# Assessing the effectiveness of goal-oriented modeling languages: A family of experiments

Silvia Abrahão[a], Emilio Insfran[a,*], Fernando González-Ladrón-de-Guevara[b], Marta Fernández-Diego[b], Carlos Cano-Genoves[a], Raphael Pereira de Oliveira[c]

[a] Department of Computer Science, Universitat Politècnica de València
c/ Camino de Vera, s/n, València, Spain
{sabrahao, einsfran, carcage1}@dsic.upv.es

[b] Department of Business Organisation, Universitat Politècnica de València
c/ Camino de Vera, s/n, València, Spain
{fgonzal, marferdi}@omp.upv.es

[c] Federal Institute of Sergipe
Rua Café Filho, 260, Estância, Brazil
raphael.oliveira@ifs.edu.br

## ABSTRACT

*Context*: Several goal-oriented languages focus on modeling stakeholders' objectives, interests or wishes. However, these languages can be used for various purposes (e.g., exploring system solutions or evaluating alternatives), and there are few guidelines on how to use these models downstream to the software requirements and design artifacts. Moreover, little attention has been paid to the empirical evaluation of this kind of languages. In a previous work, we proposed value@GRL as a specialization of the Goal Requirements Language (GRL) to specify stakeholders' goals when dealing with early requirements in the context of incremental software development.

*Objective*: This paper compares the value@GRL language with the i* language, with respect to the quality of goal models, the participants' modeling time and productivity when creating the models, and their perceptions regarding ease of use and usefulness.

*Method*: A family of experiments was carried out with 184 students and practitioners in which the participants were asked to specify a goal model using each of the languages. The participants also filled in a questionnaire that allowed us to assess their perceptions.

*Results*: The results of the individual experiments and the meta-analysis indicate that the quality of goal models obtained with value@GRL is higher than that of i*, but that the participants required less time to create the goal models when using i*. The results also show that the participants perceived value@GRL to be easier to use and more useful than i* in at least two experiments of the family.

*Conclusions*: value@GRL makes it possible to obtain goal models with good quality when compared to i*, which is one of the most frequently used goal-oriented modeling languages. It can, therefore, be considered as a promising emerging approach in this area. Several insights emerged from the study and opportunities for improving both languages are outlined.

***Keywords*:** Requirements engineering, Goal modeling, GRL, i*, Controlled experiments.

[*] Corresponding author.

# 1. Introduction

The increasing complexity of software systems has led to the emergence of modeling languages with which to increase the understanding between stakeholders and software engineers. It has been recognized that goal-oriented modeling is useful to understand the organizational context of a system and the objectives that the system should achieve through cooperation among the actors in the intended software and its environment [1]. Goal models make it possible to specify why systems are being constructed by providing the rationale required to justify the need for the software requirements. The specification of goals also provides a criterion for requirements completeness, i.e., the requirements can be judged as complete if they are sufficient to establish the goals that they are refining [1].

Goal modeling is being used in projects in various domains (*e.g.*, data warehouses or security) and with a particular purpose (*e.g.*, reasoning or alternative selection). The corresponding modeling language, therefore, often needs to be extended in order to incorporate new modeling elements related to a particular domain or to adjust it to practical situations during early requirements modeling [2]. For example, the i* language [3] has been extended to support early requirements modeling in the domain of autonomic computing systems [4].

In a previous work [5], we extended the Goal Requirements Language (GRL) to deal with early requirements in the context of incremental software development[†]. GRL is a simplified variation of the i* framework [3] and, together with Use Case Maps (UCM), constitutes the URN (User Requirements Notation), which is an ITU-T international standard [6]. The variations made to the i* language were mainly motivated by the need to reduce its complexity and ambiguity, and to align objectives and intentions to business processes and scenarios as part of the standardization process. The contribution of our approach (value@GRL) is the specialization of GRL by using a subset of its modeling elements and a set of guidelines with which to model and prioritize intentional elements. The prioritization of intentional elements consists of determining which elements (i.e., goal, soft-goal, and task) are more important for a given actor. In this context, we consider a prioritized goal model to be a "value model" that can be used to select the business processes that will be included in a particular increment.

Although goal models have principally been used in requirements analysis, their usefulness may be enhanced if exploited during other phases of the software development process and used as part of the entire system lifecycle [7], e.g., architectural design, code development, and testing. Horkoff et al. [8] performed a systematic review of goal modeling languages in order to better understand how they can be integrated into downstream system development. They concluded that, although much work has been done in this area, the work is still fragmented, follows separate strands of goal-orientation, and is often in the early stages of maturity. Moreover, little attention has been paid to the empirical evaluation of this kind of languages. According to Carver et al. [9] and Campbell and Stanley [10], experiments in Software Engineering (SE) need to be replicated in different contexts, at different times and under different conditions before they can produce generalizable knowledge.

In this paper, we present a family of controlled experiments whose objective is to compare value@GRL and i* [3] with respect to the quality of goal models, the participants' modeling time and productivity when creating the models, and their perceptions regarding ease of use and usefulness after using both languages. The family consisted of a controlled experiment [11] and three replications

[†] Incremental software development involves breaking up the development plan into smaller, working pieces (i.e., increments). These increments are then developed, implemented, and tested.

carried out with students and practitioners. We selected i* because it is one of the most frequently used goal modeling languages [7], [12]. The results provide empirical evidence concerning the conditions in which these languages are most effective.

This study extends that of Abrahão et al. [11] by providing the following new contributions:

1. Three replications are presented. The value of replications has been widely recognized as a means of achieving a greater validity and reliability of experimental results [13], [14]. Here, the concept of replication has been extended to that of the 'family of experiments', in which multiple similar experiments that pursue the same goal were carried out.
2. The data analysis of individual experiments is presented in a unified manner. We have adopted the same analysis strategy for each experiment.
3. A meta-analysis aggregating the results from the individual experiments is presented.
4. A thorough discussion of the results is reported. The practical implications of our results are discussed from the perspectives of both practitioners and researchers.

This paper is organized as follows. In Section 2, we provide an overview of the two goal-oriented modeling languages being compared. We then discuss related literature concerning existing studies comparing goal-oriented languages. In Section 3, we present the family of experiments by providing an overview of the baseline experiment, along with the design and execution of the three replications. This section also highlights the differences among the experiments. In Section 1, we present the data analysis of the individual experiments, while the results of the family of experiments are discussed in Section 5. The threats to validity are discussed in Section 6, while Section 7 presents our conclusions and future directions.

