Requirements Comprehension: A Controlled Experiment on Conceptual Modeling Methods

Mirko Morandini, Alessandro Marchetto and Anna Perini
Fondazione Bruno Kessler – CIT
38123, Povo - Trento, Italy
{*morandini,marchetto,perini*}*@fbk.eu*

***Abstract*—Several Requirements Engineering (RE) methods have been proposed to analyze and model requirements specifications. However, these methods have often been only partially evaluated and few attempts exist in literature to study and evaluate RE methods through experiments.**

In this paper, we document an empirical study that has been performed to evaluate the comprehension of requirements which were expressed in Tropos4AS. Tropos4AS specializes the goal-oriented software engineering methodology Tropos for the case of self-adaptive systems. In the experiment, we asked subjects to perform comprehension tasks on requirements specifications expressed in Tropos4AS and in Tropos, respectively.

Results show that Tropos4AS is more effective than Tropos in describing requirements of self-adaptive systems, especially when the models are used by novice requirements engineers.

***Keywords*-Requirements engineering, empirical studies, goal-oriented requirements engineering, self-adaptive systems.**

I. Introduction

Requirements Engineering (RE) is a well-known discipline that studies processes, methods, languages and tools to support system engineers during the analysis of the requirements of the system they need to develop and maintain. Several goal-oriented modeling languages and methods (e.g., *KAOS*, *i** and Tropos, for an overview see [12]) have been proposed to analyze requirements and to generate clear and meaningful requirements specifications.

A relevant effort has been devoted by the RE community in particular for proposing methodologies and specification methods for specialized system domains (e.g., agent-oriented systems, service-oriented systems, and self-adaptive systems), since general purpose RE methodologies and methods are often inadequate to capture and model domain specific features. For instance, for analyzing the requirements of service-oriented systems it may be of benefit to exploit modeling languages that provide notions of services, service providers and consumers (e.g., as proposed in [6]), rather than general concepts such as actor roles, goal and tasks (as e.g. in *i**). Analogously, for describing self-adaptive systems[1], concepts for specifying how to manage normal and exceptional behaviors are key-factors [5]. We might thus claim that a modeling language (e.g., *Tropos4AS* [10]), which lets explicitly model conditions, goal satisfaction dynamics and errors occurring in a system, should be preferred to a general purpose specification method (e.g., Tropos), not providing such constructs to explicitly manage these features, even though it adds a considerable amount of complexity.

But how can we collect empirical evidence at support of this claim? We believe that empirical studies with subjects can play a fundamental role in assessing the effectiveness and viability of RE methods. In fact, human subjects (e.g., stakeholders, requirements engineers, designers, and developers) create the requirements artifacts and also need to comprehend them, to exchange information and to find agreements on how to satisfy their needs.

Despite their potential relevance, in literature only few attempts exist to study and validate RE methods through experiments (see Section VI). Two characteristics of RE methods can mainly limit the possibility to validate them by means of empirical studies: (i) RE methods often involve iterative processes characterized by different phases that are difficult to isolate and check individually; and (ii) they aim at capturing the stakeholders' needs, expressed in informal or semi-formal languages, usually reflecting a *subjective* understanding of the domain.

An additional element of complexity in realizing such empirical studies concerns the definition of meaningful evaluation measures. A recent work [2] discusses a set of challenges in evaluating *model comprehensibility*, including information equivalence in two different representations; accessibility of study subjects to the modeling methods under investigation; and the researchers' bias in designing and conducting empirical studies which involve methods they created. Speaking about *communication effectiveness* in requirements models, [9] recommends to consider issues of visual syntax besides language semantics, when evaluating conceptual modeling languages.

In this work, we present an experimental study with subjects we conducted to assess the use of Tropos4AS [10] for the comprehension of requirements for self-adaptive systems. Tropos4AS is a design methodology that extends

[1]Self-adaptivity is defined as the ability of a system to automatically take decisions about the actions to be done, based on its knowledge of what is happening in the operating environment and guided by objectives it has been designed for.

978-1-4577-1076-6/11/$26.00 ©2011 IEEE

the goal-oriented software engineering methodology Tropos with specific elements, which are helpful to model self-adaptive systems. Therefore, on one side, we performed an experiment to validate the use of Tropos4AS by asking several subjects to comprehend the behavior of two software systems by analyzing their requirements specifications (provided in the form of textual requirements complemented with Tropos/Tropos4AS models). On the other side, we used our experiment for a comparative evaluation between Tropos and Tropos4AS, with respect to the domain of self-adaptive systems.

