, & Poelmans, 2014). The concept of **validation effort** is therefore operationalized as the number of simulation activities throughout the modeling process. To identify effort related to cross-validation between the different subviews of the model, activities relating to the UML class diagram have been labeled as pertaining to the structural model view, whereas activities relating to the statecharts and interaction diagram have been labeled as pertaining to the behavioral model view.

Finally, we refer to **modeling patterns** as a sequence of modeling activities that can potentially affect a modeling process outcome with respect to its semantic quality in the context of learning.

4. DATA COLLECTION

4.1 SAMPLING

Students were randomly assigned into groups of 2-4 students. In 2013-2014 academic year, this resulted in 22 groups, out of which 20 handed in a solution. In the academic year of 2014-2015, there were 20 groups, out of which 19 handed in a solution. "Outlier" groups, such as students working alone that were retaking the course or that didn't participate to the lectures for reasons of illness (marked with an asterisk sign in *Table 1*) were left out. This resulted in 36 cases for our analysis: 19 cases for the original study in the academic year 2013-2014 and 17 cases for the replication study in 2014-2015. The outcomes of group works were scored with respect to the semantic quality of the submitted model solution. Based on the final score (min. score = 0; max. score = 20) we further classified the cases into **best performing** and **worst performing** groups, leaving out one group with a performance just on the border of the selection criteria for the cluster (marked with a ° in *Table 1*). This resulted in two clusters in each study consisting of 5 cases which we will further refer to as best performing (scores above 15 in explanation column of *Table 1*) and worst performing groups (scores below 11.5 in explanation column of *Table 1*). In the replication study we additionally made use of intermediate solutions (scores between 12 and 15 in explanation column of *Table 1*) and make use of this cluster for gaining more insights on how worst performing students may be guided to achieve the minimum threshold to pass the course. We further refer to these clusters of the replication study as **satisfactory performing** groups (5 blue cases).

Table 1: Group performance and cluster composition

2013-2014		2014-2015		Classification	
Group	Final Score	Group	Final Score		Explanation
3	19	7	19	B	Score ≥ 15
19	18	5	17.2	B	
2	16	2	16.6	B	
8	16	11	16.6	B	
15	15	8	15.8	B	
18	14	9	14.3	S	Score ≥ 12 in the replicated experiment
20	14	1	13.8	S	
4	13	15	13.4	S	
11	13	19	12.6	S	
12	13	10	12.4	S	
10	13				
13	13				
16	13				
9	12				
		3	11.6		Score < 11.5
		6	11.4	°	
14	11	4	11.4	W	
6	10	18	8	W	
5	8	14	7.6	W	
7	8	12	6.8	W	
1	6	13	5.2	W	Outliers: problematic students, working alone or not handing in task
21	5	20	8.2	*	
13	/	17	5	*	
17	/	21	/	/	

/ groups that didn't hand in a task
 * outlier groups - students working alone that were retaking the course or that didn't participate to the lectures for reasons of illness
 ° group just on the boundary of the selection criterium, left out to keep 5 groups per cluster

4.2 EVENT LOGS

In order to observe the modeling process (how the novices created their models) interactions with the modeling tool have been logged. As modeling manifests itself in the creation of modeling elements, in our logs we capture the modeling process as a sequence of create, edit, delete, undo, redo, and copy events. These events are further abstracted into the categories CREATE and EDIT (grouping events edit, delete, undo, redo, copy). In the replication study the logging functionality was expanded to additionally capture model testing and validation activities. This was achieved by integrating a simulation feature into the modeling environment (Sedrakyan & Snoeck, 2013c, 2014c) that allows verifying the semantic conformance between a model and requirements. The inclusion of the simulation feature allowed to additionally log the simulation cycles (each time a model simulation was performed by a user) in the model creation process, logged as SIMULATE event. *Table 2* shows a sample of the data used (called event log) with each row representing one event.

Table 2: Extract from event logs (the complete data set contains 28.455 events from 36 cases from 2 academic years)

Timestamp	Group ID	Session ID	Session Type	Score	Event	Element	Model View	Diagram Type
10/11/14 10:54:00 AM	1	Session 1	Early	8	create	object	S	EDG
10/11/14 10:54:16 AM	1	Session 1	Early	8	create	attribute	S	EDG
10/11/14 10:55:55 AM	1	Session 1	Early	8	create	association	S	EDG
15/11/14 11:05:05 AM	1	Session 2	Early	8	create	event	B	OET
15/11/14 11:05:23 AM	1	Session 2	Early	8	create	state	B	FSM
15/11/14 11:06:01 AM	1	Session 2	Early	8	create	transition	B	FSM
17/11/14 01:02:34 PM	1	Session 3	Early	8	edit	object	S	EDG
17/11/14 01:02:41 PM	1	Session 3	Early	8	edit	attribute	S	EDG
12/12/14 01:08:43 PM	1	Session 4	Late	8	simulate	-	-	-

Events can be defined at different levels of abstraction by making use of a combination of one or more attributes (see columns "Element" in *Table 2*). This enables analysis of different aspects of the modeling process at different

levels of abstraction which makes it easier to devise intuitive visualizations using process mining techniques. For example, at a high abstraction level, we can specify events according to the model view (column “Model View” in *Table 2*), i.e. whether the element type can be attributed to modeling structural (S) or behavioral (B) characteristics. The classification as structural (S) or behavioral (B) (column “Model view” in *Table 2*) was achieved by grouping element types according to the diagram type they belong to and subsequently the model view that the diagrams represent. According to the MERODE methodology the structural representation of a model includes the class diagram (EDG) and the dynamic representation of a model includes the statechart (FSM) diagrams and the interaction model (OET). Consequently, the elements representing the EDG (e.g. objects, attributes, associations) are classified under structural view (S), and the behavioral view (B) combines the elements representing OET (business events indicated as “event” under the column “Element”), and the elements for FSM view (states and transitions). For a more specific view, event names can be supplemented with other information fields, for instance with the modeling diagram type (column “Type” in *Table 2*) where the class diagram⁶ (S//EDG) represents the creation of business objects and their properties, the object event table (B//OET) represents business events and rules for interactions, and finite state machines (B//FSM) represent lifecycles of objects with sequence constraints as basis for the behavior of a system. At the most detailed level, we make use of the activity name (column “event” in *Table 2*) and the targeted modeling construct (column “element” in *Table 2*) potentially in combination with other information, for a fine-grained quantitative and qualitative analysis of modeling activities and patterns.

Groups are identified by group id (column ‘Group ID’ in *Table 2*) and the score they obtained for the final solution (column “Score” in *Table 2*). Finally, so as to be able to distinguish between modeling phases, we further identified sessions based on the events’ timestamps to make a distinction between EARLY and LATE sessions. This allows identifying data at two different levels: based on (1) group id and (2) session level, by combining group id and session id (column “Session ID” in *Table 2*).

4.3 THREE-DIMENSIONAL ANALYSIS

Event logs of students’ group works have been analyzed using process mining techniques. Event data of 36 cases (28.455 events in total) have been subjected to a three-dimensional analysis (see further).

- 1) **Hierarchical**: an investigation of top-level models discovered from the data where cases are regarded as sequences of structural (S) and behavioral (B) activities (either fully abstracted or appended with EDG/OET/FSM diagram views and CREATE/EDIT activity types, followed by a session-level, fine-grained analysis of both structural and behavioral activities in isolation. For the replication study we additionally made use of SIMULATE event to observe testing and validation behavior.
- 2) **Modeling performance**: a contrast analysis was performed to identify differences between best and worst scoring groups. For more insights in the replication study we also specifically investigated the difference between insufficient (worst) performance and “just” sufficient (satisfactory) performance, as this allows identifying the guidance required to help students achieving the minimum required knowledge to obtain a credit for the course.
- 3) **Time trend analysis**: by making a distinction between “early” and “late” sessions.

