Assessing the effectiveness of feedback enabled simulation in teaching conceptual modeling


Abstract
It is commonly accepted that simulation contributes to a better learning quality while also promoting successful transfer of the skills to real-world environments. However, the practical use of simulation is hampered by the difficulty of interpreting simulation results. This paper demonstrates the learning benefits in conceptual modeling of business requirements when using feedback-enabled simulation. The effects of feedback-enabled simulation on learning outcomes of novice learners were observed by means of experimental empirical studies. Three studies were conducted in the context of two master-level courses from two different study programs spanning two academic years. The findings show a significant improvement in students’ conceptual model understanding and validation capabilities when using feedback-enabled simulation.

Keywords: teaching conceptual modeling, modeling knowledge, model validation / validity, simulation / prototyping, augmented feedback, requirements testing

1. Introduction
The modern world depends so much on information systems (IS) that a small failure can cause significant and sometimes disastrous consequences. In the context of ICT-driven economies the quality of the requirements engineering process has become very critical. Designing high quality information systems however is a complex and challenging task that requires rigorous analytical skills and experience to externalize business requirements into high-quality formal representations – models. With the growing importance of compliance between business strategy and ICT realizations, conceptual modeling gains more relevance. This is because first, conceptual models can significantly reduce the complexity of a problem domain by the use of highly abstract representations thus making it easier to integrate business domain and ICT expertise in the system design process, and secondly because conceptual models contain the critical information that is needed for designing and applying effective organizational strategies and a necessary foundation for constructing an organization’s information systems (Embley & Thalheim, 2012; TOGAF, 2014; Wand, Monarchi, Parsons, & Woo, 1995).

Important key factors affecting the quality of a conceptual model are knowledge of modeling concepts, of the modeling language and of the domain to be modeled (Nelson, Poels, Genero, & Piattini, 2012). Teaching such knowledge and skills to novice modelers is a challenging task considering that system analysis is by nature an inexact skill. Good modelers rely mainly on their personal experience, and the tacit knowledge (Bradley, Paul, & Seeman, 2006; Schenk, Vitalari, & Davis, 1998) they have developed over time is difficult to transfer to junior modelers. Transferring the academic knowledge and skills to real world businesses is yet another concern as the classroom and real world situations are not identical (Damassa & Sitko, 2010). As stated by (Schenk, et al., 1998), in their early careers novice modelers produce incomplete, inaccurate, ambiguous, and/or incorrect information requirements. Errors occurring early in the systems analysis process are much more expensive and time-consuming to resolve when only detected later in the engineering process than those that may occur at any other time in systems engineering (Schenk, et al., 1998).

While on the one hand conceptual modeling is crucial for the quality of information systems, on the other hand several reasons make conceptual modelling skills very difficult to teach:

- The cognitive aspects of modeling: Studies on comparing model quality checking approaches of novices and experts indicate the poorly adapted cognitive schemata of novice modelers to identify relevant triggers for verifying the quality of models (such as discovering omitted requirements) (Schenk, et al., 1998). Previous research on observing the differences in modeling process indicates the linear problem-solving pattern of novices focus-
ing on one task at a time vis-a-vis experts’ frequent switches between modeling activities and simultaneous cross-validation cycles (80% of their design activity) (Wang & Brooks, 2007).

- **The complexity of modeling tools**: The complexity of industry modeling tools introduce another challenge making them less effective in supporting a teaching process (Siau & Loo, 2006). Furthermore, (Wilmont, Hengeveld, Barendsen, & Hoppenbrouwers, 2013) point out that most of the modeling languages (including the UML) are too “noisy” with various concepts, which can result in misusing concepts and creation of unintended models (Erickson & Siau, 2007).

- **Lack of domain specific knowledge**: Absence of intensive trial and error rehearsals in the classroom is considered the major source of limitation in novices modeling experience (Schenk, et al., 1998).

- **The lack of validation procedures and tool support**: In addition, the lack of established validation procedures (Shanks, Tansley, & Weber, 2003) makes the conceptual modeling for novices very difficult to learn. Additionally, several researchers correlated novices learning achievements in modeling with the lack of technical insights considering the absence of technical components (such as computer-assisted learning) from education as a major contributing factor to the lack of preparedness of their skills (Barjis, et al., 2012).

As a result, the effectiveness of novice modelers becomes an important aspect for IS education. To address this challenge, IS education needs to introduce innovative teaching practices and integrated learning environments to facilitate the progression of novice modelers to advanced levels of expertise.

It is commonly accepted that simulation contributes to better understanding of modeling decisions offering a new standard of learning quality (Barjis, et al., 2012) allowing the learner to “learn by experiencing” (Kluge, 2007). Simulated environments are also known to promote successful transfer of the skills learned in classroom to real-world environments by allowing to simulate real-life situations where learners improve their technical and problem-solving skills (European Commission, 2013; Kluge, 2007). However, to our knowledge no research publications have been written describing courses that use conceptual model simulation in the context of requirements engineering, nor empirically proven learning benefits have been reported for a certain simulation tool. The reason is that the existing standards for simulation technologies also introduce a number of shortcomings. The major disadvantages include being too complex and time consuming to achieve by novice modelers whose technical expertise is limited. Another important disadvantage is connected with the difficulty of interpreting the simulation results.

This paper investigates the effectiveness of a feedback-enabled simulation environment which assists novice learners to validate a conceptual business model in a fast and easy way while also allowing to gain solid insights behind modeling decisions through experimenting with a concrete form of an information system. The learning effectiveness is enhanced by automated feedback incorporated in a prototype that visually links the test results to their causes in the model's design. The goal of the method is twofold: 1) providing assistance in validating semantic aspects while also gaining deeply understood knowledge from hands-on experience 2) allowing to detect design errors that result from misinterpreted use of modeling language constructs. The work proposes an extension to the techniques previously presented by (Haesen & Snoeck, 2004; Sedrakyan & Snoeck, 2012, 2013a, 2013b; Snoeck, Haesen, Buelens, De Backer, & Monsieur, 2007; Snoeck, Michiels, & Dedene, 2003). The methodology uses the UML as modelling language, but underneath it relies on the concepts of MERODE, an Enterprise Information Systems engineering methodology developed at the university of Leuven, which follows the Model-Driven Architecture and Engineering approach. The goal of this work is to observe the effects of feedback-enabled simulation cycles on understanding of UML model statements by novice modelers and identify the needs for future research. Subsequently our research question will be formulated as: “Does a feedback-enabled simulation improve modeling knowledge of a novice modeler in terms of his/her capability of assessing a model’s semantic quality”?

In order to answer the research question, we opt for an empirical study approach according to pre/post-test control group experimental design (Trochim, 2000) with subsequent replications. The findings show a significant improvement in students’ conceptual model understanding and validation capabilities when using feedback-enabled simulation.

---

The remainder of the paper is structured as follows. The second section describes the educational context and assumptions used within this paper. Section 3 gives an overview of related work and the research contribution. Section 4 gives a brief overview of the simulation environment subsequently highlighting the learning benefits of the proposed validation cycle. Section 5 describes the research method for measuring the effects of the proposed simulation technique on the learning outcomes of novice modelers. Section 6 describes the data analysis and subsequently reports on the results. Section 7 discusses the contributions and limitations of the work. Finally, section 8 concludes the work proposing some future research directions.