A. Goal-Oriented Requirements Engineering

Goal-oriented requirements engineering (GORE) [12] roots in organizational modeling, and more generally in conceptual modeling for requirements specification and analysis. GORE methods use the human-oriented notion of *actor* with its *goals* and *dependencies* for the description of the system-to-be, where the *goals* are objectives the system under consideration should achieve [12]. They are concerned with the use of goals for eliciting, elaborating, structuring, specifying, analyzing, negotiating, documenting, and modifying requirements.

The stakeholders' objectives and users' preferences become high-level requirements for the definition of the system behavior, including possible variability in the problem space. To go toward a detailed definition of the *system-to-be*, the requirements engineer refines the high-level goals to address more and more concrete concerns, by decomposing goals, by finding alternatives tailored to a specific context, and finally by discovering means for their achievement. This kind of analysis is supported by *Goal Models*, built by hierarchical AND/OR decomposition of high-level goals, to describe the *rationale* of a system.

A variety of approaches have been developed, which adopt goal models for requirements modeling, mostly rooting on the visual notation and analysis techniques proposed in the *KAOS* or *i** modeling languages.

The agent-oriented methodology Tropos (for an introduction see [3]) proposes a software development process, which uses GORE concepts to define and to detail the requirements of a system. The actors in an organization delegate their goals to the system-to-be. For defining the system, these goals are decomposed and detailed, until it is possible to define concrete *plans* as a means for goal achievement.

Tropos4AS [10] extends Tropos goal models, targeting the description of self-adaptive systems, that shall be aware of their environment and able to adapt to environmental changes. Tropos4AS introduces a way to model an actor's perceived environment and possible failures that can be identified and prevented by recovery activities. Moreover, goals can be annotated to define the run-time goal achievement behavior, with various conditions related to the environment, e.g. for goal creation, achievement and failure.

In the present work, we focus on modeling aspects of the Tropos and Tropos4AS languages, without considering the complete development processes.

	Number of Elements		Number of Dependencies	
	Tropos	Tropos4AS	Tropos	Tropos4AS
PMA	30	37	35	50
WMM	23	33	31	47
avg	26.5	35	33	48.5

Table I
SIZE OF THE MODELS (NUMB. OF CONCEPTS)

II. EXPERIMENT PLANNING AND DESIGN

This section describes the design of the performed experiment, following the guidelines by Wohlin et al. [13]. For replication purposes, the experiment package is made available at the experiment website[2].

The *goal* of the experiment is to study requirements modeling methods with the *purpose* of evaluating their effectiveness in supporting the comprehension of requirements for self-adaptive systems. Hence, Tropos and Tropos4AS are the two *treatments* that we consider in the experiment. The *quality focus* of the experiment concerns the capability of the treatments in supporting the analysts in requirements comprehension. The *perspective* is both of researchers, proposing and comparing requirements engineering methods, and of project managers, evaluating the possibility to adopt a specific requirements method in their organization. The *context* of the experiment consists of two *objects*, requirements specifications of two software systems, and of twelve *subjects*, researchers and Ph.D. students working in a research center. With this premise we investigate on the following research question:

Rq : Do Tropos4AS models have a significant impact on the comprehension of the requirements for a self-adaptive system, with respect to Tropos models?

This research question is translated into the corresponding null (H_0) and alternative hypothesis (H_A):

H_0 : Tropos4AS models do not significantly improve the requirements comprehension, in comparison to Tropos models.

H_A : Tropos4AS models significantly improve the requirements comprehension, in comparison to Tropos models.

Assuming that the extensions proposed by Tropos4AS might have a positive impact on the comprehensibility of system requirements, the hypotheses have been formulated with a clear direction, i.e., they are *one-tailed*.

[2]http://selab.fbk.eu/morandini/comprehension.html

978-1-4577-1076-6/11/$26.00 ©2011 IEEE

A. Object

The objects of the study are the specifications of two software systems, which have some features of self-adaptivity and are thus suitable for modeling with both treatments: a Patient Monitoring Agent (PMA) and a Washing Machine Manager (WMM). PMA is a system installed in a smart home (equipped with sensor networks) and monitors elderly people in a non invasive way. PMA is aware of guests' specific food diets and medicines, and automatically alerts caregivers if critical events happen with reference to the daily plan for meals and medicine of the individual guests. WMM is an intelligent washing machine controller, which self-adapts the washing settings to clothes and the user's preferences about energy saving and cleanness. For both systems, the textual requirements specifications have been recovered from the applications by doing a trade-off between the complexity of fully describing a self-adaptive system, and the adequateness to the experimentation.