## 2. Background and related work

As this study focuses on comparing two goal-oriented languages, we shall first introduce these two languages, after which we shall discuss existing studies comparing goal-oriented modeling languages.

### 2.1 The goal-oriented modeling languages compared

Several languages with which to model requirements have been proposed over the last 25 years. These languages employ different approaches, including scenario-based and goal-oriented modeling approaches. The main goal-oriented approaches discussed in literature include KAOS [15], GBRAM [16], NFR framework [17], i* [3], and variations of i* (e.g., GRL [6] or Tropos [18]). These approaches have been discussed in a survey of existing goal modeling languages [19]. The concept of goal is a first class entity in these languages, and is usually defined as a condition or state of affairs in the world that the stakeholders would like to achieve.

#### 2.1.1 i* (iStar)

The i* framework [3] was originally developed in order to model and reason about organizational environments and their information systems, which are composed of heterogeneous actors with different and possibly competing goals. We have employed [3] and [20] to summarize the main concepts of i*, which are: actors, intentional elements, intentional links, and dependencies.

*Actors* can be humans, hardware, software, or combinations thereof. The central idea of i* is that actors depend on each other for goals to be achieved, for resources to be provided, for tasks to be performed, and for soft-goals to be satisfied. An actor can be classified as:

- *Role*, which represent an abstract characterization of the behavior of a social actor within a particular specialized context or domain of endeavor.
- *Agent*, which represent an actor with concrete, physical manifestations, such as a human individual.
- *Position*, which represent a set of roles typically played by one agent. We say that an agent occupies a position and a position is usually said to cover a role.

In addition, actors are often not isolated and may be linked through actor links: plays, is-part-of, or is-a, which represent the concepts of responsibility, composition, and inheritance, respectively.

*Intentional elements* are used to represent the actors' intentionality within their boundary. The boundary accurately delineates what is under the actor's control; whateer needs that are not inside the boundary must be fulfilled in collaboration with other actors through dependencies. Five types of intentional elements can be defined:

- *Goal*, which represents a state of the world that is sought to be achieved.
- *Soft-goal*, which represents a goal whose fulfillment is not clear-cut; instead, its satisfaction condition is subject to interpretation. This subjectivity is the difference between a goal and a soft-goal.
- *Task*, which represents an activity whose execution is prescribed according to certain established procedures.
- *Belief*, which represents a condition about the world that an actor holds to be true. The difference between a goal and a belief is that the latter is not a condition that an actor wishes to achieve.
- *Resource*, which represents a physical or intentional entity that is produced or provided by an actor.

*Intentional links* are used to connect the intentional elements of cooperating actors. There are three types:

- *Means-end link,* which offer a way in which to identify alternative means to achieve a goal.
- *Decomposition link,* which allow tasks to be decomposed into simpler intentional elements.
- *Contribution link,* which express how intentional elements contribute to the satisfaction of a soft-goal. Contribution can be positive or negative, and can be an implication or simply a connection, yielding seven types of contribution links (Make, Some+, Help, Unknown, Break, Some-, and Hurt).

*Dependencies* are connections between actors. One of them, denominated as the *depender*, depends on a second actor, denominated as the *dependee,* for the accomplishment of a particular internal intention. The dependency is characterized by an intentional element (*dependum*), which represents the reason for dependency. Dependencies can be defined for goals, soft-goals, tasks and resources.

Finally, in order to deal with large models, i* proposes two kinds of models:

- The *Strategic Dependency model (SD)*, which depicts external relationships among actors, while remaining silent as regards the internal makeup of the actors.
- The *Strategic Rationale model (SR)*, which allows the goals, tasks, resources, and soft-goals of each actor to be modeled as internal elements to be achieved.

Fig. 1 shows an excerpt from the i* goal model (Strategic Rationale) for the Green Route system, which was taken from Estrada et al. [21]. This application proposes the ideal route for a user, avoiding

routes with high levels of pollution, floods, pollen, etc., thus making it possible to, for instance, obtain the preferred routes for people with respiratory diseases.

This goal model includes four actors: *User*, *Green Route*, *FIWARELab*, and *geographic information system (GIS)*. The *User* actor has the goal of discovering the best route. A *User* could achieve his/her main goal by providing his/her profile, historical information, and real time environmental data. The *Green Route* actor's main goal is to determine the best route for the *User* actor. To achieve this goal, the *Green Route* actor uses the *User'*s available information and requests additional environmental and geographical data from the FIWARE Lab and GIS actors, respectively. In this i* model, several external goals and resources (outside the actors' borders) are required in order to attain the *User*'s main goal (e.g., the resource *User information*, the goal *Determine route based on a user profile*).