2. Educational context and assumptions

The proposed simulation method has been developed by the Management Informatics research group at the faculty of Business and Economics, University of Leuven. The tool has been subsequently tested and validated within the course “Architecture and Modeling of Management Information Systems” over a 5-years period of teaching, with participation and constant feedback from 500 students overall. The course targets at master level students with heterogeneous backgrounds from the Management Information Systems program. 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, to let them understand the relation between an information system and the organizational aspects of an enterprise, and to let them acquire sufficient skills of developing an enterprise model as basis of an enterprise information system. Focus of this work is the observation of simulation cycle effects on learning outcomes of novice modelers, more specifically with respect to model understanding.

For further referencing some basic concepts used throughout this paper will be briefly discussed.

A model is a simplified representation (abstraction) of a system from a particular perspective (Seila, 1995). A conceptual model (also known as domain model) is a complete and holistic view of a system based on conceptual but precise qualitative assumptions about its concepts “entities” and their interrelationships (Embley & Thalheim, 2012). Conceptual model of an information system is defined as an “abstract model” of an enterprise and conceptual modeling in information systems development as the creation of an enterprise model for the purpose of designing the information system (Wand, et al., 1995). Analysis transforms a perceived real-world system into a conceptual model of that system and design transforms the conceptual model of the subject world, into a model of the information system (Wand, et al., 1995). A model is often represented visually as a diagram, by the use of a modeling language. In this paper the modeling language used is UML (Unified Modeling Language) motivated by the fact that UML is the widely accepted standard used for modeling systems throughout software engineering processes. UML class diagram is the main structural diagramming approach widely used to visually represent an information system’s components and relationships (Szlken, 2006) that are used both in high level conceptual modeling as well as in more detailed for a lower level programming code (Berardi, Calvanese, & De Giacomo, 2005; Marshall, 2000; Szlenk, 2006).

Model validity is an important aspect of model quality. To assess the effects of simulation on model quality a definition of quality parameters is also required. We will refer to the quality dimensions of the CMQF (Conceptual Modeling Quality Framework) (Nelson, et al., 2012) which presents a unified view of conceptual model quality. Within this framework, teaching conceptual modeling involves different types of modeling quality. The final objective is to achieve the capability of producing physical models with high external quality. External model validity - also called 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). In order to check a model for validity, a person needs to read and understand the model (which entails pragmatic quality aspects), and compare his/her understanding of the model with his/her understanding of the real world. On the knowledge side, this requires an appropriate level of model knowledge, language knowledge and representation knowledge, hence requiring pedagogical quality (understanding the modeling concepts), linguistic quality (understanding the graphical notation) and pragmatic quality (understanding a model) (Nelson, et al., 2012). In particular, pragmatic quality captures the extent to which the stakeholder completely and accurately understands the statements in the representation that are relevant to them.

There is still not a universal definition of simulation. Nevertheless, most of the existing definitions follow the same general concept that simulation is an imitation of a system (Banks, 1999; Ingalls, 2008) with a purpose to support modeling

---

2 The course’s page can be found on http://onderwijsaanbod.kuleuven.be/syllabi/e/D0I71AE.htm
decisions based on the dynamic representation of a system (how the prospective system would behave in a real world). In this paper we will refer to simulation of a conceptual model as a process of generating prototype applications using a conceptual model as an input. Among different types of simulation (symbolic or graphical animation, execution, prototyping), the method of prototyping is capable of achieving the most concrete form of a prospective system. In our works we refer to simulation of a conceptual model as a process of generating prototype applications using a conceptual model as an input. We will therefore use the terms “simulated model” and “prototype” interchangeably. Among the variety of forms of prototypes within this study we refer to the definition of a prototype as “fully functional to prove a concept” (Hess, 2012). Prototyping is also thought of as a type of design language (Yang, 2005), the learning context of which includes testing of a function of a prototype with a purpose to identify potential issues concerned with problem understanding with respect to its design (Hess, 2012).

3. Related work

Simulation has been used in a variety of contexts. In the context of information systems previous researches focused specifically on information workflow based on techniques such as symbolic animation or execution of models (Lalioti & Loucopoulos, 1993; Olivé & Sancho, 1996) as an instrument to assist analysts in validating large and complex models (Silva, et al., 2010). Prior studies on model understanding on conceptual level focused mainly on process models (control logic flow, deadlocks, etc.) (Recker, Reijers, & van de Wouw, 2014) and perceptual properties of graphical notation by novices (Mendling, Reijers, & Cardoso, 2007; Moody, 2009; Petre, 1995; Recker, Safrudin, & Rosemann, 2010). Semantic prototyping was introduced by (Lindland & Krogstie, 1993) with a goal to improve conceptual model comprehensibility aiming at communication with stakeholders. Simulation tools have been developed and tested in various educational areas such as science education (Rutten, van Joostingen, & van der Veen, 2012), mining engineering (Akkoyn, 2014), aerospace engineering (Okatsu, 2013), biological engineering (Datta, 2013), etc.; subsequently leading to improved learning outcomes. However, as we already stated, to our knowledge no research publications have been written describing courses that use conceptual model simulation in the context of software requirements engineering, nor empirically proven learning benefits have been reported for a certain simulation tool.

Studies on learning quality improvements indicate a self-regulative approach as major source of impact on learning outcomes (self-observation of one’s performance with the goal to reflect on own progress by appropriately adjusting actions to maximize performance) (Nicol & Macfarlane-Dick, 2006; Zimmerman, 2008). Studies on self-regulative learning approaches in turn are closely intertwined with feedback research, i.e. for all self-regulative activities, external feedback is considered as an inherent catalyst (Barber, et al., 2011; Butler & Winne, 1995). As proposed by the constructivist approach (Hadjerrouit, 2005) the method of dialogue is the most optimal way to address learning difficulties by delivering personalized feedback. Usually feedback is not available during modeling activities but is given after a task has been completed. In the feedback literature this is referred to as outcome feedback, the simplest form of feedback, indicating whether or not results are correct, thus providing minimal external guidance (Butler & Winne, 1995). (Butler & Winne, 1995; Narciss, 2008; Shute, 2008) highlight the effectiveness of more informative types of feedback paired with content-related information that guide the process of cognitive activities. This suggests that combining simulation cycles with cognitive feedback can result in improved learning outcomes. Our approach thus differs by introducing a cognitive feedback incorporated in a simulation cycle as a response to a failure of a triggered event to facilitate the analysis of simulation results targeting at validity of structural aspects (such as associations, cardinalities, chains of associations). In addition this approach benefits from an environment maximally adapted to a novice modeler whose technical expertise is limited (Sedrakyan & Snoeck, 2013b).

