

Assesing the Understandability of Collaborative Systems Requirements Notations: an Empirical Study

Miguel A. Teruel, Elena Navarro, Víctor López-Jaquero, Francisco Montero, Pascual González

LoUISE Research Group, Computing Systems Department
University of Castilla – La Mancha, Albacete, Spain
{miguel, enavarro, victor, fmontero, pgonzalez}@dsi.uclm.es

Abstract— As for single user systems, a proper specification of software requirements is a very important issue to achieve the quality of the collaborative systems. Nevertheless, many of these requirements are from a non-functional nature because are related to the user's need of being aware of other users, that is, the *workspace awareness*. In order to model these special kind of requirements, CSRML, an extension of i^* has been proposed. In this paper, we present a controlled experiment to assess the understandability of this notation compared to i^* . The specification of two different systems was used as experimental material and undergraduate students of Computer Science with an average of two years experience in Requirements Engineering were the experimental subjects.

Keywords- collaborative systems, awareness, empirical study, controlled experiment, understandability, requirements engineering, CSRML, i^*

I. INTRODUCTION

Computer Supported Cooperative Work (CSCW) systems, facilitate that “multiple individuals, situated in different work settings and situations, with different responsibilities, perspectives and propensities, interact and are mutually dependent in the conduct of their work” [1]. These systems have attracted much attention in the last years being most of the work focused on the supporting technology or the social aspects of their development. Unfortunately, no much work has been done about a critical activity of the development process: Requirements Engineering (RE, [2]). Similarly to other systems, this activity should help to develop systems that satisfy users’ expectations. However, while CSCW systems are being developed, additional endeavours have to be made because the high constraints they impose in terms of non-functional requirements (NFRs). Among them, special attention must be paid to the *awareness requirements*, that is, to clearly specify *where* others are working, *what* they are doing *now*, and *what* they are going to do *next* following the Gutwin et al.’s recommendations [3]. However, the specification of this kind of requirements is not a trivial issue, because of the high number and diversity of requirements they are related to, and their high impact in terms of the final architecture of the system. Therefore, the proper selection of the requirement specification technique becomes a challenging and important decision.

Whenever SE technique has to be selected/evaluated, the introduction of the Empirical Software Engineering (ESE, [4]) approach emerges in a natural way as it provides appropriate quantitative and qualitative mechanisms for this aim. For this reason, it has been used to guide this evaluation process. In a previous work [5], we analyzed, by using DESMET [6], which technique, Goal-Oriented (GO) [7], Use Cases [8] or Viewpoints [9] is more suitable to specify the requirements of CSCW systems and found that GO provides more facilities to model the requirements of this kind of systems. Once we determined GO as the most suitable one, we analyzed using DESMET as well which GO proposal deals with CSCW systems in a better way [10]. The proposals analyzed, paying special attention to awareness requirements, were NFR Framework [11], i^* Framework [12] and KAOS [7]. In this study, we found that i^* is the most promising approach to model CSCW systems requirements. However, during this study we detected that it exhibits some problems for the specification of CSCW systems. This led us to extended i^* in order to endow it with certain aspects related to collaboration between users and focusing on awareness characteristics. As result of this extension we developed the Collaborative Systems Requirements Modelling Language (CSRML, [13]).

In order to evaluate whether CSRML is more suitable than i^* for the specification of CSCW systems, a *controlled experiment* has been conducted, on the basis of the guidelines and advices presented in [4,14], that is illustrated in this paper. *Understandability* has been evaluated in this experiment because of its relation with two important quality factors, *usability* and *maintainability*. The first factor is necessary to facilitate the proper use of the notation and its relation to understandability has been stated in standards such as SQUaRE [15]. Maintainability is also a very important factor to reduce the effort of software modification and/or extension and its relation to usability has been previously recognized [16] as well.

This paper is structured as follows. After this introduction, the related works are presented in Section II. In section III, we offer an introduction to i^* and CSRML, our proposal for dealing with requirements engineering in CSCW systems. In section IV, we present the experimental context of this empirical study. Sections V, VI, VII and VIII describe how the experiment has been designed, conducted and analysed, respectively. Section VIII analyses the threats

to validity of the experiment. Finally, in Section IX, conclusions and future works are presented.

II. RELATED WORKS

Empirical Software Engineering (ESE) is gaining more and more attention by the Software Engineering community for validating both emerging and existing proposals. Indeed, it has been already applied to different processes, methods and tools of the development process such as design [17], implementation [18], or testing [19].

Related to the duo ESE-Requirements Engineering (RE), increasingly, the number and variety of works are being presented. For instance, there are works related to the evaluation of different techniques for software requirements inspections [20], techniques for negotiation in distributed software projects [21], techniques for prioritization of requirements [22], the effectiveness of the introduction of screen mockups to improve the understandability of the specification [23], the introduction of metrics for evaluating the completeness and granularity of functional requirements [24], etc. As can be observed all these works are related to different techniques that can be applied during the RE process.