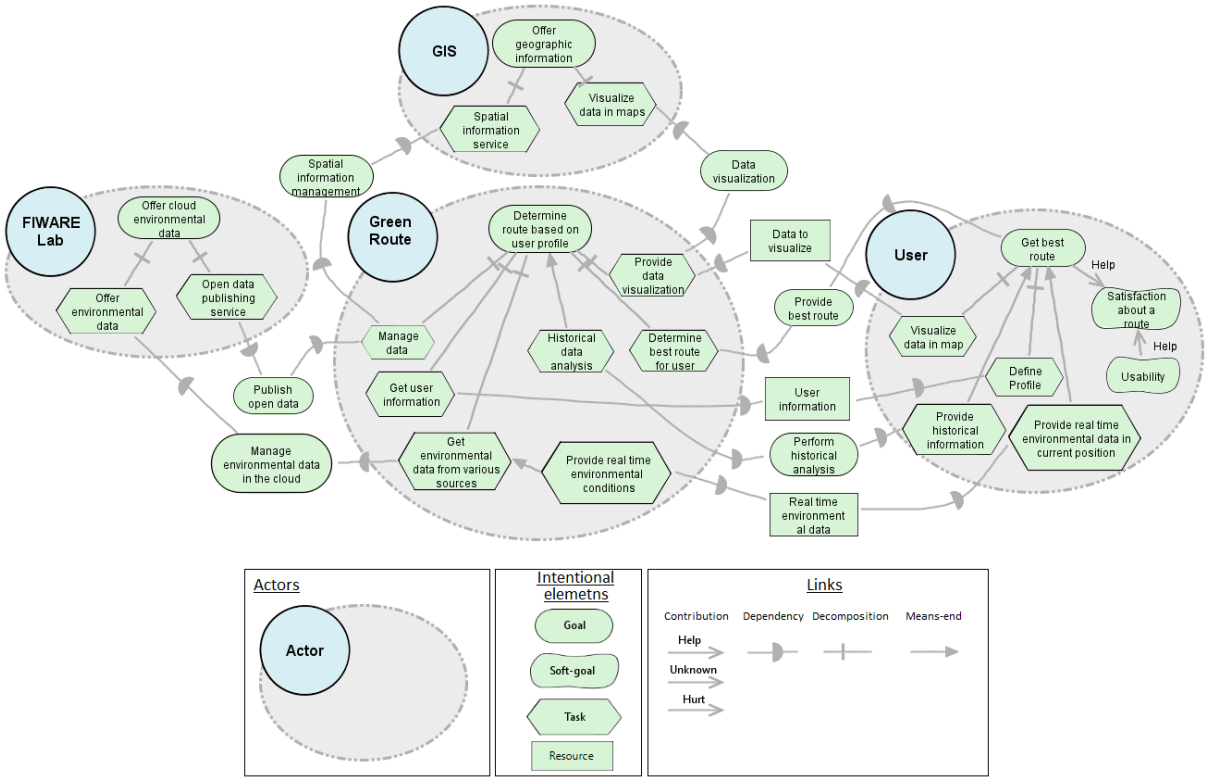



**Fig. 1.** i* goal model for the Green Route system.

### 2.1.2 Value@GRL

The *value at Goal-Oriented-Requirements Language* (value@GRL) is a specialization of the GRL language created in order to support the modeling of goals when dealing with early requirements in the context of incremental software development. This specialization consists of using a subset of the GRL modeling elements and providing guidelines for the modeling and prioritization of the intentional elements of the stakeholders involved. We consider a prioritized goal model to be a *value model*, i.e., a model that represents intentional elements that have a prioritized value from a stakeholder's point of view. This value model may be used to prioritize business processes and features that will be included in a given increment as part of an incremental software development process.

The process employed to model with value@GRL includes four main activities: goal modeling, goal model prioritization and propagation, high-level business process modeling and business process prioritization.

It is recognized that nearly half (47%) of unsuccessful software development projects fail to meet their goals owing to inaccurate requirements management [22]. In addition, the quality of a software system depends to a great extent on the degree to which it fulfills its requirements [7]. These requirements are often captured, modeled and analyzed as (stakeholder) goals [7]. It is, therefore, important to ensure the quality of goal models, as these models may be used to provide a criterion for requirements completeness. They can also be used as input to the forthcoming software development activities (*e.g.*, business process modeling, conceptual modeling, software design). In this work, we consequently focus on the first activity in the value@GRL modeling process by comparing the goal models obtained with value@GRL and i*. For more details regarding the other activities in the value@GRL process, please refer to [5].

Value@GRL is oriented toward specifying the stakeholders' interests with regard to the system to be developed. There are three categories of concepts in value@GRL: actors, intentional elements and intentional links.

*Actors* represent entities (stakeholders or systems) in the domain of interest, which have intentions and may perform actions to achieve their objectives. This concept is similar to that defined for i* (see Section 2.1.1). However, we identify three different types of actors:

- *Main*, which represents the stakeholder that will drive the specification of the goal model. This is the main stakeholder for which the system is to be developed. This actor is labeled with the tag «main».
- *External*, which represents collaborators or affected stakeholders who have goals that the system actor may take into consideration in order to satisfy its own values. These actors are labeled with the tag «external».
- *System*, which represents the system to be developed, including the set of goals and operations needed to satisfy the objectives of the actors involved. This actor is labeled with the tag «system».

The different types of actors help identify the boundaries of the system to be developed for the *main* actor and, how the *external* actors will assist the *system* actor in achieving its goals.

*Intentional elements* describe an actor's intention and capabilities. This concept is similar to that defined for i* in Section 2.1.1. However, we consider only three types of elements: *goal*, *soft-goal*, and *task*, and they are always represented inside the boundary of a given actor.

*Intentional links* are used to relate intentional elements to each other. This concept is similar to that defined for i* in Section 2.1.1. However, we consider only the *decomposition* and *contribution* links with the same semantics as i* (see Section 2.1.1). In addition, we also consider the *dependency* link in this category, since it allows us to establish relationships among intentional elements but from different actors (see Section 2.1.1).

Fig. 2 shows an excerpt of the goal model using value@GRL for the Green Route system, which was previously modeled with i* and was introduced in Section 2.1.1. The goal model includes four actors: the *User* (main actor) interested in discovering the best route; the *Green Route* (system actor) interested in determining the optimal route based on the user profile and the characteristics of the route; the *FIWARELab* (external actor) interested in providing access to environmental data and

publishing open data for other users; and finally, a *GIS* (external actor) that provides geographical information regarding the routes. The *Green Route system* (system actor) must explicitly take into account these actors' goals in order to know which one of them will be considered during the development of the software system, and to what extent and priority.

Some of the most noteworthy syntactical differences between value@GRL and GRL are: it distinguishes among different types of actors (system, main and external); considering only three types of intentional elements (goal, soft-goal, and task); and it uses only three types of intentional links (contribution, decomposition, and dependency). Despite the syntactical similarities between value@GRL and GRL (and also i*), the main difference lies in the purpose of modeling, which is to represent the intentional elements for actors regarding the software system to be developed. This purpose led us to reduce the number of modeling elements to be used by focusing only on those intentional elements and intentional links that affect or are affected by the software system to be developed.