Despite a tool’s benefits, user acceptance however can be another important factor affecting its success. Prior studies on simulation are based on implications of intended utility (McHaney & Cronan, 2000), which includes semantic compliance (is it the right simulation model?) to support its intended use acceptably, as well as relevant complete functionality to support a task completion. Previous studies have identified important variables dealing with user acceptance for computer-assisted learning environments such as the Technology Acceptance Model (Fred D. Davis, 1989; Fred D. Davis, Bagozzi, & Warshaw, 1989; Poelmans & Wessa, 2013), which suggests (amongst others) perceived usefulness as an important factor contributing to user acceptance. In this study we aim to assess the effects of feedback-enabled simulation on the model understanding capabilities of novices in correlation with user acceptance as well as personal characteristics of users to ensure the reliability and validity of the results.
4. Description of the simulation environment

The simulation feature used in this study is the improved version of the tool used in the course of several years of teaching. In our previous papers we already discussed a set of shortcomings and subsequent improvements (Sedrakyan & Snoeck, 2012, 2013a) for conceptual model simulation. We will therefore dismiss the details related to the simulation environment as well as its implementation3. To facilitate further reading of this paper we will however present a brief overview of the simulation benefits offered by the simulation tool used in this paper in the context of model validation followed by an example scenario.

It was observed that both UML and MDA require a set of simplifications to be useful in educational context (Sedrakyan & Snoeck, 2012, 2013a, 2013b). We therefore use MERODE for the following advantages:

- using a limited subset of UML relevant for conceptual modeling that allows removing or hiding details irrelevant for a conceptual modeling view
- providing integrated environment for modeling and simulation
- models designed are readily transformable for execution
- transformation to final prototype is easy and fast using one-click approach
- simulated models (prototypes) are augmented with cognitive feedback linking to a designed model thus making it easy to analyze simulation results

The proprietary modeling tool4 of MERODE is well adapted for educational purposes which uses a limited subset of UML diagrams to design a system: the constructs irrelevant for conceptual modeling are filtered away. To minimize required input skills the tool allows managing consistency between the views in an automated way (Snoeck, et al., 2007; Snoeck, et al., 2003) meaning that each time when entering specifications in one view, specifications that can be derived for other views are automatically generated by the tool. For example, when an object is created the required creation and ending events are automatically generated. A model can be simulated at any step of the modeling process. The minimal required input is at least one business object along with creating and ending events which are automatically prompted for. A minimal attribute set is provided by the simulation tool by default. This approach ensures maximal assistance for self-regulative learning: it allows starting to use simulation having a minimal level of knowledge and progressing through hands-on experience.

4.1. Simulation cycle

The tool uses a one-click approach to transform a conceptual model (an output of the modeling tool) into a fully functional prototype application: the modeler receives a compiled Java application with a built-in persistent data store and an easy-to-use interface allowing to create, modify and end object instances. Objects can be associated with one another according to the relationship rules specified in a model (association cardinalities). Understanding the interface (input and output models) of the generated prototype is quite intuitive. The graphical interface of a prototype application includes a main window and a set of input/output popup windows. Business entities are presented across tabbed views in the main window each containing corresponding properties of object instances (such as attributes and associated objects) presented in a tabular format. Each view of a tab panel also contains buttons corresponding to the business events. MERODE prototypes offer basic functionality like triggering the creating and ending of objects, and triggering other business events. Object properties such as attributes are collected by means of an input window. This same window also requires specifying associated objects by proposing a list of available object instances to choose from. The entire interaction process is guided by user-friendly messages in case of an invalid input or a failure of an event execution (e.g. creating, ending or modifying object instances) thus ensuring maximum transparency between a prototype and its design model. A sample erroneous model and validation scenario is described below. An example of a modeling task would be to validate a given model solution for a requirement “To buy wholesale products customers need to place an order. However ordering is not required for buying a retail product.” While with a static model the novice’s validation cycle would be limited to assumptions within the scope of his/her knowledge of modeling and modeling language, the generated prototype application would suggest an easier and faster way of validation by enforcing model constraints in a “live” scenario. When testing a prototype of a model a student will be confronted with the following scenario: trying to register a sale for a retail

3 A short demo of simulation using a simple model can be viewed at http://merode.econ.kuleuven.ac.be/demo.aspx
4 http://merode.econ.kuleuven.ac.be/mermaid.aspx
product (according to a designed model this will be by means of triggering a creating event of an invoice) a popup window will request an input for a specified in a model attributes as well as a choice for instances of the associated mandatory objects “customer”, “retailProduct” and “order”.

![Diagram](image.png)

Figure 1: a sample erroneous model for validation

While discovering the availability of the order object among the requested associations, a students’ analytical problem-solving ability is stimulated to pursue a correct design solution in order to fix the error (s)he detected in the model. In the example above, a student intended to make the object “invoice” optional by assigning a cardinality of 0..* (see Figure 1) to enable the flexibility of either participating or not in either retail or wholesale.

![Image](image2.png)

Figure 2: validation through a simulated model with a sample feedback for mandatory 1 constraint violation

After detecting the wrong use of this construct a student would try to remove the invoice’s dependency on the order, for example by switching the cardinalities of association between “invoice” and “order” objects. In the next simulation cycle however s/he would detect that while the model works fine for selling a retail product, the selling of a wholesale product still involves an instance of a retail product since the invoice object has a mandatory relationship with the object “retailProduct”. As a next possible solution a student might consider supplying an optional intermediate object “retailSale” in
such a way that registering “retailSale” would require instances of both “retailProduct” and “invoice” but not vice versa (see Figure 3) thus ensuring the flexibility of a model solution to allow invoicing for retail and wholesale products and ordering for only wholesale products as required by the business requirement. By doing so a student is involved in a self-regulative learning process through trial and error rehearsals while testing what-if scenarios. The generated feedback facilitates the process of achieving the intended behavior by explaining the reasons each time the execution of intended behavior is refused. As a result a student builds a deeply understood knowledge that is developed from own practice (modeling knowledge and knowledge of domain to be modeled as well as knowledge on modeling language).

![Figure 3: possible correct solution](image)

### 4.2. Augmented feedback

Another aid to facilitate the model validation is the use of feedbacks in a simulation cycle. A simulated model described in this paper is augmented by cognitive feedbacks that explain the reasons of execution failures followed by graphical visualizations that link to the model used for simulation, e.g. when an event is refused (because of failed precondition checks) the user is informed of the refusal with a message that explains the reason of rejection by indicating what constraint of a model is violated (Sedrakyan & Snoeck, 2012, 2013a). In the example above a student’s attempt to omit a required input for order instance will result in a failure message followed by a visual explanation (see Figure 2) thus revealing a misinterpreted use of an association and/or its cardinalities that resulted in a mandatory constraint. The set of implemented visualizations for structural aspects include feedbacks on failures that can result from: (1) Mandatory one cardinality violation: an object is attempted to be created before the object it needs to refer to is created or chosen to be associated with the newly created object; (2) Cardinality violation: a create-event execution fails due to a cardinality constraint of maximum 1; (3) Referential Integrity: ending-event execution fails due to existing referring objects. An example of automated feedback report for mandatory one cardinality violation is shown in Figure 2.