With the aim of creating models that ensure a fair comparison, the following procedure has been applied to create the corresponding graphical models: (1) We asked two experienced modelers (not involved in the experiment) to model the textual requirements of WMM and PAM using both Tropos and Tropos4AS. (2) We analyzed the resulting eight models (four by each expert), using ideas and parts from them, creating a single model for each considered treatment and system. (3) We refined the obtained models for creating a set of "gold standard" models that try to correctly capture a high number of requirements. Table I summarizes the size of the considered models[3] in terms of number of instances of concepts (e.g., actors, goals, plans) and dependencies among them. (4) We checked the models to guarantee that they were not tailored toward one of the considered treatments, but they were challenging enough to prompt the use of requirements design methods.

B. Population

The subjects of the experiment are 12 persons (6 researchers and 6 Ph.D. students) working at or visiting the FBK Center for Information Technology[4]. According to their scientific work we expected that all subjects have a fairly good knowledge of software engineering methods and of systems design and development, while only a part of them have previously used goal-oriented modeling languages. All the subjects have been trained on the use of both treatments before performing the experiment, by following a presentation and by individually performing two complex modeling tasks, to train and assess their modeling abilities.

C. Design

We adopted a paired, counterbalanced experiment design based on two laboratory sessions. In this design, each subject has to perform the experimental task twice, once with each object and treatment. Specifically, the subjects have been randomly divided into 4 groups of 3 people. Each group was given the two treatments and objects (PMA and WMM), following the scheduling summarized in Table II. This design mitigates learning effects between the two treatments and between the two objects. Moreover, it enables the application of proper, accurate statistical tests suitable to analyze the effect of multiple factors.

	Group 1A	Group 1B	Group 2A	Group 2B
Lab 1	PMA-Trop	WMM-T4AS	PMA-T4AS	WMM-Trop
Lab 2	WMM-T4AS	PMA-Trop	WMM-Trop	PMA-T4AS

Table II
EXPERIMENTAL DESIGN (TROP=TROPOS AND T4AS=TROPOS4AS)

D. Procedure and Material

The experiment consisted of an initial training session, a pre-questionnaire, two laboratory sessions concerning a comprehension task, and a final questionnaire with few post-experiment questions. During the training session (lasting about 2 hours), the subjects have been trained on the use of Tropos, Tropos4AS, and on the tasks to be performed during the experiment itself. The laboratory has been conducted under the supervision of the authors, in an overall time frame of approximately 40 minutes.

To perform the experiment, each participant received the following material, for each laboratory (the complete sample can be found at the experiment website):

- A pre-questionnaire containing few questions for the purpose of identifying the experience of subjects in using GORE languages and their professional position (researcher or Ph.D. student);
- A summary of the modeling language to be used (either Tropos or Tropos4AS);
- The textual requirements specifications of the system (WMM or PMA);
- The model of the WMM or PMA system, built either with Tropos or Tropos4AS;
- 5 questions about model comprehension, either for WMM or PMA;
- A questionnaire containing 4 questions about the experiment settings and the perceived feedback.

We asked the subjects to answer the five comprehension questions on the object assigned, by looking at the requirements specifications. Samples of the questions are shown in Table III (top). They are open questions concerning the comprehension of the requirements specifications, asking for capturing normal or exceptional behavior of systems. By performing a pilot study before the experiment run (with two master students not involved in the experiment), we recognized that 30 minutes represent an appropriate time frame for conducting the comprehension task.

[3]Detailed requirements specifications available at the experiment website.
[4]http://cit.fbk.eu

978-1-4577-1076-6/11/$26.00 ©2011 IEEE

PAM	In which occasions can the dinner be skipped?
PAM	With which sensors does the system have to interact to get the necessary information?
WMM	Which sensor interface are needed?
WMM	With which user settings and conditions will the system restart a new washing cycle?
cq1	I had enough time for accomplishing the tasks
	1 strongly agree 2 agree 3 not certain 4 disagree 5 strongly disagree
cq2	The comprehension questions were clear
	1 strongly agree 2 agree 3 not certain 4 disagree 5 strongly disagree
cq3	I was able to extract the asked information from the goal model
	1 nearly all 2 most 3 half 4 somewhat 5: nearly nothing
cq4	I needed to search the information in the textual requirements
	1 nearly all 2 most 3 half 4 somewhat 5: nearly nothing

Table III
EXAMPLES OF COMPREHENSION (TOP) AND POST-QUESTIONNAIRE (BOTTOM) QUESTIONS.

