Comparing the comprehensibility of requirements models expressed in Use Case and Tropos: Results from a family of experiments

Irit Hadar a,*, Iris Reinhartz-Berger a, Tsvi Kuflik a, Anna Perini b, Filippo Ricca c, Angelo Susi b

a Department of Information Systems, University of Haifa, Carmel Mount, 31905 Haifa, Israel
b Center of Information Technology (CIT-Irst) of FBK, Povo, I-38123 Trento, Italy
c Dipartimento interscuola di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi (DIBRIS), University of Genova, Via Dodecaneso 35, 16146 Genova, Italy

ARTICLE INFO

Article history:
Received 18 March 2012
Received in revised form 9 May 2013
Accepted 14 May 2013
Available online 6 June 2013

Keywords:
Empirical studies
Controlled experiment
Requirements models
Model comprehension
UML use case
Tropos

ABSTRACT

Context: Over the years, several modeling languages for requirements have been proposed. These languages employ different conceptual approaches, including scenario-based and goal-oriented ones. Empirical studies providing evidence about requirements model comprehensibility are rare, especially when addressing languages that belong to different modeling approaches.

Objective: This work aims to compare the comprehensibility of requirements models expressed in different but comparable modeling approaches from a requirements analysts’ perspective. In particular, in this paper we compare the comprehensibility of requirements models expressed in two visual modeling languages: Use Case, which is scenario-based, and Tropos, which exploits goal-oriented modeling. We further compare the effort required for comprehending the different models, and the derived productivity in each case.

Method: Requirements model comprehensibility is measured here in the context of three types of tasks that analysts usually perform, namely mapping between textual description and the model elements, reading and understanding the model irrespective of the original textual description, and modifying the model. This experimental evaluation has been conducted within a family of controlled experiments aiming at comparing the comprehensibility of Use Case and Tropos requirements models. Three runs of the experiment were performed, including a first experiment and two replications, involving 79 subjects overall (all of which were information systems students). The data for each experiment was separately analyzed, followed by a meta-analysis of the three experiments.

Results: The experimental results show that Tropos models seem to be more comprehensible with respect to the three types of requirements analysis tasks, although more time consuming than Use Case models.

Conclusions: Measuring model comprehensibility by means of controlled experiments is feasible and provides a basis for comparing Tropos and Use Case models, although these languages belong to different modeling approaches. Specifically, Tropos outperformed Use Case in terms of comprehensibility, but required more effort leading to a similar productivity of the two languages.

* Corresponding author. Tel.: +972 4 8288508; fax: +972 4 8288522.
E-mail addresses: hadari@ts.haifa.ac.il (I. Hadar), iris@ts.haifa.ac.il (I. Reinhartz-Berger), tsvikak@ts.haifa.ac.il (T. Kuflik), perini@fbk.eu (A. Perini), filippo.ricca@dist.unige.it (F. Ricca), susi@fbk.eu (A. Susi).

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Requirements specification has received considerable research attention as it provides methods, tools, and techniques for representing requirements clearly, consistently, and unambiguously. In this context, requirements models enable knowledge sharing and integration among the different stakeholders in projects, such as customers, analyzers, designers, developers, maintainers, and users. Over the years, different modeling approaches have been proposed for specifying requirements. Particularly, goal-oriented (GO) and scenario-based (SB) approaches gained recognition from both the research community and practitioners [44,61]. Each modeling approach offers its own specific set of concepts to represent the properties of the systems that are being analyzed. The basic building blocks in the GO approach, for example, are the goals of the system’s stakeholders, which are used for the purpose of analysis and to capture the various requirements demanded from the system, to be developed at different levels of abstraction [61,64]. The SB approach, on the other hand, promotes the notion of a scenario, which is a sequence of interaction events between a system and its environment in the restricted context of realizing some implicit purposes [58].
GO and SB can be seen as complementary approaches. However, in spite of the complementary nature of the two approaches, in real projects, frequently only one approach is selected, since subjective criteria, such as the project manager’s previous experience and the level of the analysts’ familiarity with a specific modeling language, are taken into consideration. An example is given in [5], where the members of two teams working on the same project selected different approaches for the analysis of the requirements of two different areas of the domain of which they were in charge. One team selected Use Case modeling, which is considered SB, while the other team selected Tropos [10], which is based on the GO approach. Deciding which approach should be adopted for a given project calls for the definition of measurable properties that can provide data useful to compare them.

In our study, we focus on comprehensibility as a basis for comparing modeling languages, since the abstract goal of modeling is to formally describe some aspects of the physical and social world around us for the purpose of understanding and communication [42]. Comprehensibility may represent the bottom line: how well is the domain knowledge that is captured and represented in a model communicated to different stakeholders (e.g., developers and customers)? Assessing comprehensibility of requirements models is known to be challenging, but it is considered an important aspect of requirements engineering research [6,55].

Several previous research works investigated how to assess conceptual schema comprehensibility, requiring participants to address a series of comprehension tasks (e.g., [33,34]). In particular, empirical studies that focus on examining the comprehensibility of requirements models may provide insights and facilitate the decision as to which modeling approach, and associated language, should be adopted for a given (phase of a) development project. This information can complement other information relevant to this type of decision: clear documentation of a language, as well as tutorials and supporting tools. Such comparisons may lead to further enhancement of the languages themselves by indicating extensions that may suit the needs of their users better.

In this paper, we describe an empirical work devoted to measuring the comprehensibility of requirements models expressed in different visual modeling approaches. Specifically, we explore how well requirements models expressed in the well-known Use Case (UC), which is an SB technique, and Tropos [10], which is a widely used GO method [7], are understood. To this end, we consider comprehension in the context of three basic types of tasks that analysts usually perform, namely, mapping between the textual description of the intended system and the corresponding model elements, reading the model irrespectively of the textual description, and modifying the model. To show these tasks at work, we selected two simple systems, textually described the intended usage of the systems (hereafter called system stories), modeled the system stories in the two different languages separately, and prepared a questionnaire consisting of 14 items to test comprehensibility. We further measured the effort required by subjects when conducting the aforementioned three types of tasks. Based on this design, we executed a family of controlled experiments, including a first experiment and two replications, which together involved 79 subjects. The data collected in these experiments were first analyzed separately for each experiment, followed by a meta-analysis of the three experiments.

The rest of the paper is structured as follows. Section 2 provides a background about SB and GO approaches, elaborates on UC and Tropos through their metamodels, and discusses relevant related work on modeling approach comparisons and visual models’ comprehensibility. Section 3 sketches the family of experiments. Section 4 describes the experimental design of the individual experiments, following the guidelines and terminology of Wohlin et al. [63]. Section 5 presents the results, and Section 6 analyzes and discusses them. Finally, Section 7 concludes and refers to future directions.

2. Background and related work

As this work focuses on the comprehensibility of Tropos and UC models, this section provides some necessary background and related work. First, the main characteristics and differences of GO and SB approaches are presented. Then, we clarify to what extent Tropos and UC are comparable through discussing their metamodels. Finally, this section discusses the issue of model comprehensibility and its centrality in deciding which modeling approach may be used.

2.1. Goal-oriented vs. scenario-based approaches

The specific characteristics of the two approaches can be briefly summarized as follows. GO methods focus on why systems are constructed. They express the rationale of and justification for the proposed systems and aim at: (1) Understanding the current organizational situation; (2) understanding the need for a change; (3) providing the deliberation context within which the requirements engineering process occurs; (4) relating business goals to functional and non-functional system requirements; and (5) validating system specifications vis-à-vis the stakeholders’ goals. Two main frameworks in this category are KAOS [17] and i* [64]. One of the most widespread i*-based methods is Tropos, which intends to “support all analysis and design activities in the software development process, from application domain analysis down to the system implementation” [10].

SB methods, on the other hand, focus on how systems are used and promote the notion of a scenario, which is a sequence of interaction events between a system and its environment in the restricted context of realizing some implicit purposes [58]. SB approaches have been proposed for requirements elicitation and validation purposes. Jacobson’s UC technique, which has been incorporated into UML, is a widely used SB requirements specification language [9]. Among UML’s diagrams, UC was found to be a pivotal diagram used most extensively in attempts to comprehend system functionalities [28].

The effectiveness of GO and SB approaches is analyzed in several works that illustrate the application of different methods to case studies (e.g., [10,14,35]), or compare the strengths and limitations of the approaches according to different criteria (e.g., [15,50]). However, to the best of our knowledge, no experimental comparisons of these requirements modeling approaches using different visual languages exist. A possible reason for this may be rooted in the difficulty of defining the appropriate measures of the communication qualities that yield unbiased results [6]. It should be noted, though, that GO and SB are complementary in nature. GO modeling is proposed to describe the actors, their goals, and the dependencies among actors, while UCs are exploited to describe the usage scenarios [29,38]. SB analyses are used to guide the development of goal modeling [51]. Recently, the User Requirements Notation (URN) [3], which is a combination of a GO Requirements Language (GRL) based on the i* framework and UC Maps (UCM), received final approval as an international standard from the ITU (the United Nations specialized agency for information and communication technologies). This notation aims to prevent possible consistency and mapping problems related to the use of the two different approaches. However, methods that

1 In this research we only refer to the language aspects of Tropos and not to the methodological ones that include information on how to develop Tropos models.
combine GO and SB principles are not widely used. Thus, in this work we chose to compare a pure SB notation and a pure GO notation.

2.2. Tropos vs. Use Cases

Mapping the concepts of Tropos and UC directly is difficult, since they belong to different modeling approaches. In order to determine whether these languages are comparable, we follow Gemino and Wand’s suggestion [23] that conceptual modeling techniques should be compared in terms of their underlying grammars. Thus, we first compare the modeling languages according to their metamodels in order to identify their overlapping expressiveness. We consider UC models as composed of UC diagrams and UC narratives: UC diagrams [9] allow visual representation of the interactions between external actors (i.e., users and other systems) and the system, while UC narratives complement the diagram with a textual description, commonly represented by structured templates. Prior research has demonstrated that the use of a combination of text and diagrams can generate synergy in capturing domain requirements [12,13,22]. Moreover, Anda et al. [4] empirically evaluated three different sets of guidelines for constructing and documenting UC models and concluded “guidelines based on templates support the construction of use case models that are easier to understand for the readers, than guidelines without specific details on how to document each use case.” Thus, we use in our work template-based UC modeling.