### 5. Methods

In this work we follow the principles of Design Science in Information Systems research which targets at building and evaluating innovative artifacts to help understanding and solving knowledge problems (Hevner, March, Park, & Ram, 2004). An empirical experimental study approach was used to assess the effectiveness of feedback-enabled simulation (the design artifact) using a randomized pre/post-test control group experimental design. Three experiments have been conducted: the original in academic year 2012-2013 (Sedrakyan & Snoeck, 2013a) and its replications in 2013-2014. During the experiments students had to validate a proposed erroneous model solution against given requirements by answering a set of questions (requirements reformulated as questions). The effectiveness of the proposed simulation method was measured by means of comparison of the test results of students between experimental cycles (without and with a use of a simulation). Context information about the students was collected: students had to complete a questionnaire about their personal characteristics such as gender, previous knowledge of data modeling, the level of computer self-efficacy level in terms of their ability to learn and use a computer software and general ICT experience in terms of previous programming experience (Compeau & Higgins, 1995; Fred D. Davis, 1989; Fred D. Davis, et al., 1989; Keller, 2009; Poelmans & Wessa, 2013; Venkatesh, Morris, Davis, & Davis, 2003). A six-position Likert-type scale was used to score subjective judgment and satisfaction of students about the method (such as easiness of use, perceived utility, satisfaction) (Fred D. Davis, 1989; Fred D. Davis, et al., 1989; Poelmans & Wessa, 2013). A classical pre/post-test control group experimental design was used. A combination of two-group and factorial designs (Trochim, 2000) was used for the replication experiments. Single-factor confirmatory analysis has been conducted to assess the validity of hypothesized effects. Exploratory correlation analysis have been performed to study the correlation of the results with user acceptance and personal characteristics. To isolate the impact of pro-social behavior (Mitchell & Jolley, 2012) the anonymity of participants was ensured by not disclosing any identifiable information both in the tests and evaluation questionnaires in the
initial experiment. Following the advice of (Lelkes, Krosnick, Marx, Judd, & Park, 2012) though, students were asked to identify themselves in the replicated experiment.

5.1. Testable dimensions
Assessing the semantic quality of a model requires a combination of model understanding and a comparing model statements with requirements. Model-reading knowledge can be assessed at different levels of understanding:

Level 1: reading a single association: Understanding the lower and upper bounds of a multiplicity, for example, understanding an implication of optionality or minimum/maximum cardinality of an association.

Level 2: reading a chain of associations: Understanding combinations of lower bounds and combinations of upper bounds, for example, understanding that chaining mandatory associations results in a mandatory (chained) association.

Level 3: combining multiple single associations in parallel: for example, understanding that when from a given class A two other classes B and C can be reached by means of mandatory associations, each object of class A will be associated with an object of class B and one of class C.

Level 4: combining multiple chains of associations in parallel, for example, understanding that when starting from a class A a class B can be reached by two different chains of associations, two different objects of class B could be associated with each object A.

In this study we target at binary associations, their cardinalities and understanding of their semantics. To ensure that participants have at least basic model reading capacities, a pre-experimental test was performed. The pre-experimental test targeted at levels 1 and 2 testing the understanding of basic concepts. Thus the experiment cases targeted at model understanding at levels 3 and 4.

5.2. Experimental design and variables
Based on the goals an experimental model was designed to guide the study. The internal validity was ensured using a pre/post-test control group experimental design. The effects of unknown factors that could potentially affect the posttest results were minimized through randomized assignment. To establish baseline measurement for unknown variables control groups were used. Distribution normality tests within the entire population as well as each usable sample were performed to ensure the statistical equivalence of groups. A replication method was used to ensure the validity and reliability of the results. Experimental variables and hypothesis used in this study are presented below.

5.2.1. Dependent variables
In this experiment the dependent variable is model validation capability, measured through the score on a set of true/false questions. To test the model understanding and model validation capabilities, students were given a short requirements statement and an erroneous model solution. They were asked to assess whether or not the model reflected a particular requirement statement correctly by responding to a set of true/false questions (requirements rephrased into test questions), e.g. “in this model solution invoicing is required to buy a retail product ( TRUE/FALSE? )”. They were also asked to motivate their answers. For each correct answer 1 point was attributed, and 0 for each wrong answer. In total 8 questions had to be answered (min. score = 0; max. score = 8) including 4 questions per testable level. A single case was used for both pre/post-test cycles in the initial experiment which we will further refer to as “Warehouse case”. A slightly altered equivalent case was used for the replication experiments to minimize maturation effects between experimental cycles. E.g. the context of “selling goods” in the first case was altered into “selling flowers” in the second case, the word “warehouse” was paraphrased into “greenhouse”, “small quantity product” into “retail product”, “large quantity product” into “wholesale product”, etc. For the same reason we reversed some of the questions to vary between the cases (questions formulated to accept positive answers were rephrased to accept negative answers and vice versa, e. g. “ordering is required to buy a small quantity product” rephrased into “ordering is not necessary to buy a retail product”). The sets of questions were also rearranged in each case. We will further refer to the second case as “Greenhouse case” (see Appendix A: Experiment cases). Students had to complete the task within a one hour timeframe.
5.2.2. Independent variable and treatment

The independent variable used in this study is a feedback-enabled simulation. The goal of the experiment was to manipulate the independent variable (treatment) to generate a usable data set in order to measure the dependent variable. Notice that within this study we did not target the assessment of feedback separately from the simulation tool as the goal of the incorporated feedback was to enhance the applicability and thus perceived utility of the simulation method by facilitating the process of interpretation of simulation results. This was achieved by means of cognitive textual and graphical feedback when and why the execution of a triggered business event is refused (Sedrakyan & Snoeck, 2012), thus making the links between a prototype and its design explicit. The results of our previous assessments of simulation without feedback showed that the majority of students were reluctant in using the tool (an average of 3.46 of perceived usefulness on a five-point Likert scale). The improved tool extended with feedback however resulted in better perceptions (4.58 on 5) (Sedrakyan & Snoeck, 2013a).

5.3. Initial experiment

An initial experiment was conducted in 2012 with a participation of 49 students (Sedrakyan & Snoeck, 2013a). The experiment was conducted in two parts with a classical pre/post-test experimental setup. In the first part students had to answer the set of questions with “Warehouse case” without the use of the model’s prototype, by means of manual inspection of a given model solution. The goal of this part was to establish a baseline model validation capability level to measure the simulation effects in the second cycle. Then, in the second part of the experiment, the same questions had to be answered again with the use of the generated prototype. The answers had to be recorded on an answer sheet. We will further refer to cycles without simulation (a paper exercise by means of manual inspection of a given model to answer the test’s questions, see Appendix A: Experiment cases) as “withoutPT”, and cycles with the use of simulation (students were required to use laptops to run a simulated model of a given model solution to answer the test’s questions) as “withPT”.

5.4. Replicated experiments

To ensure the statistical reliability of our initial experiment we conducted a factorial two-group experiment with two cycles with participation of 66 students in the 2013-2014 academic year (44 in Leuven campus and 22 in Brussels campus). In this factorial design, the inter circle learning effect and group composition influence are neutralized by the cycle rotation in two groups (Grubišić, Stankov, Rosić, & Žitko, 2009; Trochim, 2000). Between the cycles within each group the cases were altered. Students were randomly assigned to two sub-groups which we will further refer to as group A and group B.

