Pre-print version

Please cite as:

Zubcoff, J., Garrigós, I., Casteleyn, S., Mazón, J. N., Aguilar, J. A., & Gomariz-Castillo, F. (2019). Evaluating different i*-based approaches for selecting functional requirements while balancing and optimizing non-functional requirements: A controlled experiment. Information and Software Technology, Vol. 106, 68 – 84

Evaluating Different i*-based Approaches for Selecting Functional Requirements while Balancing and Optimizing Non-Functional Requirements: A Controlled Experiment

Jose Zubcoff[1], Irene Garrigós[1], Sven Casteleyn[2], Jose-Norberto Mazón[1], Jose-Alfonso Aguilar[3], Francisco Gomariz-Castillo[1]

[1] *University of Alicante, Spain*

[2] *University Jaume I, Spain*

[3]*Universidad Autónoma de Sinaloa, México*

Abstract

Context: A relevant question in requirements engineering is which set of functional requirements (FR) to prioritize and implement, while keeping non-functional requirements (NFR) balanced and optimized.

Objective: We aim to provide empirical evidence that requirement engineers may perform better at the task of selecting FRs while optimizing and balancing NFRs using an alternative (automated) i* post-processed model, compared to the original i* model.

Method: We performed a controlled experiment, designed to compare the original i* graphical notation, with our post-processed i* visualizations based on Pareto efficiency (a tabular and a radar chart visualization). Our experiment consisted of solving different exercises of various complexity for selecting FRs while balancing NFR. We considered the efficiency (time spent to correctly answer exercises), and the effectiveness (regarding time: time spent to solve exercises, independent of correctness; and regarding correctness of the answer, independent of time).

Results: The efficiency analysis shows it is 3.51 times more likely to solve exercises correctly with our tabular and radar chart visualizations than with i*. Actually, i* was the most time-consuming (effectiveness regarding time), had a lower number of correct answers (effectiveness regarding correctness), and was affected by complexity. Visual or textual preference of the subjects had no effect on the score. Beginners took more time to solve exercises than experts if i* is used (no distinction if our Pareto-based visualizations are used).

Conclusion: For complex model instances, the Pareto front based tabular visualization results in more correct answers, compared to radar chart visualization. When we consider effectiveness regarding time, the i* graphical notation is the most time consuming visualization, independent of the complexity of the exercise. Finally, regarding efficiency, subjects consume less time when

using radar chart visualization than tabular visualization, and even more so compared to the original i* graphical notation.

Keywords Controlled experiment, i*, requirements engineering, Pareto efficiency.

1. Introduction

Several studies have shown that effective requirements engineering (RE) is a critical success-factor in software projects, e.g. (Verner et al. 2005; Nasir et al., 2011). Generally, two types of requirements are discerned: functional requirements (FRs) that describe the system services, behaviour or functions to be provided, and non-functional requirements (NFRs), which include (quality) attributes or constraints on the application to build or in the development process (Glinz. 2007). Taking NFRs into account while elaborating the FRs from the early design phases significantly improves the end-user satisfaction (Ameller et al. 2010). Approaches that incorporate goal-oriented techniques, such as i* (Yu, 1997) used as a framework in this article, are an ideal candidate for this purpose, as they explicitly represent FRs (as tasks), and NFRs (as softgoals), and allow to denote the impact of FRs on NFRs (using contribution links). Like other modelling languages, i* has an accompanying graphical notation, which enables modellers to easily create and communicate requirement models, and gain an overall insight in the requirements of the system. Despite the importance of the graphical notation, and its effect on model interpretations (e.g., Nordbotten and Crosby 1999), very few empirical evidence is available regarding their efficiency and effectiveness (see Related Work), nor works comparing different visualizations that consider different purposes to interpret a model[1].

A pertinent challenge in requirements engineering is to identify an optimal subset of all identified FRs to implement, within the scope of available resources, that satisfy the demand of customers (Zhang et al. 2008). This requires requirements engineers to correctly interpret and compare models varying in the set of FRs they define, while maintaining a balanced satisfaction of NFRs according to priorities determined by the user. This is important in several contexts: during the requirement specification and negotiation phase, where requirements are agreed

[1] According to (Atkinson & Kuhne, 2003), models have two main characteristics: on one hand, concepts available for creating models and the rules governing their use (i.e., abstract syntax), and on the other hand the notation to use in depicting models (i.e., concrete syntax). For the sake of clarity, from now on, in this paper we use "model" when we refer to the abstract syntax of models, while "visualization" is used when we refer to concrete syntax of models.

upon between developers and clients within time and budgetary constraints, and ensuring compliance with the priorities of the clients in terms of NFRs; during the requirements elicitation and analysis process, when FR alternatives need to be compared and decided upon, with respect to NFRs; and in incremental development, such as according to the SCRUM Agile methodology (Schwaber and Beedle 2002), where subsets of requirements are chosen in each iteration ("sprint") to implement; when conceiving a new release of existing software, where a list of new features (FRs) needs to be chosen to include in the next release.

To tackle this challenge, we introduced in our previous work (Aguilar et al. 2011) a method based on Pareto efficiency (Szidarovszky et al. 1986), which post-processes, in an automatable way, an i* model and produces two Pareto-front based visualizations: a tabular and a radar chart. This helps designers to make informed decisions by understanding the trade-offs that are necessary to obtain a well-balanced, optimized NFRs satisfaction, and allow them to more easily prioritize FRs. Recognizing the importance of model visualization (Nordbotten and Crosby, 1999), and considering that the original i* graphical notation was not developed with comparing and balancing alternative requirements specifications in mind, we developed two custom visualizations to support our Pareto efficiency-based model: a tabular (Aguilar et al. 2011) and radar chart (Aguilar et al. 2012) visualizations, that both capture NFR optimization while allowing to easily compare different subsets of FRs to implement.

In this article, we shortly recap our Pareto efficiency-based RE approach to compare subsets of FRs while balancing NFRs, and present an extensive experimental evaluation of the original i* graphical notation (based on the original i* model) and two novel visualizations (based on our i* post-processed model) (tabular and radar chart visualization). The objective is to determine which visualization is better (regarding efficiency and effectiveness) under which conditions, and provide supporting empirical evidence. We hereby focus on efficiency (time spent to correctly answer solve a problem), and the effectiveness (regarding time: time spent to solve a problem, either correct, partially correct or wrong; and correctness of the solution, independent of time) of these visualizations when being used by the designers. The evaluation was set up as a controlled experiment consisting of a set of modelling exercises with two levels of complexity. The level of expertise of the designers participating in the experiment was heterogeneous (i.e. beginners and experts in goal-oriented modelling).

Moreover, due to the different types of visualizations (i.e., graphical and tabular), it is important to study the relation between the learning style of the subjects and the notation used. We have determined the learning style of the subjects by performing a Felder test (Felder and Silverman, 1988).

Specifically, contributions of this article are as follows: (i) an empirical comparison between our Pareto front model (and two different visualizations based upon the model), and the original i* model (with the corresponding original i* graphical notation), thus showing the convenience of using an i* postprocessed model for solving specific tasks (in our case, selecting FRs while balancing and optimizing NFRs); and (ii) a detailed discussion on the results of our empirical evaluation, thus giving insight in the variables and conditions under which visualization performs better (or worse) and considering the level of complexitiy of the models (i.e., complementing qualitative evidence found in literature with additional quantitative evidence of the performance of the original i* model under different complexities) and other features such as relation between the learning style and the visualization used.

The remainder of this paper is structured as follows: Section 2 describes related work. Section 3 shortly describes the three types of visualizations evaluated by means of an example. Section 4, describes the experimental methodology to validate our hypotheses. The analysis of the data obtained is shown in Section 5. Section 6 presents the threats to validity of the experiments. In section 7 the results obtained are discussed. Finally, the conclusion and future work are presented in Section 8.

2. Related work

Requirements Engineering (RE) approaches include mechanisms to help designers to understand the trade-offs that are necessary to prioritize FRs while optimizing NFRs satisfaction.

Regarding FR prioritization, in (Salado and Nilchiani, 2015), the authors propose an Adaptive Requirements Prioritization (ARP) method that improves decision making between conflicting functional requirements using the principles of multidimensionality. Its efficiency is evaluated using Monte Carlo simulations for a variety of priority dimensions and priority levels. Bridging simulation and real-world experiments, an interesting work is the one proposed in (Benestad and Hannay, 2012), where an artificial and a field experiment are performed using the

same prioritization techniques to assess whether they affect stakeholders' selection of software product features. Furthermore, in (Duan et al, 2009), the authors present an approach for automating a significant part of the prioritization process. The method applies data-mining and machine learning techniques to prioritize functional requirements according to stakeholders' interests, business goals, and cross-cutting concerns such as security or performance requirements. Recent efforts in order to simplify the prioritization process are focused in an algorithm to prioritizing FRs in incremental software development model according to dependency relationships between requirements (Alzyoudi et al, 2015). In (Jackson, 1999) the author addresses the importance of theoretically sound and practical methods for classifying and prioritizing product requirements by developing a structured approach to gather, analyze, and aggregate stakeholder input. In (Shannin and Zairi, 2009) the authors present how the Kano model and a questionnaire are used for classifying and prioritizing customer requirements in the international airlines industry.

There are also efforts that focus on NFRs, and more specifically on optimizing and balancing them. For example, in (Broster and Coombes, 2011) some of the challenges of measuring performance and timing behavior of reliable embedded systems are shown. Additionally, this work explains techniques and strategies for optimization of software reliability and compares different techniques for measuring and analyzing software including tracing methods, in-memory analysis and using hardware support. It shows how those techniques can be used for verification of non-functional properties, such as satisfying the requirements for safety in automobiles in the ISO26262 standard. This work is interesting since it introduces optimization at a high and low level, studying strategies and trade-offs that occur in reliable software development. To do this, the authors introduce a process that supports optimization with the idea that it is possible to have the maximum benefit for the minimum effort. In the Model-Driven Development (MDD) field, in (Xuan and Geihs, 2010) the authors propose an approach to address optimizing NFRs through trade-offs by combining MDD with Evolutionary Algorithms (EA). The framework proposed in (Douglas, 2010) defined 11 major relationships between FRs and NFRs focused on cost, risk, schedule, communication, and quality perceptions. A dynamic decision-making infrastructure to support both NFRs representation and monitoring, and to reason about the degree of satisfaction during runtime is presented in (Almeida et al, 2015). The infrastructure is composed of an extended feature model aligned with

a domain-specific language for representing NFRs to be monitored at runtime; a monitoring infrastructure to continuously assess NFRs at runtime and a flexible decision-making process to select the best available configuration based on the satisfaction degree of the NFRs. Therefore, this work allows to quantify the level of satisfaction with respect to NFRs specification.

In several situations, post-processing of modeled requirements may be beneficial to better interpret the model instances for a particular purpose. This is important in order to know how to visualize, in the best form, the stakeholders' needs to obtain an optimal final product. There are several approaches that post-process requirements models for a particular purpose. E.g. in (Buarque et al 2013) a new approach is introduced, called OOM-NFR, which processes initial requirements expressed in terms of i* diagrams to get OO-Method conceptual models (Pastor et al 2001) that allow designer to easier consider both FRs and NFRs in order to define the appropriate configuration of the application to be generated. Also, in (Abirami et al 2015), the authors present a framework for post-processing of RE models in order to automatically detect and segregate the FRs and NFRs, thus obtaining an improved conceptual model with more information about NFRs in order to be considered in later stage of development (e.g., for defining constraints).

For the sake of NFRs optimization, two important dimensions of requirement engineering are visualization of requirements, as stated by (Pohl, 2013); they form a relevant research topic (see e.g. the survey of (Cooper et al. 2009)). There are some papers that deal with approaches for visualizing NFRs trade-off. In (Rahimi et al. 2014), authors focus on adequately capturing NFRs such as security, performance, and usability, and present a data mining approach for automating the extraction and subsequent modelling and visualization of NFRs from requirement documents. (Zhang et al. 2008) explain, in their position paper, advantages of using Pareto front to identify optimal choices and trade–offs for stakeholders once an initial set of requirements has been gathered. A survey on Pareto-optimal Search-Based Software Engineering is presented in (Sayyad and Ammar 2013). Finally, (Zhang et al. 2011) highlight the problem for requirement engineers to find a set of requirements that reflect the needs of several different stakeholders, while remaining within budget. The authors introduce and evaluate two multi-objective evolutionary optimization algorithms for the automated analysis of requirements assignments when multiple stakeholders are to be satisfied by a single choice of requirements. In this paper, the authors use radar charts to

illustrate the tensions between the stakeholders' competing requirements in the presence of increasing budgetary pressure, i.e., they are used as a visualization mechanism to easily understand trade-offs between budget and satisfaction of stakeholders.