Fig. 1 depicts the metamodel of the UC modeling language used in our experiment. According to this metamodel, UCs are connected to actors via participation association, and to other UCs via inclusion dependencies. Extension dependencies between UCs are not explicitly visualized, but handled in the UC template as alternative actions (or scenarios). The template, whose properties were selected based mainly on the comparison presented in [4] regarding the popular content of UC templates, also refers to the UC type (primary, i.e., directly invokable by an actor, or secondary, i.e., invokable by another UC), the pre- and post-conditions of the UC, and its main and alternative actions.

As for Tropos [10], we considered late requirements models that elaborate on the intended system, its own goals, and the plans it is intended to execute for achieving these goals. This information is modeled using actor and goal diagrams. An actor diagram is a graph whose nodes represent actors (agents, positions, or roles), while its edges represent dependencies among them. A goal diagram focuses on the goals and the plans of the intended system and on the relationships among them (e.g., AND/OR decomposition). Fig. 2 is the metamodel of Tropos, as derived from [54]. The main concepts in Tropos are Actor, Goal, Plan, Resource, and Dependency: actors represent human stakeholders (individuals and organizations) and software systems; goals represent states of affairs an actor wishes to achieve; plans represent means to operationalize goals; resources represent artifacts that can be used by actors; and dependencies between actors represent the fact that actors depend on others to fulfill some goals (for using resources or for having plans executed). The goals and plans of the actors can be refined into sub-goals/plans via AND/OR decomposition relationships.

As noted, a direct mapping between the metamodels of UC and Tropos is difficult to achieve. However, Mylopoulos et al. [43] have already shown how the concepts of Tropos can be accommodated within UML. In particular, introducing different stereotypes, Tropos actors can be modeled as UML actors, Tropos plans (termed tasks in [43]) as UML use cases, Tropos dependencies – as associations, and Tropos decomposition relations – as UML aggregation relations. Tropos goals are further modeled in [43] as UML classes, and Tropos resources are not explicitly handled but only mentioned in the context of resource dependencies. As UC models are the main means for achieving client involvement [20], which is very important for requirements analysis, and in order to keep the complexity of models comparable to that of Tropos models, we did not use class diagrams in conjunction with UC models. Information described in Tropos plans is thus represented as “main actions” and “alternative actions” in the UC Template, and Tropos resources are represented in the “preconditions” and “postconditions” sections of the UC Template.

2.3. Empirical studies on model comprehensibility

It is commonly agreed that the evaluation of modeling languages, techniques, and methods is challenging, mainly due to the lack of standard evaluation frameworks, as noted by numerous researchers, e.g., [40] and [56]. Siau and Rossi [56] surveyed and analyzed systematically a variety of techniques for evaluating analysis and design methods. They analyzed three groups of studies. The first group, the most simplistic and the first to be applied, is based on “feature comparison,” i.e., comparisons according to a certain yardstick. The drawback of these techniques is their subjectivity in both the methods with ontological constructs; and (6) cognitive evaluation, which focuses on the cognitive aspects of modeling. The third group comprises empirical evaluation techniques that include surveys to gather data on the attitudes, opinions, and beliefs of subjects via questionnaires. They are characterized by a high degree of representativeness but a low degree of control. Furthermore, laboratory experiments that allow a high degree of control are criticized for being simplistic and artificial, whereas field experiments in a natural setting are not controllable.

In the context of empirical studies that aimed at assessing communication qualities, such as comprehensibility, two recent surveys [12,26] demonstrate the centrality of this aspect in software engineering research. In [26] a classification of research works on UML quality is presented. This classification refers to three major qualities: UC diagrams [9] allow visual representation of the interactions between external actors (i.e., users and other systems)
categories according to their focus (with some overlap between categories), namely: (i) Syntactic quality: correctness; (ii) semantic quality: consistency, completeness, correctness; and (iii) pragmatic quality: maintainability, analyzability, understandability, testability, functionality, executability, reusability, complexity, dependability. About 50% of the reviewed research works focused on semantic aspects and about 40% focused on pragmatic aspects.

In the rest of the section, we review research studies whose goal was to compare empirically the comprehensibility of two or more modeling languages, techniques, or methods. We chose to concentrate in this review on studies that compare methods from different modeling approaches or involve the specific languages we used in our experiment, namely UC or Tropos. We used the outcome of this review to examine the reported influencing factors and consider their adoption in our experiment. Table 1 lists the reviewed studies, including information on the compared methods, the type of participants, the sample size, the measures, and the conclusions.

The first group of studies focuses on the comparison of relational or structural approaches to object-oriented ones [18,19,21]. The second group of studies emphasizes the differences between object-oriented approaches and process-oriented or hybrid (object-process) methods [12,16,45,48]. It is important to note that all the subjects in the reviewed studies of the two first groups are students having different levels of skill (i.e., either undergraduate or graduate students). Some of the studies replicated the same experiment in order to obtain more results and increase the sample size. Two main measures can be found in these reviewed studies, namely, comprehension level and time taken to complete tasks. The comprehension level is measured in the reviewed studies differently, e.g., as F-measure, number of errors, or (subjective) scores. A third measure that is mentioned in these studies is the perceived model comprehensibility as reported by the subjects themselves. Based on this review, in our study we examined three types of measures: comprehension level (the correctness and completeness of the answers), the time to complete the tasks, and the subject perception of different aspects. Our subjects were students, since this allowed us to replicate the experiment several times, even though this practice is often criticized due to students’ lack of industrial experience. However, it is worth noting in this context that Svahnberg et al. [59] investigated the ability of students to understand and assess multiple perspective involvement in requirements selection processes and found that their perceptions are comparable with those of experienced industrial professionals.

The third group of reviewed studies concentrates on UC modeling, including diagrams, guidelines, and narratives [46,42,45]. These studies refer to the different communication qualities of UCs and their capacity to promote a good understanding of the requirements among the stakeholders. The results in [4] indicate that guidelines based on templates support the construction of UC models that are easier for readers to understand than are guidelines without specific details on how to document each UC. As noted, based on this part of the review, we chose to use for our comparison a version of UC modeling that includes both diagrams and narratives written using templates.

The fourth and last group in Table 1 includes a single study that empirically evaluated the comprehensibility of Tropos, or more accurately, a variant of Tropos called Tropos4AS [41]. Tropos4AS extends the Tropos method “to capture and detail at design time the specific decision criteria needed for a system to guide self-adaptation at run-time, and to preserve the concepts of agent and goal model explicitly along the whole development process until runtime” [41]. The study reports preliminary results according to which Tropos4AS is more expressive and comprehensible than Tropos. Moreover, the modeling effort (except for looking up in the language specifications) seems not to be significantly higher than for Tropos. Nevertheless, the study was conducted using a very small number of subjects (12) and it compared two variants of Tropos. To summarize, comparing the comprehensibility of methods, techniques, and languages that belong to different modeling approaches is feasible. While there are empirical studies regarding UC’s comprehensibility, we found only one empirical study regarding Tropos’ comprehensibility and to the best of our knowledge there is no study that compares the comprehensibility of the two modeling languages. Given the fundamental difference between the GO and SB approaches and that usually only one of them is adopted in spite of their complementary nature, it is interesting to compare Tropos with UC, as both are intended to document and communicate requirements.

3. The family of experiments

It is well known that in order to achieve greater validity of experimental results, replications are necessary. The concept of replication is smoothly extended to that of “family of experiments” [8]. A family of experiments is composed of multiple similar experiments that pursue the same goal in order to build the knowledge that is needed to extract significant conclusions that can be applied in practice. An exact replication is one in which the procedures of an experiment are followed as closely as possible to determine whether the same results can be obtained [57]. Replications that reuse the original procedures, e.g., the study design and the experimental steps, but modify the subject pool in order to gain more insight into the original results (as in our case), fall into the category of exact replications.

This section presents the family of experiments that we have performed. The main elements are summarized in Table 2. These elements will be elaborately explained in Section 4.

Fig. 3 shows the features of the family of experiments, including the context of each experiment, the number of involved subjects, and the place where the experiments took place. This figure also
highlights the execution order of the experiments (e.g., the original experiment took place during the spring semester of 2009) and the labels of the experiments.

As presented in Fig. 3, the experiments were conducted at the University of Haifa, Israel, with different subjects and in different courses. Sampling was conducted by convenience; full courses in which eligible students participated were selected; the subjects in all executions of the experiments were Information Systems (IS) undergraduate students in the beginning of the 3rd (last) year (Exp1 and Exp2) or towards the end of the 2nd year (Exp3) of their studies. The subjects had already studied all the basic IS courses in previous semesters including “Introduction to IT,” “IS Design and Implementation,” and “IS Analysis” (the subjects in Exp3 were towards the end of this course). The participation in the experiments was voluntary; it was highly recommended by the lecturers as useful training in course-related material, but was not mandatory. The subjects were awarded in compliance with the respective syllabus of each course and could receive up to three additional points to their final grades in the course, depending on the correctness of their answers. The three researchers from the University of Haifa were known to all subjects and they all served as their lecturers either prior to or during the semesters of the experiments (including the courses in which the experiments were conducted).

In addition to analyzing each experiment separately, we aggregated the results of the individual experiments. This approach has been used previously in other studies in the literature, e.g., in [49] and [62], and is detailed in [63]. Furthermore, we performed meta-analysis on the aggregated results [27] to compute the size of the possible effect of the independent variable. This size, Cohen’s “d”, is defined as the standardized mean difference between the treatments in terms of a dependent variable [15]. Typically, it is considered negligible for d < 0.2, small for 0.2 < d < 0.5, medium for 0.5 < d < 0.8, and large for d ≥ 0.8. Although there are several ways to combine effect sizes from a series of experiments, the focus here is on the weighted linear combination of estimators from different studies [30]. To perform the meta-analysis, we followed [30] and used the meta-analysis tool named META 5.3. The result of the tool is the mean effect size (\(d^*\)).