Unfortunately, no much attention has been paid to the evaluation of languages for requirements specification. Some works are oriented to this issue, such as that presented by Celko et al. [25] or Bente et al. [26], but as far as we know, only Al-subaie and Maibaum have carried out an empirical evaluation of Goal-Oriented based methods [27]. This situation is even more necessary if we take into account the need of evaluating RE specification techniques regarding to the domain where they have to be used. In this context is where this work is presented.

III. NOTATIONS OVERVIEW

In this section, a brief introduction about the analyzed notations is presented.

A. i^* (i -Star)

The i^* Framework [2,12] distinguishes between two kinds of elements: objects (Figure 1) and relationships (Figure 2). The objects considered in i^* are:

- An *actor* is a person or a system that has a relationship to the system to be developed. i^* identifies three kinds of actors: *Agent* (an actor who has a concrete physical representation, e.g. a person or a system), *Role* (defines the behaviour of an actor within a specific context, and *Position* (a set of roles that can be typically played by one agent. An agent can play several positions). It is worth noting that an actor can play several roles, and a role can be assigned to multiple actors.
- *Goal*: A goal answers “why?” questions. It describes a certain state in the world that an actor would like to achieve. However, a goal does not prescribe how it should be achieved.
- *Task*: A task specifies a particular way of doing something. Typically a task consists of a number of steps (or sub-tasks) that an actor must perform to execute it.

- *Resource*: A resource is a (physical or informational) entity that the actor needs to achieve a goal or perform a task. The main concern about a resource is whether it is available and from whom.
- *Softgoal*: A softgoal is a condition in the world that the actor would like to achieve, but unlike a (hard) goal, the condition to achieve it is not sharply defined. A softgoal is typically a quality attribute that constrains other elements such as a goal, a task, or a resource. A softgoal is considered to be fulfilled if there is sufficient positive evidence for its fulfilment and little evidence against it.

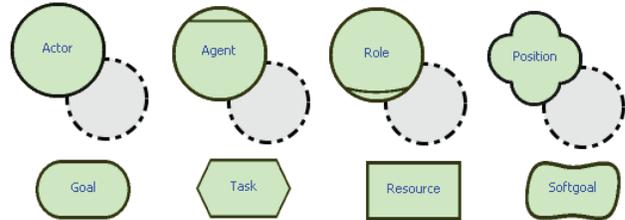


Figure 1. Objects of i^* Framework

The previous objects are related between them through this set of relationships:

- *Dependency*: A dependency in i^* documents a relationship between a *dependor* and a *dependee* are actors. The dependor depends on the dependee for achieving a goal, performing a task, or using a resource. The dependum is the object which the dependee must deliver and which the dependor depends on. It can be a goal, a task, a resource, or a softgoal. If the dependee fails to deliver the required dependum, the dependor’s ability to achieve its own goals is affected. In other words, it becomes difficult or impossible for the dependor to achieve a goal, perform a task, or use a resource. The criteria to determine how to achieve the softgoal are not clearly defined. Typically, the dependee offers several alternatives to achieve the softgoal, and the dependor determines whether the softgoal is achieved or not.
- *Means-end link*: A means-end link documents which softgoals, tasks, and/or resources contribute to achieve a goal. A means-end link also facilitates the documentation and evaluation of alternative ways to satisfy a goal, i.e., different decompositions of goal into subgoals, tasks, and resources.
- *Task decomposition link*: A task decomposition link documents the essential elements of a task. A task decomposition link relates the task to its components, which can be any combination of sub-goals, sub-tasks, resources, or softgoals. The decomposition of a task can thus comprise sub-tasks that must be performed, sub-goals that must be achieved, resources that are needed, and softgoals that typically define quality goals for the task.
- *Contribution link*: A contribution link documents which tasks or softgoals have a positive (+) or negative (-) influence on other softgoals. A contribution link describes whether a task or a softgoal contributes to

satisfy a softgoal positively or negatively. It does not define precisely which kind of support is offered or the extent of the given support.

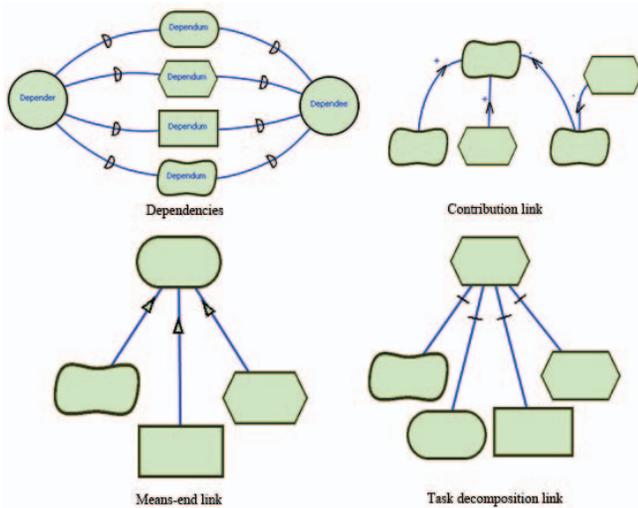


Figure 2. Relationships of the i^* Framework

B. CSRML

In order to deal with the special kind of requirements of CSCW systems, and based on the two studies aforementioned [5,10], CSRML have been developed (Collaborative Systems Requirements Modelling Language [13]). This language is an extension of i^* that includes some elements to model the special features of CSCW systems. The CSRML elements (Figure 3), excluding those whose meaning is the same as in i^* , are:

- **Role:** A role is a designator for a set of related tasks to be carried out. The difference between i^* and CSRML is that an actor playing a role can participate in individual or collaborative tasks (through participation links) and can be the responsible for the accomplishment of a goal (through responsibility links). In addition, the graphical notation is also different from the i^* role (the concept of role/actor boundary is not used in CSRML).
- **Actor:** An actor is a user, program, or entity with certain acquired capabilities that can play a role in executing (using devices) or being responsible for actions [28]. An actor has to play a role (specified by means of a *playing link*) in order to participate in the system.
- **Task:** The concept of task in CSRML is the same as in i^* . They only differ in the introduced notation to define the importance of a task: one, two or three exclamation signs, depending on the importance of the task. Two kinds of CSRML tasks have been identified: (i) *Abstract task*: This kind of task is an abstraction of a set of concrete tasks and, possibly, other elements. We are not able to assign *participation links* directly to this kind of tasks; (ii) *Concrete task*: These are the kind of tasks the participants are involved to. The abstract tasks are refined in these ones. Participants are assigned to concrete tasks by means of participation links. There are

four types of concrete tasks: *Individual task* (a task that an actor can perform without any kind of interaction with other actors) and *Collaboration / Communication / Coordination task* (two or more actors are involved in order to perform any kind of collaboration / communication / coordination among them).

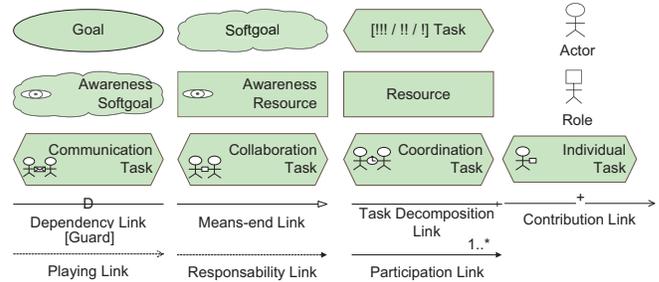


Figure 3. CSRML elements

- **Awareness softgoal:** CSRML refines the i^* concept of softgoal into a new specialization: awareness softgoal, that represents a special need of perception of other user's presence/actions, without which the task the user wants to perform would be negatively affected or even could not be done.
- **Awareness resource:** This special kind of resource corresponds to an implementation or a design solution to accomplish an awareness softgoal.
- **Playing link:** A playing link is used to represent when an actor assumes a role. This link has a guard condition that represents when a role can be played by an actor.
- **Participation link:** A participation link denotes who are involved in a task. This link has an attribute to specify its cardinality, i.e., the number of users that can be involved in a task.
- **Responsibility link:** A responsibility link assigns a role (played by an actor) to a (soft-)goal or a task. This link represents who is the stakeholder responsible for a goal/task accomplishment. It is not necessary that this stakeholder is involved in the goal/sub-task. Nevertheless, if the role is responsible for a goal or task, this role is also responsible for the elements it is decomposed, unless a responsibility link reaches one of the composing elements.

IV. EXPERIMENTAL CONTEXT

In order to assess the *understandability* of CSRML to model the software requirements of CSCW systems, a controlled experiment to compare CSRML with i^* , the requirements language it is based on, was performed based on the guidelines described in [14]. It is worth noting that, *understanding*, according to [29], "is a psychological process related to an abstract or physical object, such as a person, situation, or message whereby one is able to think about it and use concepts to deal adequately with that object". Therefore, this experiment is oriented to evaluate the ability of the subjects to think and use concepts of CSRML/ i^* to deal adequately with a CSCW software requirements.

The Goal Question Metric template [30] was used to define the goal of the experiment as follows: **analyze i^* and CSRML requirements specification language for the purpose of evaluating their understandability from the viewpoint of requirement engineering researchers in the context of undergraduate students.** To address this goal the following null hypothesis was defined:

- H_0 : CSRML has the same understandability than i^* when modelling CSCW systems requirements. H_1 : $\neg H_0$
It is worth noting that this experiment is not isolated, but three previous works have been carried out:
- First, we analyzed which is the RE technique most suitable to deal with CSCW system requirements, comparing Use Cases, Viewpoints and Goal-Oriented [5]. This analysis concluded that Goal-Oriented technique is the most suitable requirement specification approach to model CSCW system requirements.
- Next, an analysis of different Goal-Oriented techniques (NFR Framework, KAOS and i^*) was conducted to identify the most suitable one to be used as the foundation of our proposal [10]. Although none of them satisfied all the expressiveness needs for CSCW system requirements, i^* was found the most powerful one to specify this kind of systems.
- Finally, we extended i^* notation with a set of features related to collaborative tasks specification. The result was CSRML notation.