Group A

- Warehouse without PT → H1 → Warehouse with PT
- Greenhouse without PT

Group B

- Warehouse without PT → H1 → Warehouse with PT
- Greenhouse with PT

Figure 4: The experimental setup for initial experiment

Figure 5: The experimental design for replications

Group A started the experiment with “warehouse case” without the use of a prototype followed by the “greenhouse case” with the use of a prototype. Group B started with the “warehouse” case without the use of a prototype followed by the “greenhouse” case with the use of a prototype. The experimental design is presented in Figure 5. By reversing the cycles
in two groups, a potential case maturation effect could be eliminated. The green arrows represent the hypotheses about the effect of feedback-enabled simulation on model validation capabilities.

5.5. Extraneous variables (confounding variables)

Most relevant extraneous factors based on previous researches have been chosen to be controlled in this study. Among them personal characteristics such as gender, previous knowledge and skills, computer self-efficacy in terms of general ICT experience and average usage of computer per day/week. Insufficient user acceptance can be another factor affecting learner performance (Venkatesh, et al., 2003). To measure the user acceptance we considered dominant models and constructs from the technology acceptance literature (Fred D. Davis, 1989; Fred D. Davis, et al., 1989; Poelmans & Wessa, 2013; Wixom & Todd, 2005) and adapted them to fit our particular context. Following dimensions within this variable have been considered: ease of use and perceived usefulness based on (Fred D. Davis, 1989), preference and satisfaction (Bourgonjon, Valcke, Soetaert, & Schellens, 2010; Wixom & Todd, 2005). Ease of use and usefulness are widespread and validated acceptance beliefs from the TAM, referring to the required effort to interact with an IS and its efficiency and effectiveness respectively. We used the concept of preference as another success dimension, as proposed by (Hsu & Lu, 2007) and (Bourgonjon, et al., 2010). Preference is defined as “the positive and preferred choice for the continued use of simulation tool in the classroom”. User satisfaction is another key success measure that has been defined as the feelings and attitudes that stem from aggregating all the efforts and benefits that an end user receives from using an information system (Ives, Olson, & Baroudi, 1983; Wixom & Todd, 2005). A questionnaire was used to collect data on this (see Appendix B: Evaluation). Per each dimension 3 questions were asked. Reliability and validity of the acceptance measures were assessed by factor analysis using SPSS.

5.6. Hypotheses

The following observations with corresponding hypotheses were targeted:

- If the treatment positively influenced the dependent variable? (green arrows on Figure 5)
- If a maturation effect from the use of a prototype in the first experimental cycle could be observed (purple arrow on Figure 5)
- If any external factors such as particular personal characteristics caused temporary distortion? (regression and Anova analysis with demographical data collected by post-study questionnaire)
- If the perceptions of the tool influence the positive effects on the posttest scores (variables such as perceived utility, perceived ease of use, satisfaction and preference)

Subsequently we hypothesized these questions into the following testable assertions:

- **Hypothesis 1 (H1):** Feedback-enabled simulation significantly improves model validation capabilities of a novice business analyst.
  - We also target at testing potentially distorting effects such as maturation from the case, group and cohort effects.
- **Hypothesis 2 (H2):** The use of the prototype has a persisting learning effect on student’s test scores when is no longer used.
- **Hypothesis 3 (H3):** The test scores are not influenced by any particular personal characteristics of users (previous knowledge and gender).
- **Hypothesis 4 (H4):** The proposed simulation method is suitable for novice business analysts (user acceptance).

6. Data analysis

6.1. Subjects and sample representativeness

Students who participated in the experiments were final year master students from two Management Information Systems programs at KU Leuven from two different courses taught by two different instructors: “Architecture and Modeling of Management Information Systems” taught at Leuven campus (initial experiment and replication) and “Design of a Business Information System” taught at Brussels campus (second replication).
Overall 104 students fully participated in the initial and replicated experiments. The first cohort consists of 39 students of the academic year 2012-2013; the second cohort entails 65 students of the academic year 2013-2014 (44 from Leuven campus and 21 from Brussels campus). Analysis of the pre-experimental test for basic modeling skills and the context information from the post-study questionnaire resulted in the demographics presented in Table 1.

<table>
<thead>
<tr>
<th>Gender</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>72%</td>
</tr>
<tr>
<td>Female</td>
<td>29%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Age distributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min age</td>
</tr>
<tr>
<td>Max age</td>
</tr>
<tr>
<td>&lt;= 25</td>
</tr>
<tr>
<td>&gt;25 &lt;= 35</td>
</tr>
<tr>
<td>&gt; 35</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Previous knowledge of data modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>No knowledge</td>
</tr>
<tr>
<td>Little knowledge</td>
</tr>
<tr>
<td>Moderate knowledge</td>
</tr>
<tr>
<td>Extensive knowledge</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pre-experimental testing for basic knowledge: min score = 0, max score = 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average (&gt; = 5)</td>
</tr>
<tr>
<td>Failure (&lt;5)</td>
</tr>
</tbody>
</table>

Table 1: Summary of demographics

The normality of distributions of pre-experimental test scores as well as the experimental test scores were confirmed both for the testable samples (both with and without the use of prototype) as well as the entire population (2012-2013 and 2013-2014). Cross validation with the entire population confirmed the statistical equivalence for each usable subsample (see further analysis).

6.2. Test Scores

Data has been analyzed by means of statistical comparison of mean scores among the experimental cycles (mean scores withoutPT denoted by $\bar{X}_{\text{without}}$, mean scores withPT denoted by $\bar{X}_{\text{with}}$). The effectiveness was assessed based on the relative advantage (positive correction), denoted by $\bar{X}_{\text{difference}}$. Blank answers as well as those without motivation have been truncated to eliminate false negative and false positive corrections. Paired t-tests comparing the mean scores of experimental cycles within each sample as well as the entire population were performed. The results are shown in Table 2 and provide evidence that the prototyping method was effective in producing positive correction, thus supporting H1. Furthermore, the replication results confirm the reliability of the conclusion.

<table>
<thead>
<tr>
<th>Experimental sample</th>
<th>$\bar{X}_{\text{without}}$</th>
<th>$\bar{X}_{\text{with}}$</th>
<th>$\bar{X}_{\text{difference}}$</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial experiment (2012)</td>
<td>3.1</td>
<td>5.43</td>
<td>2.33</td>
<td>0.000</td>
</tr>
<tr>
<td>Replicated experiments (2013)</td>
<td>3.63</td>
<td>5.23</td>
<td>1.6</td>
<td>0.000</td>
</tr>
<tr>
<td>Entire population (2012-2013)</td>
<td>3.44</td>
<td>5.31</td>
<td>1.87</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 2: Paired t-tests comparing mean scores (withoutPT - withPT)

6.2.1. Between and within group analysis of replication experiment

To examine case maturation effects between two cycles, paired t-tests within groups in the replicated experiment comparing the test scores withPT and withoutPT resulted in the statistics presented in Table 3. In both groups a positive and significant effect is observed by comparing the test scores (with and without the use of simulation) thus providing support for H1. Even though the treatment was reversed in group B (use of the simulation in the first cycle of the experiment) a positive effect is still observable within group B ($\bar{X}_{\text{difference}} = 0.93$). The mean score in group B withoutPT ($\bar{X}_{\text{without}} = 4.4$) was significantly higher than in group A ($\bar{X}_{\text{without}} = 3$) which provides evidence for a potential maturation effect from the use of the simulation in the first cycle of the experiment thus supporting H2. When analyzing the motivations of the answers to the test questions provided by group B students for the withoutPT cycle, we observed certain patterns that closely mimicked those of execution scenarios and feedback they would typically achieve with the help of a simulated
model. For instance, in group A and in the original experiment we observed that with a manual inspection of a model a student would refer to a modeling construct that is obvious in his/her opinion, or try to execute testing scenarios using his/her own words to justify his/her answer (e.g. the relationship between “invoice” and “order” is optional). In group B, the execution scenarios in the withoutPT cycle were different because of given motivations that correspond to the typical feedback message they would receive when using a prototype (e.g. “invoice” is existence dependent of “order”, to be able to create an “invoice” one needs an instance of “order” object). This indicates that experimenting with a feedback-enabled simulation in the first cycle of the experiment enabled the transfer of validation skills into a mental habit.