---

Table 1
Summary of studies that focus on comprehensibility of methods applying different modeling approaches.

<table>
<thead>
<tr>
<th>Study</th>
<th>Comparison</th>
<th>Type of participants</th>
<th>Sample size</th>
<th>Measure</th>
<th>Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>[18]</td>
<td>ER and class diagrams</td>
<td>Students</td>
<td>30/40</td>
<td>F-measure on multiple choice questions</td>
<td>UML class diagrams significantly improved comprehension</td>
</tr>
<tr>
<td>[19]</td>
<td>ER and class diagrams</td>
<td>Students</td>
<td>40/30/ 68</td>
<td>F-measure on multiple choice questions</td>
<td>UML class diagrams improved comprehension. The two notations give the same support for maintenance activities. UML class diagrams provide better support during verification activities</td>
</tr>
<tr>
<td>[21]</td>
<td>Rational Unified Process and Structured Analysis &amp; Design Technique</td>
<td>Students</td>
<td>50</td>
<td>Time required to perform the tasks</td>
<td>No significant difference in the time required for developing or maintaining a software application. Object orientation is more sensitive to subjects’ peculiarities, and it is already able to provide some reusability advantages at the analysis level</td>
</tr>
<tr>
<td>[2]</td>
<td>Object-oriented and process-oriented paradigms</td>
<td>Students</td>
<td>43</td>
<td>Score of correctness and fullness and Jaccard's similarity coefficient</td>
<td>Superior performance was observed when the process-oriented tool was applied to the process-oriented task. For the object-oriented task, there was no difference in subject performance across the two tools</td>
</tr>
<tr>
<td>[1]</td>
<td>Object class diagrams and DFD</td>
<td>Students</td>
<td>36/35</td>
<td>Score of correctness and fullness</td>
<td>For most of the simple questions, no significant difference was observed as model comprehensibility is concerned; for most of the complex questions, the process-oriented (DFD) model was found to be easier to understand than the OO model</td>
</tr>
<tr>
<td>[45]</td>
<td>Object-Process Methodology (OPM) and Object Modeling Technique (OMT)</td>
<td>Students</td>
<td>88</td>
<td>Number of errors</td>
<td>A single model methodology – OPM – is more effective than a multimodel one – OMT – in terms of synthesis. The results further showed that there were significant differences between the two methods in specific issues that refer to the comprehensibility of the specifications. OPM is better than UML in modeling the dynamics aspect of the Web applications; in specifying structure and distribution aspects, there were no significant differences. The quality of the developed OPM models may be superior to those built in UML</td>
</tr>
<tr>
<td>[48]</td>
<td>Object-Process Methodology (OPM) and UML</td>
<td>Students</td>
<td>81</td>
<td>Score of correctness and fullness</td>
<td>There was no significant difference in comprehension. It took less time to comprehend the UC model. Subjects perceived that the OO-DFD model is easier to comprehend</td>
</tr>
<tr>
<td>[16]</td>
<td>OO-DFD and UC models</td>
<td>Students</td>
<td>53</td>
<td>Number of correct classifications (true/false/cannot tell); time required to perform the tasks; perceived model comprehensibility</td>
<td>A lean set of guidelines performs at least as well as a more detailed set in terms of the ability to produce clear and accurate (comprehensible) descriptions</td>
</tr>
<tr>
<td>[46]</td>
<td>UC writing Guidelines: CREWS vs. CP</td>
<td>Students</td>
<td>60</td>
<td>Number of rules used; communicability</td>
<td>Guidelines based on templates support the construction of UC models that are easier to understand than guidelines without specific details on how to document UCs</td>
</tr>
<tr>
<td>[4]</td>
<td>UC construction guidelines</td>
<td>Students</td>
<td>139 (31 teams)</td>
<td>Number of correct responses</td>
<td>The addition of a graphic representation to the text based UC can significantly enhance understanding of a system among novice modelers</td>
</tr>
<tr>
<td>[25]</td>
<td>UCs with and without diagrams</td>
<td>Students</td>
<td>49</td>
<td>Number of correct responses</td>
<td>The process-oriented tool was applied to the process-oriented task. Superior performance was observed when the process-oriented tool was applied to the process-oriented task. For the object-oriented task, there was no difference in subject performance across the two tools</td>
</tr>
<tr>
<td>[41]</td>
<td>Tropos vs. Tropos4AS</td>
<td>Researchers and PhD students</td>
<td>12</td>
<td>Precision and recall; perceived model comprehensibility</td>
<td>Tropos4AS is statistically significantly more expressive and comprehensible than Tropos. The modeling effort (except for looking up in the language specifications) seems not to be significantly higher than for Tropos</td>
</tr>
</tbody>
</table>

---

2 See http://userpage.fu-berlin.de/health/meta_e.htm
In the next section, the three experiments are elaborated whereas the results of the experiments and the meta-analysis are presented in Section 5.

4. Individual experiments

The three experiments were carried out by following the recommendations provided in [63]. Section 4.1 elaborates on the planning of Exp1 according to the guidelines suggested in [32]. The other replications (Exp2 and Exp3) are presented in terms of differences with respect to Exp1 in Section 4.2. Finally, Section 4.3 refers to the execution of the three experiments. For replication purposes, we made available on the web the experimental material that was produced and used.

4.1. Planning of the original experiment (Exp1)

4.1.1. Goal

The goal of the experiment was to analyze the usage of two different requirements modeling languages, namely the scenario-based UC and the goal-oriented Tropos, with the purpose of comparing the comprehensibility of the respective models. The quality focus is on ensuring that requirements models have a high level of comprehensibility for requirement analysts, while the researchers’ perspective is that of comparing the effectiveness of the modeling languages during different tasks involving comprehension of the requirements.

4.1.2. Subjects

Exp 1 was executed at the University of Haifa, in the spring semester of 2009 in a course dealing with human aspects of software engineering. The subjects were 19 Information Systems (IS) undergraduate students in their 3rd (last) year.

Instead of using the information gathered in the pre-experiment questionnaires (see Section 4.1.7) to distribute high and low ability/experience subjects equally among the groups (randomized block design), we opted to divide the subjects into groups randomly (completely randomized design). We decided on random assignment because we could not consider the pre-questionnaires during the day of the experiment for dividing the subjects to groups due to time constraints. At the same time, we could not divide the subjects prior to the experiment because we did not know in advance who was going to participate (we remind that the participants in the experiment was voluntary). Nevertheless, the analysis of subjects’ background and experience as collected by the pre-experiment questionnaire provided evidence that the subject sample was satisfactorily homogenous and thus each possible partitioning is appropriate. In particular, we found that all subjects reported to have either medium (15 subjects) or high (4 subjects) average exams’ score (none reported low average); only 1 subject out of 19 affirmed to have high experience in the field of requirements engineering (11 reported medium and 7 – low); and finally, while 7 subjects out of the 19 stated that they had had an industrial experience, follow-up conversations with these subjects in which we asked specifically about their experience revealed that this experience was very limited.

4.1.3. Experimental material

The objects of the experiment are the requirements models of two systems, taken from different application domains. The first is a meeting scheduling system (called “mss” in short) that assists the user to plan and organize meetings. The second is a system that automates several tasks in a restaurant (e-Restaurant system, “eRest” for short), such as the customer’s online ordering of dishes and asking for the bill; the customer’s checking the nutritional details of dishes; the chef’s checking the status of ingredients ordered from retailers, etc. The full system story of eRest is provided in Appendix A.2, whereas the full system story of mss can be found with the experimental material on the web.

The requirements of both systems were modeled using Tropos (actor and goal diagrams) and UC (diagram and textual descriptions of the UCs following the template presented in Section 2.2). Both Tropos and UC models that were used in the experiment for the eRest case are presented in Appendix A.2, along with the legends of these modeling languages.

Due to the differences in the modeling approaches to which the two languages belong, the UC and Tropos models of the same system (mss or eRest) were different. For example, in Tropos, means-ends analysis and decomposition were applied for allocating the dependencies to the relevant entities and refining them to the level of detail represented by the story, while the UCs represented the interaction scenarios between the same entities for obtaining exactly the same services. Nevertheless, the two models represent the same system story.

Table 3 lists the number of instances of modeling concepts contained in the (full) models of the two systems, according to the metamodels of the languages. In terms of number of elements, the two models can be considered of same order of complexity.

---

3 See http://islab.fsk.eu/TroposvsUC_exp/.

4 In order to make sure the subjects refer to the template-based textual descriptions as part of the UC model, we emphasized this point on several occasions, including in the questionnaire itself.
4.1.4. Tasks
For each system, eRest or mss, the subjects were asked to fill a questionnaire regarding the content presented in the models, specified in either UC or Tropos. Appendix A.2 presents the comprehension questions included in the eRest questionnaire.

All the questions in the questionnaire were problem-solving tasks, namely tasks that assess the subjects’ ability to use knowledge represented in the schema, where they are requested to determine whether and how certain information is available from the schema [33, 52].

Following analogous studies [53], the questionnaire was composed of 14 open questions that can be divided into three categories: (1) text-model mapping (Q1–Q4) – questions in this category refer to the subjects’ ability to map between the system story and the corresponding visual requirements models and vice versa. This is important for evaluating the completeness and correctness of requirements models and vice versa. These questions were the same for the two models that specified the same system (mss or eRest); however, some specific tailoring was required depending on the modeling language. For example, when asked about relationships, the subjects had to refer separately to goal, plan, and resource dependencies in Tropos, whereas in UC they had to specify separately participation relationships and include dependencies. To ensure that we made the comparison on an equal basis, five experts checked the models with respect to the system stories, and ascertained that all the questions could be answered using each model individually.

4.1.5. Hypotheses, parameters, and variables
In accordance with our goal that was introduced earlier and similarly to studies described in Section 2.3, we focused on the following research questions and the corresponding set of null and alternative hypotheses.