V. EXPERIMENTAL DESIGN

In this study, we used the variable *understandability* (*Und*) of CSRML requirements models in relation with i^* models. For this aim, we evaluated the ability to understand the experimental material correctly by using the number of correct answers divided by the number of questions, being the student our experimental unit. More specifically, the result of his/her test.

The subjects in this experiment were third-year Computer Science students who had received several Software Engineering courses. Specifically, they were students of a specialized course of Requirements Engineering, so they were familiar with requirements modelling. It is worth noting, at that moment, that they had not studied any Goal-Oriented approach. The other main features of the experiment are outlined in TABLE I.

TABLE I. EXPERIMENT MAIN FEATURES

Null-Hypothesis	H_0 : CSRML obtains the same score for understandability than i^* when modelling CSCW systems requirements. H_1 : $\neg H_0$
Location	Univertisy of Castilla – La Mancha (Albacete, Spain)
Date	May 2011
Subjects	30 Computer Science students (15 Group 1; 15 Group 2)
Dependent Variable	Understandability of requirements modelling notations, measured by <i>Und</i>
Independent Variable	The domain to which the diagrams were related and the use of i^* or CSRML

This experiment consisted in understanding (part of) the requirements model, specified by using both i^* and CSRML, of two different CSCW systems. The first system was an e-

learning system. Specifically, it was selected the jigsaw learning activity, also known as *Experts meeting*, that is a cooperative-learning technique in which students individually do some research in a proposed problem and then they teach each other what they have learned by sharing each individual view of the problem. The second system was a conference review system with collaborative aspects. In particular, we selected the part of the system related to a process review performed in a collaborative way among three reviewers. In addition to the requirements specification (using i^* and CSRML), a set of ten questions for each system was specified that had to be answered by the students with true or false responses. The questions for each system were the same, regardless of the particular language used to specify each system. These questions revolved around collaboration between users, emphasizing about the awareness techniques. An example of both the requirements specification and questions can be seen in Appendix 1¹. In addition, it was decided to ask the students for the time necessary to finish the test, only for statistical purposes.

In order to assign the systems' specification to the students, it was decided to distribute the subjects into two different groups, G1 and G2, avoiding that two subjects belonging to the same group were close. The above mentioned systems were used, using for each one, two conceptually identical diagrams specified with i^* and CSRML. It was decided that G1 will perform the experiment first trying to understand the i^* model of the jigsaw activity and just after that, the CSRML model of the review process. G2 made right the opposite, to understand the jigsaw activity modelled with CSRML, and later the review process specified using i^* . In addition, additional experimental material was elaborated to provide the students with detailed description about i^* and CSRML as well as a brief introduction to workspace awareness. The experiment design is summarized in TABLE II. This process of assigning subjects to 4 different treatments, obtained by combining the dependent variables (System and Language) correspond to a 2x2 factorial design with confounded interaction [31]. Indeed, within a system, the variable language changes together with the group of subjects. So the *learning effect* is cancelled out.

TABLE II. EXPERIMENT DESIGN

		System	
		<i>Jigsaw</i>	<i>Conference</i>
Language	i^*	Group 1	Group 2
	CSRML	Group 2	Group 1

It was also decided those students that met at least one of these criteria would be filtered out:

- The student age exceeds in five years the class mean age.
- The student is not familiar to RE.
- The student have any previous experience with i^* or other Goal-Oriented techniques.

¹ The experimental material can be downloaded from <https://www.dsi.uclm.es/trep.php?codtrep=DIAB-11-06-1>.

TABLE III. EXPERIMENT MEAN AND STANDART DEVIATION (IN PARENTHESES) FOR THE UNDERSTANDABILITY

Domain	Jigsaw		Conference	
Notation	i^* (n=15)	CSRML (n=15)	i^* (n=15)	CSRML (n=15)
Und	0.5733 (0.1163)	0.7933 (0.1335)	0.5933 (0.2154)	0.8200 (0.1512)

TABLE IV. EXPERIMENT ANOVA RESULTS

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0,748166667	1	0,748166667	30,55174842	8,08E-07	4,006873
Within Groups	1,420333333	58	0,024488506			
Total	2,1685	59				

ANOVA calculated by using Excel 2010

Finally, it was decided that if a student wanted to drop out from the experiment, he/she would be interviewed about the cause of his/her abandon and this fact would be noted in the experiment results. Also, any possible interruption of the experiment would be also recorded.