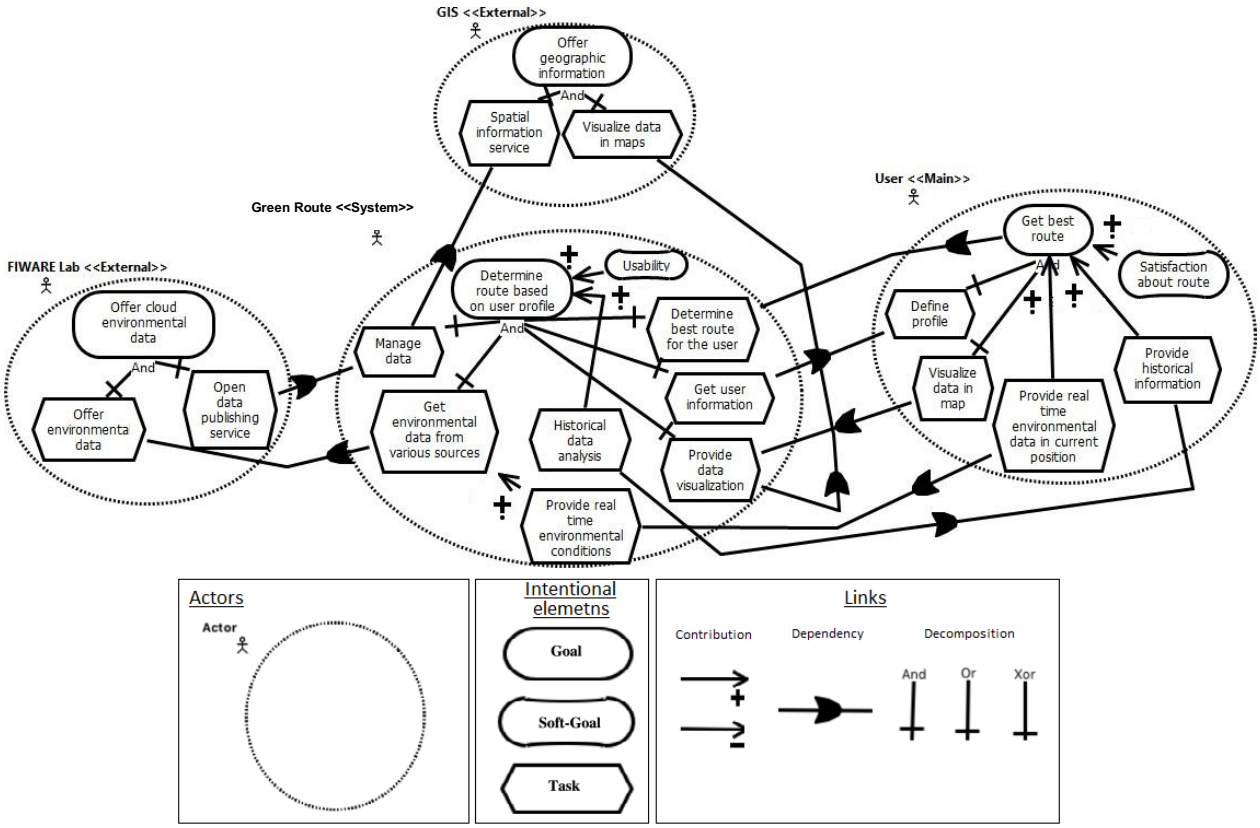



**Fig. 2.** value@GRL goal model for the Green Route system.

## 2.2 Existing studies comparing goal-oriented languages

Much attention has been paid to the area of goal-oriented Requirements Engineering (RE) [2], [8] in the last two decades, and several studies that compare goal-oriented languages have been published. According to Siau and Rossi [23], these studies can be classified in three groups: feature comparison [19], [24], [25], [26], theoretical and conceptual evaluations [27], [28], [29] and empirical studies [12], [30], [31], [32], [33].

### 2.2.1 Feature comparison

The first group of studies is based on ‘‘feature comparison’’, *i.e.*, a comparison of goal-oriented languages according to a certain set of criteria. One example is the study performed by Kavakli and Loucopoulos [19] in which the authors selected 15 goal modeling languages and classified them according to four dimensions: usage (regarding the objectives of using goal modeling in RE), subject (revealing the notion of a goal and its nature), representation (regarding how goals are represented), and development (concerning how goal models are created and evolve). This comparison provides a broader view of the goal-oriented modeling area.

Regev and Wegmann [24] compared several meanings of goal and related concepts presented in GRL, KAOS, and GBRAM. They proposed a set of principles to explain the goal-oriented behavior and established more precise definitions for goal-oriented languages. As a result, new concepts for goal-oriented RE terms were defined, such as: achievement goal, maintenance goal, soft-goal and belief. Although this was a first step toward defining more precise definitions, more research is still required in order to study the relationships among norms, beliefs, and goals.

Horkoff and Yu [25] surveyed available approaches for goal-oriented modeling (e.g., i*, GRL, Tropos, NFR, KAOS) and classified them according to several criteria (*e.g.*, satisfaction analysis, metrics, planning, simulation). They also proposed guidelines that would assist in the use of these approaches, grouped into domain understanding, communication, model improvement, scoping, requirement elicitation, requirements improvement, and design, but the proposed guidelines need to be validated in practice.

Teruel et al. [26] performed a comparison of three goal-oriented approaches (i.e., the NFR framework, i*, and KAOS) to determine which is the most suitable to model requirements for Computer Supported Cooperative Work systems (CSCW). The evaluation of the approaches was carried out by using 9 features, including: functional and non-functional requirements representation; collaborative systems characteristics; awareness representation; importance of requirements; model complexity and traceability. These features are based on the DESMET evaluation framework. They were evaluated and a score was computed for each feature and goal-oriented approach. According to the results obtained, the approaches analyzed are not fully appropriate to model collaborative system characteristics. i* is the only approach that attained a positive score for the features related to collaborative systems analyzed.

Overall, the drawback of this type of studies is their subjectivity when developing the comparison criteria and their interpretation.

### 2.2.2 Theoretical and conceptual evaluation

The second group of studies that compares goal-oriented languages comprises theoretical and conceptual evaluation, which includes: (1) metamodeling – comparing the languages by mapping them onto an abstract language; (2) metric analysis, which focuses on analyzing the aspects of the languages as regards their complexity; and (3) ontological evaluation, which focuses on matching the languages with ontological constructs.

With regard to the metamodeling category, Ayala et al. [27] compare i*, GRL, and Tropos with the objective of defining a reference framework based on noises, silences, ambiguities, and the contradictions of these languages. These languages were compared according to fourteen criteria (e.g., types of models, types of actors, intentional elements, relationships) and a metamodel with which to

embrace the commonalities was proposed. The findings are useful for identifying characteristics and guiding the selection of these languages.

In the case of the metrics analysis category, Al-Subaie and Maibaum [28] performed a qualitative evaluation of KAOS and its supporting tool, Objectiver. The method's effectiveness was measured as the degree of coverage of KAOS in relation to RE objectives (e.g., pertinence, correctness, traceability, and understandability). An objective that was "fully achieved", therefore, scored A; otherwise, it scored E, if it "failed to be achieved". Although this study provides an initial qualitative evaluation of KAOS, the authors did not perform any statistical analyses.