Finally we asked the subjects to fill in a post-questionnaire after each laboratory (Table III bottom). The four questions are defined on a Likert scale [1-5]. In detail, questions cq1 and cq2 concern the perceived adequacy of the experiment setting while cq3 and cq4 concern the information source (model or textual requirements) mainly used to answer the comprehension questions.

E. Detailed Research Questions and Hypotheses

To evaluate the impact of the two modeling languages on requirements comprehension, we defined ways to measure the subjects' effectiveness in retrieving correct information from the requirements specifications. To achieve a fair comparison, we provide the models together with the textual requirements descriptions. This also represents a reasonably realistic scenario in which a modeler uses both the models and a description to understand the system requirements.

The experiment design bases on the assumption that the comprehension tasks can be carried out faster by looking at the models than by fully reading and comprehending the textual specification. However, since we also hand out the textual requirements descriptions, we have also to evaluate this assumption, by asking the subjects an estimation about the amount of information extracted from the models and from the textual descriptions. Hence, the research question **Rq** has been tackled by the following questions that will be evaluated in the experiment:

Rq1 : Is the effectiveness in retrieving correct information (in a limited time) from textual descriptions and Tropos4AS models greater than from the same textual descriptions and Tropos models?

Rq2 : Is a Tropos4AS model more useful than a Tropos model, to extract the information?

For the sake of conciseness, we do not report here the null- and alternative hypotheses derived for **Rq1** and **Rq2**, which are similar to those for **Rq**.

In order to give an answer to these questions, we analyze the following aspects: (a) the correctness of the information extracted by the subjects from the models and the textual specifications; and (b) the amount of information extracted from the model with respect to the information extracted from the textual requirements specifications.

F. Variables and Measures

The *independent variable* of the study is the modeling language used, considering the treatments Tropos and Tropos4AS. The *dependent variables* (i.e., main factors) are the correctness of the subjects' answers to the comprehension questions and the amount of information extracted by the subjects from models and/or textual requirements.

To measure the comprehension level and to test the related hypothesis, we assess the answers given to the comprehension questions. As previously done in other works (e.g, [11]), we expected each answer in terms of a list of conceptual modeling elements (possibly embedded in natural language sentences). These elements are matched conceptually with the elements defined in the gold standard. The results thereof are evaluated for each answer, by calculating *Precision*, *Recall* and *F-measure*. In particular, considering $Answ_{s,i,t}$, the set of elements mentioned in the answer given with treatment t by a subject s in question i, and $ExpAnsw_i$, the set of elements we expected to be in the correct answer for question i, we calculated:

$$Precision_{s,i,t} = \frac{|Answ_{s,i,t} \cap ExpAnsw_i|}{|Answ_{s,i,t}|}$$

$$Recall_{s,i,t} = \frac{|Answ_{s,i,t} \cap ExpAnsw_i|}{|ExpAnsw_i|}$$

$$F - measure_{s,i,t} = \frac{2 \cdot Precision_{s,i,t} \cdot Recall_{s,i,t}}{Precision_{s,i,t} + Recall_{s,i,t}}$$

These three measures represent continuous variables in the range [0,1]. *Precision* indicates the fraction of correct elements out of the elements in the answer, whereas *Recall* indicates the fraction of answers in the expected answer set, which were successfully retrieved. The *F-measure* combines these two measures, by calculating their harmonic mean, into a single measure which represents the effectiveness of answer retrieval. The average for the 5 questions, of each of these measures, by subject, has been used for the statistical analysis of the result:

$$Avg(Precision)_{s,t} = \frac{\sum_{i=1}^{5} Precision_{s,i,t}}{5}$$

where $Avg(Precision)_{s,t}$, for subjects $s = 1 \ldots n$ and treatment t (e.g. Tropos) is the set of data in input to the statistical test.

Note that precision, Recall and F-measure cannot be defined if the respective denominator is 0. This could happen in two cases: (i) the set $ExpAnsw_i$ is empty (i.e., no correct answer exists for question i); or (ii) the set $Answ_{s,i}$ is empty (i.e., the subject s gives no answer to question i).

978-1-4577-1076-6/11/$26.00 ©2011 IEEE

This problem was encountered and tackled also by other studies, e.g., [1]. In our case, $ExpAnsw_i$ is never empty, thus only the second case could happen, leading to an undefined precision value. In this case we decided to set $Precision_{s,i} = 0$, with the aim of preserving the meaning of the precision measure and also, at the same time, to take into account unanswered questions.

To have an idea about the source (model and/or textual specifications) used by subjects to extract the information required to answer the comprehension questions, we asked the two questions cp3 and cp4, with answers defined on a Likert scale variable ranging from 1:*nearly all* to 5:*nearly nothing*, asking about the amount of information extracted by subjects from the models and from the textual requirements, respectively (Table III bottom).