RQ1: Which requirements model, the one that was expressed in Tropos (in terms of goal and actor diagrams) or the one that was expressed in UC (in terms of diagrams and templates), is more comprehensible to analysts?

RQ2: Which requirements model, the one that was expressed in Tropos (in terms of goal and actor diagrams) or the one that was expressed in UC (in terms of diagrams and templates), requires less effort (or more precisely less time) from the analysts to achieve comprehension?

RQ3: Which requirements model, the one that was expressed in Tropos (in terms of goal and actor diagrams) or the one that was expressed in UC (in terms of diagrams and templates), results in higher productivity of the analyst (derived from the ratio between comprehension level and effort)?

These research questions bear the following null hypotheses respectively:

H01. There is no difference between the comprehension levels of Tropos and UC requirements models;

H02. There is no difference in terms of effort required for comprehending UC and Tropos requirements models.

H03. There is no difference in terms of productivity in comprehending UC and Tropos requirements models.

When the null hypotheses can be rejected with relatively high confidence, it is possible to formulate alternative hypotheses as follows:

Ha1. There is a difference between the comprehension levels of Tropos and UC requirements models;

Ha2. There is a difference in terms of effort required for comprehending UC and Tropos requirements models.

Ha3. There is a difference in terms of productivity in comprehending UC and Tropos requirements models.
The independent variable of this experiment is the modeling language, which can assume one of the two values in (Tropos, UC). The dependent variables are the comprehension level of models, the effort invested by the subjects for comprehending the models, and the derived dependent variable – productivity.

To measure the comprehension level, we counted: (1) The set of elements mentioned in the answer to a question “i” by a subject “s” and (2) the number of elements in the answer that were expected as the correct answer to question “i”. Based on the above, and similarly to [19,49], we computed precision and recall for each answer [39]. Precision measures the fraction of items in the answer that are correct, while recall measures the fraction of items that are in the answer out of the number of the expected items. Since these metrics measure two different concepts, we added a derived measure, F-measure that combines the two, which is a standard metric defined as the harmonic mean of precision and recall: \( F = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \). We used the mean of the F-measure over all the questions in the questionnaire for each subject. The higher the F-measure is, the more comprehensible the model is.

The effort invested in answering the questionnaire is measured as the time in minutes it took the subjects to answer all questions, directly recorded by the subjects (as start and stop time of answering the whole questionnaire). The dependent variable productivity was calculated as the subjects’ comprehension level/effort \( \times 100 \), which may support trade-off analysis for modeling language selection. One modeling language is preferable to the other if it allows requirements analysts to do their tasks correctly in the least possible time [47].

As comprehensibility is our main focus, we further analyzed the difference in the comprehensibility of Tropos and UC requirements models taking into consideration the different questions categories discussed above. Thus, we split the null hypothesis \( H_{01} \) into three null sub-hypotheses:

- \( H_{01\text{\,map}} \): There is no difference between the comprehension levels of Tropos and UC requirements models with respect to text-model mapping;
- \( H_{01\text{\,read}} \): There is no difference between the comprehension levels of Tropos and UC requirements models with respect to model reading;
- \( H_{01\text{\,mod}} \): There is no difference between the comprehension levels of Tropos and UC requirements models with respect to model modifying.

The alternative sub-hypotheses corresponding to these latter three null ones can be easily deduced along the line of those presented in this section. In the data analysis, we tested \( H_{01} \) globally, as well as \( H_{01\text{\,map}}, H_{01\text{\,read}} \) and \( H_{01\text{\,mod}} \) using F-measure, as described above. In the subset of the questions classified as model modifying (Q11–Q14), precision and recall could not be calculated according to that formula, as more than a single possible solution existed. In these questions, each answer was evaluated as fully correct, partially correct, or incorrect/missing, according to a pre-defined scale for each question which had been agreed upon by all six researchers prior to the executions of the experiment. These evaluations were then assigned corresponding numeric values, to enable statistical analysis.

4.1.6. Experiment design

Table 4 summarizes the counter-balanced experimental design [63] that we adopted. The experiment consisted of two subsequent work sessions, called Lab1 and Lab2, of about 1.5-h each. Each Lab referred to a different system, eRest or mss, and different modeling language, Tropos or UC. The subjects were divided into four groups, each involved in both sessions.

To limit learning effects, the experimental design ensured that each subject worked on different systems in the two Labs, receiving a different treatment (modeling language) each time. In addition, the design permits us to consider different combinations of systems and modeling languages in a different order across Labs.

4.1.7. Procedure

The experiment took place in a single room. All the subjects attended a short introductory lesson in which detailed instructions for the tasks to be performed were presented. The experiment was presented as a laboratory activity and only the high-level goal of the experiment was highlighted, while details of the experimental hypotheses were not provided. The subjects were informed about the procedure to follow in the execution of the experiment. Finally, we explained the fact that the subjects have to play the role of requirements analysts.

Before the experiment, all the subjects were requested to fill the pre-experiment questionnaire, which was composed of 14 questions. The pre-experiment questionnaire, presented in Appendix A.1, had two parts. The purpose of the first part (PR1–PR6) was to obtain general information about the subjects (e.g., learning and working experience) and to assess their individual ability level and experience (e.g., in understanding the requirements). In particular, this part provided data on the homogeneity of the subjects. The purpose of the second part (PR7–PR14) was twofold: (i) To capture the previous knowledge and experience of the subjects regarding UC and Tropos, and (ii) to find out whether the subjects felt they understood the languages that would be used in the experiment (see, for example, question PR12 which dealt with the clarity of the chosen UC template).

Once the subjects concluded the introductory lesson and completed the pre-experiment questionnaires, they started the

<table>
<thead>
<tr>
<th>Table 3. Requirements models under study.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>Tropos</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>UC</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
are the two treatments. Finally, the requirements models of mss and eRest are the experimental objects.

<table>
<thead>
<tr>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
<th>Group 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab1</td>
<td>UC; mss</td>
<td>Tropos; eRest</td>
<td>UC; eRest</td>
</tr>
<tr>
<td>Lab2</td>
<td>Tropos; eRest</td>
<td>UC; mss</td>
<td>Tropos; mss</td>
</tr>
</tbody>
</table>

For each Lab, we provided the subjects with the following experimental material:

- A story describing the intended use of the software system;
- The corresponding Tropos or UC models;
- A questionnaire on the system and its model;
- The slides of the lessons (which they could consult when answering the questionnaire);
- A final post-lab questionnaire.

As already mentioned, the system story, models, and comprehension questions for the eRest case can be found in Appendix A.2. The post-lab questionnaire can be found in Appendix A.3.

After receiving the material, the subjects went through the following procedure for each Lab:

1. Specify name and start-time on the main questionnaire (that with the comprehension questions).
2. Answer individually the comprehension questions by surfing the delivered material.
3. Mark the end-time on the main questionnaire.

At the end of each Lab, the subjects were requested to fill in a final post-lab questionnaire composed of 9 questions for UC and 12 for Tropos, with the extra questions aimed at comparing the newly-learned language, Tropos, to the previously known language, UC (see Appendix A.3). This post-lab questionnaire was intended to attain qualitative insights. The first common group of questions (P1–P4) concerned the allocation of sufficient time to complete the tasks, as well as the clarity of the objectives, the system stories, and the tasks. Question P5 was devoted to the subject’s comprehension of the modeling language (Tropos or UC). Then, the following three questions (P6–P8) examined the subject’s difficulty in understanding the questions in the three question categories. Finally, for Tropos, four extra questions (P9–P12) were added with the aim of measuring possible differences in the subjects’ skill in using the two languages and in model expressiveness as perceived by the subjects. All the questions included closed answers on a five point Likert scale: 1 – strongly agree, 2 – agree, 3 – not certain, 4 – disagree, and 5 – strongly disagree.

4.1.8 Analysis procedure

In this study, we used boxplots to summarize succinctly the outcomes of the experiment (i.e., the dependent variables). A boxplot [63] is a compact representation of the distribution of a variable: the thick horizontal line is the median, the box stands for the 2nd and 3rd quartiles, and the whiskers represent the 1st and 4th quartiles. Outliers (if any) are represented by circles beyond the whiskers.

To be conservative, because of the sample sizes (only 19 subjects in this particular experiment) and the shapes of distribution (in several cases the Shapiro–Wilk normality test gave small p-values), we adopted non-parametric tests to reject the null hypotheses. In particular, we selected the Mann–Whitney test [63] because of its robustness and sensitivity. Moreover, we used two-tailed statistical tests due to the non-directionality of the hypotheses (there is no difference in terms of the comprehension levels and effort required between Tropos and UC requirements models). In all our statistical tests, we decided to accept a probability of 5% of committing a Type-I-error [63], i.e., rejecting the null hypothesis when it is actually true. The analyses of the hypotheses are presented in Sections 5.1.1–5.1.3. Finally, we measured the effect of other factors on the dependent variables, namely the Lab (i.e., whether a result was obtained in the first or second working session, to evaluate a possible learning effect), the Application Domain (mss or eRest), Order of treatments (first Tropos or first UC), Order of application domains (first eRest or first mss) and possible interactions between them using a two-way Analysis of Variance (ANOVA) [63]. The analyses of the effect of these other factors on the dependent variables are presented in Appendix B. Concerning the post-lab questionnaires, we computed only medians for the sake of simplicity, as presented in Section 5.1.4.

All the individual experiments’ data analyses were done using STATISTICA 8.0.5

4.2 Replications (Exp2 and Exp3)

Exp2 and Exp3 were exact replications of Exp1: the study design, the experimental steps, and the objects are exactly the same. They differ only in the number of subjects (19, 31, and 29 in Exp1, Exp2, and Exp3, respectively) and the courses in which the experiment took place (3rd year course on human aspects of software engineering, 3rd year software engineering course, and 2nd year systems analysis course in Exp1, Exp2, and Exp3, respectively).

