Guided Derivation of Conceptual Models from User Stories: A Controlled Experiment

Maxim Bragilovski[1](✉), Fabiano Dalpiaz[2], and Arnon Sturm[1]

[1] Ben-Gurion University of the Negev, Beer Sheva, Israel
maximbr@post.bgu.ac.il, sturm@bgu.ac.il
[2] Utrecht University, Utrecht, The Netherlands
f.dalpiaz@uu.nl

Abstract. [Context and Motivation] User stories are a popular notation for representing requirements, especially in agile development. Although they represent a cornerstone for developing systems, limited research exists on how user stories are refined into conceptual design. **[Question/Problem]** We study the process of deriving conceptual models from user stories, which is at the basis of information systems development. We focus our attention on the derivation of a holistic view of structural and interaction aspects, represented via class diagrams and use case diagrams, respectively. In this paper, we examine whether providing *guidelines* has an effect on the ability of humans to derive complete and valid conceptual models. **[Principal Ideas/Results]** We design example-based guidelines for the derivation of class and use case diagrams from user stories. Then, we conduct a two-factor, two-treatment controlled experiment with 77 undergraduate students serving as subjects. The results indicate that the guidelines improve the completeness and validity of the conceptual models in cases of medium complexity, although the subjects were neutral on the usefulness of the guidelines. **[Contribution]** The guidelines may assist analysts in the refinement of user stories. Our initial evidence, besides showing how the guidelines can help junior analysts derive high-quality conceptual models, opens the doors for further studies on the refinement of user stories, and to the investigation of alternative guidelines.

Keywords: Requirements engineering · Conceptual modeling · Use cases · Derivation process · Guidelines · Class diagram · User stories · Controlled experiment

1 Introduction

User stories are a popular technique for expressing requirements from a user perspective [8]. Through their simple notation, they represent who expresses a need, what feature is requested, and the rationale behind the feature. The so-called Connextra notation [8] "*As a* ⟨*role*⟩ *I want to* ⟨*feature*⟩ *so that* ⟨*benefit*⟩" is widely used for the representation of the elicited requirements in agile development projects [16,21].

User stories are a central artifact for the subsequent stages of software development [2,26]. In particular, user stories may be *refined* into lower-level specifications.

© Springer Nature Switzerland AG 2022
V. Gervasi and A. Vogelsang (Eds.): REFSQ 2022, LNCS 13216, pp. 131–147, 2022.
https://doi.org/10.1007/978-3-030-98464-9_11

One way to do so is to derive *conceptual models*; this is at the basis of model-driven engineering [5] and, in general, of information systems development.

Conceptual models may represent system functionality; for example, use case diagrams [6] define the roles and the functionality they expect when interacting with the system. Conceptual models can also depict structural aspects by summarizing the major entities and relationships that are referred to in the high-level requirements [15,25,34]. In addition to their use in model-driven engineering [19], conceptual models have been employed in requirements engineering to provide a holistic overview of the product domain and functionality [1,23], for the identification of potential ambiguity [11], and for analyzing quality aspects such as security and privacy [24].

In previous research, we have conducted empirical studies in which we compared user stories and use cases as a starting point for the derivation of structural conceptual models [10,12]. Our results revealed that user stories are better in time-constrained settings [12], while in absence of time constraints, the notations are equivalent and other factors have shown to have a large(r) impact [10], including the complexity of the domain and the use of a systematic derivation process.

Based on these premises, we investigate whether a human analyst's ability to derive conceptual models is influenced by guidelines that illustrate how to construct such models from user stories. While following a systematic derivation process was an emerging factor in previous research [10], here we foster such a systematic approach by providing guidelines. Like in previous research, we study the derivation of a functional conceptual model (use case diagram) and of a structural conceptual model (class diagram). Our research question is as follows: *MRQ. How does the provisioning of guidelines to information systems developers affect the quality of the derived conceptual models?*

In particular, we are going to investigate guidelines that are expressed in the form of examples [14]. Also, we use information systems developers as a general term for system analysts, designers and programmers. To measure the quality, we used the previously adopted metrics of model validity and completeness [10,13].

We answer the MRQ via a controlled experiment in which senior undergrad students were asked to derive conceptual models starting from the user stories for two systems. Half students were provided with the guidelines, half were not. These students serve as a proxy for our target population, which consists of analysts, designers, and developers of information systems. As already mentioned, we assess model quality by measuring the validity and completeness of the models [10,13]. To enable that, the researchers built gold standard conceptual models prior to the experiment's conduction. Furthermore, we assess the students' opinion on the usefulness of and need for guidelines. The results show that the guidelines lead to improved results in terms of validity and completeness, although this is mainly visible in the more complex specification.

Thus, this paper makes two contributions to the literature: (i) we propose example-based guidelines for the derivation of structural and functional conceptual models from user stories; and (ii) we assess the effectiveness and perceived appreciation of the guidelines through an experiment that compares to a baseline group without guidelines.

Organization. In Sect. 2, we set the background for this research and we review related studies. In Sect. 3, we present our devised guidelines. In Sect. 4, we describe the design of our experiment. In Sect. 5, we elaborate on the experiment results whereas in Sect. 6 we interpret and discuss those results. In Sect. 7, we evaluate the threats to validity. We conclude and set plans for future research in Sect. 8.