With regard to ontological evaluations, Matulevičius et al. [29] compared the syntax and semantics of GRL and KAOS using the Unified Enterprise Modeling Language (UEML) approach. The authors defined the semantics of both languages on top of the UEML ontology and provided a path toward automated transformations with which to translate GRL into KAOS models, and vice-versa. One drawback of this study is that the proposed semantics still need to be evaluated with users of these languages.

### 2.2.3 Empirical studies

The third group comprises empirical studies that include individual controlled experiments or a family of experiments, whose goal was to compare two or more goal modeling languages empirically. Table 1 lists the studies reviewed, including information on the languages compared, the type of participants, the sample size, the measures, and the conclusions.

**Table 1.** Summary of studies comparing goal-oriented modeling languages.

| Study | Languages | Type of participants | Sample size | Estimated construct | Main conclusions |
|---|---|---|---|---|---|
| [12] | - i*<br>- KAOS | − Undergraduate students | 19 | − Quality | KAOS had a higher quality to create models, although i* goal models had more quality. |
| [30] | - i*<br>- CSRML (i* extension) | − Undergraduate students<br>− PhD students | 84 | − Understandability | CSRML improves the understandability of CSCW requirements models when compared to i*. |
| [31] | - i*<br>- KAOS | − Undergraduate students | 38 | − Understandability | The understandability of i* is higher than that of KAOS for modeling TR systems. |
| [32] | - i*<br>- TRiStar (i* extension) | − Undergraduate students<br>− Software developers | 69 | − Understandability Effectiveness<br>− Understandability Efficiency | TRiStar has a higher effectiveness and efficiency than i* for specifying TR systems requirements. |
| [33] | - i*<br>- i* variant (with modules) | Non-experts data warehouse<br>Experts on i* | 49 | Understandability<br>Manageability | The i* variant (with modules) increases the modularity and scalability of the models which, in turn, increases the error correction capability, and makes complex models easier to understand. |
| [11] | - i*<br>- value@GRL (GRL specialization) | − Master CS students<br>− Master Business Management students | 40 | − Quality<br>− Productivity<br>− Perceived Ease of Use<br>− Perceived Usefulness | value@GRL obtained goal models with a higher quality than i* although their productivity is similar. Participants perceived value@GRL to be easier to use and more useful than i*. |

Matulevičius and Heymans [12] performed an empirical study that compared i* and KAOS in order to discover which language was of better quality. The authors adapted the semiotic quality framework [34] to evaluate the quality of the language used to create models and the quality of the models created by the languages. The results showed that KAOS had a higher quality to create models (although the statistical tests were not significant) and i* goal models had a better quality. In addition, there is a lack of methodological guidelines with which to assist users in using the languages.

Teruel et al. [30] proposed an i* extension called CSRML (Collaborative Systems Requirements Modeling Language) and performed a family of three experiments to analyze the understandability of RE languages for CSCW (Computer Supported Cooperative Work) systems. The goal was to test which language (i* or CSRML) has a better understandability to model CSCW systems' requirements. The authors measured understandability using a comprehension questionnaire. According to their results, CSRML improves the understandability of CSCW requirements models when compared to i*.

Morales et al. [31] evaluated i* and KAOS to determine their understandability levels when specifying Teleo-Reactive (TR) systems. They performed a controlled experiment in which understandability was measured by employing true/false questionnaires regarding two TR systems specified with both languages. The results showed that i* has a better understandability than KAOS when modeling reactive systems requirements.

In a similar study, Morales et al. [32] reported a family of three experiments whose objective was to evaluate the understandability when modeling TR systems with i* and TRiStar, which is an approach proposed by the authors. Two variables were used to evaluate the models' understandability: effectiveness (the number of correct answers attained by the subjects) and efficiency (the number of the subjects' correct answers divided by the time needed to understand a TR diagram). The results show that TRiStar has both a higher effectiveness and efficiency as regards specifying TR systems requirements when compared to i*.

Finally, Maté et al. [33] proposed the inclusion of modules in i* to improve the goal-oriented analysis for data warehouse systems. These modules are included in i* according to a set of guidelines. The authors evaluated their proposal by performing two questionnaire-based experiments, the first carried out within 28 participants and the second with 21 participants. According to their results, even when applying modularity concepts, the scalability of models increased, as did the time required to perform different tasks on the models. Furthermore, they reported a reduced error rate when identifying the scope of an element present in the model. Finally, their results showed that most participants had a tendency to group elements into packages at different levels of abstraction, so as to avoid adding them to a global scheme.

### 2.2.4 Discussion

Most of the evaluations performed in the goal modeling area to date are studies that compare characteristics of the languages. These studies provide a global view of the goal modeling languages and their characteristics. However, empirical evidence is required in order to understand which language is better in a given context.

To the best of our knowledge, only a few studies [12], [30], [31], [32], [33] have performed an empirical evaluation when comparing goal-oriented approaches. This information coincides with the recent results of a systematic mapping of goal-oriented RE approaches [8], in which, of a set of 246 papers, only 7% of them presented evaluations in the form of controlled experiments. Most of the studies compared i* with another language, but these results are not conclusive, suggesting that more

experimentation is needed. Overall, the participants in the studies reviewed are mainly students and the sample size of the experiments is small. What is more, most of the studies used only one or two measures to assess the language's effectiveness.

In order to improve the body of knowledge concerning goal modeling approaches, we performed an experiment [11] to evaluate the quality, productivity, perceived ease of use and perceived usefulness of i* and value@GRL. Unlike other experiments, we involved software engineers and business analysts who are the typical users of these languages. The goal of the current study is to validate the results of the previous experiment [11] by performing three replications in different settings and a meta-analysis that aggregated the empirical findings obtained in the individual experiments.

# 3. The family of experiments

A family of experiments is useful to answer questions that are beyond the scope of individual experiments and permits the generalization of findings from various studies, thus providing evidence with which to confirm or reject specific hypotheses [13]. We, therefore, conducted a family of experiments to compare value@GRL and i*.