VI. CONDUCTING THE EXPERIMENT

The experiment took place in a single class room. In order to avoid the interaction between the subjects, they were properly distributed in the class room. In addition, the instructor was a professor not teacher not related to the CSRML development to avoid a potential bias of the explanation. All the subjects received an introductory session (about 60 minutes), in which the instructor presented both languages and systems. These explanations were neutral regards to the independent variable (whether using i^* or CSRML). Before giving the models to the students, we gathered the following information from them:

- Previous knowledge about RE experience (in years).
- Previous experience with Goal-Oriented approaches.
- Average mark.
- Age.
- Gender.

In order to increase the motivation and interest of the subjects, the instructor also explained the students that thanks to their participation in the experiment they would half of an extra point in the final mark. He also advised them that their evaluation would not be related to their performance in the experiment so that their answers would be evaluated anonymously. They were also informed that there was not a time limit to carry out the task in order to avoid a possible ceiling effect.

During the experiment, none of the subjects dropped out from it. Only two students asked some questions that were related to the specification of the jigsaw activity with i^* .

VII. RESULTS ANALYSIS

After the experiment, the data were collected in four Excel 2010 tables (one for each specification) and were analysed to exclude any observation that was not complete, or was stated by a subject who satisfied any of the criteria described in section VI, or had not written down any of the requested data. All the data were properly described so that we did not have to discard any observation.

In TABLE III we can see that apparently, the subjects' groups that was given the CSRML models obtained a better score than those who try to understand the i^* one (mean values of 0.7933 and 0.8200 vs. 0.5733 and 0.5933). Despite the clear difference of both systems, the results for the understandability between them were very similar, which means that the system has no influence on the results.

In addition, we also preformed an ANOVA test to refute the null hypothesis H_0 , which is the most appropriate test for exploring the results of a 2x2 factorial design with interaction confounded [31,32]. For an α of 0.05, with $df=(1,58)$, F must be at least 4,006873 to reach a P-value <0.05 , so F score is statistically significant. Then, with a P-value < 0.05 , we can conclude that there is a statistically significant difference between the results obtained with CSRML and i^* . In addition, the difference between both results can be graphically seen in Figure 4.

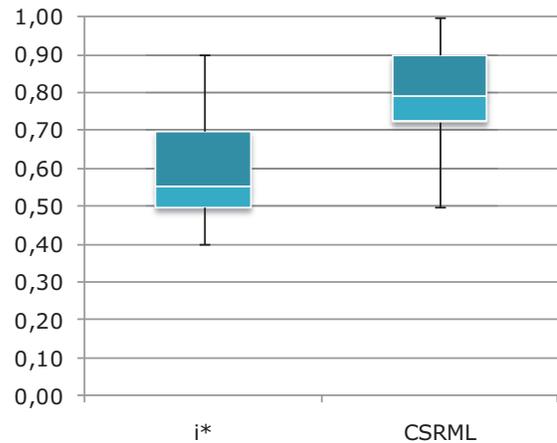


Figure 4. Box plot of obtained results

Once we have shown that CSRML obtains a better score, in the following we analyze which factors contribute to this advantage by analyzing the results grouped by questions. For this aim, the global questions results are shown in two graphs, Figure 5 and Figure 6, to describe the results related to each system.

The first notorious result that can be observed is that CSRML has higher marks than i^* for the three first questions. It is especially relevant that, for the first question,

CSRML obtains a success ratio of 100%. These questions were related to collaboration task, so it can be stated that the collaboration representation is a strongpoint of CSRML, being this a cornerstone for the specification of CSCW systems.

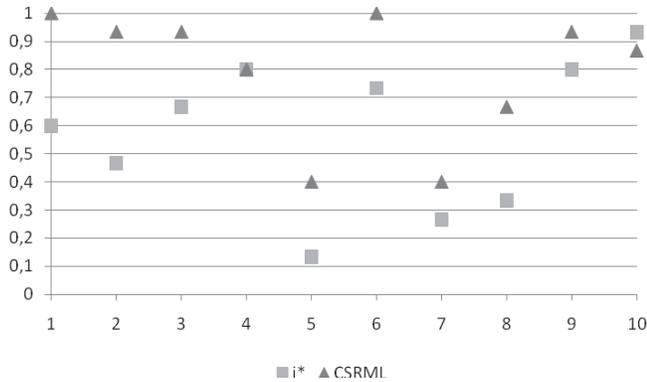


Figure 5. Results for *Jigsaw* questions

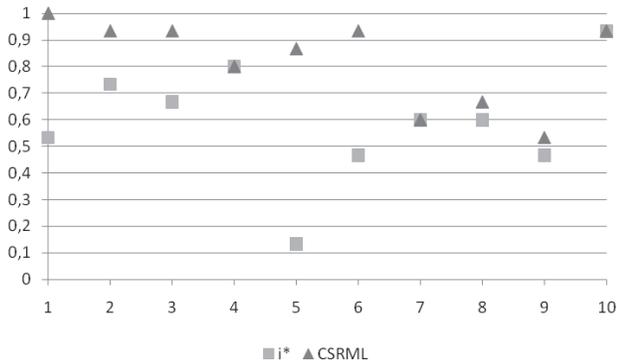


Figure 6. Results for *Conference* questions

Question number four obtains the same result for both notations and domains. This question was related with a task in which only one user is needed to its accomplishment, something easy to represent with both notations.