The post-questionnaire also contains two questions (cp1 and cp2) concerning the adequacy of the experimental setting: the time required to fill the questionnaire and the clearness of the proposed questions. They are based on an ordinal Likert scale variable ranging from 1 to 5, as follows: 1:*strongly agree*; 2:*agree*; 3:*not certain* (neutral answer); 4:*disagree*; 5:*strongly disagree* (Table III center).

Additional co-factors considered in the experiment since they can potentially impact the observed results are: the objects, the subject experience and position, and the laboratory. Conversely, the time spent to answer the questions, on the contrary, cannot be considered as a dependent variable in this experiment, since we fixed it.

G. Statistical Tests

To analyze the results with respect to the main factor – considering the nature of Precision, Recall and F-measure (continuous in the range [0,1]), the paired experimental design and the limited number of data points – we used the paired, non-parametric *Wilcoxon* test, adopting a 5% significance level. Moreover, we used the *Cohen.d effect size* to estimate the magnitude of the results obtained.

For each of the four questions cq1-cq4 in the post-questionnaire, considering that the variables assume values on an ordinal scale [1-5] and the limited number of data points, we applied the same non-parametric *Wilcoxon* test and performed two analysis: (i) a comparison of the answers with the threshold "3", representing the neutral answer in the Likert scale used; and (ii) a comparison of the answers given by subjects using Tropos with those given using Tropos4AS models. A further analysis was performed by grouping the answers of each subject to cq3 and cq4. We grouped the subjects according to their answers: group GS_1 containing all the subjects answering < 3 in cq3 and, at the same time, > 3 in cq4; and GS_2 containing all the remaining subjects. Hence, we applied a Chi^2-test to evaluate the distribution of answers. This analysis identifies those subjects that found the models (of a given treatment) sufficient to answer the questions, using the textual requirements only partially.

Question		median	p-value
cq1	adequacy of time	2	0.049
cq2	clearness of questions	2	0.000015

Table IV
EXPERIMENTAL SETTINGS: RESULTS OF THE STATISTICAL TESTS.

	median Tropos	median Tropos4AS	p-value	Cohen-d
Precision	0.7	0.9	0.021	0.74 (medium)
Recall	0.58	0.75	0.020	0.82 (large)
F-measure	0.65	0.81	0.008	0.87 (large)
cq3 (model)	3	1	0.015	0.8 (large)
cq4 (text)	3	4.5	0.038	1.2 (large)

Table V
MAIN FACTOR ANALYSIS (PAIRED WILCOXON TEST).

Finally, to check the impact of co-factors (e.g. laboratory, objects, subject experience and position) on the results, we applied a two-way ANOVA (analysis of variance) test.

III. EXPERIMENT RESULTS

In this section, we present the obtained results[5] in terms of the adequacy of the experimental settings, the analysis of the main factor and the impact of various co-factors.

A. Adequacy of the experimental settings

First, we investigated whether the experimental settings were perceived as adequate by subjects, analyzing the questions cq1 and cq2 to evaluate if the participants worked under time pressure, and if the task descriptions were clear for them. Table IV shows the results. Observing median and p-value, we conclude that: (i) The time has been perceived as adequate, even if the subjects have the feeling that additional time would have been more adequate for answering the questions. (ii) The comprehension questions have been perceived to be clear.

B. Main factor: Comprehensibility of requirements specifications

Table V (top) shows the statistical data collected for Precision, Recall and F-measure with respect to the comprehension task realized by subjects, while Figure 1 (top) shows the boxplots for Precision and Recall.

By looking at the results, we observe that the use of Tropos4AS models significantly improved the precision of the answers (median increased from 70% to 90%). This result is statistically relevant and supported by a medium effect size. In other terms, from a Tropos4AS model (together with the textual requirements specifications), in a limited time, "more accurate" information can be extracted, in comparison to a Tropos model (together with the same textual specification). Tropos4AS also improved the recall of the subjects' answers (from 58% to 75% for the median).

[5]http://selab.fbk.eu/morandini/T4AScomprehensionexperiment_rawdata.zip.