However, although the literature provides a large number of works related to techniques for visualization of requirements, less effort has been reported on empirical evaluation of different visualizations of requirements. Only some works did a limited theoretic analysis using visual notation theory of respectively the KAOS (Matulevičius et al. 2007) and i* (Moody et al. 2009) visual notations. The authors identify some weaknesses, and propose improvements, such as the introduction of a mnemonic colour scheme.

Next to the original graphical notations, several alternatives or extensions have been proposed, addressing various aspects of RE. In this respect, the workshop series on Requirements Engineering Visualization (REV) produced several interesting ideas and proposals. For example, (Feather et al. 2006) illustrate the use of software visualization techniques applied to requirements engineering, and propose the use of several visualization techniques (bar charts, tree maps, Kiviat charts) for various RE concerns (e.g., listing requirements, displaying risk). (Gotel et al. 2007) describe some visualization techniques to enumerate requirements along with some quality attributes, e.g., a smiley face visualization denoting if requirements' necessary data is present, or a volcanic world visualization denoting requirements stability. Heim et al. (2009) propose a graph-based visualization to better show the relationships between requirements. (Gabrysiak et al. 2009) combine formal requirements models and prototyping in the form of scenario-based prototyping, which enables the interactive visualization for elicitation and validation of requirements in the business domain by means of a simulation and animation. The prototype tool allows stakeholders to visualize the FRs by means of a simulation and animation; NFRs are not considered. As far as we can verify, all these proposals remained ideas, in few cases accompanied with a prototype tool, but never validated or evaluated.

Also in other contexts, several techniques and tools have been proposed for requirements visualization. PaladinRM (Austin et al. 2006) is a tool for requirements visualization established by the NASA Goddard Space Flight Center. Requirements are organized into layers for team development and graph structures are used to describe compliance of requirements, and define relationships among requirements. In (Horkoff et al. 2010), the authors developed

an approach for visualizing reasoning through i* models in order to help analysts in understanding conflicts among alternatives (e.g., goals with conflicting paths). These visualization mechanisms were tested with some case studies that suggest that further visualization mechanisms could support analysis. In (Ernst et al. 2006), the authors present graphical and textual annotations in i* diagrams to denote four quality attributes (NFRs): degree of certainty, feasibility, trustability and performance of a goal. For example, the degree of trust is denoted by thickness of delegation links. A theoretical evaluation of this visualization technique is presented. In the context of the object-oriented analysis (OOA) method, (Gemino 2004) performed an empirical validation to assess the effectiveness of animations and narrations to complement textual descriptions and static OOA diagrams when validating requirements, and concluded that in particular the latter might have a positive effect.

In conclusion, while several efforts have been done to improve requirement visualization at different stages and for different purposes in the RE process, there is a clear lack of rigorous, empirical evaluations of the proposals. To the best of our knowledge, there are no empirical studies that show how the cycle: prioritizing, post-processing and visualization of FRs considering NFRs has been integrated in one framework, or compare different visualizations with a single purpose in mind (in our case: NFR optimization). This gap is bridged in this article by conducting a controlled experiment on different visualizations for NFR trade-offs: one visualization is based on i* model, while the remaining visualizations are based on our post-processed Pareto efficiency-based model (a tabular and a radar chart visualization).

3 Goal oriented Requirements models

In this section, we shortly describe the two i* models used in this article, the original i* model and our post-processed Pareto efficiency-based model, and focus on their visualization that have been evaluated in the experiment, by means of an example: the original i* diagram for the former, and a tabular and radar chart visualization for the latter.

3.1. I* modeling

The i* modeling framework is a goal-oriented requirements engineering (GORE) technique that incorporates social analysis by modeling the relationships between

different actors. It consists of two models: the strategic dependency (SD) model to describe the dependency relationships among various actors in an organizational context, and the strategic rationale (SR) model, used to describe actor interests and concerns and how they are addressed. The SR model provides a detailed way of modeling internal intentional elements and relationships of each actor. Intentional elements are goals, tasks, resources and softgoals (see Figure 1 (A)). Intentional relationships are means-end links representing alternative ways for fulfilling goals; task-decomposition links representing the necessary sub-components for a task to be performed; or contribution links in order to model how an intentional element contributes to the satisfaction or fulfilment of a softgoal (see Figure 1 (B)). Possible labels for a contribution link are “Make”, indicating complete satisfaction of a softgoal, “Some+”, indicating a strong positive contribution and “Help”, indicating a smaller positive contribution; their negative counterparts are “Break”, “Some-”, “Hurt”. Finally, the label “Unknown” indicates an unknown (positive or negative) strength of the contribution.

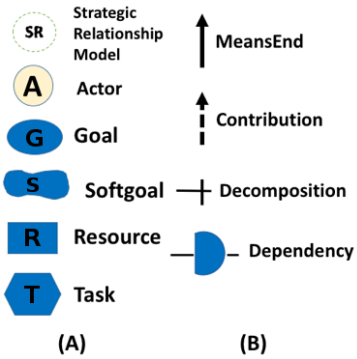

Fig. 1. Graphical notation of i* model.

Figure 2 shows an example of a simple i* diagram from our experiment. This i* diagram describes a system for managing surveys, having a main goal "Survey to be performed", which can be achieved by means of three FRs (“tasks” in i* terminology), in this case navigational requirements: “Interactive interview”, “Perform interview”, “Private questionnaire”. Each of these tasks is further decomposed into sub-tasks and required resources, and affects one or more NFRs. For example, “Interactive interview" is decomposed into sub-tasks "Establish chat

connection" and "Perform interview", it requires the resource "SurveyRepository", and it helps usability and hurts reliability.

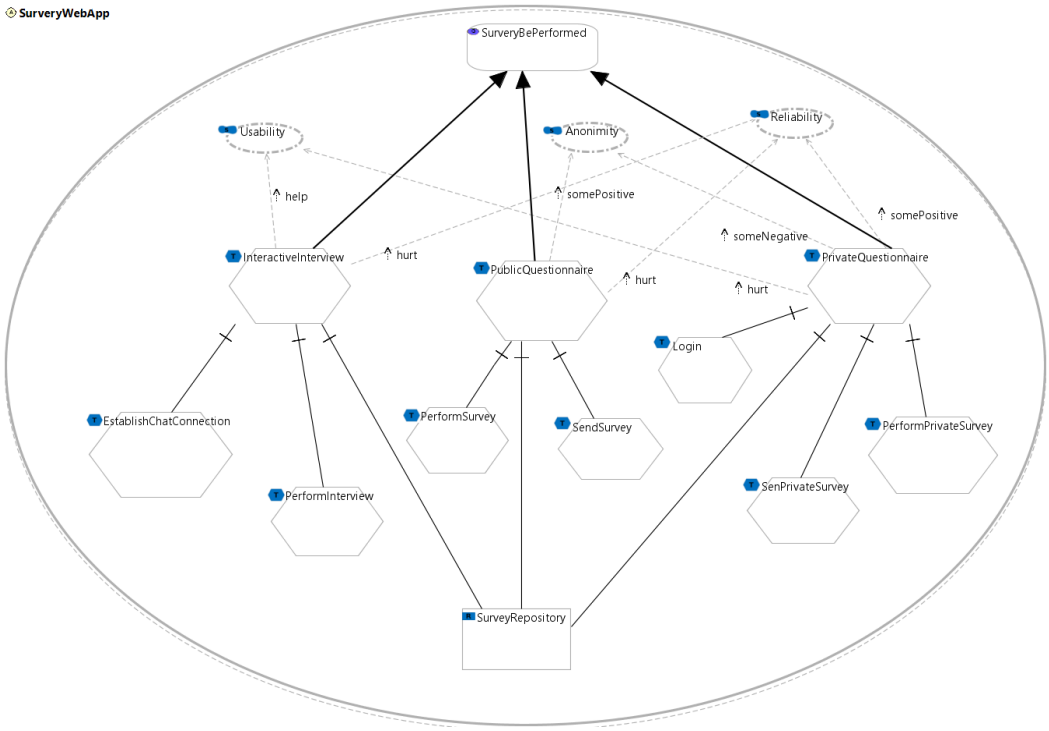

Fig. 2. Visualization of an example i* model with Complexity “simple” and type “i* diagram”

3.2.Tabular visualization of Pareto Front

Our Pareto-based approach assists requirement engineers to evaluate and prioritize FRs while NFRs trade-off improves the application’s quality. To do so, our approach builds upon the goal-oriented RE approach i*, with the aim of supporting and improving it using Pareto efficiency. The Pareto efficiency algorithm is based on computing the Pareto front, which is useful when there are multiple competing and conflicting objectives that need to be balanced (Sayyad & Ammar 2013). The Pareto efficiency is a notion from economics widely applied to engineering, which can be described as follows: “given a set of individuals, a set of alternative allocations, and a set of allocation-dependent valuations, an allocation A is an improvement over allocation B only if A can make at least one particular valuation better than B, without making any other worse”. Intuitively, the Pareto front is the set of allocations that cannot uniformly be improved.

Applying this principle to our setting, the set of individuals refers to the set of FRs, the alternative allocations corresponds with sets of FRs in a certain state (implemented or not implemented), i.e. a configuration, and improving a

particular valuation refers to better satisfying a particular NFR. Therefore, a Pareto front configuration is an allocation of states for FRs (implemented or not implemented) so that no other configuration better satisfies any single NFR, while satisfying the other NFRs equally. The Pareto front typically contains several configurations (i.e., subsets of implemented requirements), each representing an optimal trade-off between the NFRs. While individual NFR may be satisfied better or worse in the different Pareto front configurations, none of the Pareto front configurations can be changed without negatively affecting at least one NFR. The Pareto front can thus be used by requirement engineers to assess the impact of implementing requirements on individual NFRs, and allows them to make a well-informed decision which of the well-balanced configurations best satisfies all NFRs as prioritized by the stakeholders.

Finding the set of Pareto optimal configuration can be defined as the problem of finding a (decision) vector of decision variables X (i.e., a valid implemented/not implemented requirements configuration), which maximizes a vector of M objective functions fi(X) (i.e., the satisfaction of softgoal i in configuration X) where i = 1..M (with M the amount of softgoals). To do so, the concept of domination between vectors is defined as follows: a decision vector X is said to dominate a decision vector Y (also written $X \succ Y$) if and only if their corresponding objective vectors of objective functions fi(X) and fj (X) satisfies:

$\forall i \in \{1...M\}: fi(X) \geq fi(Y)$ and $\exists i \in \{1...M\}: fi(X) > fi(Y)$, it is then said that all decision vectors that are not dominated by any other decision vectors form the Pareto optimal set, while the corresponding objective vectors are said to form the Pareto front. For a more detailed explanation of the Pareto algorithm we refer the reader to our previous work (Aguilar et al. 2011; Aguilar et al. 2012).

Table 1 shows the tabular visualization of the Pareto front of the goal-oriented requirements model instance for the system for managing surveys example, introduced in Fig. 2. Three FRs are shown as columns (R0 = PublicQuestionnaire; R1 =PrivateQuestionnaire; R2 = IntertactiveInterview), and the NFRs (reliability, anonymity, usability) as the remaining columns. Different configurations are shown as rows. For each configuration, an “I” in a cell means the corresponding FR was implemented, an N means it was not implemented. The cells containing numerical values correspond with the sum of all the contribution links for that NFR in this configuration, where “make” contributes +4, “some+” contributes +2 and “help” contributes +1. Negative contribution links are correspondingly

negatively graded. The complete specification of the experiment is available at https://github.com/josezubcoff/soft_expt to allow its replication.

Table 1. An example of a tabular chart visualization of a Pareto front model: Complexity "simple" and type "tabular" (corresponding to the i* diagram in Fig. 2).

Config.	R0	R1	R2	Reliability	Anonymity	Usability
X2	I	I	N	1	0	-1
X3	I	N	I	-2	2	1
X4	I	N	N	-1	2	0
X5	N	I	I	1	-2	0
X6	N	I	N	2	-2	-1
X7	N	N	I	-1	0	1
X8	N	N	N	0	0	0

3.3. Radar chart visualization of Pareto Front

Figure 3 shows an example of a radar chart visualization of a Pareto front from a i* model for a movie provider. A radar chart is equivalent to a tabular Pareto front visualization (i.e., it is based on the same underlying model), whereas NFRs are shown as axes in the radar chart (S0 = reliability; S1 = anonymity; S2 = usability), and configurations (of implemented and not-implemented FRs) are shown as colored plots on the axes, where each particular plot on an axis denotes the total contribution of this configuration for this particular NFR. Figure 3 shows configurations X2 – X5 as a radar visualization of the Pareto front of the goal-oriented requirements model instance for the system for managing surveys example, introduced in Fig. 2.