<table>
<thead>
<tr>
<th>Experimental sample</th>
<th>( \bar{X} ) without</th>
<th>( \bar{X} ) with</th>
<th>( \bar{X} ) difference</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group A</td>
<td>3</td>
<td>5.15</td>
<td>2.15</td>
<td>0.000</td>
</tr>
<tr>
<td>Group B</td>
<td>4.4</td>
<td>5.33</td>
<td>0.93</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Table 3: Within group analysis

Notice that a group effect is detected in relative advantage: in group A the relative advantage (\( \bar{X} \) difference = 2.15) is significantly higher than in group B (\( \bar{X} \) difference = 0.93). However, in both groups, H1 is still confirmed at confidence level \( p < 0.05 \).

To test for differences between group A and group B in the replicated experiment, F-tests for variances and corresponding independent t-tests have been performed. We first compared the test scores of group A (with PT) with the scores of group B (with PT). The results did not disclose any evidence for statistically significant difference in relative advantage (\( \bar{X} \) difference = 0.17, \( p > 0.05 \)).

<table>
<thead>
<tr>
<th></th>
<th>Group A</th>
<th>Group B</th>
<th>( \bar{X} ) difference</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \bar{X} ) with</td>
<td>5.15</td>
<td>5.32</td>
<td>0.17</td>
<td>0.7</td>
</tr>
<tr>
<td>( \bar{X} ) without</td>
<td>3</td>
<td>4.32</td>
<td>1.32</td>
<td>0.01</td>
</tr>
<tr>
<td>( \bar{X} ) with-without</td>
<td>2.15</td>
<td>1</td>
<td>1.15</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Table 4: independent t-tests for unequal variance between rotated cycles of groups

The t-test comparing the scores of group B (withoutPT) with the scores of group A (withoutPT) resulted in a significant difference: \( \bar{X} \) difference = 1.32. We attribute this difference to a maturation effect within group B from the use of the prototype in the first cycle. This seems to indicate that the prototype effect persists when the prototype is no longer used, thus providing support for H2.

<table>
<thead>
<tr>
<th></th>
<th>Initial experiment (2012)</th>
<th>Replicated experiments (2013)</th>
<th>( \bar{X} ) difference</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \bar{X} ) with</td>
<td>5.23</td>
<td>5.43</td>
<td>0.20</td>
<td>0.2</td>
</tr>
<tr>
<td>( \bar{X} ) without</td>
<td>3.12</td>
<td>3.63</td>
<td>0.51</td>
<td>0.54</td>
</tr>
<tr>
<td>( \bar{X} ) with-without</td>
<td>2.31</td>
<td>1.6</td>
<td>0.71</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Table 5: independent t-tests between cohorts

The occurrence of cohort effects (2 years) has also been tested by means of independent t-tests (see Table 5). No significant differences in the mean scores with and without the use of the prototype were disclosed between the two cohorts (2012 and 2013). There was a difference though in the differential results for relative advantage \( \bar{X} \) difference = 0.71, with \( p = 0.04 \). The relative advantage in the initial experiment are higher than in the latest replications, which can be explained by the strict design used in the replicated experiment and as a consequence a maturation effect from the use of simulation in group B. For both cohorts however H1 is still confirmed.

Thus we can state that the presence of limited group and cohort effects does not undermine the positive effect of the use of the feedback-enabled simulation.

6.2.2. User Acceptance

In Table 6 we present the statistical descriptives of the acceptance measures and computer self-efficacy. Each scale item was modeled as a reflective indicator of its theorized latent construct. We assessed the validity and reliability of our multi-item empirical indicators via confirmatory factor analysis and Cronbach Alpha, as suggested by (Segars & Grover,
The measures are reliable with both Cronbach Alpha’s and factor loadings of at least 0.73. With measures on 6-point Likert scales, average scores above 3.5 can be considered positive beliefs. In general, the tool was accepted by students with mean scores above 3.6. In particular, students found it useful and prefer its use (mean scores above 4). The lowest score, attributed to ease of use, still reflects a positive perception.

<table>
<thead>
<tr>
<th>Computer Self-efficacy</th>
<th>N</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Min</th>
<th>Max</th>
<th>Cronbach Alpha</th>
<th>Factor Loadings /item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satisfaction</td>
<td>57</td>
<td>3.88</td>
<td>1.15</td>
<td>1.00</td>
<td>6.00</td>
<td>0.90</td>
<td>0.86; 0.86; 0.86</td>
</tr>
<tr>
<td>Preference</td>
<td>57</td>
<td>4.08</td>
<td>1.16</td>
<td>1.00</td>
<td>6.00</td>
<td>0.89</td>
<td>0.89; 0.96; 0.86</td>
</tr>
<tr>
<td>Usefulness</td>
<td>57</td>
<td>4.03</td>
<td>1.23</td>
<td>1.33</td>
<td>6.00</td>
<td>0.92</td>
<td>0.93; 0.95; 0.90</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>58</td>
<td>3.63</td>
<td>1.24</td>
<td>1.00</td>
<td>6.00</td>
<td>0.9</td>
<td>0.90; 0.95; 0.87</td>
</tr>
</tbody>
</table>

Table 6: Statistical descriptives of the acceptance measures and computer self-efficacy

Using regression analyses, no impact of acceptance variables on the relative advantage (with PT – without PT) was identified. The results thus confirm the validity of H4.

6.3. Personal Characteristics: Previous Knowledge and Gender

As experts or more knowledgeable students might be able to interpret the presented models well without assistance from the prototype, thus rendering the simulation less useful, we measured previous knowledge and performed a pre-experimental test score on modeling. Previous knowledge was measured as a latent variable using 3 items and with a dummy variable, distinguishing knowledgeable students from students with no or less experience: 0 = low (first 50%, score < 2 on the scale above); 1 = less low (second 50%, > 2); using the 50% percentile as a cut-off point. The pre-experimental test score consisted of 9 items, with scores between 0 and 9. The descriptive statistics of these variables is given in Table 7. In general, the students are not experienced in modeling (with a previous knowledge of 2.03 on average). Due to the fact that students had received classes in modeling in their current program, the pre-experimental test score of 5.23 (on 9), is in line with expectations.