This analysis was carried out using Excel, Disco, and ProM. From a process mining perspective, we could make use of two prominent techniques: (1) process discovery using Disco⁷ and (2) Dotted chart analysis in ProM (van der Aalst et al., 2009). We further elaborate previous findings with quantitative and qualitative fine-grained analysis.

5. DATA ANALYSIS AND RESULTS

5.1 SAMPLE REPRESENTATIVENESS

Analysis of the personal characteristics of the students resulted in the demographics presented in *Table 3*. Previous knowledge of participants was measured by means of pre-experimental tests before they started working on their group project. Different dimensions of knowledge with respect to data and behavior modeling was defined. The dimensions were further categorized into modeling knowledge levels (low, intermediate, advanced). The pre-experimental test contained two-three questions per dimension. Based on the test performance the baseline knowledge (no/little/moderate/extensive) was established. Students scoring below the basic level were classified as having no knowledge.

⁶ In MERODE we refer to class diagram as existence dependency graph (EDG) due to the fact that relationships are translated into existence dependencies

⁷ Disco is a commercial tool developed by Fluxicon: <http://fluxicon.com/disco/>

Table 3: Summary of demographics

	Original study	Replication study	Entire population
Gender			
Male	74 %	72 %	73 %
Female	26 %	28 %	27 %
Age distributions			
Min age	21 year	22 year	21 year
Max age	42 year	49 year	49 year
AVG age	25,6 year	25,8 year	25,7 year
<= 25	62 %	68 %	65 %
25 < 35	28 %	24 %	26 %
>= 35	10 %	8 %	9 %
Previous knowledge of modeling			
No knowledge	30 %	46 %	38 %
Little knowledge	33 %	22 %	27,5 %
Moderate knowledge	29 %	26 %	27,5 %
Extensive knowledge	8 %	6 %	7 %
Scores of group works (on 20 scale)			
Min score	5	5	5
Max score	19	19	19

5.2 A BIRD'S EYE VIEW ON THE MODELING PROCESS

Multiple filtering rules were applied to construct modeling process maps at different abstraction levels. The first step in our analysis approach targets the creation of a top-level map by filtering the dataset based on group number as case identifier along with timestamp and corresponding activities (Create New Model, modeling activities within the structural view (S), modeling activities within the behavioral view (B)). In contrast to statistical tools, process mining techniques make it easier to observe processes at different levels of granularity.

Figure 2. Top-level process map representing the modeling process of the 5 worst performing groups (replication study 2014).

Figure 3. Top-level process map representing the modeling process of the 5 satisfactory performing groups (replication study 2014).

One of such techniques is the path filtering in Disco. It allows to make abstraction by removing/hiding less frequent paths for a specified percentage range making the process maps visually easier to interpret. Moreover, the abstraction allows to identify a number of process differences between worst, best and satisfactory clusters since process graphs can be used that only reveal the most substantial information. Note that due to the use of this path filtering technique differences between the numbers of activities shown on incoming arrows and event boxes on the process maps may be observed, however should not be considered as wrong. It only indicates that low-frequent behavior was excluded.

Figure 4. Top-level process map representing the modeling process of the 5 best performing groups (replication study 2014).

Although the high level of abstraction prevented the extraction of ample insights in the original study as the process maps of the best and worst performing groups looked identical showing only iterations between structural (S) and behavioral (B) views, the fully abstracted view on the modeling process did reveal interesting patterns from a quantitative perspective. The number of activities for modeling the behavioral aspects in both groups exceeded the number of activities for modeling the structural aspects (approximately 60% of all activities). The second observation from the graphs revealed a higher (almost double) frequency of modeling activities for the same time period in the best scoring groups. This suggests that the more the groups are engaged in modeling activities the better the modeling process output becomes. The reader is referred to (Sedrakyan, Snoeck, & De Weerd, 2014) for more details on our previous study. From a quantitative perspective this tendency is also confirmed in the replication study. The resulting process maps for worst, satisfactory and best performing groups for the replication

study (2014) are presented in Figure 2, Figure 3, and Figure 4. From a process perspective, the original study does not allow the revelation of any significant differences between groups with better and worse learning outcomes. Both process maps in the original study for worse and best performing groups show switches between structural and behavioral views. The inclusion of the “simulate” event in the replication study however reveals an interesting observation. In the worst and satisfactory performing groups the simulation cycles mostly result in further activities for the structural view, as can be observed from the arrows going back and forth between simulation activities and activities for the structural view in Figure 2, Figure 3. In contrast, in the best performing groups the simulation cycles mostly result in further activities for both behavioral and structural views (Figure 4). It seems that the activities following the simulation target at adaptation of models to address discovered problems in their solutions. The process map for the best performing groups seems to suggest that students in the best performing groups put a lot more effort into the validation of behavioral aspects of their models by means of simulation (Figure 4). The process map of the best performing group also suggests that they initiate simulation mostly when they work with behavioral aspects presumably targeting at testing the behavior, but later they also came across problems that can be addressed by structural changes. Worse and satisfactory performing groups in contrast seem to limit themselves to testing the structural aspects of their models. As Figure 2 and Figure 3 show that the activities preceding and following simulation cycles are mostly to/from the structural view, we can deduce that their simulation effort is targeting mostly at the validation of structural consistency such as associations and cardinalities. Such approach limits testing activities to only the default behavior generated by the tool, i.e. create/end events (e.g. create an instance of object A, end an instance of the object A, ...) without further testing of a more complex behavior (e.g. sequences in which object instances can interact throughout their lifecycle, for instance constraining that an invoice should be paid once an order is delivered,...). From the quantitative perspective, the number of simulation activities in the best performing groups for the same time period is observed to be significantly larger than those of the worse performing groups for the same time period. This suggests that the more students are engaged in testing and validation activities the better the modeling process output becomes. We can also conclude that best performing groups are able to improve the quality of their solutions because by testing the behavior it is more likely that they discover global problems that result from the interplay of structural and behavioral aspects while this is not the case the other way around. Such behavior-oriented testing seems to result in a broader validation coverage. In contrast, by limiting testing to structural aspects, students in worse performing groups are in most cases ignorant about interplay issues and can in the best case only discover wrong or missing associations between objects.

5.2.1 WITHIN SESSION ANALYSIS

Next, we zoomed into the modeling process by investigating processes at the session level instead of the case level. In contrast to the top level process maps that target a modeling behavior over the entire semester, i.e. activities between “Create new model” and “Submit”, this allows to observe how students behave within “open” and “close” events of the project’s file. In this way, we make it feasible to analyze the modeling process at a more fine-grained level and we can look at evolution over time by contrasting modeling activities in earlier vs. later sessions. A first way of digging deeper into the log data is achieved by detailing the event types or activity names with information about their CREATE/EDIT categorization. The session level analysis did not reveal significant differences between clusters from the process perspective which confirms the results of our previous study. However we could discover different approaches by additionally making use of the activity filtering technique in Disco and filtering away some events in the dataset (e.g. events in which students did insignificant changes such as modifying an attribute name, redo/undo, sorting, etc.) This resulted in the process maps shown in Figure 5, Figure 6 and Figure 7 for the replication study (due to the use of activity filtering some differences for the activity numbers between top-level and session level process maps for the same clusters may be observed).

Figure 5. Session-level process map representing create, edit & simulate modeling activities of the 5 worst performing groups (replication study 2014).