2 Background and Related Work

User stories are a widespread notation for expressing requirements [16,21], especially in agile development projects [8]. They are simple descriptions of a feature written from the perspective of the stakeholder who wants such a feature. Multiple templates exist for representing user stories [31], among which the Connextra format is the most common [21]: *As a ⟨role⟩, I want ⟨action⟩, so that ⟨benefit⟩*. For example, a user story for a learning management system could be "As an enrolled student, I want to access the grading rubrics, so that I know how my assignments will be evaluated". The 'so that' part, despite its importance in providing the rationale for a user story [20], is often omitted in practice. We consider user stories that are formulated using the Connextra template, and we group related user stories into epics.

Just a few methods exist that derive conceptual models from user stories. Lucassen *et al.* [23] propose an automated approach, based on the Visual Narrator tool, for extracting structural conceptual models (i.e., class diagrams) from a set of user stories. Their work relies on and adapts natural language processing heuristics from the literature. The approach is able to achieve good precision and recall, also thanks to the syntactic constraints imposed by user stories, although perfect accuracy is not possible due to the large variety of linguistic patterns that natural language allows for. Furthermore, the Visual Narrator is limited to the identified lexicon and, unlike humans, is unable to perform the abstraction process that is a key issue in conceptual modeling [27].

Wautelet *et al.* [30] introduce a process for transforming a set of user stories into a holistic use case diagram, which integrates the user stories by using the granularity information obtained through tagging the user stories. Their work focuses on the joint use of two notations, one textual and one diagrammatic.

The same research group [32] proposed one of the few studies on the construction of diagrams from user stories. In particular, they investigate the construction of a goal-oriented model (a rationale tree) that links the who, what, and why dimensions of a user story. Their research shows differences depending on the modeler's background and other factors. While their work is highly related, we focus on a different task, which concerns the derivation of structural and functional conceptual models.

The extraction of conceptual models from natural language description requirements is one of the four types of NLP tools described by Berry *et al.* [3] and a long-standing research thread. We refer the reader to a recent literature review [35] for a comprehensive view; our focus is on humans' ability to derive models, rather than on automated techniques, without over-constraining the humans in the way they specify their requirements or by imposing computer-alike rules for the derivation process.

Very few attempts that test human's ability to extract conceptual models exist. España and colleagues [13] studied the derivation of UML class diagrams from either textual requirements or a requirements model; unlike them, we fix our notation and only study user stories. Some studies compare the effectiveness of automated tools with that of humans. Sagar *et al.* [28] propose a tool that outperforms novice human modelers in generating conceptual models from natural language requirements. This result is achieved thanks to the notational constraints that facilitate the tool; we do not set such constraints in this research.

3 Guidelines for Deriving Models from User Stories

In our earlier experiments on the derivation of conceptual models from requirements (both user stories and use cases) [10, 12], we gave limited guidance to the human participants regarding the way conceptual models should be generated from user requirements. The obtained and compared results, therefore, could have been affected by different interpretations of the derivation task. In earlier work [10], we found out that following a systematic derivation process (self-defined by the subjects) results in higher-quality models. To better control the derivation process, in this work, we set off to define a set of guidelines, with the aim of investigating their effect on the derivation process.

First, we dealt with the issue of what should be the form of the guidelines. We started with a set of linguistic rules, so that one can apply the rules easily by just following them. Our initial aim was to identify effective rules that could later be embedded into an algorithm that could automate their application. This approach was inspired by previous research on the automated derivation of conceptual models, especially the work on the Visual Narrator [23], which employs and adapts NLP heuristics from the conceptual modeling literature in order to derive domain models from user stories. For example, a rule to identify a class diagram entity was "As a ROLE, I want to ACTION on NOUN", where the NOUN would define an entity.

However, after applying the guidelines to some datasets, we encountered several cases in which the rules could not be applied correctly, due to the linguistic variety of natural text. For example, the rule "As a ROLE, I want to ACTION on NOUN" is hard to apply to a user story such as "As a teacher, I want to have an overview of the grades": the verb "to have" does not really represent an action. One could introduce an increasing number of rules, but then the guidelines would become impractical. Furthermore, we realized that applying linguistic rules requires major cognitive efforts.

Therefore, we looked for an alternative way to present the guidelines that will cover many cases, offer flexibility, and require minimal cognitive efforts. We opted for an example-based learning approach [14], which requires less cognitive effort and increases learning outcomes in less time. Such an approach best fits domains in which the tasks are highly structured [14] (such as the task of model derivation), and some background knowledge is required for making learning-by-examples effective. This is also the case we are dealing with, as the guidelines are aimed at developers who are familiar with all artifacts. We built on the principles for designing examples [14], which include focused attention, redundancy avoidance, planning the sub goals, and including a high-level explanation. For more complex cases, for instance, we split the example to have focused rules, with minor repetitions, and with some explanations.

Table 1 presents a few examples of such guidelines, both for use case diagrams and for class diagrams. For example, the first example shows how the role of the user story becomes an actor in a use case diagram, but also that some entity in the rest of the user story can be an actor; here, "researcher". The entire set of guidelines, which consists of 9 examples for use case diagrams and 13 examples for class diagrams, can be found in the experiment forms in the online appendix [4].

Table 1. Some of our example-based guidelines for the derivation of use case diagrams and class diagrams from user stories. The complete guidelines are online [4].