<table>
<thead>
<tr>
<th>Previous Knowledge (continuous)</th>
<th>N</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Mean</th>
<th>Std. Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous knowledge as a dummy</td>
<td>51</td>
<td>0</td>
<td>5</td>
<td>2.03</td>
<td>0.96</td>
</tr>
<tr>
<td>Pre-experimental score</td>
<td>67</td>
<td>2</td>
<td>9</td>
<td>5.34</td>
<td>1.66</td>
</tr>
</tbody>
</table>

Table 7: previous knowledge and pre-experimental test scores

Regression analyses with previous knowledge (as a continuous variable) and the pre-experimental scores as determinants of the test scores, did not reveal a significant impact on the relative advantage (withPT – WithoutPT). Only the test score withPT was significantly influenced but to a limited extent influenced by the pre-experimental scores (with an R² of 20%, Beta coefficient of 0.60). Consequently, we conclude that H4 is to a large extent supported. Table 8, showing the non-parametric and conservative Kendall’s tau coefficients, illustrates this pattern. No significant correlations with the relative advantage withPT-withoutPT could be detected. There is only a significant correlation between the pre-experimental test score and the test scores withPT and withoutPT. This means that, not surprisingly, student with better pre-experimental test score, performed better with higher test scores, both withPT and withoutPT, but the effectiveness of the prototype, the relative advantage, was not affected.

<table>
<thead>
<tr>
<th>Previous knowledge</th>
<th>Previous Knowledge</th>
<th>Pre-experimental test scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Pre-experimental scores</td>
<td>N.s.</td>
<td>1</td>
</tr>
<tr>
<td>Test Scores with PT</td>
<td>N.s.</td>
<td>0.33**</td>
</tr>
<tr>
<td>Test Scores without PT</td>
<td>N.s.</td>
<td>0.23*</td>
</tr>
<tr>
<td>Differential Test Scores (With-Without PT)</td>
<td>N.s.</td>
<td>N.s.</td>
</tr>
</tbody>
</table>

N.s = Non-significant; *: p < 0.05; **: p < 0.001

Table 8: correlations results with previous knowledge
The sample consists of 18 female and 44 male students (17.3% and 42.3% respectively), with 42 missing values. Using Anova, no significant differences in the test scores (with and without the prototype), the global differential test score, as well as with the acceptance measures could be detected.

Overall the results provided evidence that personal characteristics of the test participants didn’t influence the simulation effect thus supporting the validity of H4.

7. Contributions and limitations

7.1. Contributions

In the domain of conceptual modelling, not many empirical studies with an experimental design can be found that investigate dimensions of knowledge of conceptual modelling, such as model understanding, language knowledge, modeling knowledge, etc. The largest number of experimental studies are to be found in the domain of business process modelling, yet those studies did not investigate the effects of simulation. This study addresses this gap of lack of empirical and experimental studies in this domain by introducing experimental results that used a strict design with two experimental groups and two cohorts.

From a theoretical perspective this work presents two major contributions: (1) the results contribute to improving our knowledge on the cognitive aspects of conceptual modeling, and (2) the results provide empirical support for the use of augmented feedback in simulation theory.

In terms of cognitive aspects of conceptual modeling, the study provides experimentally supported results that clearly indicate the advantage of the use of feedback-enabled simulation method over traditional methods of learning conceptual modeling (H1, H2). The results also show that the learning effect from the instrument persists when it is no longer used thus improving cognitive aspects of conceptual modeling for novice modelers (H2). The study indicates that personal characteristics of students were not significantly correlated with the learning effects, i.e. different clusters based on gender, varying levels of previous knowledge of data modeling and computer self-efficacy were observed to equally benefit from the use of the feedback-enabled simulation cycle (H4). This suggests that the method can also be integrated in other contexts of data modeling such as databases systems, software engineering, etc. Overall, the results indicate that a systematic use of the feedback-enabled simulation tool can facilitate the progression of novice modelers to advanced levels of expertise at academic level by improving cognitive mechanisms of modeling and validation process. The method also contributes into innovative learning practices (EuropeanComission, 2013) by allowing a learner to learn by experiencing and thus building a deeply understood knowledge that is developed from own practice.

7.2. Limitations

7.2.1. Internal validity

In the experimental design we didn’t consider continuous variables such as timing and also omitted the frequency of the use of a prototype by limiting the experiment to a one cycle test. This could potentially result in decreased learning effect from the simulation. Alternatively, longer learning periods could be considered such as observation over a semester period between a group of students using the simulation method vis-à-vis a group taught by traditional learning method. However, such set-up would induce an ethical problem of unequal learning opportunities for the different groups of students. Although we tried to maximally isolate the maturation effects in the replicated experiments, we nevertheless observed a reversed problem. Despite the maturation effect observed from the simulation cycle within group B in the replicated experiment, we could not attribute this indisputably to a maturation effect, since the effect could not be precisely measured. A few other factors could have their share, such as the altered case or yet the reversed questions (the negative or positive reformulation of questions to avoid distortion effects). We observed this to be the major shortcoming of the experimental design described in this paper. It is also impossible from our experiments to distinguish the proportional impact of feedbacks on the one hand and from the prototype on the other hand. Although we were not able to single out the effects of simulation and of the feedback separately, the annual evaluation of the usage of the simulation environment indicates a high perceived utility of the inclusion of textual explanations and graphical visualizations of execution failures in the prototype as feedback (is not part of the dataset used in this study). In the academic year 2011-2012 (Sedrakyan & Snoeck, 2012) the failure explanations and graphical visualizations were implemented as an optional plugin students could extend their prototypes with. Although the plugin scored slightly higher by students (3.87) on perceived usefulness than the prototyping tool which did not contain feedback features (3.46), the majority of students...
seemed to be reluctant in using the feature mostly resulting in the “didn’t use” answer while evaluating the feature. We attribute this to their low technical skills to cope with the chain of simulation process requiring to first generate a prototype, and next extending it with a plugin. Evaluation by students for the improved tool extended with feedbacks in 2013 resulted in average of 4.58 on perceived usefulness (for the prototyping tool) and 4.52 (for the incorporated feedbacks) on a five-point Likert scale. A separate assessment of feedback however could be a useful extension to this work in order to identify the aspects to achieve an optimized and possibly a personalized feedback.

7.2.2. External validity
The involvement of three different groups of students from two different courses taught by different teachers, at different campuses and spread across two different academic years contributes to the external validity of the results obtained within this study. Nevertheless, the validity of the results is limited to the courses described in this work. The relatively small sample sizes could be a limitation as well. Higher statistical confidence therefore can be established using experimental samples from other universities, as well as more experienced modelers from industry. Since the method targets at educational context, assessing educators attitude could be potentially considered as another omission (Hu, Clark, & Ma, 2003).

8. Conclusions and future work
This study presented an easy and intuitive approach to validate/analyze static systems’ designs by novice analysts with the aid of simulated model (generated prototype application) and augmented feedback. The learning effectiveness was assessed by means of empirical studies observing the impact of the feedback-enabled simulation cycle on the understanding capabilities of conceptual business models by students. The results were analyzed by comparing the test results of students using the simulated model in the process of validating the proposed model solutions to the results of those who didn't. The results confirmed that the method leverages the understanding of model statements by novice learners resulting in significantly improved learning outcomes. This improved knowledge enables students to better understand a model, contributing in this way to what is called pragmatic quality in (Nelson, et al., 2012). At the same time this also enables a student to better assess the semantic quality of a model, in terms of his/her ability to judge the validity of a model with respect to its correctness of representation of the domain at hand.