The process maps of the original study showed that a linear approach of modeling was prevailing within each session both in best and worst performing clusters, i.e. students preferred to concentrate on one task at a time working on either structural or behavioral aspects of a system within one open/close delimited session. This tendency of a linear way of working is partially rediscovered in the replication study. The process map of worst performing groups (Figure 5) shows that within one open/close session they work sequentially first for modeling structural aspects and afterwards for behavioral views ($S \rightarrow B$) and/or they iterate between “create” and “edit” activities within one model view. We could partially attribute this to the sequence of topics taught in the course (modeling structural aspects, followed by modeling behavioral aspects), i.e. whatever the students learned in class they tried that out without much variation of applying knowledge from different topics or revisiting their models when more advanced concepts have been introduced. This could also indicate their lack of skills to identify right triggers for verification of their models for interplay issues between structural and behavioral views, i.e. understanding that changes in one view of a model will have propagation effects in other view(s) of a model (e.g. if an association between A and B states that

each B-object is mandatorily related to exactly one A-object, this implies a sequence of first creating an A-object, for which then a B-object can be created).

Figure 6. Session-level process map representing create, edit & simulate modeling activities of the 5 satisfactory performing groups (replication study 2014).

Figure 7. Session-level process map representing create, edit & simulate modeling activities of the 5 best performing groups (replication study 2014).

In the sessions in which simulation was used, simulation cycles are starting activities presumably targeting a general testing of the model. Simulation cycles are further followed by “create” activities for the structural view (*SIMULATE* → *S*) most likely targeting to add discovered missing elements. For satisfactory and best performing

groups (Figure 6 and Figure 7 respectively) the tendency of working on one task within a session is dominant for “create” activity types both for structural and behavioral views. However for “edit” activities the session level process map shows a “loop” between views, with intermediary use of simulation: the resulting process map for these groups shows modifications both for structural as well as behavioral views within open/close sessions that presumably result from validation (simulation) activities in the same session.

In contrast to worse performing groups this sequence of activities suggests that best performing groups have an intention to validate recent changes in a model. Moreover, when validating their solutions better performing groups seem to target at either testing the interplay aspects between views or revealing interplay issues while testing their solutions. This tendency of testing an interplay between views is in addition revealed from the sequence in which preceding/following activities of simulation cycles happened in better groups. The pattern $B \rightarrow SIMULATE \rightarrow S$ suggests that they could discover structural issues while testing. As in the case with top-level models in section 5.2 the findings of within-session analysis suggest as well that validation activities are significant in revealing differences in the modeling process between better and worse performing groups. The prevailing difference between better and worse performing groups as can be observed from the session level process maps is the *sequence* of activities within sessions in which simulation is applied. In comparison with the top level process maps however, this pattern, namely frequent switching between different views with a simulation activities positioned in between these activities, manifests with a lower frequency at session level.

In summary, we can conclude that the sequence patterns of activities in sessions where simulation occurs suggest that worse and better performing groups differ in terms of their intention of using validation techniques. The sequence of activities in worse performing groups ($OPEN \rightarrow SIMULATE$) suggests that they mostly use simulation intending at a general test and as a consequence discover missing elements as can be concluded from “create” activities following simulation cycles. Ultimately, both the session level and top-level process maps of the replication study also suggest that *the frequency and the way modeling activities are performed* is not the *sole indicator* for learning outcomes (as suggested by the results of the original study). As can be observed by comparing the process maps of the groups with worst and satisfactory performance, they put almost similar modeling effort in their modeling task. Moreover, even less validation effort can be observed in the groups with satisfactory performance as compared to the groups with the worst performance. Based on the top level and session level maps we can however conclude that it’s not only the *frequency of engagement* in modeling activities but rather the *validation process pattern (how validation of models is performed)* in terms of *intended target* (general test vs. targeted test of a specific behavior) that can potentially lead to differences in learning outcomes. This result confirms the added value of process mining compared to purely descriptive statistics.

5.2.2 GLOBAL PROCESS ANALYSIS

Next, we further subdivided the structural and behavioral views by adding event types based on the S/EDG, B/OET, B/FSM typification. The resulted process maps of the original study are presented in Figure 8 and Figure 9 and for the replication study in Figure 10, Figure 11 and Figure 12.

Figure 8. Top-level process representing modeling activities for class diagram (EDG), business events (OET) & event sequence constraints (FSM) – Worst 5 groups (original study 2013)

Figure 9. Top-level process map representing modeling activities for class diagram (EDG), business events (OET) & event sequence constraints (FSM) – Best 5 groups (original study 2013)

Figure 10. Top-level process map representing modeling activities for class diagram (EDG), business events (OET) & event sequence constraints (FSM) – Worst 5 groups (replication study 2014)

Figure 11. Top-level process map representing modeling activities for class diagram (EDG), business events (OET) & event sequence constraints (FSM) – Satisfactory 5 groups (replication study 2014)

Figure 12. Top-level process map representing modeling activities for class diagram (EDG), business events (OET) & event sequence constraints (FSM) – Best 5 groups (replication study 2014)

being centred in between the activities for class diagram (structure) and statecharts (behavior) in best performing groups.

The second difference between the worse and better performing groups that can be observed from the process maps is the difference of the *sequence* of activities. The activity sequence of $EDG \rightarrow OET$ in best performing groups suggests that better performing groups are more focused on capturing high level behavior from requirements

The process maps of the original study that capture the modeling behavior of the entire semester (with open/close events filtered away) show a sequential modeling pattern within the worst performing groups for the entire semester (Figure 8). In contrast, the process map for the best performing cluster showed alternations between modeling activities, i.e. students worked on model views in parallel or revisited and adapted different views of the system afterwards (see Figure 9). In the original study these differences led us to the conclusion that best performing groups were more verification and cross-validation oriented, thus conveying a more iterative modeling approach. However, from our original study we were not able to identify what could cause such “extra” iterative activities observed in the process map of the best performing groups. The additional observable events added in the replication study (more specifically the simulation cycles) however allow to zoom into these loops. In the replication study we are not able to rediscover the pure sequential pattern as was discovered from the process map of the worse groups of the original study. However, a sequential approach of working was found to be dominant in terms of sequencing between the views ($EDG \rightarrow FSM \rightarrow OET$). The first notable difference with the process map of the better performing groups is the concentration of simulation activities around class diagram (structure) vs. simulation cycles

documents into their models, i.e. identifying business events per business object as we can conclude from the nature and definition of OET⁸. In contrast, the dominance of the *FSM* → *OET* sequence in the process map of the worse performing groups suggests that they are more focused on fine-grained behavior and are more reactive in their modeling approach, i.e. focusing on individual statecharts and supplying business events mostly to support statechart modeling.

This tendency of supplying events to support statechart modeling in addition can be revealed from the fact that the sequence *EDG* → *OET* (i.e. intention to supply business events per business object) is absent too in worse performing groups. While we can only make an assumption for this tendency in their modeling approach, we were able to find an evidence later with qualitative analysis. Interestingly, the process diagram of the satisfactory groups (Figure 11) shows some kind of “mix” of modeling approaches of worse (*EDG* → *FSM* loop) and best performing groups (*EDG* → *OET* loop, *FSM* → *SIMULATE* → *EDG* validation pattern). In contrast with worse performing groups, the process diagram also shows a closed iterative loop between all three model views. In summary, such dominant process patterns confirm our findings from top-level and session-level model analysis presented in previous sections, that it is not purely the modeling approach but, rather in addition the way novices validate their models (both in terms of frequency, the target view of their validation activities and the sequence in which validation activities happen) that can potentially make a difference in a quality of a modeling process outcome and learning achievements.