Analyzing the pre-questionnaires, no big differences in subjects’ average exams’ scores, industrial experience and experience in the field of requirements engineering were identified among the experiments. Thus, we can conclude that all the subjects over the three experiments have roughly the same background, skill, and experience.

Experience from the first execution indicated the need to consider the possibility that the subject’s confidence in answering questions may vary depending on the question and the languages under consideration, affecting the quality of the measures collected within the experiment. A similar warning is discussed also in [6]. We therefore inserted in Exp2 and Exp3, for each comprehension question in the questionnaire, two additional questions to measure the subjects’ confidence (i.e., their individual certainty in their answers), and how difficult they perceived each single question. These values were used to aid our understanding of the results obtained for the comprehension level variable. All the questions about confidence and perceived difficulty were evaluated on a five point Likert scale (1 – very low, 2 – low, 3 – medium, 4 – high, 5 – very high). Similarly to the post-lab questionnaires analysis, we computed only medians of the confidence and perceived difficulty variables, for the sake of simplicity.

4.3 Execution (preparation and deviations)

About two weeks before each experiment, the subjects attended a training session on UC and Tropos. All the subjects had previous knowledge of UC modeling, which they had gained and applied in previous courses at the university. The training included a short refresher session in UC modeling, using an example of a home banking application, and a longer introduction to Tropos session with a parallel example. The examples were constructed interactively with the subjects and the entire training session took 4 h (0.5 h – a UC refresher, 1.5 h – a tutorial on Tropos, and 2 h – an exercise lesson on Tropos).

As explained earlier, we divided the subjects in the experiment itself into four groups randomly (see Table 4) and the subjects worked on the two Labs according to their allocation to groups. No deviations from the plan (a.k.a. protocol) occurred; no subjects dropped out from the study (it was not allowed) and the 3 h to complete the experiment were respected and sufficient (this was also recognized by the subjects in the post-lab questionnaire, as will be presented in the next section).

5. Results

This section reports the data analysis of the family of experiments. For the reasons explained in Section 3 (exact replications, subjects with, roughly, the same background and experience), we further present the results from analyzing the aggregated data and from the meta-analysis. Additional analyses concerning the other factors (see Table 2) are presented and discussed in Appendix B. Finally, we discuss in this section the threats to validity that could affect the results we obtained.

5.1. Data analysis

5.1.1. Comprehension level

Table 5 reports a summary of the subjects’ comprehension level, measured by the F-measure, in the three executions of the experiment according to Treatment (Tropos or UC): first the general comprehension level in each experiment (or overall) is presented (first row in each experiment/overall), followed by the level measured separately in each question category. In addition, p-values of unpaired Mann–Whitney tests are presented. Statistically significant values are shown in boldface.

In Fig. 4 and in the “Overall” row in Table 5, we can see that the mean and median values of F-measure are higher for Tropos than for UC (respectively, 0.63 and 0.63 for Tropos and 0.55 and 0.55 for UC). The practical meaning is that the subjects gave better answers to the questionnaires when using Tropos models than when using UC models. This difference of 8 points (i.e., 14.5%), irrespectively of whether the mean or the median is considered, is significant, as the Mann–Whitney test results in a p-value < 0.01. Thus, globally, the first null hypothesis $H_0$ can be rejected, meaning that a difference exists between the two languages in terms of comprehensibility, where Tropos is more comprehensible than UC.

However, considering the three experiments separately, we can see in Table 5 that for Exp 1 and Exp 3 the difference, in favor of Tropos, is significant (respectively, p-value = 0.037 and p-value = 0.008), while for Exp 2 it is not significant (p-value = 0.46), even if the direction is the same as for Exp 1 and Exp 3 (i.e., Tropos outperforms UC also in this case).

Concerning the effect size, we obtained Cohen’s $d = 0.58$ for Exp 1 (medium), $d = 0.35$ for Exp 2 (small) and $d = 0.58$ for Exp 3 (medium). Applying meta-analysis, the mean effect size $d^*$ is 0.48, which can be considered small (almost medium) in practical terms.

A closer look at the results, and separating the questions according to the three different categories, yields the following interesting results (see Fig. 5 and the rows labeled “Mapping”, “Reading”, and “Modifying” under “Overall” in Table 5). For text-model mapping and model reading, Tropos outperforms UC and the differences are significant (p-values < 0.01) while for model modifying, Tropos is still slightly better but the difference is not significant (p-value = 0.31). Thus, globally, we can reject $H_0$ but not $H_0$ but not $H_0$.

Note that both text-model mapping and model reading questions require only “reading” capabilities [24], i.e., consulting the requirements model without modifying it, while the model modifying questions, in which no significance was perceived, requires both “reading” and “writing” capabilities. Considering the three experiment executions separately, we obtained only the following significant differences: text-model mapping in Exp 1 (p-value = 0.002), model reading in Exp 2 (p-value = 0.048) and both text-model mapping (p-value < 0.01) and model reading (p-value < 0.01) in Exp 3.

5.1.2. Effort

Table 6 provides a summary of the effort, measured as time in minutes needed to answer the whole questionnaire of the executions of the family of experiments according to treatment (Tropos or UC). p-Values of unpaired Mann–Whitney tests are presented in the last column of Table 6.

The diagram in Fig. 6 shows boxplots of the time needed to answer the entire questionnaire (without separation into question categories), as presented in the “overall” row in Table 6.

From this Figure, it is clear that the time it took the subjects to answer the questionnaire was slightly longer with Tropos than with UC. Considering the medians, the difference in time between the two languages is 5 min (6.2 min considering the mean, i.e., 13.5%). This difference was found to be significant according to an unpaired Mann–Whitney test (p-value = 0.02). Thus, overall,
the second null hypothesis \( H_0^2 \) can also be rejected. A possible explanation for this outcome may be the subjects’ prior knowledge of the UC modeling language. This knowledge could be exploited to perform the tasks more quickly, but not necessarily more correctly (as the F-measures analyzed in the previous sub-section indicate). It can be seen from the results of the individual experiments that the situation is similar to that presented in Fig. 6 for each experiment. The difference, considering the median, is respectively of 10, 9, and 5 min for Exp1, Exp2, and Exp3, in favor of UC.

Concerning the effect size, we obtained Cohen’s \( d = 0.58 \) for Exp1 (medium), \( d = 0.51 \) for Exp2 (medium) and \( d = 0.31 \) for Exp3 (small). Applying meta-analysis, the mean effect size \( d^+ \) is 0.44, which can be considered small in practical terms.

5.1.3. Productivity

The comprehension level and the measured effort show different directions, as previously discussed. Thus, we considered the derived measure of Productivity (F-measure/Time \(* 100\)). A substantive draw for productivity is presented in Table 7. The medians of productivity are quite similar: 1.33 for Tropos and 1.29 for UC (overall \( p\text{-value} = 0.84 \)). A substantive draw for productivity is also evident when separating by experiment (see Table 7); in all the experiments, we cannot reject \( H_0^3 \).

Concerning the effect size, we obtained Cohen’s \( d = 0.29 \) for Exp1 (small), \( d = 0.02 \) for Exp2 (negligible) and \( d = 0.08 \) for Exp3 (negligible). Applying meta-analysis, the mean effect size \( d^+ \) is 0.10, which can be considered negligible in practical terms.

5.1.4. Post-lab questionnaire analysis

As noted above, the post-lab questionnaire was intended to retrieve qualitative insights about the experiment. Table 8 presents...
the aggregated medians, for all three-experiment executions, of the subjects’ perceptions according to their perceived level of agreement in the post-lab questionnaire. Answers are on a Likert scale from 1 (strongly agree) to 5 (strongly disagree). Values different from 3 (not certain) are highlighted.

Globally, we can observe that for UC, subjects judged the time allotted to complete the task sufficient (P1); they found the objectives of the laboratory (P2), the description of the system story (P3) and the tasks (P4) clear. They experienced no difficulty in reading both UC diagrams and UC descriptions (P5). It is interesting that the perceived difficulty (P6, P7 and P8) reflects the obtained results. In fact, subjects experienced no difficulty in answering text-model mapping and model reading questions (P6 and P7) where they performed better, while they were not certain for model modifying questions (P8) where they performed worse (see actual results in Fig. 5 and subjects’ perceptions in the row UC in Table 8).

For Tropos, as for UC, subjects judged the time allotted to complete the task sufficient (P1); they found the objectives of the laboratory (P2), the description of the system story (P3), and the tasks (P4) clear. However, they were not certain regarding their understanding of the Tropos diagrams (P5). Concerning answers related to the perceived difficulty, the situation for Tropos is less clear-cut than for UC. Subjects experienced no difficulty in answering text-model mapping questions (P6), but they were not certain in model reading (P7) and model modifying questions (P8). In the Tropos case, the perceived difficulty does not reflect the quantitative results obtained in our experiments (see the Tropos row in Table 8 and Fig. 5). This may explain the extra time it took subjects to answer the questions: they were less familiar with Tropos, which resulted in their having less confidence and perceiving the level of difficulty as higher.

Regarding the extra questions (which appeared in the Tropos forms – see Appendix A.3 and Table 8), we obtained 3 (not certain), for questions P10 and P12, and 2 (agree), for question P11. This suggests that most subjects considered the Tropos diagrams more informative than UC diagrams.

5.1.5. Summary of the results

Table 9 summarizes the main aggregated results of comparing UC and Tropos. There are significant differences between Tropos and UC regarding comprehension level and effort: The comprehensibility of Tropos models is significantly higher than the comprehensibility of UC models; at the same time, Tropos requires a significantly higher effort investment for comprehension than UC. As a result, there is no significant difference between the productivity achieved using the two modeling languages.

5.2. Threats to validity

In this section we discuss the main threats to the validity of our results, according to [63].