Despite the positive results the relatively lower indicator on ease of use suggests further improvement. We observed this lower ease of use was due to the limited testing capabilities of students which also resulted in insufficient frequency of what-if scenario trials in their testing patterns. We therefore believe that stimulating testing skills of a novice modeler in parallel with modeling knowledge acquisition can result in even better learning outcomes. To close this gap, tool support to enable automated generation of test scenarios can be investigated. A set of experiments and subsequent extensions are also planned to check the effects for feedback-enabled simulation cycle on understanding the dynamic aspects of a model using UML statecharts, the combination of dynamic and structural aspects of a conceptual model as well as models using inheritance. We also plan to replicate the experiment at industrial seminars to investigate the effect of feedback enabled simulation on the model validation capabilities of (more experienced) practitioners.

References


EuropeanCommission. (2013). Opening up education: Innovative teaching and learning for all through new technologies and open educational resources. Communication from the commission to the European parliament, the council, the European economic and social committee and the committee of the regions.


Appendix A: Experiment cases

Case A: Warehouse

“A warehouse sells textile products and accessories for making clothes (like buttons, zippers, sewing needles, thread, scissors, etc.). The warehouse is a wholesaler of fabrics and accessories. As a result, most of their clients are other textile product warehouses or clothes manufacturers. The warehouse does however also sell to private customers. Some of its products can be bought in small quantities; others are only sold in large quantities. The latter products need to be ordered at least two days in advance. Ordering is not required for buying products that are sold in small quantities.”

A student proposes the following solution:

Which of the following statements is supported by the model? Use the answer sheet do record your answer + a short motivation.

0) (Example) There are no anonymous orders: an order cannot be created without specifying a customer.
1) Ordering is not required for Private Customers buying products that are sold in small quantities.
2) Large Quantity Products will be invoiced to Business Customers only.
3) Product A can be registered only once and then sold both in large and small quantity.
4) When a Large Quantity Product is ordered one needs to create an invoice first.
5) When an order is invoiced, all the invoice lines on that invoice are for the order for which the invoice is created.
6) A Private Customer can buy a Large Quantity Product without ordering it beforehand.
7) If business customer A orders some products, then it is possible that business customer B pays the invoice for these products.
8) Paying an invoice (= state change of that invoice) involves exactly one customer.

Case B: Flower greenhouse

“A flower greenhouse produces and sells premium quality flowers. The greenhouse sells most of its flowers in large quantities to wholesale stores, e.g. tree flowers, flowers in pot, etc. These flowers can be sold via online orders. Online orders should be placed at least a 4 business days in advance. Invoices are sent by the end of the month. The greenhouse also gets advantage from retail sale by selling some of its short-lasting flowers to walk-in customers e.g. cut flowers, bouquets. For walk-in buyers ordering is not required: the sales to these customers are immediately invoiced.”
A student proposes the following solution:

Which of the following statements is supported by the model? Use the answer sheet do record your answer + a short motivation.

0) (Example) In order to register an online order, the wholesale customer needs to be registered first.
1) A same flower needs to be registered twice to be sold to all types of customers.
2) WholeSale Flowers can only be invoiced to Wholesale Customers.
3) A WholeSale Flower cannot be ordered without having to create an invoice.
4) The WholeSale Items that are put together on one invoice could be from different orders than the order that is invoiced.
5) When wholesale customer A orders some products, it is the same wholesale customer A that pays the invoice for these products.
6) Ordering is not required for selling Retail Flowers to Walk-in Customers.
7) Ordering is not required for selling WholeSale Flowers to Walk-in Customers.
8) Paying an invoice (= modification of the state of the invoice) may involve more than one customer.

Appendix B: Evaluation

<table>
<thead>
<tr>
<th>I find the prototype easy to use.</th>
<th>O 1 O 2 O 3 O 4 O 5 O 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>It was easy to learn and understand the prototype.</td>
<td>O 1 O 2 O 3 O 4 O 5 O 6</td>
</tr>
<tr>
<td>I can without effort get the prototype to do what I want it to do.</td>
<td>O 1 O 2 O 3 O 4 O 5 O 6</td>
</tr>
<tr>
<td>Using the prototype improves my understanding of a data model/class diagram.</td>
<td>O 1 O 2 O 3 O 4 O 5 O 6</td>
</tr>
<tr>
<td>Using the prototype makes me understand data models/class diagrams much faster.</td>
<td>O 1 O 2 O 3 O 4 O 5 O 6</td>
</tr>
<tr>
<td>Using the prototype enables me to remove my uncertainty about the presented data models/class diagrams.</td>
<td>O 1 O 2 O 3 O 4 O 5 O 6</td>
</tr>
<tr>
<td>If had the choice, or opportunity I would use prototyping to learn data modeling.</td>
<td>O 1 O 2 O 3 O 4 O 5 O 6</td>
</tr>
<tr>
<td>If I had to vote, I would vote in favor of using prototyping in the classroom</td>
<td>O 1 O 2 O 3 O 4 O 5 O 6</td>
</tr>
<tr>
<td>I am enthusiastic about using the prototyping in this kind of courses</td>
<td>O 1 O 2 O 3 O 4 O 5 O 6</td>
</tr>
</tbody>
</table>
Overall, I am very satisfied with using the prototype

My experience with the prototype meets my general expectations

Using the prototype was a positive experience

Previous knowledge (in terms of having data modeling and/or programming course(s) before)

1 = no knowledge/experience at all; 2 = little knowledge (a few hours course); 3 = moderate knowledge (intermediate level course); 4 = extensive knowledge (advanced course(s))

<table>
<thead>
<tr>
<th>Previous knowledge on data modeling in previous degree</th>
<th>O 1  O 2  O 3  O 4</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Previous knowledge on programming in previous degree</th>
<th>O 1  O 2  O 3  O 4</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Previous knowledge on testing a software in previous degree</th>
<th>O 1  O 2  O 3  O 4</th>
</tr>
</thead>
</table>

Years of programming experience (if applicable)

<table>
<thead>
<tr>
<th>I could use a new software application well . . .</th>
</tr>
</thead>
</table>

1 = not at all confident; 2 = probably not; 3 = rather not; 4 = rather yes; 5 = likely yes; 6 = Totally Confident: Yes

<table>
<thead>
<tr>
<th>... even if I had never used an application like it before.</th>
<th>O 1  O 2  O 3  O 4  O 5  O 6</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>... if I had just the built-in-help facility or manual for assistance.</th>
<th>O 1  O 2  O 3  O 4  O 5  O 6</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>... if I had first seen someone else using it before trying it myself.</th>
<th>O 1  O 2  O 3  O 4  O 5  O 6</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>... using only the internet for assistance.</th>
<th>O 1  O 2  O 3  O 4  O 5  O 6</th>
</tr>
</thead>
</table>

On Average, I use computers (laptop, desktop, tablet) per day:

- [] less than one hour
- [] one to two hours
- [] three to five hours
- [] six to eight hours
- [] eight or more hours