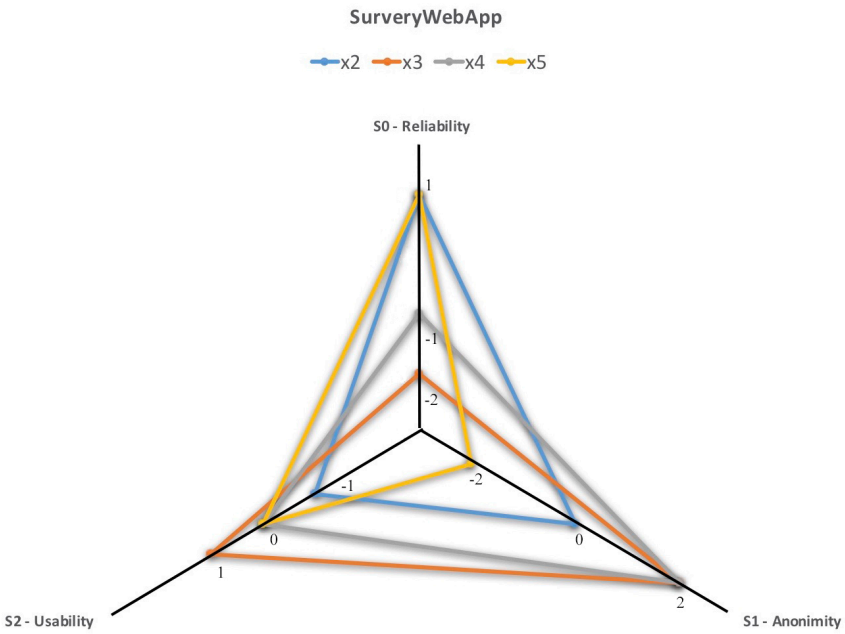

Fig. 3. An example of a radar chart visualization of a Pareto front model: Complexity "simple" and type "tabular" (corresponding to the i* diagram in Fig. 2)

It is worth noting that the i* and Pareto efficiency-based model, along with the three discussed visualization, are all supported by means of our WebREd CASE tool (Aguilar et al 2012). The tool allows regular i* modeling, and is subsequently capable of automatically calculating the Pareto Front, and generating the tabular and radar chart visualizations (only Pareto Front configurations are shown). Furthermore, the tool allows interactively adding/removing Pareto Front configurations to/from the radar chart.

4 Experiments

In this section, we describe the definition, design, and settings of the controlled experiment we conducted. The context of the controlled experiment is described following the guidelines described in (Kitchenham et al. 2002) and the subjects as proposed by (Höst et al. 2000) to perform empirical studies in Software Engineering.

4.1. Experiment definition

The overall aim of our experiments was to compare three different visualization techniques used within goal-oriented modeling to select which FRs to implement, while optimizing NFRs. We considered and compared the following techniques: (i) i* graphical notation, based on the original i* model, (ii) a tabular visualization of FRs and NFRs configurations, based on the i* post-processed Pareto front model, and (iii) a radar chart visualization of these configurations, based on the i* post-processed Pareto front model. For these three techniques, we have tested the efficiency (time to correctly solve a problem), and effectiveness of results according to time (independent of the score) and according to correctness (independently of time), under two different levels of complexity (i.e., simple and complex). In particular, we evaluated the following main hypotheses:

- Is there a relation between the complexity of an i* model, the type of visualization, and the correctness of selected configurations? We studied this hypothesis separately by classifying correctness into three levels (i.e., correct, partially correct and incorrect).
- Is there a relation between the complexity of an i* model, the type of visualization and the time required to select a FR configuration? We studied this hypothesis separately for the three different visualizations (i* diagram, tabular and radar chart) and for the different complexities (easy and complex), and subsequently determined if there is any interaction between the type of visualization and the complexity of model instances.

After obtaining the results of the analysis, and due to the different nature of every visualization of the i* model instances (textual or visual), we performed a follow-up study to verify an additional hypothesis:

- Is there a relation between the subject's learning style (textual or visual) and the time required to select a FR configuration? We also determined the subject learning style to see its possible influence in the time required to select a FR configuration within our experiments.

To this aim, we set up a controlled experiment with two fixed factors: the three types of visualizations and two kinds of complexity levels. Specifically, simple and complex model instances differ by the number of intentional elements that directly influence the decision for NFR optimization: softgoals, contribution links, and tasks with outgoing contribution links. All simple model instances follow the

same pattern: 3 softgoals, 7 contribution links and 3 tasks with outgoing contribution links (as well as 1 goal, 7 other tasks and 1 resource). For complex model instances, we considered instances that have 5 softgoals, 14 contribution links and 6 tasks with outgoing contribution links (as well as 3 goals, 3 other tasks and 2 resources). The complex model instances of this experiment represent partial cases of real world scenarios.

Consequently, we have developed six exercises to be solved by the subjects of the experiments (one per type of visualization and per level of complexity). For each exercise, we observed two variables: time (measured in seconds) and score (ranging from 0 to 2). Time is measured by using the subjects' response time in accomplishing the required tasks. The time variable can bring us a measure for assessing the efficiency, as wasting more time to correctly solve an exercise indicates a less efficient notation. Score is measured by expert-judging the effectiveness of the subjects' answers: 0 if the result is wrong, 2 if the result is right, and 1 when the given solution contains the right result but it is incomplete. Subsequently, score serves as a useful measure for analyzing the effectiveness regarding correctness. Finally, we analyzed the time spent for all scores to assess the effectiveness regarding time and the efficiency (considering only correct answers). Consequently, we can assess the efficiency and effectiveness of each visualization by statistically analyzing the time and score variables.

In addition, due to the different nature of the visualization techniques used (textual and visual), we studied the relation of the learning style (textual or visual) of the subject and the type of visualization. To this aim, we performed a Felder test (Felder and Silverman, 1988) to determine the learning style of each subject.

Finally, the experience of the subject dealing with FRs was considered when studying the different types of visualization. As some of the subjects have extensive previous i* experience, and others have little i* experience, the efficiency could show different behavior according to the experience of the subject.

4.2. Experiment context

The context of the controlled experiments is described next (following the guidelines described in (Kitchenham et al. 2002).

4.2.1 Subjects

In order to ease the generalization of the results, the subjects are identified. The subjects are master students (Official Master of Web Technologies) from the University of Alicante (Spain), researchers and PhD students of the Polytechnic University of Valencia (Spain) and Jaume I of Castellon (Spain). Specifically, 32 subjects participated in total: 20 from Alicante (18 of them were master students and 2 PhD students), 12 from Valencia and Castellon (6 PhD students and 6 professors).

The subjects from the University of Alicante are experts in the software development domain and they had previous knowledge of the i* modeling framework (all of them have been enrolled in a 30-hour i* course). Therefore, they have experience with the i* modeling framework, although, they did not know anything about Pareto efficiency. The remainder of the subjects have previous experience with i* ranging from experts with deep understanding of i* to beginners with little i* experience. None of the participants had knowledge of Pareto efficiency. Consequently, a training session about Pareto efficiency took place to provide the subjects with the necessary knowledge to carry out the tasks required in the experiment.

4.2.2 Objects

As previously stated, in our experiment we defined six exercises: three having a "simple" complexity level, and three having a "complex" level. Each exercise of each complexity level uses one of the following types of visualization: (i) i* diagram, (ii) tabular visualization of Pareto front, and (iii) radar chart visualization of Pareto front.

Each single exercise contains three questions with the aim of asking the subject which is the best set of FRs to implement, while optimizing certain NFRs. For each exercise, the subject is asked to write down the answers to three questions: the first question asks for the best configuration for satisfying one NFR, the second aims to satisfy two NFRs (according to a specific priority) while the third asks for satisfying three NFRs (according to a specific priority). For example, in the survey system exercise represented in Figure 2, the following three questions are asked:

Question 1: which tasks do you need to implement to maximize usability?

Question 2: which tasks do you need to implement to maximize usability and reliability at the same time (equal priority)?

Question 3: which tasks do you need to implement to maximize usability (1st priority), and then reliability and anonymity (both 2nd priority)?

The model instances were different for each type of visualization (i* diagram, tabular Pareto front or radar chart) and level of complexity (simple or complex). Furthermore, we distributed the experiment's solving sequence randomly. This was necessary to avoid a repeated measure experiment (i.e., subjects learning from a previous model/visualization); however, we ensured that for each level of complexity, the underlying model instances for every visualization (type) had exactly the same difficulty. This was done by assuring that each model had the same amount of FRs, NFRs, tasks, resources, means-end and contribution links and the same hierarchy of tasks. Furthermore, we had an equal amount of positive and negative outgoing contribution links from tasks, and an equal amount of positive and negative incoming contribution links for each softgoal.

4.3. Hypothesis formulation

The main objective of our experiments was comparing three different visualizations for optimizing NFRs and how they respond to the complexity. Therefore, we tested the relation of the visualization used (i.e. i* notation, tabular and radar chart visualization) and the complexity in order to assess the effectiveness regarding correctness of the answers. The null-hypotheses were then formulated for correct answers:

- $H0_1$: There is no interaction between the type of visualization and the complexity level, for correct answers (score=2).
- $H0_2$: There are no differences between the types of visualization, for correct answers (score=2).
- $H0_3$: There are no differences between the complexity levels, for correct answers (score=2).

We also proposed to compare the differences of visualization measured according to effectiveness regarding time. The null hypotheses were:

- $H0_4$: There is no difference in the time spent to identify a set of FRs to implement while optimizing NFRs between type of visualization.
- $H0_5$: There is no difference in the time spent to identify a set of FRs to implement while optimizing NFRs between simple and complex levels of complexity.

- $H0_6$: There is no interaction between type of visualization and complexity levels, when measuring the time spent.
- $H0_7$: There is no interaction between type of visualization, complexity levels and score when measuring the time spent.

Furthermore, we performed a similar analysis on the efficiency where we considered time spent for correct answers. The null hypotheses for this variable were:

- $H0_8$: There is no difference in the time spent to correctly identify a set of FRs to implement while optimizing NFRs between different types of visualization.
- $H0_9$: There is no difference in the time spent to correctly identify a set of FRs to implement while optimizing NFRs between simple and complex levels of complexity.
- $H0_{10}$: There is no interaction between type of visualization and complexity level, when measuring the time spent for the correct answers.

In addition, two further analyses were done to study the impact of the subjects' experience on modeling FRs and the learning style (visual or textual) identified by the Felder test (Felder and Silverman, 1988). To analyze if a relation exists between the learning style of subjects and their performance measured in time spent to solve the exercise, we tested the following hypotheses (addressing both effectiveness regarding time and efficiency):

- $H0_{11}$: There is no difference to identify a set of FRs to implement, while optimizing NFRs, in the time spent, for different learning styles of subjects.
- $H0_{12}$: There is no difference in the time spent to correctly identify a set of FRs to implement while optimizing NFRs, for different types of visualization.
- $H0_{13}$: There is no interaction between type of visualization and learning style, when measuring the time required to correctly identify a set of FRs to implement, while optimization NFRs.

Finally, we compared the results obtained from experts and beginners to be able to evaluate the influence of the subjects' experience when testing the effectiveness regarding time of the type of visualization. The formulation was:

- $H0_{14}$: There is no difference in time required between experts and beginners when using i*, tabular and radar chart visualization to identify a set of FRs to implement while optimizing NFRs.

If the null hypothesis can be rejected with a low margin of error, we may accept an alternative hypothesis, which admits a positive effect of the type of

visualizations and/or complexity/learning-style/experience on the effectiveness/efficiency of the model.

4.4 Identification of Main factors and cofactors

In our experiment the main factors were: the *type of visualization* used to optimize NFRs (i* diagram, tabular and radar chart visualizations) and the *complexity* of the represented model (easy or complex). We also assessed if there is interaction among the main factors, which is the combined effect of both factors on the dependent variable (time or score). In addition, to better assess the effect of type of visualization it was needed to control other factors (called co-factors) that may have effect on the dependent variables. Those co-factors were the subjects previous *i* experience* and *learning style*.