5.3 TIME-TREND ANALYSIS: EARLY VS. LATE SESSIONS

Next, we performed a session level analysis but additionally subdivided the modeling period into early and late sessions. This was achieved by using an identifier for the first and second half of the period between the start date when the task was made available to students and its submission deadline. The procedure resulted in 25 early and 53 late sessions (number on the start arrows on the process maps) in the worst performing groups (Figure 13, Figure 14), 16 early and 41 late sessions in the satisfactory performing groups (Figure 15, Figure 16) and 38 early and 61 late sessions in the best performing groups (Figure 17, Figure 18). The resulting process maps are shown below.

Unlike the top level process maps (frequent switches between structural (S) and behavioral (B) views), a sequential working pattern was observed for session-level process maps within both early and late sessions for all clusters: iterations are found within one view (e.g. structural or behavioral) but not so much between the views (ingoing or ongoing arrows but not loops (Figure 13, Figure 14). In general this confirms the earlier results found in the original experiment. What is found to be different between the clusters in the replication experiment is the position of simulation activities. Simulation activities in worst performing groups within both early and late sessions are preceded by “open” activity and followed by activities either for structural views (early sessions) or for behavioral view (later sessions). The process maps of satisfactory and best performing groups (Figure 15, Figure 16, Figure 17, Figure 18) seem to suggest that the difference with worse performing groups is that within one open/close session their validation effort is targeted at both model views. Moreover, the positioning of simulation between views (*B* → *SIMULATE* → *S* and *S* → *SIMULATE* → *B*) seem to suggest that their validation activities are centered around interplay issues between model views, which is indicative for a more cross-validation oriented approach. In comparison with the satisfactory performing cluster the best performing cluster shows a stronger support for the pattern. As discussed earlier in section 5.2.1, such sequence of activities also suggests that best performing groups might have targeted at testing recent changes rather than a general test. However in contrast to top level analysis this pattern (simulation in between views or targeting both views) does not show a significant dominance at session level.

Best performing groups, in addition, seem to be more validation oriented both in their early and late sessions (Figure 17, Figure 18) as can be concluded from the distribution of simulation activities across the early and late sessions (almost equal validation effort of 31 vs. 47 cycles) as opposed to worst (12 vs. 39 cycles) and satisfactory performing groups (6 vs. 30 cycles) whose validation effort seem to be mostly concentrated in their late sessions.

⁸ Object event table (OET) is specific to MERODE methodology that allows to populate a CRUD table as a list of business events by also specifying their type, such as “create” event, “end” event or “modify” event.

Figure 13. Session-level process map representing create, edit & simulate modeling activities of early sessions – Worst 5 groups (replication study 2014)

Figure 14. Session-level process map representing create, edit & simulate modeling activities of late sessions – Worst 5 groups (replication study 2014)

Figure 15. Session-level process map representing create, edit & simulate modeling activities of early sessions – Satisfactory 5 groups (replication study 2014)

Figure 16. Session-level process map representing create, edit & simulate modeling activities of late sessions – Satisfactory 5 groups (replication study 2014)

Figure 17. Session-level process map representing create, edit & simulate modeling activities of early sessions – Best 5 groups (replication study 2014)

Figure 18. Session-level process map representing create, edit & simulate modeling activities of late sessions – Best 5 groups (replication study 2014)

In terms of the distribution of modeling effort the process maps show that the most prevailing difference is the distribution of modeling effort put into the structural view. The process maps show that the best performing groups

(Figure 17, Figure 18) are more active in their earlier sessions ($S \backslash \backslash \text{CREATE} = 608$, $B \backslash \backslash \text{CREATE} = 536$, with a moderately higher number of “create” activities found for structural view) while the number of modeling activities showed a moderate tendency to decrease in later sessions (the largest number of “create” activities found for behavioral view $B \backslash \backslash \text{CREATE} = 516$, while there is a significant decrease for structural view $S \backslash \backslash \text{CREATE} = 203$). The only exception is their validation effort showing a moderate increasing effect in their late sessions. It seems the students first targeted at capturing relevant data needs from textual requirements into their model and continued to model/adapt their model in later sessions as can be concluded from the intensive use of validation activities in their late sessions. In contrast to the best performing groups, worse and satisfactory performing groups remained active in later sessions while also showing some increased effort (increased number of create and edit activities as compared to their early sessions). This confirms the earlier result of the original study where we also noticed a decrease of modeling activities for the best groups and an increased effort for the worse performing groups.

5.4 DISTRIBUTION OF MODELING EFFORT OVER TIME

We next applied a dotted chart diagram to observe the differences in terms of frequencies and gaps between session activities over the semester. The context information necessary to read the figure is presented in **Table 4**. The dots show the modeling sessions in two colors: green (prevailing number of structural aspects) and blue (prevailing number of behavioral aspects).

Table 4: Context information

	Original study 2013	Replication study 2014
Assignment available (AA)	08/11/2013	24/10/2014
Deadline for intermediate solution (D1)	22/11/2013	20/11/2014
Exercise session on testing peer solution (T)	22/11/2013	21/11/2014
Peer feedback deadline (PD)	02/12/2013	01/12/2014
Last class (LC)	16/12/2013	15/12/2014
Deadline for solution (D2)	20/12/2013	19/12/2014

The red dots represent the “create new model” event (the first event in every case) which in some cases are not shown because of overlap (Figure 19). Figure 20 shows the chart for the replication study (“create new model” marked with grey, simulation activities with red color). Figure 21 shows the distributions of simulation/validation activities over time (simulations marked with red, start and submit activities marked with grey). The worst performing groups are indicated by the label “W”, and the best performing groups by “B”. For the replication study the additional satisfactory performing groups are indicated as “S”.

Context information is shown by vertical arrows labeled with the abbreviations from **Table 4**. The number of modeling sessions in the worst performing groups is significantly less than that of best performing groups. In addition significant time gaps can be observed between the sessions for worse performing groups. In contrast, best performing groups are distinguished by more frequent modeling sessions.

Figure 19. Distribution of modeling effort over time: original study 2013

Figure 20. Distribution of modeling effort over time: replication study 2014

The context information (see **Table 4**) shows that both best and worst performing groups are sensitive to deadlines (submission for peer review and final submission) in terms of being more intensively engaged in modeling activities right before the deadlines. In the best performing group however more engagement is also found in between the deadlines.

Figure 21. Distribution of validation activities over time (replication study 2014)

groups are more systematic in their validation activities as opposed to little validation cycles in worse performing groups. A reason behind the lesser use of simulation could presumably be a lack of capabilities to identify right triggers for model verification and developing scenarios for testing as suggested by research results on simulation-based testing (Sedrakyan & Snoeck, 2014a).

Table 5: Modeling process time statistics in Disco

Process time indicators	Original study (2013)			Replication study (2014)		
	Worse cases	Satisfactory cases	Best cases	Worse cases	Satisfactory cases	Best cases
Mean case duration (d)	18.6 d	25.9 d	26.5 d	27.3 d	21.5 d	40.9 d
Start of modeling activities	13.11.13 14:21:38	16.11.13 14:47:00	08.11.13 14:36:12	09.11.14 18:25:17	15.11.14 15:13:07	28.10.14 15:22:42
End of modeling activities	20.12.13 12:14:21	20.12.13 12:16:57	20.12.13 07:36:41	19.12.14 11:33:46	19.12.14 10:18:27	19.12.14 14:02:15
Mean modeling time	7.9 h	15.5 h	22.2 h	10.7 h	27.6 h	13.2 h
Mean simulation duration (d)	NL	NL	NL	11.9 d	15.6 d	30 d
Start of simulation activities	NL	NL	NL	16.11.14 13:16:58	17.11.14 14:01:08	03.11.14 09:32:44
End of simulation activities	NL	NL	NL	18.12.14 18:34:20	19.12.14 03:47:20	18.12.14 19:18:04
Mean simulation time	NL	NL	NL	31.1 h	46.8 h	49.3 h

d=day; h=hour;

NL = Not logged : Simulation environment was not part of modeling environment in the original study.