Construct validity threats concern the relationships between theory and observation. They are mainly due to the method used to assess the outcomes of tasks. Similarly to [11,49], we used questionnaires to assess the comprehension level of the subjects; the answers were evaluated using an information retrieval-based approach in order to avoid subjective evaluation as much as possible. In addition, and as mentioned before, the models used as experimental objects were checked by five experts. We believe that this operation was useful for eliminating any bias potentially introduced into the models by the researchers involved in the development of Tropos (two of the six authors of this paper). Effort was measured by means of proper time sheets and ensured by the researchers who were present during the experiment. Although this may not be a very accurate method, it is widely adopted for measuring the effort required to accomplish tasks in controlled experiments. The post-lab questionnaire was designed based on previously designed questionnaires (e.g., [49]) and following recognized guidelines (e.g., grouping similar questions to make the questionnaire easier to comprehend/complete and using the Likert scale) [37].

Internal validity threats concern external factors that may affect a dependent variable. They may be due to learning and fatigue effects experienced by the subjects between Labs. The learning effect is mitigated by the experiment design that we chose (a counter-balanced experiment design), while the fatigue effect is mitigated by the mandatory break that we imposed between the two Labs. Indeed, the two-way ANOVA analysis by Treatment and Lab, reported in Appendix B, reveals no effect on the comprehension level. However, the effort was found to be affected by the Lab factor. The subjects’ skill in using the two languages was initially different in all the experiments. We included a training session on Tropos, as well as a refresher session on UC, with the purpose of mitigating any possible effect of this difference in the subjects’ background knowledge. We measured a posteriori the difference, as perceived by the subjects, in language usage skills (see question P5 in Appendix A.3 and Table 8). The data provide evidence that, after the training session and before the experiment, the subjects’ knowledge of the two languages was quite similar. Moreover, the possible effects of skill differences on the quality of the answers to the questions whose aim was to measure model comprehensibility were measured, via the questions on perceived difficulty and confidence associated with each comprehension question. The results (see Appendix B) seem to confirm that there is no significant difference in skill between languages.

Another factor that may have had an effect on the dependent variables is the selection of the compared language variants. We used a scenario-based language that employs UC diagrams and a specific template. In the future, it will be interesting to repeat

<p>| Table 8 |
| Medians of Post-lab questionnaire for the questions P1–P12 (scale: 1 – strongly agree, 2 – agree, 3 – not certain, 4 – disagree, and 5 – strongly disagree) |</p>
<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
<th>P8</th>
<th>P9</th>
<th>P10</th>
<th>P11</th>
<th>P12</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Tropos</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

| Table 9 |
| Summary of results. |
| Tropos | UC | p-Value |
| Mean | Med | SD | Mean | Med | SD |  |
| Comprehension level | 0.63 | 0.63 | 0.21 | 0.55 | 0.55 | 0.14 | <0.01 |
| Effort | 52.0 | 47.5 | 18.6 | 45.8 | 42.5 | 15.6 | 0.02 |
| Productivity | 1.21 | 1.33 | 0.49 | 1.20 | 1.29 | 0.46 | 0.84 |
our experiment and compare Tropos and languages that use different kinds of UC templates, or a language that uses both UC and class models for specifying requirements. Analogously, it will be worth investigating possible differences in using other goal-oriented modeling languages, such as that exploited in i* [64].

**Conclusion validity** concerns the relationship between the treatment and the outcome. The statistical analysis is performed using mainly non-parametric tests (i.e., the Mann-Whitney test for unpaired analysis) that do not assume data normality and are also well suited for use on small samples. Moreover, two-way ANOVA was used mainly to detect possible interactions between each co-factor (e.g., application domain and experiment's labs schedule) and the Treatment. Even if all the assumptions/conditions (i.e., that the samples should be normally distributed) for using ANOVA were not valid, this test is quite robust [31]. It has been often used in the literature (e.g., in [11,60]) to study the influence of a factor on one dependent variable, and (possible) interactions between factors.

Another concern relating to conclusion validity in this research is the analysis and discussion of the data aggregated from all three experiments. The motivation for this aggregation is that while each experiment includes a smaller number of subjects, aggregating data to include results obtained from a larger sample contributes to statistical robustness and generalizability of the conclusions. However, aggregating data from different individual experiments may threaten validity due to differences between the settings of the experiments and the groups of subjects. These threats were mitigated in this research by the design of the family of experiments as exact replications (see Section 4.1.2). Moreover, the results obtained in the three experiments were similar in their direction, and the only difference (if any) was in the level of significance, which is to be expected in small numbers of subjects.

**External validity** concerns the generalization of the results. The main threats in this area stem from the simple tasks and models used in the experiment (because the controlled experiment had to be completed within a standard 3-h university lesson) and from the type of subjects. Regarding the first point, we can say that the models chosen are simple but realistic and they belong to two different application domains that are quite well known by all the subjects. Regarding the second point, we can say that the subjects were undergraduate students with little experience in requirements engineering; however, they represented a population specifically trained on Tropos and UCs (this is true in all the three experiment executions). More generally, Kitchenham et al. [36] argue that using students instead of software engineers as subjects is not a major issue, provided that the research questions are not specifically designed for experts. Moreover, it has been shown, in a specific context of requirements engineering, that students have a good understanding of the way industry behaves, and may work well as subjects in empirical studies in this area [59]. Further studies may confirm whether or not our results can be generalized to more experienced subjects (e.g., professional software analysts) and/or additional stakeholders’ types who may be potential users of requirements models and are not IT professionals (e.g., users or managers).

### 6. Discussion

In this study, we aim to compare the comprehensibility of UC and Tropos requirements models, providing evidence that these languages, although belonging to different modeling approaches, are comparable with respect to model comprehensibility. This was done by defining, collecting, and analyzing measurements of model comprehensibility in a family of controlled experiments. Accordingly, our discussion is threefold. In Section 6.1, we discuss the challenges in empirical evaluations of model comprehensibility and their manifestation in this study. Section 6.2 summarizes and discusses the results obtained regarding the comparison of UC and Tropos in terms of comprehensibility level and required effort. Finally, in Section 6.3 we briefly note on the practical implications of our findings.

#### 6.1. Challenges in empirical evaluations of model comprehensibility and their manifestation in this research

In [6] the authors present a framework for empirical evaluation of model comprehension that is grounded in the underlying theory of the language to be evaluated and in frameworks from cognitive science. They further refer there to challenges to define the comprehensibility construct and challenges to empirically evaluate comprehensibility. The critical questions that these challenges raise include: Does the experiment compare modeling languages having the same level of expressiveness and abstraction? Does it evaluate them in models having the same level of complexity (in terms of size, coverage of language constructs, and application domain)? Are the experimental results biased by the subjects’ skill level and by their (possible varying) confidence in performing the different tasks? Some of these issues correspond to well-known threats to the validity of controlled experiments (e.g., subjects’ expertise) and have been addressed in Section 5.2. Others are briefly discussed in the following.

#### 6.1.1. Definition of the comprehensibility construct

As suggested in [6] we referred to correctness of understanding and the time required to understand the representation as the main affected comprehensibility variables. Issues about the subjects’ confidence and their perception of the difficulty of the questions, which may vary during the experiment, have been addressed.

---

**Fig. 7. Tropos legend.**

---

Actor dependency diagram (social dependencies perspective)

- actor
- soft-goal
- goal
- plan
- resource

Actor goal diagram (actor perspective)

- actor
- goal
- plan
- resource
- actor dependency
- resource dependency
- goal dependency
- plan dependency

Goal or Plan Decomposition

- AND dec.
- OR dec.
- Plan to goal means-ends
- Positive contribution

The "why" of actor dependency

- why
in Exp2 and Exp3 by enhancing the design of Exp1. The analysis of the collected data (as presented in Appendix B) confirms that these issues do not affect the results.

We further took a special care of the affecting comprehensibility variables presented in [6] when designing the experiment. In particular, the types of task we used were carefully chosen to represent typical tasks requirements analysts usually perform. We tried to balance the language expertise by devoting more time to teach and tutor the new language – Tropos and only refreshing the principles of the already known language – UC. In order to make sure that the lack of domain expertise is not an obstacle for comprehension, we chose two relatively simple application domains (eRest, mss) with which the subjects were expected to be familiar to some extent. Finally, we carefully controlled the problem size: the stories were chosen to be realistic, but not too complicated to be completed in the experiment’s timeframe. Model complexity was also checked (see Table 3 in Section 4.1.3), in order to obtain comparable coverage of model constructs, a comparable number of their instances, and relationships of similar types, across the models.

6.1.2. Empirical evaluation of comprehensibility

The authors of [6] raise three challenges to empirically evaluate comprehensibility. The first challenge refers to the information equivalence. Although the expressiveness of the languages in the study is not equivalent (see the metamodels in Figs. 1 and 2 in
Fig. 10. UC diagram and template legend.

**USE CASE**

<table>
<thead>
<tr>
<th>Label</th>
<th>Primary/Secondary</th>
<th>What should be true before</th>
<th>What will be true after</th>
<th>Description of the principal task realizing the UC</th>
<th>Description of task(s) alternative to the main one (0 if any; 1 for each alternative behaviour)</th>
</tr>
</thead>
</table>

**Fig. 11.** Use Case diagram for the e-Rest system.

**USE CASE TEMPLATE**

| Get Dishes and Food Information | Secondary (system function not directly invoked by the Customer) | The function realizing the Browse Menu use case has been activated | All information requested has been retrieved and browsed | The requested information about the ingredients of the dishes are searched by the e-Restaurant system | The nutritional information about the ingredients of the dishes are searched by the e-Restaurant system |

**Fig. 12.** Use Case template for the UC “Get Dishes and Food Information”.
Section 2.2), both languages are used for requirements specification. We examined only aspects that can be expressed in both modeling languages. The second challenge, accessibility of participants, was addressed by carefully selecting students whose background represent to some extent the skills of requirements analysts. Nevertheless, as mentioned in Section 5.2 this selection is also a threat to the validity. Finally, to eliminate or minimize researcher bias and agenda, four of the authors of this paper are not involved in the development of the examined languages.