4.5 Measurement of dependent variables

We have considered two dependent variables: score and time, as follows:

- Time is measured in seconds, by using the subjects' response time in accomplishing the required tasks. To do so, the subjects' starting and end time of each exercise are recorded. Therefore, time is a continuous variable.
- Score is ranging from 0 to 2 as discrete values (categorical variable). Score is measured by expert-judging the correctness of the subjects' answers:
    - A score of 0 is obtained if the result is wrong
    - A score of 2 is obtained if the result is right
    - A score of 1 is obtained when the result contains the right solution but is incomplete

4.6 Experimental trials

The experiments are performed using the test subjects. There are six exercises in total, two exercises for each type of visualization (i* diagram, Pareto front based tabular and radar chart), and, for each of them, one with complexity level easy and the other complex. All subjects solve all exercises. In order to avoid learning effects, each exercise is related to a different case study (for more details see Section 6 Threats to Validity).

Before the experiments, subjects were trained on the different approaches. Specifically, all individuals had previous knowledge on requirements engineering with i*. Regarding the other visualizations (table and radar chart), we did an ad-

hoc 30-minutes training session before conducting experiments. Due to the fact that the visualizations are based on the i* diagrams, it is enough time for the subjects to understand them (since all of them know i*). We also gave the subjects detailed instructions related to the tasks to be performed.

Each exercise was done individually. Subjects could take as much time as was needed to solve each exercise (i.e., no fixed time). The subject recorded the starting and ending time for each answer in hours, minutes and seconds, and the answer to the questions, i.e., one or more configurations of FRs.

4.7 Data analysis

The diagram in figure 4 presents an overview of the analysis strategy followed. We have divided the data analysis results in four subsections, results for: score, time, visual-preference and experience. A descriptive analysis is presented in each of the four sections as first step. Afterwards, we include the results of the relevant statistical tests to verify the formulated hypotheses.

Figure 4. Data analysis strategy

To analyze effectiveness regarding correctness, after a descriptive analysis, we tested the effect of type of visualization and complexity on the score (Fig. 4) by using a logistic Generalized Linear Mixed Model (GLMM) (Hair and Anderson, 2010). For the model selection, we tested all possible models (GLM & GLMM) against the null model and finally selected the complete logistic GLMM (Chi-square p-value=8.89e-05; Area under ROC=0.758). The complete logistic GLMM model includes score as response variable, Type and Complexity as fixed factors, subject and exercise as random effects, and Time as covariate; no transformation was required for this model. Here, we eliminate the level 1 of the score variable

for comparing only correct vs. incorrect answers. Then, the estimated GLMM model can be easily interpreted by comparing the ratio between score=2 (correct answers) vs. score=0 (incorrect answers) probabilities.

We have analyzed the effectiveness regarding time by testing the time spent to solve the exercise considering type and complexity as fixed factors in a multifactorial ANOVA. Here, we used all answers, to understand the behavior across the type of visualization between the two levels of complexity. Due to the heterogeneity of variability around the mean time across these factors, a logarithmic transformation was needed to be able to assure the homogeneity of variances. After that analysis, we selected only the correct answers (with score = 2), to go in deep with the efficiency analysis. We used the Tukey HSD test (Jaccard et al. 1984) for the post hoc analysis when needed.

The analysis for the time spent considering only those answers with the maximum score, shows the heteroscedasticity (Fig. 5). The time does not show homogeneity of variances between the type, score and complexity levels (p-value under 0.05 even with square root or logarithmic transformations). In this case, we proceed to analyze the ANOVA setting the significance to 0.01.

Analysis of subject learning style preference

To analyze the subject's textual or visual learning preference, we performed a Felder test, with which we obtain the subject's learning style preference. Then, ANOVA was used to detect if there is any evidence that the relationship between the preference of the subjects and the type of visualization has any influence on the time spent.

Analysis of experience

We tested if the experience of the subjects (with previous knowledge of i*) has any effect on time. We used the experience and type of visualization as fixed factor and the time as a dependent variable. Then, we analyze the experience effect by testing with ANOVA, and the Tukey HSD when a significant difference was found.

5. Results

We have divided the results into four subsections, focusing respectively on effectiveness, efficiency, visual or textual preference and influence of experience analysis.

5.1 Analysis of score

The results for score, which represent effectiveness regarding correctness, are summarized in Table 2 in which the total amounts of correct/incorrect/partially-correct answers by type and complexity levels are shown. For example, each participant had to solve 3 questions on easy (complexity) i* graphical notation (type), therefore the total amount of answers for that type of exercise should be 96. However, not all participants answered all questions, and we got 84 answers. Considering the score results, we applied an analysis strategy separating the score levels in: correct (score = 2), incorrect (score = 0) and partially correct answers (score = 1).

Table 2. Total amount of score by type of visualization and complexity level

Type	i* Graphical Notation		Pareto tabular visualization		Pareto radar chart visualization	
Complexity	Easy	Complex	Easy	Complex	Easy	Complex
Score=2	41	23	41	39	28	24
Score=1	8	6	12	12	22	26
Score=0	35	19	15	13	20	11
	84	**48**	**68**	**64**	**70**	**61**
	132		**132**		**131**	

From Table 2, we highlight some results: (i) overall, there is a similar amount of correct (64) versus incorrect and partially correct answers (68) for i* graphical notation; more correct answers (80 versus 52) for tabular visualization; less correct answer for radar chart visualization (52 versus 79); ii) i* graphical notation has a larger amount of wrong answers compared to tabular and radar chart visualizations, even for easy model instances; iii) when the model is getting complex, the amount of correct answers for the i* graphical notation decreases sharply (-44%), while only mildly for radar chart visualization (~ -15%) and quite similar for tabular visualization (< -5%); iv) the total amount of partially correct answers is much lower than that of the correct answers for i* graphical notation and tabular visualization; for the radar chart the correct and partially correct answers are more evenly spread (52 and 48 respectively); and v) the Pareto tabular

visualization has similar values of correct answer for easy and complex model instances.

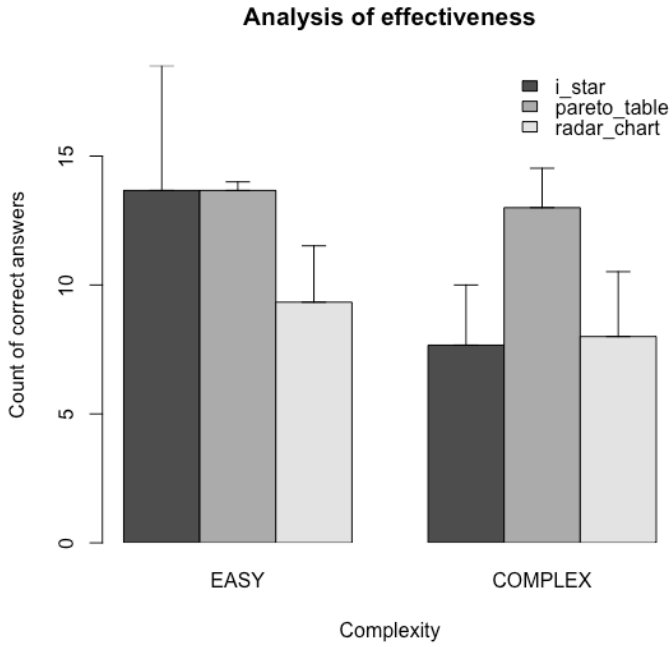

Fig. 5. Barplot for the amount of correct answers by exercise for type and complexity.

The dependent variable on ANOVA was the amount of correct answers (score=2) by exercise. ANOVA was not able to detect significant interaction between Type and Complexity ($F_{2,12}$=0.597; p=0.566) (hence, $H0_1$ cannot be rejected). Subsequently, the differences on the combined effect were not significant. This is mainly due to the high variability in the number of correct answers on i* graphical notation (type of visualization). This behavior can be hiding the tendency shown in Figure 5 where i* graphical notation sharply decreases the number of correct answers on complex models, while there is no clear pattern on the other two types of visualizations.

We found no significant differences in mean of the number of correct answers by Type ($F_{2,12}$=1.550; p=0.252) ($H0_2$ cannot be rejected). Also regarding Complexity, there are no significant differences ($F_{1,12}$=1.508; p=0.243), and complexity does not seem to affect the radar chart and tabular visualizations (Fig. 5) (hence, $H0_3$ cannot be rejected). We note that, as for interaction between Type and Complexity, the lack of individual effect may be due to the high variability on the behavior of i* graphical notation, which may hide differences on decrease in the number of correct answers.

Furthermore, we have done a Logistic Generalized Linear Mixed Models (GLMM) for score as a dependent variable. The objective behind this analysis is

to understand the effects from the fixed factors (type and complexity), as well as the possible random effects (individuals and exercises), considering the time as covariate. The results of the estimates, standard error, Wald test statistic, p-value and Odd Ratio (OR) from the GLMM model are shown in Table 3.

Table 3. Results of the complete logistic mixed model GLMM.

	Estimate beta	**Std.error**	**Wald value**	**Pv Wald**	**Odds ratio (OR)**
Intercept	0.4385	0.3866	1.134	0.2567	1.5504
TYPE.Radar_chart	0.2537	0.4393	0.578	0.5635	1.2889
TYPE.Pareto_table	1.2568	0.4252	2.956	**0.0031***	3.5143
COMPLEXITY.Easy	1.4146	0.4197	3.370	**0.0007***	4.1150
Radar_chart & Easy	-0.6027	0.6417	-0.939	0.3475	0.5473
Pareto_table & Easy	-1.4264	0.6294	-2.266	**0.0234***	0.2402
Time	-0.0070	0.0020	-3.483	**0.00049***	0.9930

Positive values in estimates from Table 3 (or greater than 1 in OR) indicates an increase in the probability of the score=2 vs score=0 ratio. The Intercept OR is 1.5504, in other words it is 1.55 times more probable to solve the exercise correctly, compared to not solving it correctly (Score=0) by using the i* graphical notation with complex model instances (intercept), however is not significant (Wald test p=0.2567). The tabular visualization of the Pareto front has significant behavior compared with the intercept (Wald test p=0.0031), and, its OR indicates that it is 3.51 times more probable to solve the exercise correctly using the tabular visualization of Pareto front than using the i* graphical notation, when working with complex model instances. The complexity, in global terms, has a significant effect (p=0.0007): for easy model instances, it is 4.115 times more probable that the exercise is solved correctly (score=2) compared to incorrectly (score=0). For radar chart there is no significant effect with respect to easy or complex model instances (p=0.3475). Pareto front for easy model instances has a (significant) negative effect compared to i* complex model instances: the estimate from table 3 is -1.4264 and the OR indicates that it is almost 4 times more probable to solve an easy Pareto exercise than an i* complex model, an evident result. The time, considered as covariate in the analysis of effectiveness, has a significant negative effect, which is interpreted as follows: as long as the time increases, the probability of getting correct answers diminishes. As the estimate is close to zero

(estimate=-0.0070), this is only a minor behavior, however, it is significant (p-value=0.00049). This behavior is further confirmed in the efficiency analysis.

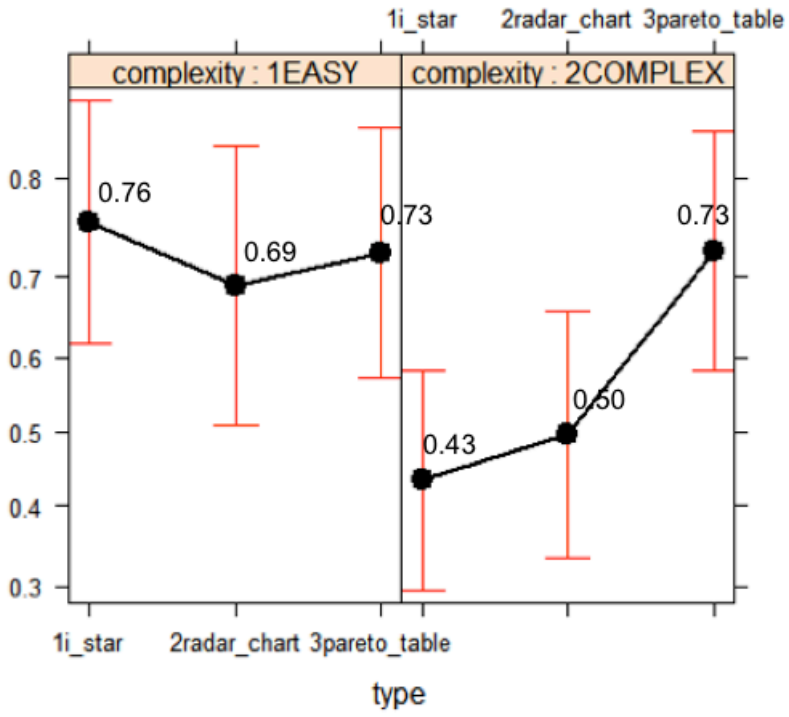

Figure 6: Estimation of probability for score=2 by Complexity and Type

The interaction plot in figure 6 shows that all three types of model visualization obtain similar probabilities for solving the exercises when the model is easy. However, when modeling becomes complex, the probability to correctly solve the problem when using the i* graphical notation is halved. Although the radar chart visualization seems to have similar behavior for complex model instances, the probability is much lower and is not significantly different from the probability of solving easy exercises. Finally, the tabular visualization of the Pareto front seems not to be affected by the complexity, because its probability remains high for easy and complex model instances (around 0.73 of probability to correctly solve the exercise).