Use Case Diagrams

Example	*Outcome*	*Remarks*
As an administrator, I want to have researchers reset their own passwords, so that I don't have to send passwords in cleartext.	Actors: administrator, researcher	"researcher" is an actor, although not the role of the user story
As an assistant archivist, I want to upload and tag staff generated working papers, so that staff and researchers are able to easily access them.	UCs: (1) upload staff generated working papers; (2) tag staff generated working papers	Two desired actions in the I want part. The so that part does not lead to a use case, as it represents a non-functional property (easily access)

Class Diagrams

Example	*Outcome*	*Remarks*
As an archivist, I want to apply a license or rights statement, so that I know what I can do with a file.	Class: License, Rights statement, File	There may be multiple classes in one user story, also in the so that part
As a researcher, I want to check whether a document has a citation information, so that I can cite accurately in a publication.	Class: Document, Citation Association: Document, Citation.	The "has" verb denotes the association

Note that, for class diagrams, we did not provide guidelines for fine-grained aspects such as multiplicity, association types, and navigation, because we are primarily interested in the derivation of high-level models rather than low-level data models.

4 Experiment Design

We investigate how user stories can be translated into conceptual models with and without providing guidelines. We refer to the manual/human derivation of two types of conceptual models: use case diagrams and class diagrams.

Hypotheses. To compare the differences among the two experimental conditions (i.e., with and without provided guidelines), we measure *validity* and *completeness* [13,18] with respect to gold standard solutions. Furthermore, we collect and compare the perceptions of the subjects with respect to the guidelines (desired or missing).

Although working with guidelines is expected to be easier than using linguistic rules based, e.g., on part-of-speech tags (as per the Visual Narrator heuristics [23]), our example-based guidelines cannot cover all cases: they are incomplete and the analysts using them will have to decide how to adapt them to unseen cases. These observations lead us to the following hypotheses:

- Deriving a use case diagram from user stories with and without guidelines results in equal diagram completeness ($H_0^{UC\text{-}Completeness}$) and validity ($H_0^{UC\text{-}Validity}$)
- Deriving a class diagram from user stories with and without guidelines results in equal diagram completeness ($H_0^{CD\text{-}Completeness}$) and validity ($H_0^{CD\text{-}Validity}$)

Independent Variables. The first variable indicates whether the guidelines were provided (*IV1*). The second independent variable is the case used (*IV2*). It has two possible values: Data Hub (DH) and Planning Poker (PP). These cases are obtained from a publicly available dataset of user story requirements [9]. DH is the specification for the web interface of a platform for collecting, organizing, sharing, and finding data sets. PP are the requirements for the first version of the *planningpoker.com* website, an online platform for estimating user stories using the Planning Poker technique. Table 2 presents a few metrics that characterize the size of the cases.

Table 2. Metrics concerning the user stories and the models.

		Data hub	Planning Poker
User stories	Number of user stories	22	20
Class diagram	Number of entities	15	9
	Number of relationships	16	13
Use case diagram	Number of actors	3	2
	Number of use cases	24	20
	Number of use case relationships	24	22

Dependent Variables. There are two dependent variables, taken from conceptual modeling research [13,18], that we use for measuring the quality of a generated conceptual model. These variables are specified by comparing the elements in the *subject solution* (the conceptual model derived by a subject) against the *gold standard solution*:

- *Validity (DV1)*: the ratio between the number of elements in the subject solution that are in the gold standard (true positives) and the true positives plus the number of elements in the subject's solution that do not exist within the gold standard solution (false positives). In information retrieval terms, validity equates to precision. Formally, $\mathit{Validity} = |TP|/(\,|TP| + |FP|\,)$.
- *Completeness (DV2)*: the ratio between the number of elements in the subject solution that also exist in the gold standard (true positives) and the number of elements in the gold standard (true positives + false negatives). In information retrieval terms, completeness is recall. Formally, $\mathit{Completeness} = |TP|/(\,|TP| + |FN|\,)$.

To measure completeness and validity, we use various ways of counting the elements of a model. For the *use case diagram*, we count the number of use cases and actors, and we ignore the number of relationships. For the *class diagram*, we first count only the number of classes. Next, we count the classes and the attributes. In all these metrics, we consider the importance of the appearance of each element equally to avoid bias: we did not favor a class or an attribute. Since relationships can only be identified when the connected entities are identified, we use an *adjusted* version of validity and completeness for the relationships [10], which calculates them with respect to those relationships in the gold standard among the entities that the subject has identified.

Subjects. In an optimal setting, we would have used experienced analysts, designers, and developers of information systems as subjects. However, this is a practically challenging task. Thus, we followed convenience sampling and we involved third-year undergraduate students taking a project workshop that follows a course on *Object-Oriented Analysis and Design* at Ben-Gurion University of the Negev. The course teaches how to analyze, design, and implement information systems based on the object-oriented paradigm. In the course, the students-subjects were taught about modeling techniques, including class and use case diagrams. The instructor of the course was the third author of this paper. The students learned user stories and use cases for specifying requirements as part of the development process. They also practiced class diagrams, use cases, and user stories through homework assignments, in which they achieved good results, indicating that they understood the concepts well. All subjects were taught the same material and the guidelines were not included as part of the course. Recruiting the subjects was done on a volunteering basis. Nevertheless, they were encouraged to participate in the experiment by providing them with additional bonus points to the course grade based on their performance. Before recruiting the subjects, the research design was submitted to and approved by the department's ethics committee.

Task. We designed the experiment so that each subject would experience the derivation of the two conceptual models following one case (either with or without provided guidelines). For that purpose, we designed four forms (available online [4]), in which we alternate the treatment and the case.

The form has three parts: (1) a pre-task questionnaire that checks the subjects' background and knowledge; (2) the task, in which subjects receive the user stories of one application (DH or PP), with or without the guidelines and were asked to derive the conceptual models - one class diagram and one use case diagram for the entire set; We asked the subjects to derive a use case diagram and a class diagram that would serve as the backbone of the system to be developed, as taught in the course. (3) questions about the subjects' perception regarding the task they performed.