## 3.1 Goal

On the basis of the Goal-Question-Metric (GQM) template [35], the goal of our family of experiments was to *analyze* goal models specified with value@GRL and i* *for the purpose of* assessing them *with respect to* the quality of the resulting models, the participants' modeling time, productivity, perceived ease of use, and perceived usefulness *from the point of view of* novice software engineers and business analysts and software industry professionals *in the context of* Undergraduate and Master's degree students in Computer Science (CS) and Master's degree students in Business Management.

Although experienced modelers and practitioners would have been preferable, we focused on the profile of novice goal modelers since one of our objectives is to provide a goal language that will help less experienced modelers to specify high-quality models. The research questions addressed are:

- RQ1: Which language allows modelers to create goal models with a higher quality?
- RQ2: Which language allows modelers to be faster?
- RQ3: Which language allows modelers to be more productive?
- RQ4: Which language is perceived to be easier to use?
- RQ5: Which language is perceived to be more useful?

## 3.2 Context selection

The context of this study is the specification of two goal models created by novice software engineers and business analysts, and software professionals. The context is defined by (i) the goal-oriented languages selected, (ii) the experimental objects (i.e., goal models to be specified); and (iii) the selection of participants.

### 3.2.1 Goal-oriented languages compared

We compared value@GRL and i*, which is one of the most frequently used goal-modeling languages [7], [12]. We focused on the first activity of value@GRL (i.e., goal modeling), which is concerned with the specification of a goal model. This activity corresponds to the main purpose of i*. Specifically, we focused on the i* Strategic Rationale (SR) model because it shows all the internal

elements of actors, including goals, soft-goals, tasks, and resources that contribute to the analysis of alternatives and the fulfillment of dependencies. Note that i* is undergoing standardization and that an updated version of the language, denominated as i* 2.0 [36] has been recently proposed. This version has minor differences with respect to the seminal one (e.g., richer types of contribution links). However, i* 2.0 was released at the time when the pilot study of our family was executed (May 2016). The baseline experiment was executed in November 2016 (see Fig. 3). In this work, we, therefore, used the original version of i*, as at that time there was still no evidence regarding the use and likely adoption in practice of the new version.

### 3.2.2 Experimental objects

The goal models to be specified were selected from requirements engineering literature:

- *O1 – Green Route* [21]: the purpose of this system is to help a user determine the best route to follow to reach a destination, taking into account the user profile (e.g., health conditions, disabilities) and preferences. This system was presented in Section 2.1.1.
- *O2 – Lattes Scholar* [37]: the purpose of this system is to present publications from an author's curriculum and their citations by searching in the Brazilian Scientific CV repository, entitled Lattes, and the Google Scholar databases.

### 3.2.3 Participants selection

The following groups of participants were identified in order to facilitate the generalization of results:

- *Novice Software Engineers*: 28 Master's degree students enrolled on a Computer Science Master's degree program at the Department of Computer Science, Universitat Politècnica de València (UPV), Spain, and 124 undergraduate students, all Computer Science students at the UPV.
- *Novice Business Analysts*: 12 students enrolled on a Master's degree in Business Management at the Faculty of Business Administration and Management at the UPV.
- *Software Industry Professionals*: 20 software designer and developer practitioners who participated in a professional Master of Science (MSc) degree program in Software Engineering at the National University of Asunción (UNA) in Paraguay.

The participants included in this study were selected by means of convenience sampling. Since we focused on the profile of novice modelers, we selected groups of participants with no previous knowledge in goal modeling. Nevertheless, we verified this assumption by means of a pre-questionnaire intended to determine the respondents' demographics and experience with goal modeling. All the participants were volunteers and were aware of the practical and pedagogical purposes of the experiment, but the research questions were not disclosed to them. The participants were not rewarded for their effort.

## 3.3 Design of individual experiments

Fig. 3 summarizes our family of experiments, including the context of each experiment, the number of participants involved, and the place where the experiments took place. The figure also shows the execution order of the experiments.

Since experimental conditions are hard to control in Software Engineering, one way in which to satisfy the statistical requirement of replications is that of running internal replications (in the same place and by the same experimenters) [38]. Having more internal replications of the same experiment considerably reduces the Type I error, and identical replications are also required to be able to

estimate the effect size under study [38]. A Type I error (α-error, false positives) occurs when the null hypothesis ($H_0$) is rejected in favor of the alternative hypothesis ($H_1$), when the 'null' hypothesis is actually true. The effect size indicates the magnitude of the observed effect or relationship between variables.

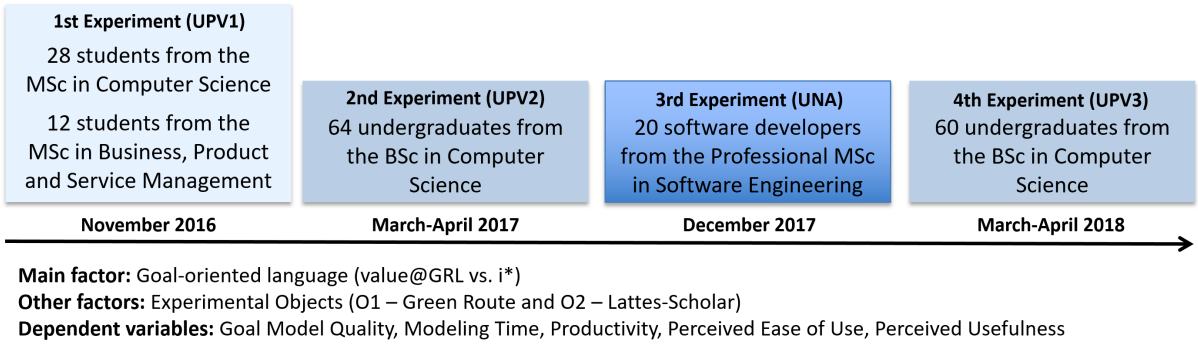



**Fig. 3.** Overview of our family of experiments.

Our family was, therefore, composed of the baseline experiment that was conducted at the Universitat Politècnica de València (UPV1) and three differentiated internal replications (i.e., UPV2, UNA and UPV3) performed in different settings. These replications were operational, as we varied some dimensions of the experimental configuration [39]. In UNA and UPV3, we varied the experimenter and the population, while only the population was varied in UPV2. This allowed us to verify whether the results are independent of the participants' profile and the experimenters.

In the following, we present the experiments, which were designed according to the experimental process proposed by Wohlin et al. [40]. With regard to the replications, we discuss only their differences with respect to the baseline experiment. For replication purposes, the experimental materials can be found at https://goo.gl/wMPH1e.