5.2 Analysis of time

The strategy for the analysis of time starts with the descriptive analysis of the time variable followed by the analysis of the effect of type and complexity (fixed

factors) on the time. We consider all responses (effectiveness regarding time) and only correct responses (efficiency).

Table 4. Summary of mean and (standard deviation) of time in seconds by type and complexity for all answers (effectiveness regarding time) and for score=2 (efficiency)

ALL ANSWERS	**EASY**	**COMPLEX**
i* graphical notation	141.85 (112.51)	133.36 (84.78)
Tabular visualization	69.29 (42.21)	100.08 (43.11)
Radar chart	70.38 (48.05)	76.36 (44.59)
SCORE=2		
i* graphical notation	119.29 (87.32)	106.09 (62.48)
Tabular visualization	70.15 (40.68)	94.18 (45.92)
Radar chart	71.35 (62.00)	61.71 (33.68)

Considering the effectiveness regarding time (see Table 4), the i* graphical notation obtained the worst results, both for easy and complex models. On the other hand, radar chart overall performs best regarding time (lowest time), yet for easy models, tabular visualization obtains similar results (note that it is not possible to statistically significantly differentiate between tabular and radar chart visualization for easy models, even though the mean for the former is slightly lower). Interestingly, the complexity of models doesn't seem to significantly affect the time needed to solve exercises for i* graphical notation and radar chart visualization (only minor differences); for tabular visualization on the other hand, the time spent seems to significantly increase as models become more complex. Finally, as a disclaimer, we must mention that there is a high variability in time to solve the exercise for all visualizations (see standard deviation in Table 4). This variability is higher for i* graphical notation than for the tabular and radar chart visualizations.

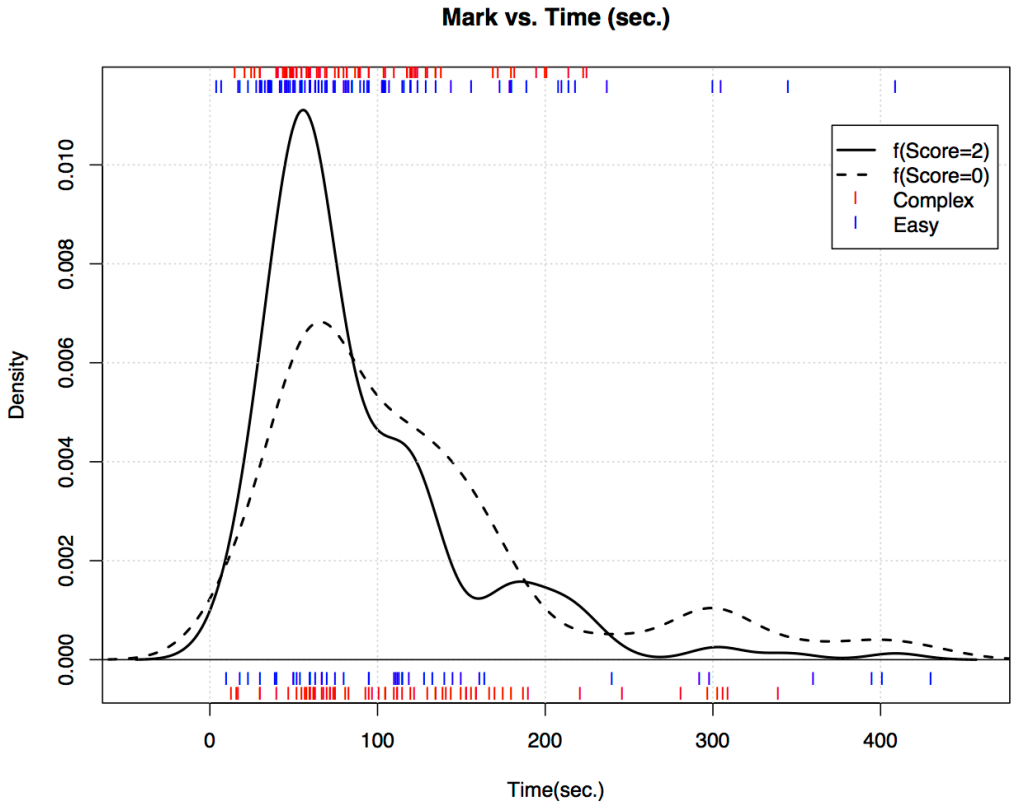

Figure 7. Density plot for time variable. Lines for score=2 and dashed-line for score=0. Tick marks on top are for score=2, bottom for score=0. Red marks are for complex model instances, blue marks for easy model instances.

Time is not normally distributed as shown in Figs 7 and 8. There is a peak below 100 seconds, and there is another small peak around 200 seconds (around 300 seconds if considering the corresponding wrong answers). The tick marks on the top represents score=2, for easy and complex model instances. The tick marks on the base of x-axis represents score=0 (for easy and complex model instances too). The tick marks of score=2 are more concentrated around the mean than tick marks of score=0. The wrong answers are distributed along the range of time observed.

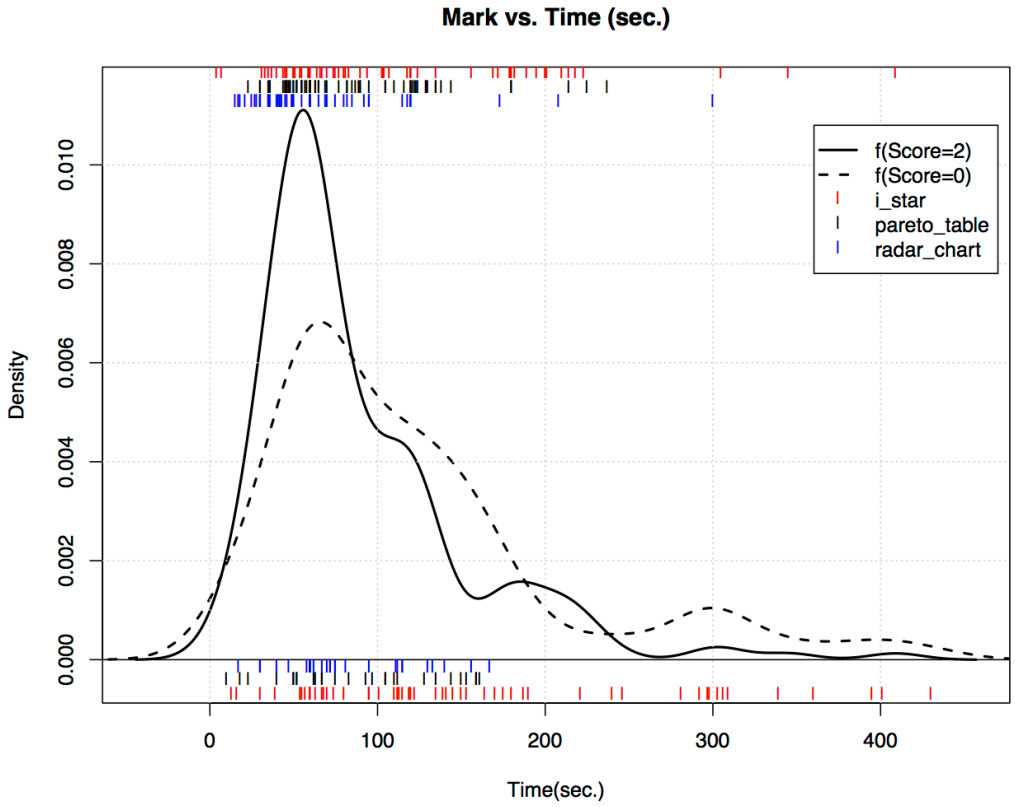

Fig. 8. Density plot of the Time variable. Black line for score=2 and dashed-line for score=0. Tick marks on top are for score=2, bottom for score=0. Red marks are for i*, black marks for Pareto tabular and blue marks for radar chart.

The variability in time was mainly produced by the time required to solve the exercises using the i* graphical notation (see Table 4 and figures 7 and 8). The lack of normality and homoscedasticity (large dispersion for score=0 and 1, and more concentrated times for score=2) discourages the use of parametric test. However, ANOVA is robust under the lack of normality, also in presence of heteroscedasticity (Lix et al. 1996), when applied on balanced datasets where the sample size is large enough (n>30), but it is recommended to reduce the significance used to 0.01. The alternative tests do not address the main problem, namely the inequality of variances. Moreover, the heterogeneity can result in a lack of effect detection, while it is important to deal with heterogeneity observed over time, and explain its effects. Given these restrictions, the significance was set to 0.01 and we proceed with the multifactorial ANOVA analysis.

The ANOVA results (Table 5) hint an interaction effect between the type and the complexity ($F_{2,377}$=5.005; p=0.00716) (hence, $H0_6$ can be rejected). The lines of the interaction plot (Fig. 9) show different trends in time by type and complexity levels. The more complex the model, the more time is required for solving the

exercise when using the Pareto table (see Fig.9). When using the other notations, complexity seems to have no effect on the time spent for solving the exercise. The type of visualization has a significant effect on the time to solve the exercise ($F_{2,377}$=34.619; p=1.57e-14) ($H0_4$ can be rejected). There is a lack of effect of complexity on the effectiveness regarding time ($H0_5$ is not rejected) ($F_{1,377}$=0.872; p=0.35113). In addition, the interaction between type, complexity and score was not significant to the effectiveness regarding time ($H0_7$ is not rejected) ($F_{4,377}$=0.870; p=0.48210).

The score has significant effect on effectiveness regarding time ($F_{2,377}$=5.289; p =0.00543). The post-hoc Tukey test showed differences between score=2 and score=0 (TukeyHSD test p=0.00554). Furthermore, the Tukey test showed that for score=2 (special case for the $H0_{8\text{-}10}$), measuring efficiency, the time required using the i* graphical notation was significantly higher than radar chart (p=0.0043) and Pareto tabular (p=0.0766). Considering efficiency, the tabular and radar chart visualizations have similar time values on correct answers (p=0.9474) (Fig. 9b). In addition, for any score, the i* graphical notation requires more time to solve the same exercise than any other visualization (Fig. 9). Based on the results of the Tukey test, we can thus reject $H0_8$: subjects using the i* graphical notation spent (significant) more time (p=0.00000) than radar chart and Pareto table. The complexity does not seem to have an effect on time (p-value=0.35113), and thus $H0_9$ cannot be rejected. The interaction between type and complexity has significant effect on time ($F_{2,377}$=5.005; p=0.00716) ($H0_{10}$ is rejected). Figure 9 shows the time spent considering all data (Fig.9a) and by score (Fig. 9bcd).