Process time indicators in Disco seemed to suggest a positive correlation of modeling time variables with score (**Table 5**). While on average best performing groups show better time indicators, their total modeling time suggested by Disco is significantly less than modeling time of satisfactory groups (“Mean modeling time” in **Table 5**). According to the process time statistics best performing groups are however distinguished by earlier engagement in the modeling part of their solutions (1-2 weeks in the original study, and 2-3 weeks in the replication study) than worst and satisfactory performing groups (“Start of modeling activities” in **Table 5**). Best performing groups are also distinguished by much earlier engagement in validation techniques both for semantic quality aspects (“Start of simulation activities” in **Table 5**). Overall, best groups also spent more time on validation activities (“Mean simulation time”) and are more systematic in using validation techniques (“Mean simulation duration”). “Mean case duration” in **Table 5** shows that modeling sessions of best performing groups are spread across a larger coverage of days (26.5 and 40.9 days in original and replication studies respectively) as opposed to worse performing groups (18.6 and 27.3 days in original and replication studies respectively). This suggests that the modeling process of best performing groups could have benefited from frequent engagement and exploiting their more recent knowledge of modeling and validation concepts learned in class that is still active in their working memory. Interestingly, as in case of process maps, the process statistics of satisfactory groups in the replication study seem to suggest a “mix of approaches” from the best and worst performing groups: their mean case duration is less than of the worse performing groups (21.5 vs. 27.3 days), however this is compensated by more actual time spent on modeling (27.6

In addition, after the peer review, best scoring groups seem to react to the given feedback and to adapt their models to address the comments, while the worst performing groups didn’t “react” to peer comments. While activities in both groups were observed to increase right before the deadlines, the distributions of activities over the entire semester shows that best performing groups model and simulate systematically as opposed to worse groups whose modeling and simulation activities are less frequent and in the majority of cases concentrations of activities are observed right before the deadlines.

Likewise, the distributions of pure simulation activities (see Figure 21) throughout the semester in the replication study suggest that better performing

hours vs. 10.7 hours) and more use of validation techniques (mean simulation duration of 15.6 days vs. 11.9 days and mean simulation time of 46.8 hours vs. 31.1 hours). Further experiments and analysis of process variables would be required to examine time/score correlations at a more granular level (e.g. active modeling time vs. viewing,... time indicators per event/type/element...), however such statistical analysis is outside the scope of this work. Process statistics demonstrate how process mining techniques can be informative about learning processes by suggesting/identifying perspectives that can be of further interest to identify feedback needs and develop feedback strategies (e.g. how far have students advanced in mastering and/or applying a certain concept, i.e. if they have reached the recommended amount/time of use for mastering?, is there a need for a teacher interference or adapting a teaching process?, etc.).

Figure 22. Relative modeling process effort per event type: early vs. late sessions: Worst 5 cases (original study 2013)

Figure 23. Relative modeling process effort per event type: early vs. late sessions: Best 5 cases (original study 2013)

Figure 24. Relative modeling process effort per event type: early vs. late sessions: Worst 5 cases (replication study 2014)

Figure 25. Relative modeling process effort per event type: early vs. late sessions: Satisfactory 5 cases (replication study 2014)

Figure 26. Relative modeling process effort per event type: early vs. late sessions: Best 5 cases (replication study 2014)

Distribution of relative modeling effort spent for “create” and “edit” activities as percentage of total modeling effort (Figure 22 - Figure 26) shows that in the late stages of modeling, activities in the best performing groups show a tendency to decrease (decrease in create-activities of approximately 14% in the original and 13% in the replication study as compared with their early sessions) while the frequency of activities in worst and satisfactory performing groups show a continuously active modeling process (increase in create-activities of approximately 25% in the original and 6 % in the replication study as compared with their early sessions). This tendency of continuous modeling can be also observed for satisfactory groups (increase of 27% and 2% in the original and replication studies respectively). Prevailing number of “create” events in early sessions suggests that best performing groups first targeted at capturing relevant information from textual requirements into their model and continued to adapt the model in later sessions (proportional difference of create/edit activities). This seems to indicate that later phases in best performing groups had already reached a satisfying solution and presumably had a more verification/validation-oriented character. In contrast to the best performing groups worst performing groups remained active in later sessions, with a larger number of create events compared to their early sessions.

5.5 FINE-GRAINED ANALYSIS: DETAILING WITH ELEMENT TYPE INFORMATION

To better understand the differences between best and worst performing groups early and late sessions, we “zoomed” into the modeling processes by detailing create-events in each view with the type of element being created, i.e. object, attribute, association, event, state chart, state, transition, or constraint. An overview of the relative modeling effort spent per each element as a percentage of total effort spent on the modeling task is

depicted in Figure 27 - Figure 31. The charts of both the original and replication study suggest that best performing groups put the largest part of their effort in simultaneously modeling transitions and associations in their early (blue) sessions. While these modeling constructs represent different model views both are responsible for interplay aspects of a system. Associations define collaboration channels between objects and transitions define how objects react to interactions.

Figure 27. Modeling process effort per element: early (blue) vs. late (red) sessions: Worst 5 cases (original study 2013)

Figure 28. Modeling process effort per element: early (blue) vs. late (red) sessions: Best 5 cases (original study 2013)

Figure 29. Modeling process effort per element: early (blue) vs. late (red) sessions: Worst 5 cases (replication study 2014)

Figure 30. Modeling process effort per element: early (blue) vs. late (red) sessions: Satisfactory 5 cases (replication study 2014)

Figure 31. Modeling process effort per element: early (blue) vs. late (red) sessions: Best 5 cases (replication study 2014)

This modeling pattern is different from the pattern of effort in the worse performing groups whose effort seem to be targeted at different constructs but never concentrate simultaneously on elements representing interplay aspects in both views. It seems that in their early (blue) sessions best performing groups try to “glue” model views together and refine the “agreement” (consistency) between their model views.

While these tendencies captured by process mining techniques could be observed at a more granular level to obtain insights on the causes (e.g. how increase/decrease effects can be correlated with other process variables), statistical analysis is however not the main target of this research. The goal is rather to demonstrate how process mining techniques can be useful in suggesting points of interest for further focus and/or identifying feedback needs by making it visually easier to extract relevant information from learning process data.

6. SYNTHESIZED MODELING PROCESS PATTERNS

The results of this research suggest that the way novices model can be indicative for learning outcomes. Furthermore, the findings also suggest that it is not only the way novices model that determines the quality of modeling process outcome but that in addition the way how they validate their outcomes can in fact affect the quality of their model, and thus the learning process and outcomes. Based on the findings of the two empirical studies, we further synthesized modeling patterns indicative for worse/better learning outcomes.