For questions related with roles (5 and 6), CSRML supersedes i^* one more time, because of its mechanisms to represent when an actor can play a role, and its task's cardinalities representation. The same occurs with awareness questions (7, 8 and 9) due to the new awareness elements.

Nevertheless, the last question obtains a better (or the same for the *Conference* system) result when using i^* . Due to this, in future experiments we will try to analyze the impact of determining the task's importance by using alternative representations.

VIII. THREATS TO THE VALIDITY OF THE EXPERIMENT

We are going to explain in this section the issues that can threaten the validity of this experiment, considering the four types of threats proposed in [4].

A. Conclusion validity

In this experiment, as can be observed in Section VII, the score obtained by the ANOVA analysis is statistically

significant, so it allow us to reject the initial null hypothesis with a large degree of certainly.

B. Internal validity

The number of subjects that participated in the experiment was large enough ($=30$), according to the central limit theory [33], the distribution function of data approximates to a normal distribution, something necessary in order to obtain a statically significant set of data.

Also, the subjects had enough maturity level to participate in the experiment due to their previous coursed subjects of Software Engineering and Requirements Engineering.

Additionally, in order to avoid a possible source of bias, the questionnaire given to the subjects were reviewed by several experts in collaborative systems not related to the CSRML development. They helped us to elaborate the questionnaire taking into account both the main features of CSCW systems and its objectiveness.

C. Construct validity

We used a measure based on correct responses divided by the total number of answers to quantify the understandability of the models. Nevertheless, in further works, we consider to use some formal guidelines such as that presented in CTML [34].

D. External validity

The models presented in this experiment were relatively easy to understand by the students. We will consider increase the complexity of models in further works in which we will use Software Engineering practitioners. In these experiments we will use more industrial-focused domains.

IX. CONCLUSIONS AND FURTHER WORKS

In our previous works [5,10], we found that GO techniques (and especially i^*) can be used to deal with CSCW systems requirements modelling. Nevertheless, we also found that this ilk of specifications suffer from an important lack of expressiveness for some CSCW characteristics such as user collaboration or awareness representation. To address these shortcomings, we have proposed CSRML, an extension of i^* to model CSCW systems requirements.

In this paper we present an experiment to assess the *understandability* of the proposed CSRML notation related to the original i^* notation, which it is based on. The participants in the experiment was thirty Computer Science students that tried to understand two models that belong to two different systems (an e-learning system and conference review system with support for collaboration) based on CSCW systems.

As a result of the study, we have observed that the CSRML notation improve the understandability of requirements model respect to i^* . We have also found some important facts analyzing the students' answers. For example, we found that a CSRML model is more understandable due to its collaboration representation features, the new awareness elements added and the roles

management techniques improved over the original i^* . This fact can be seen in the results of every test's questions (except for the tenth one). Nevertheless, we have found out an important issue when establishing the importance of a task that has to be solved in future versions of the language.

As on-going work, we have performed two replicas of this experiment in the Polytechnic University of Valencia (Spain) and University of La Plata (Argentina) whose results are being analyzed using meta-analysis techniques. We also plan to carry out more experiments with the cooperation of practitioners in industrial contexts.

Also, as a further work, we are planning additional evaluations with new criteria, such as usefulness, correctness or completeness, as in the experiments described in [35] [36]

ACKNOWLEDGMENT

This work has been partially supported by a grant (DESACO, PEII09-0054-9581) from the Junta de Comunidades de Castilla-La Mancha and also by a grant (TIN2008-06596-C02-01) from the Spanish Government. We also want to thank Professor José Eduardo Córcoles Tendero his collaboration as instructor during this experiment.