Table 5. ANOVA results for the model (Time ~ Type * Score * Complexity)

	Df	**Sum Sq**	**Mean Sq**	**F value**	**Pr(>F)**
Type	2	309615	154808	34.619	1.57e-14 ***
Score	2	47300	23650	5.289	0.00543 **
Complexity	1	3897	3897	0.872	0.35113
Type x Score	4	23300	5825	1.303	0.26850
Type x Complexity	2	44762	22381	5.005	0.00716 **
Score x Complexity	2	17677	8839	1.977	0.13999
Type x Score x Complexity	4	15558	3889	0.870	0.48210
Residuals	377	1685866	4472		

Significance codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

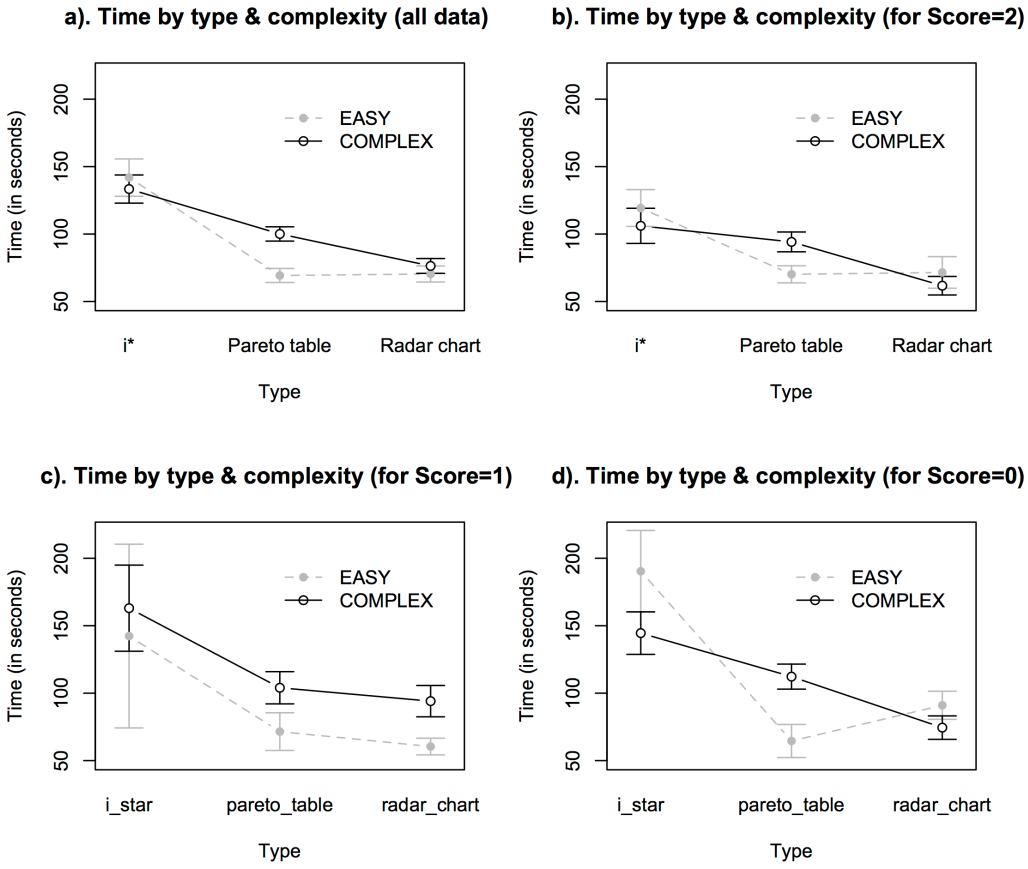

Fig. 9. Interaction plot for time by type and complexity with a) all answers, b) only Score=2, c) Score=1, and d) Score=0

5.3 Analysis of subject learning style preference

As stated, we also performed a Felder test (Felder and Silverman, 1988) to assess if there is any evidence on the relationship between the preferred learning style of the subjects (visual, textual or balanced) by complexity and the time spent to solve the exercises. Table 6 contains the total amount of correct versus incorrect or partially correct answers by subject learning style preference. We hereby note that not all the subjects performed the Felder test and not all the participant answered all the questions. Note that for our experiment, the Felder test did not reveal any subjects with "textual" learning style preference; only "visual" and "balanced". Considering the ratio between visual vs balanced subject style preference, both for correct versus incorrect and partially correct answers (Table 6), we observe that for i* graphical notation and for radar chart visualization, the ratio is close to 1 both for correct and incorrect/partially correct answers, while for tabular visualization the ratio shows slightly different behavior for correct (ratio=1.2) versus incorrect and partially correct answer (ratio=0.7) answers.

Table 6. Total amount of answers by type and subject learning style preference

Type	i*			Tabular			Radar chart		
Sub. pref.	Visual	Balanced	Ratio	Visual	Balanced	Ratio	Visual	Balanced	Ratio
Score=2	30	27	1.1	43	37	1.2	21	26	0.8
Score=1 or Score=0	30	33	0.9	17	23	0.7	38	34	1.1
	60	**60**		**60**	**60**		**59**	**65**	
	120			**120**			**124**		

The ANOVA test did not detect any significant difference in mean time (effectiveness regarding time) for the interactions between type, complexity and subject learning style preference. Considering the main factor, the subject's preference has no effect on the time spent to solve an exercise ($F_{1,347}$=0.100; p=0.753) ($H0_{11}$ is not rejected). Observing only those well-answered exercises (score=2), measuring efficiency, the subject learning style preference does not seem to have an effect on the time spent in answering ($F_{1,172}$=1.063; p=0.304) ($H0_{12}$ is not rejected). The interaction between the type and subject learning style preference was also not significant ($F_{2,172}$=0.997; p=0.371) ($H0_{13}$ is not rejected). The interaction plot on figure 10 shows similar slopes for visual or balanced subject learning style preference for any type and score. Thus, the subject learning style preference does not seem to have any effect on the efficiency.

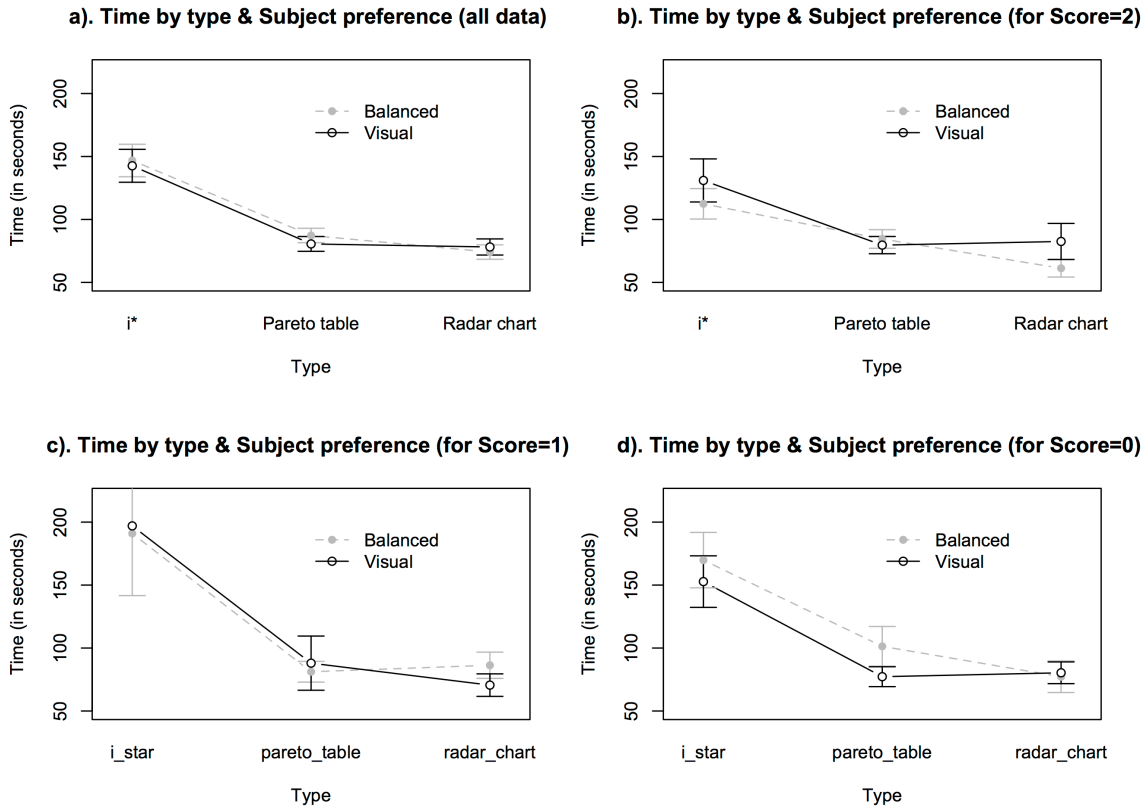

Fig. 10. Interaction plot for time by type and subject learning style preference with a) all answers, b) only score=2, c) score=1, and d) score=0

5.4 Analysis of experience

We analyzed effectiveness regarding time and efficiency based on the subjects experience. The subjects experience was not balanced between the beginner and expert levels, mainly due to the fact that few experts were present among the students, and almost all were beginners. Nevertheless, using the i* graphical notation requires more time to interpret model instances and formulate an answer for beginners than for those that are experts (p-value=4.09e-14) ($H0_{14}$ can be rejected). Radar chart and tabular visualizations of Pareto front have a similar effect on time for experts and beginners (Fig. 11A). Assessing the efficiency, the ANOVA test did not find significant differences on any source of variation. Also, the trend to increase the time to correctly solve the exercise observed for Pareto table in (Fig. 11B) was not significant.

.

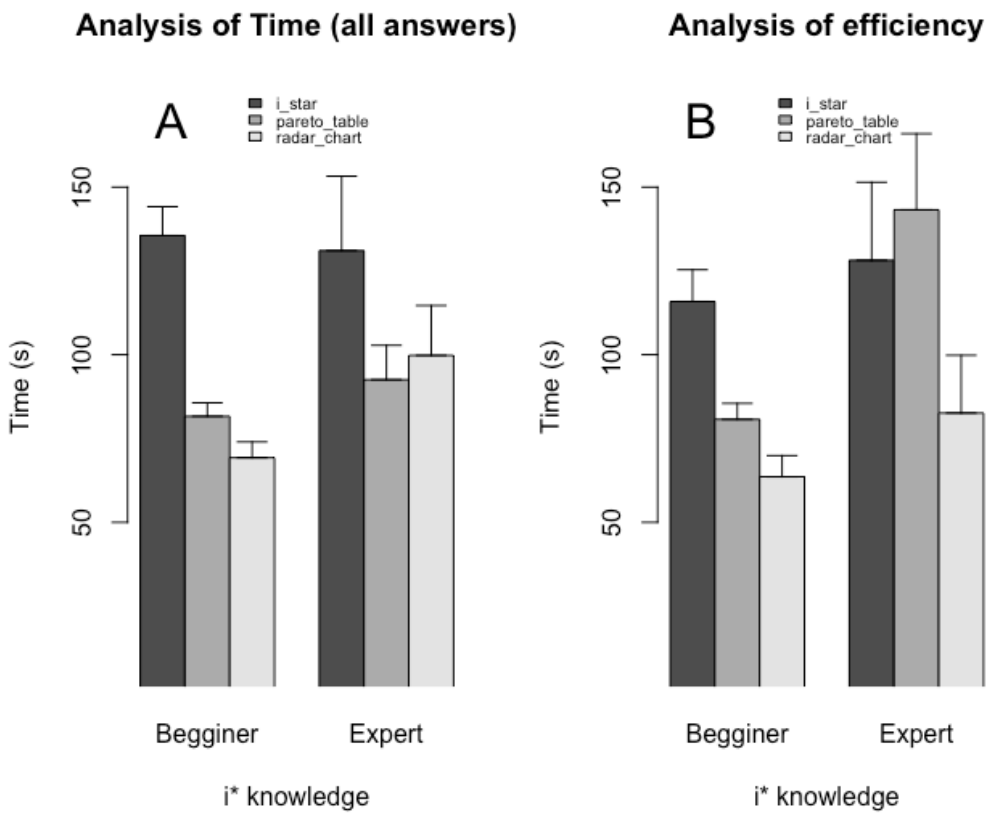

Fig. 11. Barplot for time by type and experience with all answers (A) and the correct answers (B).

There is a high number of outliers for the beginners (Fig. 12). This means that some beginners take a lot of time to solve some questions. Most of those outliers come from i* graphical notations. This is different from the experts' behavior, where the maximum (correct) solving time was lower than 200 seconds.

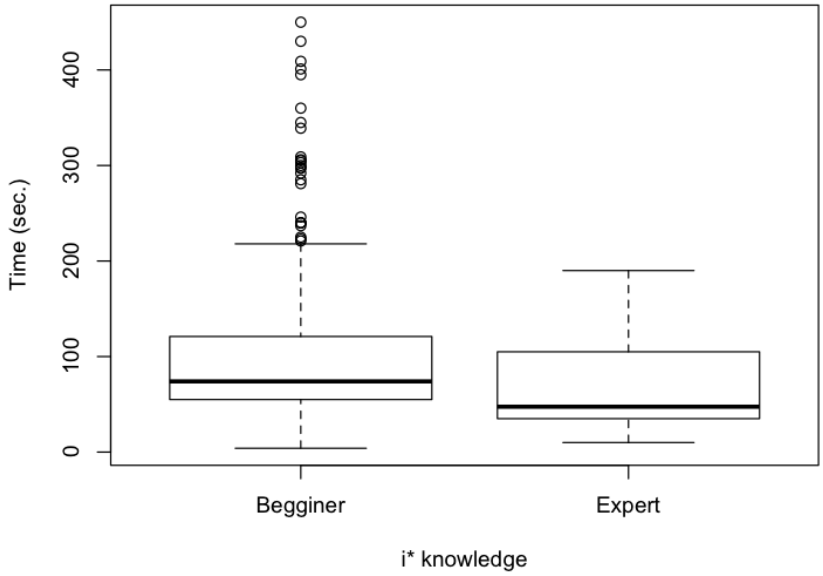

Fig. 12 Boxplot representing time by experience.