To create the gold standard (in the online appendix), the second and third authors applied the guidelines and independently created four conceptual models: a class diagram and a use case diagram for either case. Then, these authors compared the models and produced the reconciled versions, involving the first author for a final check.

Execution. The experiment took place in a dedicated time slot and lasted approximately 1 hour, although we did not set a time limit for the subjects. The assignment of the groups (i.e., the forms) to subjects was done randomly. The distribution of groups was as follows: (i) DH, guided: 19 students; (ii) DH, not-guided: 18 students; (iii) PP, guided: 21 students; and (iv) PP, not-guided: 19 students. Note that the students that were provided with the guidelines have seen them for the first time in the experiment.

Analysis. The paper forms delivered by the students were checked against the gold standard by one researcher who was unaware of the purpose of the experiment, so to avoid confirmation bias. When checking the forms we were flexible regarding the alignment with the gold standard. In essence, the gold standard served as a proxy for the examination. For example, we allowed for synonyms and related concepts. This led to the spreadsheet in our online appendix; there, each row denotes one subject, while each column indicate elements in the gold standard; we also count how many additional elements were identified by the subjects. The statistical analysis was conducted mostly using Python, while the effect size was calculated using an online service at https://www.socscistatistics.com/effectsize/default3.aspx.

5 Experiment Results

We present the results by comparing the groups through their responses in the background questionnaire in Sect. 5.1. We statistically analyze the validity and completeness of the models in Sect. 5.2, then present the students' opinion in Sect. 5.3. Finally, in Sect. 5.4, we provide additional qualitative insights by reviewing in depth the results.

5.1 Background Questionnaire

We run a series of analyses over the results (all materials are available online [4]). In order to determine whether the groups are balanced, we compare their background. Table 3 compares the groups according to four criteria. For each criterion, it presents the arithmetic mean ($\overline{x}$), the standard deviation (σ), the number of participants (N) that responded to the pre-questionnaire, and whether the groups are significantly different. We adopt this structure also for all the following tables. In some rows, the number of participants differs from what was listed earlier because some participants did not complete all the tasks in the experiment. With respect to the background questionnaire, all the responses were self-reported. Familiarity questions were ranked using a 5-point Likert-type scale (1 indicates low familiarity and 5 indicates high familiarity), while the (up to date) GPA is on a scale from 0 to 100. For the familiarity criteria, since they deviate from the normal distribution (following Kolmogorov-Smirnov test), we perform the Mann-Whitney test while for the GPA we perform the T-Test.

Table 3. Pre-questionnaire results: mean, standard deviation, significance.

	PP							DH						
	GUIDED			!GUIDED			Sig.	GUIDED			!GUIDED			Sig.
	$\overline{x}$	σ	N	$\overline{x}$	σ	N		$\overline{x}$	σ	N	$\overline{x}$	σ	N	
CD Familiarity	2.15	0.67	21	2.26	0.65	19	0.926	2.44	0.62	18	2.17	0.62	18	0.177
UCD familiarity	2.75	0.72	21	3.00	0.67	19	0.203	2.89	0.76	18	3.06	0.94	18	0.530
US familiarity	2.80	0.52	21	2.53	0.77	19	0.144	2.74	0.73	19	2.33	0.69	18	0.100
GPA	82.30	5.05	21	82.63	3.98	19	0.759	83.00	4.88	19	80.00	3.74	16	0.052

The results of the statistical tests evidence that the random assignment of the subjects to the four groups, as explained in Sect. 4, does not yield any statistically significant difference that may influence the validity of the results.

5.2 Completeness and Validity of the Derived Models

We analyze the completeness and validity of the conceptual models derived by the students. To do so, we perform the analysis for each case separately due to the different complexity of the domains and of the conceptual models. Table 4 and Table 5 present the results of the DH and PP cases, respectively. For each group, we report the mean, the standard deviation, and the number of responses for the related metric. Bold numbers indicate the best results for a given metric. We also report statistical significance (applying T-Test) and denote statistically significant results (with $p < 0.05$) via gray rows. Finally, we report effect size using Hedges' g. For the qualitative interpretation, we refer to Cohen [7]: small effect when $g > 0.2$, medium effect when $g > 0.5$, large effect when $g > 0.8$.

Table 4. Data hub results.

	GUIDED			!GUIDED			sig.	*Effect size*
	$\overline{x}$	σ	N	$\overline{x}$	σ	N		*(Hedges' g)*
UC Completeness	**0.76**	**0.14**	**19**	0.58	0.14	16	p<0.001	1.296
UC Validity	**0.89**	**0.08**	**19**	0.87	0.09	16	0.473	0.250
CD Class Completeness	**0.37**	**0.12**	**18**	0.36	0.09	18	0.917	0.038
CD Class Validity	**0.67**	**0.18**	**18**	0.62	0.19	18	0.454	0.253
CD Class+Att Completeness	**0.31**	**0.16**	**18**	0.29	0.09	18	0.698	0.131
CD Class+Att Validity	**0.58**	**0.18**	**18**	0.44	0.15	18	0.012	0.880
CD Class+Att+relationships Completeness	**0.36**	**0.13**	**18**	0.33	0.10	18	0.425	0.272
CD Class+Att+relationships Validity	**0.50**	**0.12**	**18**	0.37	0.12	18	0.002	1.083

Table 5. Planning poker results.