978-1-4577-1076-6/11/$26.00 ©2011 IEEE

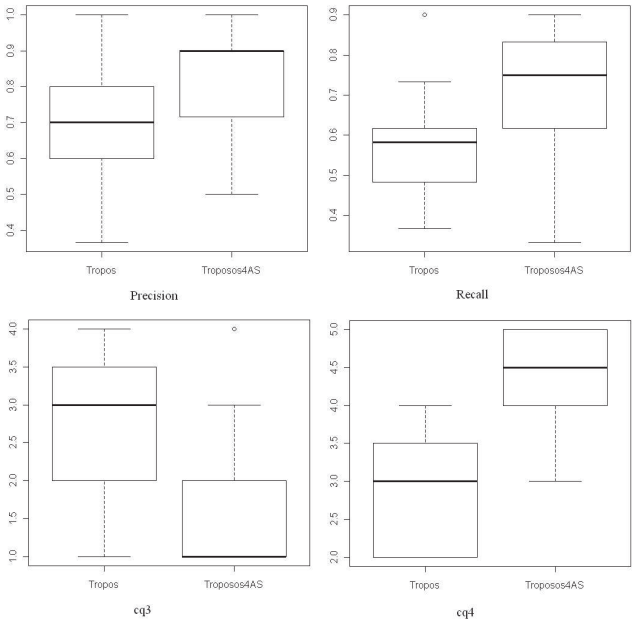

Figure 1. Boxplots of Precision/Recall (top) and cq3/cq4 (bottom)

This result is also statistically supported. In other terms, information extracted from a Tropos4AS model can be expected to be "more complete" with respect to information extracted from the relative Tropos model. Correspondingly, Tropos4AS improved, with statistical significance, also the F-measure of the subject answers (from 65% to 81% for the median), denoting hence the general effectiveness of Tropos4AS in information retrieval.

Overall, we can reject with statistical evidence the null-hypothesis and give an affirmative answer to the research question **Rq1**, hence: *A Tropos4AS model, together with the textual specifications, is more effective for retrieving correct information than a Tropos model, together with the same textual specifications.*

To understand if the Tropos4AS models were used to a greater extent than the Tropos models, for extracting the required information (**Rq2**), we analyzed the answers given to cq3/cq4. In other words, these questions try to understand if the textual requirements specifications were used more heavily when only the Tropos models were available, rather than the Tropos4AS models.

We observed that, according to the answers given to question cq3, the amount of information extracted from Tropos4AS models is (quantitatively) larger than the amount extracted from Tropos. Similarly, the results for question cq4 confirm that the amount of the information extracted from textual requirements is (quantitatively) lower when using Tropos4AS than when using Tropos models. These results, shown in Table V (bottom) and as boxplots in Figure 1 (bottom), are confirmed by the statistical significance (p-value < 0.05) and a large effect size (≥ 0.8).

	Treatment	Experience	Treatment:Experience
Precision	0.07	0.62	0.13
Recall	0.0.34	0.8	**0.08***
F-measure	0.031	0.67	**0.08***

Table VI
CO-FACTOR ANALYSIS (ANOVA TEST)

To cross-check this outcome we further analyzed the received answers by combining cq3 and cq4 (see Section II-G). This analysis confirms the previous findings. In particular, we observed that 10 out of 12 subjects in group GS_1 found the Tropos4AS model sufficient to answer the questions and used only partially the requirements, while only 2 subjects did not find enough information in the Tropos4AS model. This result is statistically significant, with a p-value of the Chi^2-test < 0.004. Conversely, by analyzing GS_2, we observed that only 3 out of 12 subjects found the Tropos model sufficient to answer the questions, while the remaining subjects needed the textual requirements specifications to properly answer the comprehension questions. Also this difference is statistically significant (p-value < 0.021).

Therefore, we can confirm that in the experiment the Tropos4AS model was used to a bigger extent than the Tropos model to extract the required information, and thus reject also the null-hypothesis of **Rq2**.

Summing up, we can answer in an affirmative way to our main research question **Rq**, rejecting the null-hypothesis $\mathbf{H}_0$ with statistical evidence: *By using Tropos4AS models, the comprehensibility of requirements was significantly improved, in comparison to Tropos models.*

C. Co-factors

We investigated the impact of four main co-factors in the experiment: the laboratory (first vs. second), the position of the subject (researchers vs. Ph.D. students), the experience of the subjects in working with Tropos (low vs. high), and the objects of the experiment (WMM vs. PMA). No statistically relevant impact on requirements comprehensibility was observed, with respect to the treatment, thus we do not expect any relevant influence of these factors on the obtained results. Table VI details the results obtained for the interaction between treatment and subject experience, were we observed a small, but not statistically confirmed impact on the experiment. The interaction plot in Figure 2 shows details related to the F-measure and confirms the small interaction between subject experience in working with Tropos and the treatment: subjects with different experience gained different benefits from the use of the requirements modeling methods. Overall, from the interaction plots of Precision, Recall and F-measure we see that (i) for both experienced and non-experienced subjects the performance is lower with Tropos than with Tropos4AS; (ii) the difference of the performance for subjects with low