The GLM model (Poisson family) did not find any significant difference for the count of correct answers by experience or type of visualization. The variability on the effectiveness regarding correctness was high across all levels (Fig. 13) masking any behavior (the relative lower effectiveness regarding correctness on experts was not significant).

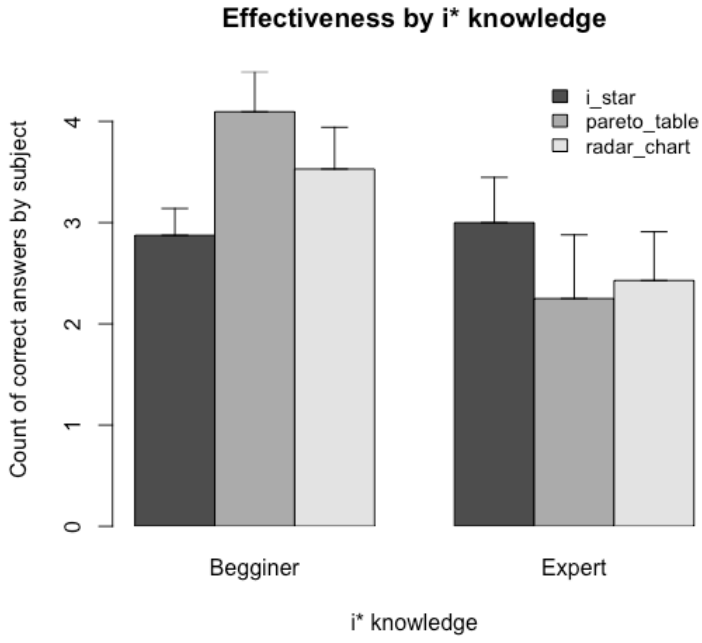

Fig. 13 Effectiveness regarding correctness by experience and type of visualization.

6 Threats to Validity

In this section, we will analyze the threats to validity related to the experiment. We can categorize them as internal, external, construct and conclusion validity threats.

6.1. Internal Validity

In this type of experiments, the main internal validity threat is the learning effect. This effect appears when performing the experiments consecutively, and subjects learn how to improve their results if they always need to solve the same problem, or always start with the same type of visualization and complexity. In this work, we deal with this effect by not using the same assignment for any exercises (one for each combination of type and complexity) and in addition, by randomly distributing the experiment's solving sequence. Furthermore, subjects were asked to optimize different NFRs, having similar difficulty for each different assignment (within easy/complex).

To avoid the communication between subjects, which could falsify results, they performed all experiments in one run, and having surveillance to make sure there was no communication between subjects.

6.2. External Validity

As was described in the experiment's methodology, all the subjects were students from a master in computing engineering, with the same short training in the modelling instruments used in the experiments. Then, the results can be generalized for all graduates. No modelling experience was required. The short training period establishes a common basis suitable for the analysis.

The main external validity threat is the relatively small model instances used in the exercises compared to business scenarios that can result in more complex model instances. However, the complexity level was restricted for the duration of the experiments, which was limited to maximum two hours, not days as in real world cases. Nevertheless, the complex level was always selected with enough level of difficulty to represent partial cases of real world scenarios. In addition, to deal with this external validity threat and to allow detecting the differences in effectiveness or accuracy related to complexity it was taken as a factor in the analysis.

6.3 Construct Validity

The validity threats related to the design mainly affect discrete or categorized variables. The relation between factors when considering discrete variables can be analyzed by a Chi square test. However, it is known that this analysis cannot deal with interaction and/or multifactorial analysis and it increasingly rejects the null hypothesis on some circumstances (Bull et al. 1992). To deal with this validity threat a logistic regression was used to explain relationships on possible hidden effects behind multifactorial models on qualitative variables.

Construct validity threats that may be present in this experiment, i.e., interactions between different treatments, were mitigated by a proper design that allowed separating the analysis of the different factors and their interactions. In particular, in each set of exercises, subjects worked on two different model instances to avoid learning effects and to ensure that the differences in instances' complexity would not bias the results (we selected two data instances of comparable complexity at each level, as described in section 4.2.2 Objects).

An important issue is the diversity on the subjects. They were selected from masters in software engineering, researchers and Phd students from three universities. These subjects share the interest of acquiring more and specialized knowledge in the topic of the experiment. Nevertheless, they come from different universities, different basic formation and different business experience. This diversity represents the real-world scenery in software engineering (Briand et al. 2005).

6.4 Conclusion Validity

We have presented the results classified by four aspects: effectiveness, efficiency, preference and experience of subjects. We have used the appropriate tests considering the analytical strategy. We presented the suitable model (and its p-value) for testing differences between means for quantitative dependent variables, and, in case of qualitative dependent variables we used a proper analytical tool (GLMM) for the experiment. We checked the validity of the model instances and the assumption required in each statistical analysis. In addition, we used the descriptive analysis preceding the statistical one. Figures included in this work were selected to improve the comprehension of the experiment results related to each analysis. Thereby, they facilitate demonstrating and reaching a conclusion for each aspect.

7. Discussion

It is well known that research in a particular field passes through several phases, depending on its maturity (we consider the classification scheme suggested by Wieringa et al (2006)). Our research contributes to the mature field of software engineering (and more specifically, requirements engineering), where solid validation/evaluation of solution proposals are expected. Our work indeed classifies as validation/evaluation research, as we: (i) compare our solution (Pareto Front model) with an existing solution (original i* model) at hand of three visualization (two for the former, one for the latter); and (ii) further qualify the comparison, and the Pareto Front solution itself, by investigating under which conditions the Pareto Front-based visualization performs better/worse (e.g., simple/complex models, learning style, time spent).

Our results shed light on some important issues. Regarding the original i* graphical notation, with respect to selecting functional requirements while balancing and optimizing non-functional requirements:

1. Considering the effectiveness regarding correctness we have observed a large amount of wrong answers. Additionally, for complex model instances, the amount of answers in general, and the amount of correct answers in particular, decreases.
2. Considering the effectiveness regarding time and efficiency, in general, using the i* graphical notation takes more time to obtain an answer (and also having more extreme values[2]). Surprisingly, the average time spent decreases as models become more complex. We can explain this unexpected result considering that mostly experts answered the complex model exercises, and beginners, which are generally slower, were not always able to complete the complex exercises. We observed that some beginners take a lot of time to solve some exercises of the experiment.

Both the above observations discourage the use of i* graphical notation in real world cases: these models tend to be rather complex, whereby fewer modelers are able to formulate an answer; modelers on average spend more time formulating an answer; and there generally is a larger amount of incorrect answers.

Regarding the tabular and radar chart visualizations:

[2] This variability was randomly observed and cannot be explained by visual preferences, previous knowledge of i* or complexity.

1. Considering the effectiveness regarding correctness, complexity does not seem to affect our tabular and radar chart visualization, as each (separately) obtains similar amounts of correct answers for easy and complex model instances. However, tabular visualization got significantly more correct answers than radar chart.
2. Considering the effectiveness regarding time, tabular and radar chart visualizations emerge as the most efficient approaches. The tabular visualization seems to be slightly affected in its efficiency on complex model instances, but the post-hoc test was not able to detect that trend. Nevertheless, for our experiment, we emphasize that the using radar chart modelers perform ~30% faster (around 30 seconds) on average for complex models, and using tabular visualization around 10% faster, compared to i* graphical notation.

Taking in mind the above facts, on complex model instances, which are especially relevant as real-world cases are indeed complex by nature, the visualization with better effectiveness regarding correctness was the tabular visualization. We can conclude then, that regarding correctness, the original i* graphical notation is not suitable for complex model instances. Nevertheless, although the tabular visualization is the one that obtains more correct answers, it performs worse regarding time on complex model instances.

An important observation in this experiment is the high variability of the i* graphical notation results across all the variables (time and correctness). This variability denotes a large number of outliers: participants spending much more or less time than the average, and a large spread between correct and incorrect answers. Together with the sharp drop of correct answer on complex models, it suggests a lack of confidence of users using the i* graphical notation, and a larger possibility that a particular modeler produces a bad result (in time and/or correctness). We could also not find an improvement with experience: even gaining experience in i* graphical notation, the results are still highly variable, or in other words, training in i* does not remedy the problem of high variability. Clearly, the high variability detected for i* graphical notation, and the consequences it implies, is an undesirable property in a real word context (i.e., in industry). On the other hand, using Pareto based visualization the results were more consistent, showing less variability, particularly on the correctness.

Furthermore, the Pareto front based alternatives tend to performs better even without any previous experience.

8. Conclusions

Requirements Engineering (RE) methods and frameworks were developed to understand, elaborate, reason about and document requirements. RE methods feature graphical notations (visualizations), which were primarily designed for ease of model construction and interpretability, and model readability. In this article, we argue that different visualizations for different purposes in the RE process might be useful. Particularly, we focus on a crucial step in RE: optimizing NFRs when selecting FRs to implement. For this purpose, we chose the goal-oriented RE approach i*, and compare its original graphical notation with two visualizations of our custom developed i* post-processed Pareto front model. More specifically, contributions of this article is twofold: (i) a controlled experiment to compare the three visualizations, to test their effectiveness regarding correctness and time, and their efficiency when being used by the designers (thus showing the convenience of using an i* postprocessed model for solving specific tasks such as selecting FRs while balancing and optimizing NFRs); and (ii) a detailed discussion on the results of our empirical evaluation, thus giving insight in the variables and conditions under which visualization performs better (or worse). From this work, we can conclude that, when selecting FRs to implement while optimizing NFRs, the original i* graphical notation is the least adequate visualization method. It performs worse regarding correctness, and scales worse when dealing with complex model instances, compared to Pareto efficiency-based radar chart and tabular visualizations. The latter two have a higher probability than the i* graphical notation to obtain a correct configuration of FR while balancing NFRs, while spending less time. Among them, and for complex model instances, the tabular visualization resulted in more correct answers compared to radar chart visualization.

When considering the time required to solve an exercise, the radar chart visualization scales better with complexity compared to the tabular visualization. According to our experiments, subjects perform better when using our Pareto-efficiency-based visualizations (tabular and radar chart) in efficiency and effectiveness regarding time. Moreover, there is no difference with regards to the

visualization preference of the subjects, nor to their previous experience, when considering time spent to solve an exercise.
As future work, we will focus on considering these results to improve visualizations mechanisms of our approach for increasing efficiency and effectiveness of NFR optimization. Also, we would like to replicate these experiments to get more valuable insights.

Acknowledgements

We thank all the anonymous participants that took part in the experiments. Sven Casteleyn is funded under the Ramón y Cajal Program of the Spanish Government, grant number RYC-2014-16606. This work has been partially supported by the Publi@City project (TIN2016-78103-C2-2-R) from the Spanish Ministry of Economy and Competitiveness.

References

Abirami S., Shankari G., Akshaya S., Sithika M. (2015) Conceptual Modeling of Non-Functional Requirements from Natural Language Text. In: Computational Intelligence in Data Mining - Volume 3. Smart Innovation, Systems and Technologies, vol 33. Springer, New Delhi.

Aguilar JA, Garrigós I, Mazón JN, Trujillo J (2010) An MDA Approach for Goal-Oriented Requirement Analysis in Web Engineering. In: *J. Univ. Comp. Sc.* 16(17), 2475–2494

Aguilar JA, Garrigós I, Mazón JN (2011) A Goal-Oriented Approach for Optimizing Non-functional Requirements in Web Applications. In: *ER workshops: Eighth International Workshop on Web Information Systems Modeling*. 14-23

Aguilar JA, Garrigós I, Casteleyn S & Mazón JN. (2012). WebREd: A Model-Driven Tool for Web Requirements Specification and Optimization. In M. Brambilla, T. Tokuda, & R. Tolksdorf (Eds.), *Web Engineering* (Vol. 7387, pp. 452-455). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-31753-8_42

André Almeida, Nelly Bencomo, Thais Batista, Everton Cavalcante, and Francisco Dantas. 2015. Dynamic decision-making based on NFR for managing software variability and configuration selection. In Proceedings of the 30th Annual

Ameller D, Gutiérrez F, Cabot J (2010) Dealing with Non-Functional Requirements in Model-Driven Development. In: *18th IEEE International Requirements Engineering Conference*, 189–198

Ruba Alzyoudi, Khaled Almakadmeh, and Hutaf Natoureah. 2015. A Probability Algorithm for Requirement Selection In Component-Based Software Development. In Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication (IPAC '15). ACM, New York, NY, USA, Article 95, 6 pages.