	GUIDED			!GUIDED			sig.	*Effect size (Hedges' g)*
	$\overline{x}$	σ	N	$\overline{x}$	σ	N		
UC Completeness	**0.64**	**0.21**	**21**	0.53	0.23	18	0.132	0.494
UC Validity	0.84	0.12	21	**0.87**	**0.14**	**18**	0.524	0.208
CD Class Completeness	0.53	0.15	20	**0.57**	**0.19**	**19**	0.480	0.226
CD Class Validity	**0.78**	**0.14**	**20**	0.74	0.11	19	0.279	0.357
CD Class+Att Completeness	0.49	0.13	20	**0.51**	**0.19**	**19**	0.622	0.161
CD Class+Att Validity	**0.60**	**0.12**	**20**	0.53	0.12	19	0.073	0.585
CD Class+Att+relationships Completeness	0.57	0.13	20	**0.59**	**0.18**	**19**	0.695	0.129
CD Class+Att+relationships Validity	**0.59**	**0.12**	**20**	0.56	0.10	19	0.435	0.251

For the DH case (Table 4), the conceptual models derived by the subjects who had the guidelines outperformed those derived by those subjects who did not have the guidelines, for all metrics. The difference was statistically significant in the case of UC completeness and in the cases of class diagrams validity including also attributes and relationships. Furthermore, the effect sizes for DH statistically significant differences indicate *a large effect* [7].

For the PP case (Table 5), the results are mixed and statistical significance is never achieved. Therefore, we cannot reject $H_0^{\mathit{UC\text{-}Completeness}}$ nor $H_0^{\mathit{CD\text{-}Validity}}$. While the guided subjects outperformed the non-guided ones for UC completeness, the non-guided ones had higher validity for the use case diagrams. The opposite situation occurs for class diagrams: completeness is higher for the non-guided ones, validity is higher for the guided subjects.

Based on the results, we can conclude that for the Data Hub case we can reject $H_0^{\mathit{UC\text{-}Completeness}}$ and $H_0^{\mathit{CD\text{-}Validity}}$ hypotheses on the equality of having guidelines or not for deriving conceptual modes for the metrics defined above (the grey rows in Table 4). In that case, introducing the guidelines resulted in better conceptual models. For the other metrics, we accept the H_0 hypotheses and can infer that no difference exists when providing the guidelines or not for deriving conceptual models.

5.3 Subjects' Opinion

Table 6 presents the participants' opinions on the performed task, which we collected via a post-questionnaire. The participants were asked to use a 5-Likert scale to rank their agreement with the various statements. With respect to deriving the conceptual model elements, no statistically significant differences were found (applying T-Test) between the guided and the non-guided groups in most cases. For PP, which has simpler models, the provided guidelines did not contribute and even blurred the process. In the case of DH, with a more complex model, the guidelines are perceived as supportive, to some

Table 6. Post-questionnaire results: mean, standard deviation, significance. We use the following abbreviations: Der. for Deriving, Guid. for Guidelines

	PP							DH						
	GUIDED			!GUIDED			Sig.	GUIDED			!GUIDED			Sig.
	$\overline{x}$	σ	N	$\overline{x}$	σ	N		$\overline{x}$	σ	N	$\overline{x}$	σ	N	
Der. UC is easy	2.95	0.76	20	3.00	0.77	18	0.851	2.95	1.03	19	2.94	0.87	18	0.987
Der. actors is easy	1.65	0.93	20	1.72	0.67	18	0.429	1.68	0.67	19	1.89	0.96	18	0.678
Der. classes is easy	2.26	0.65	19	2.72	0.57	18	0.021	3.11	0.57	19	2.78	0.65	18	0.131
Der. class att. is easy	2.75	0.79	20	3.00	0.91	18	0.334	3.47	0.84	19	3.39	0.92	18	0.923
Der. relationships is easy	3.40	0.94	20	3.11	0.68	18	0.381	3.44	0.86	18	3.17	0.99	18	0.390
Guid. for UC are required				1.83	0.62	18					2.72	1.02	18	
Guid. for actors are required				2.83	1.15	18					3.17	1.34	18	
Guid. for classes are required				2.94	1.00	18					2.71	0.85	17	
Guid. for class att. are required				2.89	1.08	18					2.56	1.20	18	
Guid. for relationships are required				2.00	0.69	18					2.33	0.97	18	
Guid.for UC were useful	2.00	0.92	20					2.32	0.95	19				
Guid. for actors were useful	2.75	1.12	20					2.00	0.82	19				
Guid. for classes were useful	2.65	0.81	20					2.58	1.02	19				
Guid. for class att.were useful	2.40	1.14	20					3.00	1.05	19				
Guid. for relationships were useful	2.40	1.19	20					3.16	1.07	19				

extent, for the derivation process. As for the usefulness of the guidelines (lines 11–15 in the table), the subjects indicate limited satisfaction (ranging from 2–3.16 out of 5) and the subjects who did not get the guidelines (lines 6-10 in the table) thought that these are of limited importance (ranging from 1.83–3.17 out of 5)

5.4 Qualitative Insights

We provide qualitative observations by drilling down into the derived conceptual models and by analyzing the alignment of the individual elements (each use case, class, relationship, attribute) with the gold standard solution. To do so, we used the spreadsheet in our online appendix that reports on the alignment of individual elements.