6.2. Explaining the obtained results

Analyzing at the results obtained for the three question categories, we found that for the “reading” tasks, namely for the text-model mapping and model reading questions, the results are significant and are in favor of Tropos, while, for the “writing” tasks, i.e., in the model modifying questions, the subjects experienced difficulties resulting in low comprehension level in both languages, as detailed in Fig. 5 (and confirmed by the assessment of the subjects in the post-lab questionnaire in Table 8). Indeed, creating and updating models is a complicated task, and it requires greater expertise in using the modeling languages and in the application domain. However, further investigations need to be performed to test this hypothesis.

In an effort to understand the reasons for the results, we examined the specific questions in each category and found two possible sources for the significant differences. First, some of the questions in the model reading category required consulting the UC template, which is textual. In Tropos, the information required to answer these questions was always found in a diagrammatic form, in the actor and/or the goal diagram. Research conducted in cognitive science has already shown that information search and inference is easier and more efficient in diagrammatic representations than in sequential representation, since in the former case information is organized by location and “often much of the information needed to make an inference is present and explicit at a single location” [38]. Indeed, in order to answer questions correctly in the model reading category, the subjects had to find the relevant location in the model (UC, actor, or goal diagrams) and then search for the relevant information. In Tropos, this information was explicitly linked to the “found location” via dependencies, whereas in the UC model the subjects had to search further the relevant UC template by its name.

Second, the answers in which significant differences were found referred to inputs, outputs, and possible execution paths. While these aspects are represented explicitly, as classes, in the Tropos metamodel (“Resource” for inputs and outputs and “Plan” for execution paths), the UC metamodel refers to them secondarily as fields in the template, making them less salient for novice users.

Another interesting result concerns the subjects’ perception of the informativeness of the modeling languages. According to the post-lab questionnaire (question P11, see Appendix A.3), the subjects judged Tropos to be more informative than UC (see Table 8). This may also be attributed to the prevalence of visual modeling concepts available in Tropos as compared with UCs that contain also a textual part. This fact can also explain the directions of the measures related to the general effort and the comprehension level. In fact, comparing the metamodels of the two languages, Tropos allows the user to model the system and its context, exploiting more structured constructs, such as goals, tasks, and AND/OR relationships between them. In UC, these concepts can be expressed using free sentences in natural language associated with the sections “main actions” and “alternative actions” in the UC template. It seems therefore that Tropos allows the subjects to manage the model in a more comprehensible way, but at the same time, the structured visual modeling concepts constrain analysts so that they have to increase their effort when using them (as shown in Fig. 6 and detailed in Table 6).

6.3. Practical implications

Our findings have also practical implications. As noted in the introduction, GO and SB can be seen as complementary approaches, but in real projects frequently only one approach is selected. When considering which language should be selected for dealing with the requirements of a specific project, Tropos may be considered on an equal basis with UC, as it has been found to be somewhat more comprehensible. Comprehensibility is very important when considering the documentation and communicating requirements. Indeed, we found that the effort required to comprehend Tropos models is greater than the effort required to comprehend UC models. However, the productivity, which takes into account both comprehension level and effort, is similar in the two modeling languages. Note that the effort required for comprehending Tropos models may decrease after intensive adoption of Tropos by an organization. However, only further research may confirm this hypothesis.

Additional implications relate to education in the field of requirements engineering. Educators also face the need to choose between different modeling languages when teaching requirements specification via models. Understanding the strengths and weaknesses of each language, based on the results of this study, may provide the basis for selecting the more appropriate language to the teaching objectives. Moreover, the results can guide educators to emphasize in class the weaknesses of the taught language and support students in overcoming related comprehension difficulties. For example, we found that Tropos is generally stronger than UC in the comprehension variable, but weaker in the effort variable. We also found that the strength in the comprehension variable is not significant in model modifying but only in text-model mapping and model reading.

7. Conclusions and future work

In this paper, we presented the design and the results of a family of experiments with the objective of comparing the model comprehensibility of two state-of-the-art requirements modeling languages, Use Case (UC) and Tropos, from a requirements analysts’ perspective. Although these languages belong to two different modeling approaches, namely scenario-based (SB) and goal-oriented (GO) respectively, we proposed a way to compare the comprehensibility of their models. The experimental design was counter-balanced, having two treatments, Tropos and UC. We conducted a family of experiments with 79 information systems students representing novice requirements analysts. We focused on three categories of problem-solving comprehension questions, namely, text-model mapping, model reading, and model modifying, and on the effort in terms of the time required to perform them. The subjects’ comprehension of the models was assessed using a questionnaire-based approach. In particular, we measured the comprehension level of each subject using an information retrieval-based approach that allowed us to achieve a balance between the correctness and completeness of the answers. The design addressed possible influencing factors, such as differences in the degree of expressiveness of the compared modeling languages, differences in model complexity, and the
subjects’ unbalanced skills in the two languages. The UC and the Tropos models used in the experiment represented the same system requirements, thus addressing the informational equivalence issue [55]. The analysis of the results showed that Tropos models are more comprehensible than UC models, a result that is somewhat unexpected, since the subjects had knowledge of UC modeling prior to the experiment; they learned Tropos in preparation for the experiment only. However, interesting findings emerged regarding differences in answering the different types of comprehension questions. These differences are worthy of further investigation. The subjects’ high comprehension level in Tropos came at the cost of a longer time spent performing the required tasks, which resulted in similar values in overall productivity. In practice, this ratio of cost/benefit should be taken into consideration when selecting a modeling language with which to work.

This study contributes to both research and practice. Our first contribution stems from designing and executing a family of experiments for comparing the two modeling languages – scenario-based UC and goal-oriented Tropos, illustrating how they can be compared in terms of comprehension level, effort, and productivity. The experimental design can be adapted and adapted for comparing any two analysis modeling languages that may be used for expressing the same set of requirements. The second contribution stems from the results of comparing UC and Tropos. By comparing these languages we were able to highlight the weaknesses and strengths of each language. Providing this information to practitioners and educators may support a more informed decision-making regarding the modeling language choice within a project or an educational program. We expect that choosing the more appropriate modeling language according to the task at hand and understanding its strength and weakness would increase the comprehension of, and communication about, models and thus improve the quality of the systems or the success of teaching the modeling language. Future research may examine this assumption.

Future research may also replicate this experiment using larger classes of experienced students and practitioners. Additional domains, specifically ones that are expected to be less familiar to the subjects, can be used in order to analyze whether familiarity with the domain affects the results. Moreover, the tasks can be extended to creating requirements models from scratch based on user requirement documents only and for evaluating different aspects of the languages. An additional direction for future research is to conduct studies in industrial settings with experienced practitioners and real customer specifications. In these settings, additional perspectives of different stakeholders’ types (e.g., customers, managers, and designers) as well as additional variables (such as the degree of training with the modeling languages) can be examined. In parallel, it would be interesting to qualitatively investigate how subjects retrieve information from the different models and their opinions about their characteristics (e.g., navigating between two diagrams in Tropos vs. a diagram and a textual template in UC). Finally, future research may compare other goal-oriented languages and scenario-based approaches to generalize further and utilize the research design and results.

Appendix A. Examples of the experiment material

This appendix describes some of the artifacts exploited in the experiment. In particular, we present the pre- and post-lab questionnaires, as well as the system story, the models, and the comprehension questions in the three categories (text-model mapping, model reading, and modifying) of the e-Restaurant (eRest) case.

A.1. Pre-experiment questionnaire (possible answers are between brackets)

| PR1. What is your year of study? [1, 2, 3, other] |
| PR2. What is your average exam score? [low, medium, high] |
| PR3. Have you ever worked as a computer programmer or analyst in industry? [no, yes (part-time), yes (full-time)] |
| PR4. What is your experience in the field of requirements engineering? [very low, low, medium, high, very high] |
| PR5. What is your experience/knowledge of UML? [very low, low, medium, high, very high] |
| PR6. I understand the meaning of “late requirements” [strongly agree, agree, not certain, disagree, strongly disagree] |
| PR7. The lecture about Tropos was sufficiently clear [strongly agree, agree, not certain, disagree, strongly disagree] |
| PR8. Graphical elements used in Tropos are clear [strongly agree, agree, not certain, disagree, strongly disagree] |
| PR9. I am able to understand Tropos Diagrams [strongly agree, agree, not certain, disagree, strongly disagree] |
| PR10. Graphical elements used in Use Case diagrams are clear [strongly agree, agree, not certain, disagree, strongly disagree] |
| PR11. I am able to understand Use Case Diagrams [strongly agree, agree, not certain, disagree, strongly disagree] |
| PR12. The chosen Use Case template is clear [strongly agree, agree, not certain, disagree, strongly disagree] |
| PR13. I am able to understand the textual Use Cases [strongly agree, agree, not certain, disagree, strongly disagree] |
| PR14. The tasks are clear [strongly agree, agree, not certain, disagree, strongly disagree] |

A.2. The main questionnaire

A.2.1. The e-Restaurant story

To improve the business, the owner of a restaurant chain in Trento and Haifa decided to introduce an e-Restaurant system for the management of several services in the restaurant.

The e-Restaurant system includes a touchscreen device placed on each table of the restaurant, which communicates with the restaurant database. Using the system the customers may choose their dishes and directly order them while the restaurant’s waiter or chefs can manage information related to these orders.

The e-Restaurant system will allow the customers to browse the available choices in the menu, inspect information on food, in particular to inspect nutritional and other information about the dishes, such as dishes’ ingredients, as well as their prices.

Using the system the customers can place the order automatically or call a waiter to place the order (they can also ask waiters to help them in using the system). The customer can also use the system to request the preparation of the bill (or again call a waiter for that).

The waiters can interact with the system to start the preparation of the bill or to check if all the customers he/she served paid the bill.
The e-Restaurant manages also the list of pending orders of ingredients to retailers and requires the connection with the ingredients Data-Base for checking available ingredients in order to allow the chef to have a picture of the available ingredients while updating the menu offered by the restaurant.

A.2.2. The Tropos model of the e-Restaurant system

The Tropos legend is presented in Fig. 7; the Tropos model of the e-Restaurant system includes an actor diagram, presented in Fig. 8, and a goal diagram, presented in Fig. 9.