978-1-4577-1076-6/11/$26.00 ©2011 IEEE

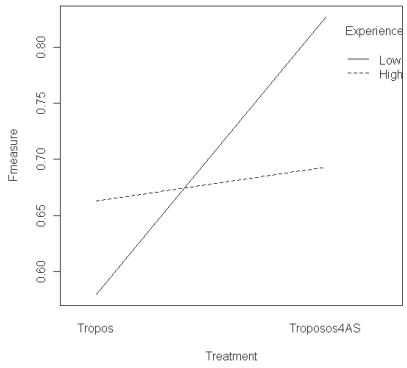

Figure 2. F-measure: Interaction plot for treatment vs. subject's experience

and high experience increases when considering Tropos4AS; and (iii) with the Tropos4AS treatment, subjects with low experience overcame (about +10%) the results obtained by subjects with high experience.

IV. Discussion

Results showed that Tropos4AS models can improve the comprehension of requirements for self-adaptive systems, over conventional Tropos models. Moreover, we also observed that Tropos4AS helps in particular the novice users, who outperformed the performance of expert Tropos modelers. Indeed, the performance of the expert Tropos modelers remained quite stable when using or not using the extensions provided by Tropos4AS.

We believe that the main reason for this is that Tropos4AS adds to the Tropos models the ability of graphically (i.e., semi-formally) specifying relevant details for self-adaptive systems (e.g., conditions raising exceptional behaviors and the goal satisfaction dynamics). Nevertheless, the additional constructs introduced by Tropos4AS bring few more complexity in the models (Table I shows that a Tropos4AS model contains, on average, +15% of structural elements with respect to the corresponding Tropos model), they facilitate requirements comprehension. Subjects, in fact, can reasonably easily extract correct information from such models without having to look-up in the textual requirements specifications, which often result to be complex, generic and ambiguous. This effect remains limited for experts usually more familiar with the interpretation of textual requirements.

However, only a further replication of the experiment, with a specific focus on subject experience, can confirm our observations.

V. Threats to validity

Internal validity concerns factors influencing the independent variable. A paired and balanced design was adopted to mitigate learning effects and group assignment. A preliminary training session was performed to train subjects about the treatments and a preliminary pilot has been conducted with students (not involved in the experiment) to set up questions and check the actual laboratory time. Setting an adequate (maximum) time was important in the experiment to avoid situations in which subjects extract the information only using the textual requirements instead of looking at the model.

Construct validity concerns the relationship between theory and observation. The performed evaluation was mainly based on comprehension questions that have been objectively evaluated, however, some degree of subjectivity involves the construction of the used models and the proposed questions. To mitigate this subjectivity, we constructed the models with the help of two expert modelers (not involved in the experiment). The analysis of the answers, instead, has been performed by mapping the elements of each answer given by subjects with a list of expected "gold standard" elements.

Conclusion validity concerns the relationship between the treatment and the outcome. With the aim of using proper statistical tests to analyze the collected data we used the non-parametric Wilcoxon test and the ANOVA test (which is quite robust with respect to the non-normality of the distribution of the samples). In the comprehension answers evaluation, we tried to limit the use of subjective scores by measuring Precision and Recall considering lists of elements. On the contrary, the post-questions capture subjective feedback. The experiment raw data will be available for the analysis repetitions.

External validity concerns the generalization of the results. The findings we obtained are limited by subjects and objects of the experiment. The subjects were researchers and Ph.D. students. This population is quite representative for a scientific community, but not for industry. With the aim of being able to perform a meaningful experiment, we had to limit to small objects and comprehension tasks. Although, they are not trivial and in fact required a quite high understanding and modeling effort, relative to the size of the experiment. Much more complex objects would not have been treatable in such a study. Finally, we have to remark that the results we obtained are limited to the use of the modeling languages in comprehension activities, thus they cannot be generalized to the whole requirements modeling process;

VI. Related Work

The relatively low impact of the works dealing with self-adaptive systems in premier software engineering conferences can be ascribed in part to the lack of adequate assessments [4]. We start filling this gap, for RE in self-adaptive systems, with this and further planned controlled experiments. Several works document experiments involving modeling techniques and their extensions. For instance, a comprehension study similar to ours has been performed by Ricca et al. [11], which compares the use of the UML

978-1-4577-1076-6/11/$26.00 ©2011 IEEE

class diagrams with one of its extensions for modeling Web applications. Different to ours, this study is based on diagrams recovered from typical Web applications and on comprehension questions objectively evaluated to measure the effectiveness of the diagrams in supporting the application understandability. Their results are in line with ours, showing that the UML extension for the Web domain help especially the novice modelers, while the performance of expert modelers remained quite stable with both languages.