Pattern 1 (Modeling approach: *Sequential vs. iterative*): Over time the modeling behavior of worst performing groups is characterized by a tendency to work in a *sequential way* (one modeling task/view at a time), while the modeling behavior of best performing groups is characterized by an *iterative approach* manifested in more frequent switches between different model views

Pattern 2 (Validation approach: *partial and/or disconnected vs. cross-validation oriented*): Self-regulative activities such as model testing and validation are found to be the central cause of the iterative working pattern. The validation approach in best performing groups is observed to have broader coverage targeting both structural and behavioral views with almost equal effort as well as being positioned between different model views. Validation

activities in worst performing groups are observed to be limited in scope (targeting either structural or behavioral aspects of a model). In contrast to worst performing groups, best performing groups are also distinguished by *earlier* engagement in validation activities in their modeling process.

Pattern 3 (Validation intention: *general testing vs. verifying recent changes*): Testing activities of best groups are observed to be positioned around the interplay effects between views. The findings suggest that this pattern corresponds to the *intention to validate recent changes*. Best performing groups put more effort in testing behavioral aspects which results in a broader test coverage for their models, while worst performing groups put more effort in testing structural aspects using minimal behavioral aspects required for that purpose (e.g. create an instance of an object, end an instance...) with no further testing of complex behaviors which could potentially uncover hidden interplay aspects. This suggests that the validation pattern of worst performing groups rather targets a *general (unstructured) test*.

Pattern 4 (Engagement styles: *Early and systematic vs. deadline-oriented engagement*): In contrast to worst performing groups best performing groups are distinguished by *earlier* and *systematic* engagement in modeling activities in their project and by also putting more modeling effort (in terms of both the number of modeling activities, modeling sessions and time spent). Both best and worst performing groups are sensitive to deadlines (submission for peer review and final submission) in terms of being more intensively engaged in modeling activities right before the deadlines. In the best performing groups however, more intensive activities are also found in between the deadlines both for modeling as well as validation activities.

Pattern 5 (Effort distribution over time: *Decreased vs. continuous or increased*): In general, while best performing groups put more effort in their modeling process, the frequency of their modeling activities is characterized by a tendency to decrease over time. In contrast, worst performing groups are characterized by putting less effort into the modeling process, however the number of their modeling activities is characterized by a tendency of continuous or increased effort presumably indicating difficulties in achieving a correct solution.

Pattern 6 (Effort distribution within modeling tasks: *Goal-oriented vs. unstructured modeling*): Session-based analysis showed that best performing groups are distinguished by attempting a broader coverage of the requirements into their models in the earlier stages of the modeling process (*prevailing number of create events in earlier sessions*) and continued to adapt the model in later sessions (*prevailing number of edit activities in later sessions*) which reveals a more goal-oriented approach. Worst performing groups in contrast seem to start with smaller models while actively expanding the model in later sessions (increasing number of create events). Moreover, in their earlier sessions best performing groups put the largest part of their effort into simultaneous modeling of constructs that are responsible for interplay aspects of a system trying to refine the “agreement” (consistency) between their model views, while worst performing group demonstrate a more disconnected approach (use of different constructs but not simultaneously making use of constructs representing interplay aspects in both model views). Additionally, the qualitative analysis revealed that worst performing groups had a more reactive approach by thinking more “what is required to adapt a discovered issue” e.g. adapting a model by adding elements rather than supplying elements in accordance with conceptual analysis by thinking about “what is required by requirement statements to achieve a correct solution” which confirms that they did not have a structured modeling approach.

To summarize, the patterns identified in this work provide insights on problem solving behavior of novices in terms of how they model and validate their solutions and how those approaches are associated with the quality of their modeling process output and learning outcomes. These insights are suggested as guidelines to be used to improve teaching practices and providing process-oriented guidance for modeling courses that adopt a multi-view modeling methodologies. It should be however noted that the patterns are limited to the scope of object-oriented domain modeling and are to be interpreted in a learning context. Identification of patterns in this work does not target at suggesting good or bad practices for object-oriented domain modeling either. For example, we do not claim that a linear way of modeling (pattern 1) would not be a good modeling approach but rather that can be indicative for poor learning approach. It is possible to achieve a good quality of a model by working sequentially without necessarily further revisiting and improving a model. Based on the findings of our empirical studies we can however state that it is less likely for a novice with a limited experience in modeling to achieve a better learning outcome when following a linear modeling approach.

With respect to previous findings in the domain of business process modeling our findings are in line with the conclusion on structured and goal-orientated approaches being associated with better model quality in general (Claes et al., 2013; Pinggera et al., 2013). Although the results of our work seem to contradict the earlier findings in terms of emphasizing “successive phases of thorough and localized modeling (i.e., within blocks) being more attractive than diverting one’s attention across different parts of a model at the same time” (Claes et al., 2012) our

findings nevertheless provide more insights on multi-view modeling approaches where consistency between views can be of importance. With respect to switching between different modeling views our findings suggest that validation activities should be considered as the link where the switches are associated with better model quality and thus learning outcomes.

7. CONCLUSION

While in the educational context mostly outcome feedback is available to novices, i.e. when a task is completed (is a solution correct? Why (not) ?), this research aimed at revealing perspectives of process-related feedback to guide a learning process (how to solve it the right way(s)?). This study was carried out in the context of one particular learning case, namely, object-oriented domain modeling. Modeling activities of novices were empirically examined using process mining techniques. The findings of two empirical studies include modeling behavior patterns of novices showing that certain behavioral patterns can indeed be associated with learning outcomes. From a theoretical perspective this work presents three major contributions: (1) the work outlines a novel approach for analyzing cognitive learning processes for complex problem solving behavior through the application of process mining techniques and process discovery maps in particular (2) the results contribute to improving our knowledge on cognitive processes in learning object-oriented domain modeling; (3) the results provide empirical support for the use and advancement of process-oriented feedback mechanisms.

While findings showed that certain behavioral patterns can indeed be associated with better/worse outcomes in terms of reaching a satisfactory model quality, further examinations are needed to evolve towards more exhaustive and generic patterns.

With respect to previous research several contributions are present: (1) we extend the scope of modeling process observations previously limited to business process modeling process observations by new observations for object-oriented domain modeling that combine processes for modeling different aspects of enterprise modeling such as the "what" (represented as objects and relationships) and the "how" (represented as individual object behavior and interactions among objects). On the methodological side (2) our research contributes to the state-of-the-art by extending previous research results that rely on statistical and data mining techniques as well as the extended version of dotted chart technique from the arsenal of process mining by emphasizing the **sequence** of activities and the application of process discovery map technique to observe sequence characteristics. Our research results also include (3) **learner behavior patterns** that are indicative for learning outcomes thus providing support for process-oriented guidance for teaching modeling.

It should however be noted that we do not suggest using patterns of more successful students as recommendations for guiding future learners. Grounding instructional advice on such a basis alone would not only be limited in scope in terms of being bounded to a specific environment and context, but would also raise ethical concerns (Bannert et al., 2014). Nonetheless the findings of this research show that process mining techniques can be highly valuable for monitoring and analyzing (cognitive) learning processes by also serving as a practical instrument for identifying and suggesting feedback needs thus providing insights for process-based teacher feedback mechanisms.

To facilitate further research and in particular for future empirical studies in this domain, certain limitations of this study should be mentioned. This study is limited to one particular modeling methodology and tool with empirical studies involving student samples from one course from one university program. For more generic patterns it would be interesting to compare the results with those obtained by using student samples from other courses and universities in the context of other modeling methods and techniques. Also, the analyzed learning behavior is limited to the modeling process of novices learning and applying modeling knowledge in solving a task throughout a semester period of time.

The research can be expanded along several dimensions in different research domains:

1. Linking modeling processes to the learning processes and phases for knowledge acquisition would provide better insights on the learning context.
2. Since simulation activities (as self-regulative approach to test and verify own solutions) were found to be significant in differentiating patterns for worse/better outcomes, experimentation with expanded observations for fine-grained analysis of testing/validation activities would allow to gain even better insights in novices' learning process. Based on the results, tool support to enable automated generation of scenarios assisting in testing a model can be further investigated to promote better validation coverage.