Data Hub. For this first case, with respect to the system functionality via the use case diagram, we observe the following:

1. As expected, all subjects were able to identify all actors in both groups.
2. It seems that the subjects who received the guidelines were able to better identify the use cases. This might be because the guidelines demonstrate the derivation of use cases from the *so that* part. See, e.g., user story E2.5: “so that I can validate the data I am about to publish”. Another contribution of the guidelines is that it explicates the important role of the *I want* part. This allows to systematically analyze the user stories without judging their perceived importance; for example, see E3.1: “see real examples of published packages” where the average completeness of the group provided with the guidelines was 0.632 whereas for the other group it was 0.125 or E3.4: “download the data package in one file” the average completeness of the group provided with the guidelines was 1 and for the other group it was 0.75.

With respect to the system structure via the class diagram:

1. The classes Site, Pricing plan, Account, Consumer, Data Package, and Publisher were identified by both groups to a medium-to-large extent (44%–94%). These are core classes in the domain, which are easy to identify even without guidelines.
2. The classes Site Deployment, Key Metric, Billing System, and Configuration Parameter were identified to a limited extent both by the subjects who received the guidelines and those who did not (0–22.2%). Our conjecture is that the subjects considered them to be technical issues; also, they appear only in epic 4.
3. The classes Data, Tag, Single download file, Example, and Published Data Package were also identified to a limited extent by both groups (0–28%). Here again, it seems that the subjects found those classes of limited importance to the domain.
4. With respect to the identification of attributes, completeness was limited in both groups. This is probably due to the fact that the subjects consider those of limited importance, focusing on giving a higher-level overview of the domain.
5. With respect to relationships, it seems that the students who received the guidelines were able to better identify the relationships between the classes when referring to the classes that were identified. This might be attributed to the provided guidelines.

Planning Poker. For the PP case, with respect to the use case diagram:

1. All the actors were identified by all subjects in both groups.
2. Use cases were identified to a satisfactory level. The subjects using the guidelines better identified use cases that appear in the *so that* part. For example, this happened for the user stories, and corresponding use cases, regarding starting the game. Another difference between the groups concerns the user stories that refer to presenting information, e.g., "show all estimates"(the completeness of the group that was provided with the guidelines was 0.571 and for the other group it was 0.157) or "accept the average of all estimates" (the completeness of the group that was provided with the guidelines was 0.619 and for the other group it was 0.389).

With respect to the system structure via the class diagram:

1. The subjects in both groups were able to identify important classes such as Game, Estimator, and Item.
2. For some reason, the class Round was not always identified (∼80%), although it appears five times in the user stories.
3. The Policy class (referring to the estimation policy) and its sub-classes defining specific policies were identified to a very limited extent, probably as they were not explicated in the user stories and appeared only once.
4. The class Estimate was less frequently identified by the subjects that received the guidelines. This may have happened since, although a concept, the user stories were often referring to this notion using the verb *to estimate*, rather than a noun.
5. Attributes were derived to a certain extent, but only limited.
6. Relationships were identified to a satisfactory level. No significant differences can be observed between the two groups.

6 Discussion

The results indicate that the guidelines support the derivation process only to some extent. It seems that, as the complexity of the derived models increases (because of their size, or because of specificity of the domain), the guidelines further improve the validity and completeness of the models.

Finding 1

The guidelines seem to lead to increased validity and completeness for more complex domains, while they do not seem effective for more straightforward domains.

As partially highlighted in Table 2, the DH models were more complex than those of the PP case. For DH, the complexity emerges due to various factors: the number of entities, the number of relationships, the introduction of an external system (for billing) with which the system under design interacts, the multiple interactions among the roles/actors, and the existence of several related roles/actors with similar names. In the DH case, in all metrics, the subjects who got the guidelines achieved better results than those who did not get the guidelines. Although only some of the results are of statistical significance, the trend is clear. In the PP case, those who received the guidelines delivered better models, but the difference was of lower magnitude. These results are in line with our previous experiments [10], in which we found complexity to be a more significant factor than the notation used as a starting point for the derivation of a conceptual model. Also, our previous research [10] pointed out how the students who followed a systematic derivation process obtained better results; here, we fostered (but could not enforce) the adoption of such a process by providing guidelines.

Finding 2

Despite leading to better results in more complex settings, the guidelines are not perceived as useful by the subjects.

The derivation of a conceptual model requires mental effort. While the guidelines create awareness about the expected output, the participants may see the guidelines as a constraining mechanism that limits their ability to analyze the requirements, to identify the relevant concepts, and to assemble those concepts into a model. In addition, the subjects were introduced to the guidelines for the first time during the experiment. They could have ignored some of these while focusing on the actual task based on their own skills. Nevertheless, the example-based guidelines shed light on parts of the user stories that might be neglected by just reading them. For example, the guidelines point to several possibilities: a role can appear in the ⟨*action*⟩ part, multiple functions may be present in the ⟨*action*⟩ part, a function can emerge from the ⟨*so that*⟩ part, consider a generalization of several user stories, multiple entities may exist in one user story, etc.

Finding 3

The inclusion of a type of concept/element in a conceptual model does not depend only on the guidelines, but also on its perceived importance for the model.

Our guidelines included references to all major concepts: use cases, actors, and associations for the use case diagram, and classes, attributes, and relationships for the class diagram. However, *attributes* were included only to a limited extent both in the PP and in the DH cases, with or without guidelines. Since the subjects were already filtering the concepts based on the perceived importance, they have probably ranked the attributes as less important than the classes, and, therefore, they could be excluded. The inclusion or exclusion of attributes depends on the task at hand: if we had specified that the class diagram would be used as a blueprint for detailed design (e.g., data structures or a database schema), perhaps they would have paid attention to attributes too. Alternatively, we could have used specific guidelines which could convey the importance of certain concept types, rather than leaving the choice to the subject's perception.