### 3.3.1 Baseline experiment (UPV1)

The aim of the experiment was to evaluate whether participants applying our proposed language in order to create goal models (i.e., value@GRL) may obtain higher performances and report better perceptions than when using i*.

#### 3.3.1.1 Context selection

We used both the systems described in Section 3.2.2 as experimental objects and applied the goal modeling languages described in Section 3.2.1. The system requirements and the corresponding goal models (taken from literature) can be found on the website, along with the experimental materials. These models were validated by an assessment group composed of one independent expert on goal modeling and two of the authors of this paper.

According to Systems Theory [41], the complexity of a system can be assessed by considering the number of different types of elements and the number of different types of relationships between them. The system complexity also depends on the problem domain being represented, because it can influence the participants' understandability.

The two object systems we chose are from two different application domains that do not require specialized knowledge to understand them. These systems have a similar complexity, based on the metrics shown in Table 2. The assessment group was also responsible for judging the complexity of

the solutions for both systems, and considered them to be comparable in terms of the number of elements and relationships, and the problem domain.

Table 2. Description of the systems.

| | #Actors | #Goals | #Soft-Goals | #Tasks | #Resources | #Links |
|---|---|---|---|---|---|---|
| Green Route (i*) | 4 | 10 | 2 | 15 | 3 | 25 |
| Green Route (value@GRL) | 4 | 4 | 2 | 15 | - | 26 |
| Lattes Scholar (i*) | 5 | 6 | 6 | 18 | 9 | 34 |
| Lattes Scholar (value@GRL) | 5 | 5 | 8 | 18 | - | 34 |

#### 3.3.1.2 Participants

The experiment included 40 participants with two different profiles:

- **Novice software engineers:** 28 Master's degree students enrolled on a Software Engineering Master's degree program at the Dept. Computer Science, UPV, Spain. A pre-questionnaire was administered to the participants in order to assess their experience. 18 participants reported that they had professional experience in software development, varying between 2 to 6 years, with an average of 3 years, but that they had no previous knowledge of goal modeling. The participants were chosen by means of convenience sampling. They attended the Fall 2016 course on Empirical Software Engineering with a focus on comparing modeling approaches. The participants were asked to carry out the experimental task as part of the laboratory exercises of the course.
- **Novice business analysts:** 12 students enrolled on a Master's degree in Business, Product and Service Management at the Faculty of Business Management, UPV, Spain. They had competencies and skills in modeling business processes involving organization areas and functions such as Logistics and Operations, Marketing, and Finances, using tools such as Quality Function Deployment to design products and services. Since they would need to interact with other organization specialists, they were motivated to communicate with IT specialists and software developers. Indeed, they attended a course on Information Systems in Organizations and were invited to carry out the experimental task as part of a workshop on goal modeling.

#### 3.3.1.3 Selection of variables

The main independent variable (or factor) was Language, which was a nominal variable that could assume two possible values: value@GRL and i*. The secondary independent variable is the experimental object, with two possible values: Green Route and Lattes Scholar.

There are two types of dependent variables: performance-based and perception-based variables. Performance-based variables assess how well the participants perform the experimental task. In our case, these variables were: quality of goal models, modeling time and productivity.

Several quality evaluation frameworks with which to evaluate conceptual models have been proposed in literature (*e.g.*, [42], [43]). The Lindland et al. [43] framework suggests that a systematic evaluation of quality considers a model's syntax (how well the model adheres to the rules of the modeling language), semantics (how well the model reflects the reality modeled) and pragmatic (how

well the model is understood and used). In this study, we deal with the first two dimensions of Lindland's framework: syntactic and semantic quality.

The **Quality** variable assesses the syntactic and semantic quality of goal models created with value@GRL and i* in terms of *correctness* (whether the model conforms to the rules of the language) and *completeness* (whether the model contains all the correct modeling elements required to represent the stakeholders' goals). This variable was measured by using an information retrieval based approach [44] that has been used in other SE experiments [45], [46], [47], and [48] to compare models created by participants with an Oracle (the correct set of models created by an expert) regarding each type of graphical elements (e.g., actors, goals, soft-goals, tasks and links). This was done by employing Equations (1) and (2):

$$precision_{element} = \frac{|P_{element} \cap O_{element}|}{|P_{element}|}, \quad (1)$$

$$recall_{element} = \frac{|P_{element} \cap O_{element}|}{|O_{element}|}, \quad (2)$$

where $P_{element}$ indicates graphical elements of a given type modeled by a participant and $O_{element}$ indicates the known correct set of expected type of graphical element that belongs to an Oracle. Accordingly, $precision_{element}$ measures the correctness of a graphical element belonging to a given goal model and $recall_{element}$ measures the completeness of a goal model with respect to its graphical elements. Precision and recall quantitatively summarize two dimensions of the quality of a model. We, therefore, used their harmonic mean [44] to attain a balance between the correctness and completeness of each graphical element within a goal model by employing Equation (3):

$$F - Measure_e = 2 * \frac{precision_{element} * recall_{element}}{precision_{element} + recall_{element}} * 100, \quad (3)$$

where $F\text{-}Measure_e$ summarizes the accuracy of each graphical element in the goal model regarding its graphical elements when compared with an Oracle. Quality is then computed as the arithmetic mean of all the F-Measures for the different types of graphical elements in the model. All the measures above assume values of between 0% and 100%. Whatever the measure is, values close to 100% mean that the participants' goal models were very similar to the Oracle. Conversely, values close to 0% indicate that the goal models were very dissimilar to the Oracle.

The quality variable was defined in order to give the same relevance to the correctness and completeness of goal models with respect to the actors, objectives and tasks, and the links among them. In order to show how the defined measure works, we provide an example of its calculation. Fig. 2 shows a goal model that corresponds to the first Oracle for the value@GRL experimental object 1. Table 3 shows a summary of $O_{elements}$ and $P_{elements}$, the intersection between $O_{elements}$ and $P_{elements}$, *precision*, *recall*, the $F\text{-}Measure_e$ for each type of element, and the mean of all F-Measures, which is the quality of the modeled solution produced by the participant. The first Oracle for value@GRL was evaluated by the assessment group before the experiment (one for each object).