3. Since the empirical studies presented in this work were conducted at a group level, investigating collaboration and co-regulative patterns within groups (how students interact and/or co-regulate each other learning processes) would be an interesting perspective for further work (Arvaja, Salovaara, Häkkinen, & Järvelä, 2007; Hadwin, Järvelä, & Miller, 2011; Molenaar & Järvelä, 2014).
4. Exploring perspectives on personalization of feedback by means of studies at individual rather than group level can be another perspective using additional experimental variables. Dimension to consider are gender, previous knowledge, the level of computer self-efficacy level in terms of their ability to learn and use software (Compeau & Higgins, 1995; Davis, 1989; Davis, Bagozzi, & Warshaw, 1989; Keller, 2009; Poelmans & Wessa, 2013; Venkatesh, Morris, Davis, & Davis, 2003).
5. Investigation of advanced feedback features by means of comparisons of novices' vs. experts patterns and possibilities of learning reinforcement algorithms could be considered as yet another possible extension of this work.
6. While the process mining and analysis part of this research was mainly done manually, algorithms for extending process mining tools for automatic process pattern discovery adapted specifically for educational and cognitive learning process observations would also be an interesting expansion of this research.
7. Finally, the findings also suggest that historical learners' patterns, in combination with new learner data, can also be exploited to provide automated process-oriented guidance in a multi-user learning environment. While this type of machine feedback will not be mature to fully replace a teacher, such process-oriented feedback can significantly alleviate teacher effort leaving the instructor more time for in-depth discussions with students.

Ultimately, the outcomes of the work can be inspirational outside of the domain of object oriented domain modeling as learning event data is becoming readily available through digital learning systems and other educational information systems by also opening up new perspectives for learning analytics and artificial intelligence research in the domain of education.

8. REFERENCES

- Arvaja, M., Salovaara, H., Häkkinen, P., & Järvelä, S. (2007). Combining individual and group-level perspectives for studying collaborative knowledge construction in context. *Learning and Instruction*, 17(4), 448-459.
- Asterhan, C. S., Schwarz, B. B., & Cohen-Eliyahu, N. (2014). Outcome feedback during collaborative learning: Contingencies between feedback and dyad composition. *Learning and Instruction*, 34, 1-10.
- Bannert, M., Reimann, P., & Sonnenberg, C. (2014). Process mining techniques for analysing patterns and strategies in students' self-regulated learning. *Metacognition and learning*, 9(2), 161-185.
- Barber, L. K., Bagsby, P. G., Grawitch, M. J., & Buerck, J. P. (2011). Facilitating Self-Regulated Learning With Technology: Evidence for Student Motivation and Exam Improvement. *Teaching of Psychology*, 38(4), 303-308.
- Bavota, G., Gravino, C., Oliveto, R., De Lucia, A., Tortora, G., Genero, M., & Cruz-Lemus, J. A. (2011). Identifying the weaknesses of UML class diagrams during data model comprehension *Model Driven Engineering Languages and Systems* (pp. 168-182): Springer.
- Bernaert, M., Poels, G., Snoeck, M., & De Backer, M. (2014). Enterprise architecture for small and medium-sized enterprises: a starting point for bringing EA to SMEs, based on adoption models *Information Systems for Small and Medium-sized Enterprises* (pp. 67-96): Springer.
- Bézivin, J. (2006). Model driven engineering: An emerging technical space *Generative and transformational techniques in software engineering* (pp. 36-64): Springer.
- Bradley, J. H., Paul, R., & Seeman, E. (2006). Analyzing the structure of expert knowledge. *Information & Management*, 43(1), 77-91. doi: 10.1016/j.im.2004.11.009
- Buckl, S., Matthes, F., & Schweda, C. M. (2010). A meta-language for EA information modeling – state-of-the-art and requirements elicitation *Enterprise, Business-Process and Information Systems Modeling* (pp. 169-181): Springer.
- Butler, D. L., & Winne, P. H. (1995). Feedback and Self-Regulated Learning: A Theoretical Synthesis. *Review of Educational Research*, 65(3), 245-281. doi: 10.2307/1170684
- Carbone, M., & Santucci, G. (2002). *Fast&Serious: a UML based metric for effort estimation*. Paper presented at the Proceedings of the 6th ECOOP workshop on quantitative approaches in object-oriented software engineering (QAOOSE'02).
- Claes, J., Vanderfeesten, I., Pinggera, J., Reijers, H. A., Weber, B., & Poels, G. (2013). A visual analysis of the process of process modeling. *Information Systems and e-Business Management*, 13(1), 147-190.
- Claes, J., Vanderfeesten, I., Reijers, H. A., Pinggera, J., Weidlich, M., Zugal, S., Fahland, D., Weber, B., Mendeling, J., Poels, G. (2012). Tying process model quality to the modeling process: the impact of structuring, movement, and speed *Business Process Management* (pp. 33-48): Springer.
- Compeau, D. R., & Higgins, C. A. (1995). Computer self-efficacy: Development of a measure and initial test. *MIS Quarterly*, 19(2).
- Crapo, A. W., Waisel, L. B., Wallace, W. A., & Willemain, T. R. (2000). *Visualization and the process of modeling: a cognitive-theoretic view*. Paper presented at the Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining.
- Cruz-Lemus, J. A., Genero, M., Manso, M. E., Morasca, S., & Piatini, M. (2009). Assessing the understandability of UML statechart diagrams with composite states—A family of empirical studies. *Empirical Software Engineering*, 14(6), 685-719.
- Cruz-Lemus, J. A., Genero, M., Morasca, S., & Piatini, M. (2007). Using practitioners for assessing the understandability of UML statechart diagrams with composite states *Advances in Conceptual Modeling—Foundations and Applications* (pp. 213-222): Springer.
- Cruz-Lemus, J. A., Genero, M., & Piatini, M. (2008). Using controlled experiments for validating uml statechart diagrams measures *Software Process and Product Measurement* (pp. 129-138): Springer.
- Cruz-Lemus, J. A., Maes, A., Genero, M., Poels, G., & Piatini, M. (2010). The impact of structural complexity on the understandability of UML statechart diagrams. *Information Sciences*, 180(11), 2209-2220.