7 Threats to Validity

Our results need to be considered in view of threats to validity. We follow Wohlin *et al.*'s classification [33]: construct, internal, conclusion, and external validity.

Construct validity concerns the relationships between theory and observation and these threats are mainly due to the method used to assess the outcomes of the tasks. We examined if the use of guidelines improves conceptual model derivation. The domains selection may affect the results; our choice is justified by our attempt to provide domains that would be easy to understand. Also, in the experiment, we adopt a fixed set of guidelines. Other sets of guidelines may lead to different results. The subjects have seen the guidelines for the first time during the experiment. Thus, it might be that they were able to absorb the guidelines only to a limited extent, and the positive effect that we identified in the experiment could be larger if the guidelines were learned beforehand. Finally, for practical reasons, we purposefully selected a small set of user stories to be analyzed by the subjects: this may not be representative of real-world tasks. Yet, earlier research has shown that generating conceptual models from many user stories may just transfer the cognitive complexity from text to models [22]. Thus, the manual derivation of such models is better suited for relatively small, cohesive collections of requirements.

Internal validity threats, which concern external factors that might affect the dependent variables, may be due to individual factors, such as familiarity with the domain, the degree of commitment by the subjects, and the training level the subjects underwent. These effects are mitigated by our experiment design. It is unlikely that the subjects were already familiar with the two chosen domains (although they were familiar with the notion of agile development, they were not taught the planning poker procedure). The random assignment that was adopted should eliminate various kinds of external factors. Although the experiment was done on a voluntary basis, the subjects were told that they would earn bonus points based on their performance, and thus we increased the motivation and commitment of the subjects, which could have led them to increase the time on task. Eventually, all subjects received the entire bonus points based on their participation (this was approved by the ethics committee).

Conclusion validity concerns the relationship between the treatment (the notation) and the outcome. We followed the various assumptions of the statistical tests (such as normal distribution of the data and data independence) when analyzing the results. In addition, we used a predefined solution, which was established before the experiment, for grading the subjects' answers; thus, only limited human judgment was required. In addition, as we allow flexibility with respect to the gold standard, it might be that further subjectivity was involved. Another matter the requires attention is that an alternative gold standard could be presented. To mitigate that threat, we discussed the used gold standard among the research team.

External validity concerns the generalizability of the results. The main threats are the choice of subjects and the use of simple experimental tasks. The subjects were undergraduate students with little experience in software engineering, in general, and in modeling in particular. Kitchenham *et al.* argue that using students as subjects instead of software engineers is not a major issue as long as the research questions are not specifically focused on experts [17]. Our main research question studies a task (the derivation of conceptual models) that is part of the educational path of students, and we, therefore, consider the students as an appropriate proxy. Nevertheless, experiments with experienced developers should be conducted to test our assumption. In addition, the presentation of the guidelines may have affected the results (for example, presenting the guidelines as a list and not as a table, maybe also with different examples). Generalization should be taken with care, as our cases are small and might differ from specifications in industry settings.

8 Summary

We provided initial evidence on the effect of providing guidelines for deriving conceptual models from (user story) requirements. This is an important task in information systems development, and we expect the task's importance to grow with the increasing interest in low-code development platforms that embrace the model-driven development of information systems.

We conducted a controlled experiment with 77 undergraduate students as part of a third-year course. The results indicate that the provision of example-based guidelines may increase validity and completeness in the case of non-trivial specifications, although the subjects were rather neutral on the perceived usefulness of the guidelines.

This work calls for further experimentation that analyzes the effect of domain complexity, involves experienced developers, considers other forms of guidelines (such as explicit rules, other examples), and offers comprehensive training before conducting the experiment. It would be important to investigate the use of refined user stories (e.g., via acceptance criteria) as a basis for the derivation process. Moreover, interactive approaches that combine the automated derivation of a model with human refinement should be considered (for example, see Saini *et al.* [29]). Finally, our research so far has relied on an assessment of a model against a gold standard; future research could consider alternative evaluation methods that measure the *quality-in-use* of the generated conceptual models.

References

1. Arora, C., Sabetzadeh, M., Nejati, S., Briand, L.: An Active learning approach for improving the accuracy of automated domain model extraction. ACM Trans. Softw. Eng. Methodol. **28**(1), 1–24 (2019)
2. Berends, J., Dalpiaz, F.: Refining user stories via example mapping: an empirical investigation. In: Proceedings of RE, Industrial Innovation Track (2021)
3. Berry, D., Gacitua, R., Sawyer, P., Tjong, S.: The case for dumb requirements engineering tools. In: Proceedings of REFSQ, pp. 211–217 (2012)
4. Bragilovski, M., Dalpiaz, F., Sturm, A.: Guided derivation of conceptual models from user stories. Online Appendix (2021). https://doi.org/10.5281/zenodo.5905846
5. Brambilla, M., Cabot, J., Wimmer, M.: Model-Driven Software Engineering in Practice, 2 edn. Morgan & Claypool Publishers, San Rafael (2017)
6. Cockburn, A.: Writing Effective Use Cases. Addison-Wesley Professional, Boston (2000)
7. Cohen, J.: Statist. Power Anal. Current directions in psychological science **1**(3), 98–101 (1992)
8. Cohn, M.: User Stories Applied: for Agile Software Development. Addison Wesley, Boston (2004)
9. Dalpiaz, F.: Requirements Data Sets (User Stories) (2018), Mendeley Data, v1. https://doi.org/10.17632/7zbk8zsd8y.1