REFERENCES

- [1] K. Schmidt and L. Bannon, "Taking CSCW seriously," *Computer Supported Cooperative Work (CSCW)*, 1992, pp. 7-40.
- [2] K. Pohl, *Requirements Engineering: Fundamentals, Principles, and Techniques*, Springer, 2010.
- [3] C. Gutwin and S. Greenberg, "A Descriptive Framework of Workspace Awareness for Real-Time Groupware," *Computer Supported Cooperative Work*, vol. 11, 2002, pp. 411-446.
- [4] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering: An Introduction*, 2000.
- [5] M.A. Teruel, E. Navarro, V. López-Jaquero, F. Montero, and P. González, "An Empirical Evaluation of Requirement Engineering Techniques for Collaborative Systems," *15th International Conference on Evaluation and Assessment in Software Engineering*, Durham, United Kingdom: 2011.
- [6] B. Kitchenham, "A methodology for evaluating software engineering methods and tools," *Experimental Software Engineering Issues: Critical Assessment and Future Directions*, H. Rombach, V. Basili, and R. Selby, eds., Springer Berlin / Heidelberg, 1993, pp. 121-124.
- [7] A. van Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour," *Proceedings of the Fifth IEEE International Symposium on Requirements Engineering*, IEEE Computer Society, 2001, pp. 249-263.
- [8] A. Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2000.
- [9] A. Finkelsetin, J. Kramer, B. Nuseibeh, L. Finkelstein, and M. Goedicke, "Viewpoints: A Framework for Integrating Multiple Perspectives in System Development," *International Journal of Software Engineering and Knowledge Engineering*, vol. 2, 1992, pp. 31-57.
- [10] M.A. Teruel, E. Navarro, V. López-Jaquero, F. Montero, and P. González, "A Comparative of Goal-Oriented Approaches to Modelling Requirements for Collaborative Systems," *6th International Conference on Evaluation of Novel Software Approaches to Software Engineering*, Beijing, China: 2011.
- [11] L.M. Cysneiros and E. Yu, *Non-Functional Requirements Elicitation (Perspectives on Software Requirements)*, Springer, 2003.
- [12] J. Castro, M. Kolp, and J. Mylopoulos, "A requirements-driven development methodology," *In Proc. of the 13th Int. Conf. On Advanced Information Systems Engineering (CAiSE'01)*, 2001, pp. 108-123.
- [13] M.A. Teruel, E. Navarro, V. López-Jaquero, F. Montero, and P. González, "CSRML: A Goal-Oriented Approach to Model Requirements for Collaborative Systems," *30th International Conference on Conceptual Modeling (ER 2011)*, Brussels, Belgium: Springer Berlin Heidelberg, 2011.
- [14] B. a Kitchenham, S.L. Pfleeger, L.M. Pickard, P.W. Jones, D.C. Hoaglin, K. El Emam, and J. Rosenberg, "Preliminary guidelines for empirical research in software engineering," *IEEE Transactions on Software Engineering*, vol. 28, Aug. 2002, pp. 721-734.
- [15] ISO/IEC, *ISO/IEC 25010:2011, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*, 2011.
- [16] L.C. Briand, C. Bunse, and J.W. Daly, "A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs," *IEEE Transactions on Software Engineering*, vol. 27, Jun. 2001, pp. 513-530.
- [17] J.A. Cruz-Lemus, A. Maes, M. Genero, G. Poels, and M. Piattini, "The impact of structural complexity on the understandability of UML statechart diagrams," *Information Sciences*, vol. 180, Jun. 2010, pp. 2209-2220.
- [18] A. De Lucia, F. Fasano, R. Oliveto, and G. Tortora, "Can Information Retrieval Techniques Effectively Support Traceability Link Recovery?," *14th IEEE International Conference on Program Comprehension (ICPC'06)*, IEEE, 2006, pp. 307-316.
- [19] G. Canfora, A. Cimitile, F. Garcia, M. Piattini, and C.A. Visaggio, "Evaluating advantages of test driven development," *ACM/IEEE International Symposium on Empirical Software Engineering (ISESE '06)*, New York, New York, USA: ACM Press, 2006, p. 364.
- [20] K. Sandahl, O. Blomkvist, J. Karlsson, C. Krysander, M. Lindvall, and N. Ohlsson, "An Extended Replication of an Experiment for Assessing Methods for Software Requirements Inspections," *Empirical Software Engineering*, vol. 3, 1998, pp. 327-354.
- [21] D. Damian, "An empirical study of requirements engineering in distributed software projects: is distance negotiation more effective?," *8th Asia-Pacific Software Engineering Conference (APSEC'01)*, IEEE Comput. Soc, 2001, pp. 149-152.
- [22] L. Karlsson, P. Berander, B. Regnell, and C. Wohlin, "An Experiment on Exhaustive Pair-Wise Comparisons versus Planning Game Partitioning," *8th Conference on Empirical Assessment in Software Requirements Prioritisation*, Edinburgh, UK: 2004.
- [23] F. Ricca, G. Scanniello, M. Torchiano, G. Reggio, and E. Astesiano, *On the effectiveness of screen mockups in requirements engineering*, New York, New York, USA: ACM Press, 2010.
- [24] S. España, N. Condori-Fernandez, A. Gonzalez, and Ó. Pastor, *Evaluating the Completeness and Granularity of Functional Requirements Specifications: A Controlled Experiment*, IEEE Computer Society, 2009.
- [25] J. Celko, J.S. Davis, and J. Mitchell, "A demonstration of three requirements language systems," *ACM SIGPLAN Notices*, vol. 18, Jan. 1983, pp. 9-14.

[26] B. Anda, D. Sjøberg, and M. Jørgensen, "Quality and Understandability of Use Case Models," *15th European Conference on Object-Oriented Programming (ECOOP 2001)*, Springer Berlin Heidelberg, 2001, pp. 402-428.