A.2.3. The Use Case model of the e-Restaurant system

The Use Case legend is presented in Fig. 10; the UC model of the e-Restaurant system includes a UC diagram, presented in Fig. 11, and a UC template, presented in Fig. 12.

The comprehension questions

1. How many actors, mentioned in the system story, have been modeled? What are their names?
2. Are there actors in the model who are not mentioned in the system story? If so, name them.
3. Are there any actors, mentioned in the system story, who have NOT been modeled? If so, name them.
4. Is there any functionality in the system story, which is NOT mentioned in the model? If so, describe it (them) in a few words.
5. How many relationships between actors appear in the model? Specify the number of dependencies per type.
6. Which info can the e-Restaurant system get from Ingredients DB, according to the model only? Please provide a short description.
7. How many relationships between the Waiter and the e-Restaurant system appear in the model? What are their types?
8. According to the model, describe the ways the Customer can place an order through the e-Restaurant system? What are they? Please provide a short description.
9. According to the model, who are the actors that provide the e-Restaurant system with the data about the available ingredients?
10. According to the model, which type of information about dishes can the Customer get from the e-Restaurant system? Please provide a short description.
11. How do you modify the model to include the fact that the e-Restaurant can give information about the history of the dishes? Make the modification directly on the model and describe here which modification you made.
12. How do you modify the model if the Customer has to complete the activity of placing an order using the e-Restaurant system, without calling the Waiter? Make the modification directly on the model and describe here which modification you made.
13. How do you modify the model if the e-Restaurant system will get information about the status of the pending orders from the Ingredients DB instead of from the Retailer? Make the modification directly on the model and describe here which modification you made.
14. How do you modify the model to represent the fact that the bill preparation will be performed by the e-Restaurant system, without the intervention of the Waiter? Make the modification directly on the model and describe here which modification you made.

A.3. Post-lab questionnaire (possible answers are between brackets)

P1. I had enough time to perform the lab tasks [strongly agree, agree, not certain, disagree, strongly disagree]
P2. The objectives of the lab were perfectly clear to me [strongly agree, agree, not certain, disagree, strongly disagree]
P3. The description of the system story was clear [strongly agree, agree, not certain, disagree, strongly disagree]
P4. The tasks were perfectly clear to me [strongly agree, agree, not certain, disagree, strongly disagree]
P5. I experienced no difficulty in understanding the Tropos/UC diagrams [strongly agree, agree, not certain, disagree, strongly disagree]
P5a. I experienced no difficulty in understanding the UC template [strongly agree, agree, not certain, disagree, strongly disagree]
P6. I experienced no difficulty to answer the first group of questions (1–5) [strongly agree, agree, not certain, disagree, strongly disagree]
P7. I experienced no difficulty to answer the second group of questions (6–10) [strongly agree, agree, not certain, disagree, strongly disagree]
P8. I experienced no difficulty to answer the third group of questions (11–14) [strongly agree, agree, not certain, disagree, strongly disagree]

Extra questions for the group working with Tropos (write what you think independently from the executed exercise. Use your knowledge and previous lectures on Tropos and UC):

P9: I think that understanding Tropos diagrams is in general hard [strongly agree, agree, not certain, disagree, strongly disagree]
P10: I think that updating Tropos diagrams to take new/changed requirements into account is easier than updating the corresponding UC model (diagrams and templates) [strongly agree, agree, not certain, disagree, strongly disagree]
P11: I think that Tropos diagrams are more informative than UC diagrams [strongly agree, agree, not certain, disagree, strongly disagree]
P12: I think that Tropos Requirements models may substitute for UML UCs in the first phases of software development dealing with requirements analysis [strongly agree, agree, not certain, disagree, strongly disagree]

Appendix B. Analysis of other factors

In addition to the analysis of the dependent variables of our study, we analyzed the following factors: experiment’s labs schedule, application domains, order of application domain, order of treatments, order of questions, and confidence and perceived difficulty. The experiment’s labs schedule factor accounts for learning or fatigue effects. The application domain factor lets us evaluate whether the application domain influences, in some way, the comprehensibility of models expressed in Tropos or UC. The order of application domains, order of treatments and order of questions in the comprehension questionnaire factors test whether or not the chosen experimental design was able to mitigate order effects. Finally, the analysis of confidence and perceived difficulty shed additional light on the obtained results.
B.1. Experiment’s labs schedule

Applying two-way ANOVA globally, we found no effect of Lab on the comprehension level (p-value = 0.26) and no interaction between the two factors. This is true also when separating by experiment. Our analyses, however, revealed an effect of Lab on the effort (p-value < 0.01). Nevertheless, we can exclude any significant interaction between the Lab factor and the Treatment (p-value = 0.96), and thus, although some learning effect on effort was observed, it does not confound the effect of the treatment. Separating the analysis by experiment, we observed an effect of the Lab on effort only in Exp2 and Exp3.

B.2. Application domains

Analyzing the results separately per application domain (mss and eRest), we found that, in the eRest case, Tropos significantly outperforms UC (0.75 vs. 0.60 as medians, p-value = 0.02), while in the mss case the situation is less clear-cut. In the latter, the difference is of only 5 points in medians in favor of Tropos and it is not significant (p-value = 0.13). This means that Tropos is more comprehensible for eRest than for mss. Further, when separating by experiment, in all experiments Tropos outperforms UC in the eRest case, while in the mss case Tropos outperforms UC for Exp 1 and Exp 3 and UC outperforms Tropos for Exp 2. Concerning the differences in the comprehension level of the two application domains, we can propose two possible causes: the size of the problem and the domain expertise of the subjects (see [6]). As for the problem size, the complexities of the two models were compared in Table 3 in Section 3, showing that they can be considered comparable. Moreover, to answer the comprehension questions the subjects had to focus on (the same set of entities in both application domains. Thus, we assumed that the differences in the comprehension level might be attributed to the prior domain expertise of the subjects.

The effect of the application domain factor is also revealed by a two-way ANOVA analysis by treatment and application domain, showing a significant effect of the application domain factor on the comprehension level (p-value < 0.01). Moreover, we can exclude a possible interaction between the two factors (p-value = 0.75). When we separated ANOVA tests for each experiment, it revealed a significant effect of the application domain factor on the comprehension level only for Exp1 (p-value = 0.03) and Exp2 (p-value < 0.01). An interaction between the two factors exists only in Exp2 (p-value = 0.01).

A two-way ANOVA analysis by treatment and application domain reveals a significant effect of the application domain (p-value = 0.02) on the effort. It also rejects the possibility of an interaction between treatment and effort (p-value = 0.23). This outcome reinforces our assumption that the subjects found the restaurant scene more familiar and intuitive than the meeting scheduling systems. The separation of ANOVA tests for each experiment revealed a significant effect of the application domain factor on the effort only in Exp1 (p-value = 0.01). No interaction between treatment and effort is revealed in each experiment.

When we analyzed the time needed to answer the questionnaire separately per application domain (mss and eRest), we found that for mss the difference is significant (p-value = 0.04) while for eRest it is not (p-value = 0.17). It is interesting to note that for both applications the students needed more time to answer the questionnaire with Tropos. Moreover, we can see that they spent more time on mss with both Tropos and UC. In particular, this is so when the Tropos treatment is administered.

B.3. Order of application domains (eRest first or mss first)

Applying two-way ANOVA globally, we found no effect of order of application domains on the comprehension level (p-value = 0.82) and no interaction between the two factors (p-value = 0.82). This is true also when separating by experiment. The same is true for the variable effort. Globally, the p-values are 0.54 for the order of application domains factor only and 0.41 for the interaction between the two factors. This means that the experimental design was able to mitigate the order of application domains effects.

B.4. Order of treatment (Tropos first or UC first)

Applying two-way ANOVA globally, we found no effect of order of treatment on the comprehension level (p-value = 0.14) and no interactions between the two factors (p-value = 0.27). This is true also when separating by experiment. Our analyses, however, revealed a significant interaction between the order of treatment factor and the treatment factor for the variable effort (p-value < 0.01) but no effect of order of treatment on the effort (p-value = 0.91). Separating the analysis by experiment, we observed an interaction between order of treatment and treatment on the variable effort only in Exp2 and Exp3. No effect of the order of treatment factor on the effort was found. In addition, in this case, since we observed only a significant interaction but not a clear contribution of the order of treatment factor, we can conclude that the chosen experimental design was also able to mitigate order of treatment effects.

B.5. Order of questions in the comprehension questionnaire

We analyzed the effect of the learning across subsequent questions within labs, using a repeated measure ANOVA. The analysis was performed separately for the two systems mss and eRest, as the questions being asked are different. Specifically, this test allows to distinguish the between subjects variance, due to the application of different treatments to different subjects, from the within subjects variance, due to: (i) different treatments received by each subject according to the experimental design; (ii) the ordering and possible different difficulty of the questions being asked; and, (iii) the interaction between these two factors. The results of the analysis indicate that in both systems we obtained a significant within subjects effect of the order of questions factor. This means that subjects exhibit different comprehension level on different questions. In particular, the number of correct answers to the questions Q3 and Q4 was very low for both systems. We also observed a difficulty to interpret interaction between the question factor and treatment. Finally, based on the analysis we cannot indicate a learning effect across subsequent questions.

B.6. Confidence and perceived difficulty

As noted, the median values of the subjects’ confidence and their perception of the difficulty of the questions were collected for Exp2 and Exp3. The median value of the perceived confidence (median of all the answers) is 4 (out of 5) for both languages in Exp2, and 3 for both languages in Exp3. This supports the conclusion that the subjects trusted their own answer quality. However, the subjects in Exp2, who were third year students, had more confidence in their answers than the subjects in Exp3, who were second year students. Interestingly, the subjects in Exp3 answered the questionnaire, in average, better than the subjects in Exp2, independently from the modeling language used, even though their confidence was lower (3 vs. 4).

The median value for perceived difficulty is 3 for the two modeling languages in both experiments, indicating that the subjects...
judged the questions to be of medium difficulty, irrespectively of the modeling language.

References