10. Dalpiaz, F., Gieske, P., Sturm, A.: On deriving conceptual models from user requirements: an empirical study. Inf. Softw. Technol. **131**, 106484 (2021)
11. Dalpiaz, F., van der Schalk, I., Brinkkemper, S., Aydemir, F.B., Lucassen, G.: Detecting Terminological Ambiguity in User Stories: Tool and Experimentation. Inform, Software Tech (2019)
12. Dalpiaz, F., Sturm, A.: Conceptualizing requirements using user stories and use cases: a controlled experiment. In: Proceedings of REFSQ, pp. 221–238 (2020)
13. España, S., Ruiz, M., González, A.: Systematic derivation of conceptual models from requirements models: a controlled experiment. In: Proceedings of RCIS, pp. 1–12. IEEE (2012)
14. van Gog, T., Rummel, N.: Example-based learning: integrating cognitive and social-cognitive research perspectives. Educ. Psychol. Rev. **22**(2), 155–174 (2010)
15. Insfran, E., Pastor, O., Wieringa, R.: Requirements Engineering-based conceptual modelling. Req. Eng. **7**(2), 61–72 (2002)
16. Kassab, M.: An empirical study on the requirements engineering practices for agile software development. In: Proceedings of EUROMICRO SEAA, pp. 254–261 (2014)
17. Kitchenham, B.A., et al.: Preliminary guidelines for empirical research in software engineering. IEEE Trans. Softw. Eng. **28**(8), 721–734 (2002)
18. Lindland, O.I., Sindre, G., Solvberg, A.: Understanding quality in conceptual modeling. IEEE Softw. **11**(2), 42–49 (1994)
19. Loniewski, G., Insfran, E., Abrahão, S.: A Systematic Review of the Use of *Requirements Engineering Techniques in Model-driven Development*. In: Proceedings of MODELS, pp. 213–227 (2010)
20. Lucassen, G., Dalpiaz, F., van der Werf, J., Brinkkemper, S.: Improving agile requirements: the quality user story framework and Tool. Requir. Eng. **21**(3), 383–403 (2016)
21. Lucassen, G., Dalpiaz, F., Werf, J.M.E.M., Brinkkemper, S.: The use and effectiveness of user stories in practice. In: Daneva, M., Pastor, O. (eds.) REFSQ 2016. LNCS, vol. 9619, pp. 205–222. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30282-9_14
22. Lucassen, G., Dalpiaz, F., van der Werf, J.M.E.M., Brinkkemper, S.: Visualizing User story requirements at multiple granularity levels via semantic relatedness. In: Comyn-Wattiau, I., Tanaka, K., Song, I.-Y., Yamamoto, S., Saeki, M. (eds.) ER 2016. LNCS, vol. 9974, pp. 463–478. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46397-1_35
23. Lucassen, G., Robeer, M., Dalpiaz, F., van der Werf, J.M.E., Brinkkemper, S.: Extracting conceptual models from user stories with visual narrator. Requir. Eng. **22**(3), 339–358 (2017)
24. Mai, P.X., Goknil, A., Shar, L.K., Pastore, F., Briand, L.C., Shaame, S.: Modeling security and privacy requirements: a Use case-driven approach. Inform. Softw. Tech. **100**, 165–182 (2018)
25. Maiden, N.A.M., Jones, S.V., Manning, S., Greenwood, J., Renou, L.: Model-driven requirements engineering: synchronising models in an air traffic management case study. In: Proceedings of CAiSE, pp. 368–383 (2004)
26. Müter, L., Deoskar, T., Mathijssen, M., Brinkkemper, S., Dalpiaz, F.: Refinement of user stories into backlog items: linguistic structure and action verbs. In: Proceedings of REFSQ, pp. 109–116 (2019)
27. Parsons, J., Wand, Y.: Choosing classes in conceptual modeling. Commun. ACM **40**(6), 63–69 (1997)
28. Sagar, V.B.R.V., Abirami, S.: Conceptual modeling of natural language functional requirements. J. Syst. Softw. **88**, 25–41 (2014)
29. Saini, R., Mussbacher, G., Guo, J.L., Kienzle, J.: Automated traceability for domain modelling decisions empowered by artificial intelligence. In: Proceedings of RE, pp. 173–184. IEEE (2021)

30. Wautelet, Y., Heng, S., Hintea, D., Kolp, M., Poelmans, S.: Bridging user story sets with the use case model. In: Link, S., Trujillo, J.C. (eds.) ER 2016. LNCS, vol. 9975, pp. 127–138. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47717-6_11
31. Wautelet, Y., Heng, S., Kolp, M., Mirbel, I.: Unifying and extending user story models. In: Proceedings of CAiSE, pp. 211–225 (2014)
32. Wautelet, Y., Velghe, M., Heng, S., Poelmans, S., Kolp, M.: On modelers ability to build a visual diagram from a user story set: A goal-oriented approach. In: Proceedings of REFSQ, pp. 209–226 (2018)
33. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer, Berlin (2012). https://doi.org/10.1007/978-3-642-29044-2
34. Yue, T., Briand, L.C., Labiche, Y.: A systematic review of transformation approaches between user requirements and analysis models. Requir. Eng. **16**(2), 75–99 (2011)
35. Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K.J., Ajagbe, M.A., Chioasca, E.V., Batista-Navarro, R.T.: Natural language processing for requirements engineering: a systematic mapping study. ACM Comput. Surv. (CSUR) **54**(3), 1–41 (2021)