Austin M, Mayank V, Shmunis N (2006) *PaladinRM: Graph-based visualization of requirements organized for team-based design*. Syst. Eng. 9, 2 (May 2006), 129-145. doi:10.1002/sys.v9:2

Benestad H. and Hannay J. 2012. Does the prioritization technique affect stakeholders' selection of essential software product features?. In Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement (ESEM '12). ACM, New York, NY, USA, 261-270.

Bimrah KK, Mouratidis H, and Preston D. (2008) Modelling Trust Requirements by Means of a Visualization Language. In *Proceedings of the 2008 Requirements Engineering Visualization* (REV '08). IEEE Computer Society, Washington, DC, USA, 26-30

Briand L, Labiche Y (2004) Empirical studies of software testing techniques: Challenges, practical strategies, and future research. *ACM SIGSOFT Software Engineering Notes*, *29*(5), 1-3

Broster I, Coombes A. 2011. How to measure and optimize reliable embedded software. In Proceedings of the 2011 ACM annual international conference on

Special interest group on the ada programming language (SIGAda '11), ACM, New York, NY, USA, 1-2.

Bull CR, Bull RM, Rastin BC (1992) On the sensitivity of the chi-square test and its consequences. *Measurement Science and Technology*, *3*(9), 789-795. http://doi.org/10.1088/0957-0233/3/9/001

Almir Buarque, Jaelson Castro, and Fernanda Alencar. 2013. The role of NFRs when transforming i* requirements models into OO-method models. In Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC '13). ACM, New York, NY, USA, 1305-1306.

Cooper JR, Lee SW, Gandhi RA, Gotel O (2009). Requirements Engineering Visualization: A Survey on the State-of-the-Art. *In 2009 Fourth International* Workshop on Requirements Engineering Visualization (REV) (pp. 46-55). http://doi.org/10.1109/REV.2009.4

Douglas, V. L. R. (2010). A case study of non-functional requirements and continuous improvement at a national communications system contractor. Available from ProQuest Dissertations & Theses Global.

Duan C, Laurent P, Cleland-Huang J, Kwiatkowski C. Towards automated requirements prioritization and triage. Requirements Engineering. May 2009;14(2):73-89. Available from: Academic Search Complete, Ipswich, MA. Accessed June 15, 2017.

Ernst, N A, Yu, Y, Mylopoulos, J (2006). Visualizing non-functional requirements. *In First International Workshop on Requirements Engineering Visualization,* IEEE, pp 2. doi: 10.1109/REV.2006.10

Feather, MS, Cornford, SL, Kiper, JD, Menzies, T (2006) Experiences using Visualization Techniques to Present Requirements, Risks to Them, and Options for Risk Mitigation. *In 1st international workshop on Requirements Engineering Visualization*, IEEE, pp 10, Doi: 10.1109/REV.2006.2

Felder RM, Silverman LK (1988) Learning and teaching styles in engineering education. *Engineering education*,*78*(7), 674-681

Gabrysiak G, Giese H, and Seibel A (2009) Interactive Visualization for Elicitation and Validation of Requirements with Scenario-Based Prototyping. In *Proceedings of the 2009 Fourth International Workshop on Requirements Engineering Visualization* (REV '09). IEEE Computer Society, Washington, DC, USA, 41-45

Garrigós, I., Mazón, J.-N., Trujillo, J. (2009). A requirement analysis approach for using i* in web engineering. In: *ICWE*, pp. 151–165

Garrigós, I., Mazón, J. N., Koch, N., Escalona, M. J., & Mylopoulos, J. (2010). Foreword: First International Workshop on the Web and Requirements Engineering. In *Web and Requirements Engineering (WeRE)*, 2010 First International Workshop on the (pp. i-ii). IEEE.

Gemino, A (2004) Empirical comparisons of animation and narration in requirements validation. In *Requirements Engineering*, 9(3), pp. 153 - 168

Glinz, M. (2007, October). On non-functional requirements. In Requirements Engineering Conference, 2007. RE'07. 15th IEEE International (pp. 21-26). IEEE.

Gotel OCZ, Marchese FT, and Morris SJ (2007) On Requirements Visualization. In *Proceedings of the Second International Workshop on Requirements Engineering Visualization* (REV '07). IEEE Computer Society, Washington, DC, USA

Heim, P, Lohmann, S, Lauenroth, K, Ziegler, J. (2008) Graph-based Visualization of Requirements Relationships. In *3rd International Workshop on Requirements Engineering Visualization*, pp. 51 – 55. Doi: 10.1109/REV.2008.2

Horkoff, J., & Yu, E. (2010). Visualizations to support interactive goal model analysis. In *Requirements Engineering Visualization* (REV), 2010 Fifth International Workshop on (pp. 1-10). IEEE.

Höst M, Regnell B, Wohlin C (2000) Using Students as Subjects-A Comparative Study of Students and Professionals in Lead-Time Impact Assessment. *Empirical Software Engineering* 5(3): 201-214

Jaccard, J., Becker, M. A., & Wood, G. (1984). Pairwise multiple comparison procedures: a review. *Psychological Bulletin*, 96(3), 589.

H. V. Jackson Jr., "A Structured Approach for Classifying and Prioritizing Product Requirements.", North Carolina State University, Ann Arbor, 1999.

Kitchenham B (1997) DESMET: A method for evaluating Software Engineering methods and tools. In: *Computing & Control Engineering Journal* , 8(3), 120-126

Kitchenham B, Pfleeger SL, Pickard L, Jones P, Hoaglin DC, El Emam E, Rosenberg J (2002) Preliminary Guidelines for Empirical Research in Software Engineering. *IEEE Trans. Software Eng. 28(8):* 721-734

Lix LM, Keselman JC, Keselman HJ (1996) Consequences of assumption violations revisited: A quantitative review of alternatives to the one-way analysis of variance F test. *Rev. Educ. Res. 66*: 579-619

Matulevičius, R., Heymans, P. (2007) Visually Effective Goal Models Using KAOS. *Advances in Conceptual Modeling – Foundations and Applications (ER 2007 workshops)*, LNCS 4802, pp. 265 – 275, Doi: 10.1007/978-3-540-76292-8_32

Moody, DL, Heymans, P, Matulevicius, R. (2009) Improving the Effectiveness of Visual Representations in Requirements Engineering: An Evaluation of i* Visual Syntax. In *17th IEEE International Requirements Engineering Conference*, pp. 171 – 180, Doi: 10.1109/RE.2009.44

Morandini M, Perini A, Marchetto A (2011a) Empirical Evaluation of Tropos4AS Modelling. In*: 5th International i* Workshop*, pp. 14-19

Morandini M, Marchetto A, Perini A (2011b) Requirements comprehension: A controlled experiment on conceptual modeling methods. in *Empirical*

Requirements Engineering (EmpiRE), 2011 First International Workshop on. 2011

Mussbacher G, Amyot D, Arajo J, Moreira A, Weiss M (2007) Visualizing Aspect-Oriented Goal Models with AoGRL. In Proceedings of the *Second International Workshop on Requirements Engineering Visualization* (REV '07). IEEE Computer Society, Washington, DC, USA

Nasir, M. H. N., & Sahibuddin, S. (2011). Critical success factors for software projects: A comparative study. *Scientific research and essays*, 6(10), 2174-2186.

Nordbotten, J.C., Crosby, M. E. (1999). The effect of graphic style on data model interpretation. *Information Systems Journal,* 9(2), 139-155.

Pastor, O., Gómez, J., Insfrán, E., & Pelechano, V. (2001). The OO-Method approach for information systems modeling: from object-oriented conceptual modeling to automated programming. Information Systems, 26(7), 507-534.

Pohl K (2013). The Three Dimensions of Requirements Engineering. In J. Bubenko, J. Krogstie, O. Pastor, B. Pernici, C. Rolland, & A. Sølvberg (Eds.), *Seminal Contributions to Information Systems Engineering* (pp. 63-80). Springer Berlin Heidelberg. doi: 10.1007/978-3-642-36926-1_5

Rahimi M, Mirakhorli M, Cleland-Huang J (2014). Automated extraction and visualization of quality concerns from requirements specifications. *In Requirements Engineering Conference (RE), 2014 IEEE 22nd International (pp. 253-262*). http://doi.org/10.1109/RE.2014.6912267

Reddivari S, Rad S, Bhowmik T, Cain N, Niu N (2013). Visual requirements analytics: a framework and case study. *Requirements Eng* 19, 257–279. doi:10.1007/s00766-013-0194-3

Rohleder C (2008) "Visualizing the Impact of Non-Functional Requirements on Variants : A Case Study," *Requirements Engineering Visualization, 2008. REV '08.* , vol., no., pp.11,20, 8-8 Sept. 2008

Salado, A. and Nilchiani, R. (2015), Adaptive Requirements Prioritization (ARP): Improving Decisions between Conflicting Requirements. Syst. Engin., 18: 472–490. doi:10.1002/sys.21324

Savio D, Anitha P, Patil A, Creighton O (2012). Visualizing requirements in distributed system development. *In 2012 IEEE Second Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems (RES4)* (pp. 14-19). http://doi.org/10.1109/RES4.2012.6347690

Sayyad AS, Ammar H (2013). Pareto-optimal search-based software engineering (POSBSE): A literature survey. In 2013 *2nd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)* (pp. 21-27). http://doi.org/10.1109/RAISE.2013.6615200

Schwaber, K., & Beedle, M. (2002). Agile software development with Scrum (Vol. 1). Upper Saddle River: Prentice Hall.

SHAHIN, A; ZAIRI, M. Kano model: A dynamic approach for classifying and prioritising requirements of airline travellers with three case studies on international airlines. Total Quality Management & Business Excellence. 20, 9, 1003-1028, Sept. 2009. ISSN: 14783363.

Szidarovszky F, Gershon M, Duckstein L (1986) Techniques for Multiobjective Decision Making in *Systems Management.* Elsevier Science Ltd, 574

Teruel MA, Navarro E, Lopez-Jaquero V, Montero F, Gonzalez P (2011) An empirical evaluation of requirement engineering techniques for collaborative systems, *Evaluation & Assessment in Software Engineering (EASE 2011), 15th Annual Conference on* , vol., no., pp.114,123, 11-12 April 2011

Tukey J (1949) Comparing Individual Means in the Analysis of Variance, *Biometrics 5* (2): 99–114. JSTOR 3001913

Ugai T, Hayashi S, Saeki M (2010). Visualizing stakeholder concerns with anchored map (pp. 20-24). In *Requirements Engineering Visualization* (REV), 2010 Fifth International Workshop on. IEEE. doi:10.1109/REV.2010.5625662

Verner J, Cox K, Bleistein S, Cerpa N (2005) Requirements Engineering and Software Project Success: an industrial survey in Australia and the U.S. In: *Australian Journal of Information Systems*, 13(1), 225-238

Nguyen Xuan Thang and Kurt Geihs. 2010. Model-driven development with optimization of non-functional constraints in sensor network. In Proceedings of the 2010 ICSE Workshop on Software Engineering for Sensor Network Applications (SESENA '10). ACM, New York, NY, USA, 61-65.

Wieringa R., Maiden N., Mead N., and Rolland C. 2006. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. Requir. Engin. 11, 1, 102–107.

Yu ESK (1997) Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering. In: *3rd IEEE International Symposium on Requirements Engineering*, 226-235

Zhang Y, Harman M, Finkelstein A, Afshin Mansouri S (2011). Comparing the performance of metaheuristics for the analysis of multi-stakeholder tradeoffs in requirements optimisation. *Information and Software Technology*, 53(7), 761-773. doi:10.1016/j.infsof.2011.02.001

Zhang Y, Finkelstein A, Harman M (2008). Search Based Requirements Optimisation: Existing Work and Challenges. *In B. Paech & C. Rolland (Eds.), Requirements Engineering: Foundation for Software Quality* (pp. 88-94). Springer Berlin Heidelberg. doi:10.1007/978-3-540-69062-7_8