- Damassa, D. A., & Sitko, T. (2010). Simulation Technologies in Higher Education: Uses, Trends, and Implications. *EDUCAUSE Center for Analysis and Research (ECAR), Research Bulletins*.
- Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3), 319-340. doi: 10.2307/249008
- Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: a comparison of two theoretical models. *Management science*, 35(8), 982-1003.
- De Weerd, J., Schupp, A., Vanderloock, A., & Baesens, B. (2013). Process mining for the multi-faceted analysis of business processes - a case study in a financial services organization. *Computers in Industry*, 64(1), 57-67.
- Embley, D. W., & Thalheim, B. (2012). *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*: Springer.
- Erickson, J., & Siau, K. (2007). *Can UML Be Simplified? Practitioner Use of UML in Separate Domains*. Paper presented at the Proceedings of the 12th Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD'07), held in conjunction with the 19th Conference on Advanced Information Systems (CAiSE'07), Trondheim, Norway.
- Genero, M., Miranda, D., & Piattini, M. (2003). Defining metrics for UML statechart diagrams in a methodological way *Conceptual Modeling for Novel Application Domains* (pp. 118-128): Springer.
- Gustas, R. (2010). Conceptual modeling and integration of static and dynamic aspects of service architectures *Ontology, Conceptualization and Epistemology for Information Systems, Software Engineering and Service Science* (pp. 17-32): Springer.
- Hadwin, A. F., Järvelä, S., & Miller, M. (2011). Self-regulated, co-regulated, and socially shared regulation of learning. *Handbook of self-regulation of learning and performance*, 30, 65-84.
- Keller, C. (2009). User Acceptance of Virtual Learning Environments: A Case Study from Three Northern European Universities. *Communications of the Association for Information Systems*, 25.
- Lindland, O. I., Sindre, G., & Solvberg, A. (1994). Understanding quality in conceptual modeling. *Software, IEEE*, 11(2), 42-49.
- Martin, J. (1990). Information Engineering, Book II: Planning and Analysis. Eaglewood Cliffs, NJ: Prentice-Hall, Inc.
- Mason, L., Pluchino, P., & Tornatora, M. C. (2015). Eye-movement modeling of integrative reading of an illustrated text: Effects on processing and learning. *Contemporary Educational Psychology*.
- Molenaar, I., & Järvelä, S. (2014). Sequential and temporal characteristics of self and socially regulated learning. *Metacognition and learning*, 9(2), 75-85.
- Mukala, P., Buijs, J., Leemans, M., & van der Aalst, W. (2015). Learning Analytics on Coursera Event Data: A Process Mining Approach.
- Narciss, S. (2008). Feedback strategies for interactive learning tasks. *Handbook of research on educational communications and technology*, 125-144.
- Nathan, M. J., & Sawyer, R. K. (2014). *Foundations of the learning sciences*. Paper presented at the The Cambridge handbook of the learning sciences.
- Nelson, H. J., Poels, G., Genero, M., & Piattini, M. (2012). A conceptual modeling quality framework. *Software Quality Journal*, 20(1), 201-228. doi: 10.1007/s11219-011-9136-9
- Otero, M. C., & Dolado, J. J. (2004). Evaluation of the comprehension of the dynamic modeling in UML. *Information and Software Technology*, 46(1), 35-53.
- Pechenizkiy, M., Trčka, N., Vasilyeva, E., van der Aalst, W., & De Bra, P. (2009). Process Mining Online Assessment Data. *International Working Group on Educational Data Mining*.
- Pinggera, J., Soffer, P., Fahland, D., Weidlich, M., Zugal, S., Weber, B., Reijers, H. A., Mendling, J. (2013). Styles in business process modeling: an exploration and a model. *Software & Systems Modeling*, 1-26.
- Pinggera, J., Zugal, S., Furtner, M., Sachse, P., Martini, M., & Weber, B. (2014). The Modeling Mind: Behavior Patterns in Process Modeling *Enterprise, Business-Process and Information Systems Modeling* (pp. 1-16): Springer.
- Poelmans, S., & Wessa, P. (2013). A constructivist approach in a blended e-learning environment for statistics. *Interactive Learning Environments*, 1-17. doi: 10.1080/10494820.2013.766890
- Recker, J., Safrudin, N., & Rosemann, M. (2010). How novices model business processes *Business process management* (pp. 29-44): Springer.
- Reggio, G., Leotta, M., Ricca, F., & Clerissi, D. (2013). *What are the used UML diagrams? A Preliminary Survey*. Paper presented at the EESSMOD@ MoDELS.
- Schenk, K. D., Vitalari, N. P., & Davis, K. S. (1998). Differences between Novice and Expert Systems Analysts: What Do We Know and What Do We Do? *Journal of Management Information Systems*, 15(1), 9-50. doi: 10.2307/40398371
- Sedrakyan, G., & Snoeck, M. (2012). Technology-enhanced support for learning conceptual modeling *Enterprise, Business-Process and Information Systems Modeling* (pp. 435-449): Springer.
- Sedrakyan, G., & Snoeck, M. (2013a). Feedback-enabled MDA-prototyping effects on modeling knowledge *Enterprise, Business-Process and Information Systems Modeling* (pp. 411-425): Springer.
- Sedrakyan, G., & Snoeck, M. (2013c). *A PIM-to-Code requirements engineering framework*. Paper presented at the Proceedings of Modelsward 2013-1st International Conference on Model-driven Engineering and Software Development-Proceedings.
- Sedrakyan, G., & Snoeck, M. (2014a). *Do we need to teach testing skills in courses on requirements engineering and modeling?* Paper presented at the CEUR Workshop Proceedings.
- Sedrakyan, G., & Snoeck, M. (2014c). Lightweight semantic prototyper for conceptual modeling *Advances in Conceptual Modeling* (pp. 298-302): Springer.
- Sedrakyan, G., & Snoeck, M. (2015). Effects of Simulation on Novices' Understanding of the Concept of Inheritance in Conceptual Modeling *Advances in Conceptual Modeling* (pp. 327-336): Springer.
- Sedrakyan, G., Snoeck, M., & De Weerd, J. (2014). Process Mining Analysis of Conceptual Modeling Behavior of Novices - empirical study using JMermaid modeling and experimental logging environment. *Computers in Human Behavior*, 41(C), 486-503. doi: 10.1016/j.chb.2014.09.054
- Sedrakyan, G., Snoeck, M., & Poelmans, S. (2014). Assessing the effectiveness of feedback enabled simulation in teaching conceptual modeling. *Computers & Education*, 78, 367-382.
- Shanks, G., Tansley, E., & Weber, R. (2003). Using ontology to validate conceptual models. *Commun. ACM*, 46(10), 85-89.
- Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, 78(1), 153-189.
- Siau, K., & Cao, Q. (2001). Unified Modeling Language (UML)-a complexity analysis. *Journal of Database Management*, 12(1), 26.
- Siau, K., Erickson, J., & Lee, L. Y. (2005). Theoretical vs. practical complexity: The case of UML. *Journal of Database Management*, 16(3), 40.
- Sikora, E., Bastian, T., & Pohl, K. (2011). *Requirements engineering for embedded systems: An investigation of industry needs*: Springer Berlin Heidelberg.
- Snoeck, M. (2014). *Enterprise Information Systems Engineering: The MERODE Approach*: Springer.
- Snoeck, M., Michiels, C., & Dedene, G. (2003). Consistency by construction: the case of MERODE *Conceptual modeling for novel application domains* (pp. 105-117): Springer.

- Southavilay, V., Yacef, K., & Calvo, R. A. (2010). *Process Mining to Support Students' Collaborative Writing*. Paper presented at the EDM.
- TOGAF. (2015). from <http://www.opengroup.org/subjectareas/enterprise/togaf>
- Trčka, N., & Pechenizkiy, M. (2009). *From local patterns to global models: Towards domain driven educational process mining*. Paper presented at the Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on.
- Trčka, N., Pechenizkiy, M., & van der Aalst, W. (2010). *Process mining from educational data*: Chapman & Hall/CRC.
- van der Aalst, W. M., van Dongen, B. F., Günther, C. W., Rozinat, A., Verbeek, E., & Weijters, T. (2009). ProM: The Process Mining Toolkit. *BPM (Demos)*(489).
- Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3).
- Wilmont, I., Hengeveld, S., Barendsen, E., & Hoppenbrouwers, S. (2013). Cognitive Mechanisms of Conceptual Modeling. In W. Ng, V. Storey, & J. Trujillo (Eds.), *Conceptual Modeling* (Vol. 8217, pp. 74-87): Springer Berlin Heidelberg.
- Zachman, J. (1987). A framework for information systems architecture. *IBM systems journal*, 26(3), 276-292.
- Zimmerman, B. J. (1990). Self-regulated learning and academic achievement: An overview. *Educational psychologist*, 25(1), 3-17.