Various works deal with the evaluation of goal-oriented modeling frameworks such as *i** and KAOS. Estrada et al. [7] present the results of an in-depth empirical evaluation of *i** on industrial case studies. The evaluation was carried out over a long time period with three teams of professional requirements engineers and mainly evaluated the coverage of different aspects of the modeling language in-the-field. Reusability and scalability were recognized to be missing aspects, which remain still present in both Tropos and Tropos4AS. Matulevicius et al. [8] compare the full *i** and *KAOS* development processes, with an experiment consisting of three steps: interviewing, creating goal models and evaluating models and languages. Groups of students interviewed persons playing the stakeholders role and delivered the requirements model created with the assigned language, within two weeks. The students then individually evaluated the used language and model of a competing group, filling a questionnaire. Conversely to our experiment, they simulated a requirements analysis process, comparing the modeling languages in the overall process. However, they performed a quasi-experiment (without controlling the context of the experiment) and results are not completely statistically significant, because of the low number of development groups.

VII. Conclusion and Future work

This paper documents an experimental study with subjects, performed to evaluate the impact of the extensions of a requirements engineering method to requirements comprehension. The results showed that Tropos4AS, an extension of Tropos tailored to self-adaptive systems, outperforms its ancestor in comprehending the requirements for such systems. Despite of the increased complexity of the models, it seems that the additional abstractions available help for comprehending the requirements, especially when used by novice requirements engineers. The experiment design can be reused for evaluating the performance of other RE methods extending existing methods for specific domains.

A further experiment which studies the role of Tropos and Tropos4AS in creating models, evaluating the modeling effort and efficiency, was also performed. Future work should consist in replicating the study with a larger number of subjects, involving people form industry. Furthermore, it would be of interest to extend the experiment for evaluating the impact of the design methods through the different phases of the engineering process.

References

[1] G. Antoniol, G. Canfora, G. Casazza, A. D. Lucia, and E. Merlo. Recovering traceability links between code and documentation. *IEEE Transactions on Software Engineering*, 28(10):970–983, 2002.

[2] J. Aranda, N. Ernst, J. Horkoff, and S. Easterbrook. A Framework for Empirical Evaluation of Model Comprehensibility. In *Workshop on Modeling in Software Engineering (MISE07)*, pages 7–13. IEEE, 2007.

[3] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini. Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236, July 2004.

[4] Y. Brun. Improving impact of self-adaptation and self-management research through evaluation methodology. In *Proceedings of Software Engineering for Adaptive and Self-Managing Systems (SEAMS10)*, pages 1–9, 2010.

[5] B. H. C. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, editors. *Software Engineering for Self-Adaptive Systems [outcome of a Dagstuhl Seminar]*, volume 5525 of *Lecture Notes in Computer Science*. Springer, 2009.

[6] H. Estrada, A. Martínez, O. Pastor, J. Mylopoulos, and P. Giorgini. Extending organizational modeling with business services concepts: An overview of the proposed architecture. In *Int. Conf. on Conceptual Modeling*, pages 483–488, 2010.

[7] H. Estrada, A. M. Rebollar, O. Pastor, and J. Mylopoulos. An empirical evaluation of the *i** framework in a model-based software generation environment. In *CAiSE*, pages 513–527, 2006.

[8] R. Matulevicius and P. Heymans. Comparing goal modelling languages: An experiment. In *13th International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2007)*, pages 18–32, 2007.

[9] D. L. Moody, P. Heymans, and R. Matulevicius. Visual syntax does matter: improving the cognitive effectiveness of the * visual notation. *Requirements Engineering*, 15(2):141–175, 2010.

[10] M. Morandini, L. Penserini, and A. Perini. Towards goal-oriented development of self-adaptive systems. In *SEAMS '08: Workshop on Software engineering for adaptive and self-managing systems*, pages 9–16. ACM, 2008.

[11] F. Ricca, M. Di Penta, M. Torchiano, P. Tonella, and M. Ceccato. How developers' experience and ability influence web application comprehension tasks supported by uml stereotypes: A series of four experiments. *IEEE Transactions on Software Engineering*, 36(1):96–118, 2010.

[12] A. van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *RE*, page 249, 2001.

[13] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering: an introduction*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

978-1-4577-1076-6/11/$26.00 ©2011 IEEE