[27] H.S.F. Al-Subaie and T.E. Maibaum, "Evaluating the Effectiveness of a Goal-Oriented Requirements Engineering Method," *Fourth International Workshop on Comparative Evaluation in Requirements Engineering (CERE'06)*, IEEE Computer Society, 2006, pp. 8-19.

[28] M. Noguera, M. González, J.L. Garrido, V. Hurtado, and M. Rodríguez, "System Modeling for Systematic Development of Groupware Applications," *International Conference on Software Engineering Research and Practice*, 2006, pp. 750-756.

[29] C. Bereiter, *Education and Mind in the Knowledge Age*, Routledge, 2002.

[30] V.R. Basili, G. Caldiera, and H.D. Rombach, "The Goal Question Metric Approach," *Encyclopedia of Software Engineering*, vol. 2, 1994, p. 528-532.

[31] B.J. Winer, D.R. Brown, and K.M. Michels, *Statistical Principles in Experimental Design*, McGraw-Hill, 1991.

[32] R.E. Kirk, *Experimental Design: Procedures for Behavioral Sciences*, Pacific Grove, CA: Brooks/Cole, 1995.

[33] C.M. Grinstead and L.S. Snell, *Introduction to Probability*, 2006.

[34] R.E. Mayer, *Multimedia Learning*, Cambridge University Press, 2001.

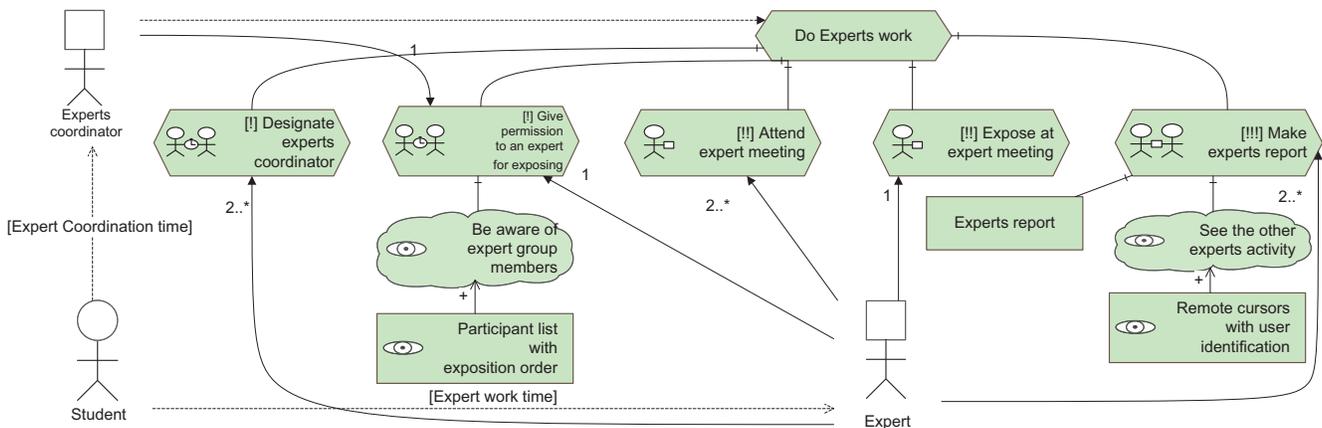
[35] A. Gross and J. Doerr, *EPC vs. UML Activity Diagram - Two Experiments Examining their Usefulness for Requirements Engineering*, IEEE, 2009.

[36] I. Menzel, M. Mueller, A. Gross, and J. Doerr, *An Experimental Comparison Regarding the Completeness of Functional Requirements Specifications*, IEEE, 2010.

APPENDIX 1: EXPERIMENTAL MATERIAL²

A. An Example of an understanding task (test 1)

The following CSRML diagram describes a jigsaw learning activity, a cooperative-learning technique in which students individually do some research in a proposed problem and then they teach each other what they have learned by sharing each individual view of the problem.



WRITE DOWN STARTING TIME (HH: MM: SS) _____

Answer the following True/false statements:

1. The task "Designate experts coordinator" is a collaborative task. _____
2. The task "Expose at team meeting" is a collaborative task _____
3. Two different roles are necessary to perform the task "Give permission to an expert for exposing" _____
4. Two different roles are necessary to perform the task "Attend experts meeting" _____
5. A "Student" actor can play the "Expert" role anytime _____
6. Only a "Expert" is needed to "Make experts report" _____
7. "Participant list with exposition order" is an implementation of "Be aware of expert group member" softgoal _____
8. "Experts report" is an implementation of "Make experts report" _____
9. "See the other users activity" is a softgoal related with awareness. _____
10. "Designate experts coordinator" task is more important than "Expose at experts meeting" task. _____

² The experimental material can be downloaded from <https://www.dsi.uclm.es/trep.php?codtrep=DIAB-11-06-1>