For i*, the first Oracle was extracted from [21], [37]. Fig. 1 shows a goal model that corresponds to the first Oracle for the i* experimental object 1. Table 4 shows a summary of $O_{elements}$ and $P_{elements}$, the intersection between $O_{elements}$ and $P_{elements}$, *precision, recall,* the $F\text{-}Measure_e$ for each type of element, and the mean of all F-Measures, which is the quality of the modeled solution produced by the participant.

**Table 3.** Example of quality calculation for Oracle 1 of the value@GRL experimental object 1.

| value@GRL elements for Oracle 1 | $O_{elements}$ in Oracle 1 | $P_{elements}$ | $P_{elements} \cap O_{elements}$ | *precision* | *recall* | $F\text{-}Measure_e$ |
|---|---|---|---|---|---|---|
| Actor | 4 | 5 | 4 | 0.80 | 1.00 | 88.88 |
| Goal | 4 | 6 | 4 | 0.66 | 1.00 | 80.00 |
| Soft-goal | 3 | 4 | 1 | 0.25 | 0.33 | 28.57 |
| Task | 14 | 13 | 11 | 0.84 | 0.78 | 81.48 |
| Positive contribution | 4 | 9 | 3 | 0.33 | 0.75 | 46.15 |
| Dependency | 9 | 6 | 1 | 0.16 | 0.11 | 13.33 |
| And | 14 | 8 | 0 | 0.00 | 0.00 | 0.00 |
| Or | 1 | 1 | 1 | 1.00 | 1.00 | 100.00 |
| *Mean of all F-Measures* | | | | | | 54.80 |

Since the first Oracle for both value@GRL and i* might have been biased by the modelers' expertise and a goal model can have several correct solutions, we only considered these first Oracles as a baseline, which may evolve by adding new correct solutions provided by the participants.

The assessment group was, therefore, also responsible for determining whether the goal models created by the participants matched one of the Oracles, or whether a new Oracle should be added to the baseline (thus increasing the total number of correct solutions for a particular experimental object). Disagreements among the assessment group members were resolved by consensus. In order to assess the quality of a goal model provided by a participant, the *F-measure* of his/her solution was, therefore, calculated by considering all the possible Oracles on the baseline, and the highest result was selected.

**Table 4.** Example of quality calculation for Oracle 1 of the i* experimental object 1.

| i* elements for Oracle 1 | $O_{elements}$ in Oracle1 | $P_{elements}$ | $P_{elements} \cap O_{elements}$ | *precision* | *recall* | $F\text{-}Measure_e$ |
|---|---|---|---|---|---|---|
| Actor | 4 | 2 | 2 | 1.00 | 0.50 | 66.66 |
| Goal | 10 | 2 | 2 | 1.00 | 0.20 | 33.33 |
| Soft-goal | 2 | 3 | 2 | 0.66 | 1 | 80.00 |
| Task | 15 | 7 | 6 | 0.85 | 0.40 | 54.54 |
| Resource | 3 | 2 | 1 | 0.50 | 0.33 | 40.00 |
| Help | 2 | 2 | 0 | 0.00 | 0.00 | 0.00 |
| Dependency | 9 | 4 | 3 | 0.75 | 0.33 | 46.15 |
| Decomposition | 11 | 5 | 3 | 0.60 | 0.27 | 37.50 |
| Means-end | 3 | 1 | 0 | 0.00 | 0.00 | 0.00 |
| *Mean of all F-Measures* | | | | | | 39.79 |

The **Modeling Time** variable was measured as the total time (in minutes) taken by a participant to create a goal model using a particular language.

The participants' **Productivity (PROD)** was calculated as the ratio between the quality of the goal model and the time taken to apply the language (Quality / Modeling Time). This measure is related to the timing of the modeling task, but also reflects the ability to create a goal model correctly and completely. A higher value of this measure reflects better productivity. Quality is measured in terms of percentages while time is measured in minutes. As an example, if a participant scored 100% in quality and took 20 minutes to do so, the participant's productivity is 5 (i.e., PROD = 100 / 20).

Furthermore, two perception-based variables were employed to assess the participants' perceptions of their performance when using value@GRL or i*. These variables are based on the Technology Acceptance Model (TAM) [49], which is a widely applied and empirically validated model [50]:

- **Perceived Ease of Use (PEOU):** the degree to which modelers believe that using a goal modeling language will be effort-free.
- **Perceived Usefulness (PU):** the degree to which modelers believe that using a specific goal modeling language will increase their job performance within an organizational context.

Table 5 shows the items defined to measure the perception-based variables. The items defined for each construct were combined in a survey, consisting of nine questions. The items were formulated by using a 5-point Likert scale, ranging from 1 (strongly disagree) to 5 (strongly agree), using the opposing-statement question format. Various items within the same construct group were randomized to prevent systemic response bias [51]. The aggregated value of each subjective variable was calculated as the mean of the answers to the variable-related questions. The questionnaire also included open questions with which to obtain feedback from the participants. The survey is available at https://goo.gl/wMPH1e.

**Table 5.** Items in the survey used to measure the perception-based variables.

| Item | Item Statement |
|---|---|
| PEOU1 | I found the procedure to apply the goal modeling language simple and easy to follow. |
| PEOU2 | The goal modeling language is easy to learn. |
| PEOU3 | In general, I found this goal modeling language ease to use. |
| PU1 | I believe this language would reduce the time and effort required to specify goal models. |
| PU2 | I believe that the goal models obtained by this language are clear, concise, and unambiguous. |
| PU3 | I believe this goal modeling language has enough expressiveness to represent the objectives and intentions of different stakeholders. |
| PU4 | I believe that this goal modeling language provides an effective means for specifying goal models. |
| PU5 | Using this goal modeling language would improve my performance when specifying goal models. |
| PU6 | In general, the goal modeling language is useful. |

#### 3.3.1.4 Hypotheses formulation

The null hypotheses of the experiment can be summarized as follows:

- $H1_0$: Quality (value@GRL) = Quality (i*)
- $H2_0$: Modeling Time (value@GRL) = Modeling Time(i*)
- $H3_0$: Productivity (value@GRL) = Productivity (i*)
- $H4_0$: PEOU (value@GRL) = PEOU (i*)
- $H5_0$: PU (value@GRL) = PU (i*)

The goal of the statistical analysis was to reject these hypotheses and possibly accept the alternative ones (e.g., $H1_1 = \neg H1_0$). All the hypotheses are two-sided because we did not postulate that any effect would occur as a result of the goal modeling language usage.

#### 3.3.1.5 Design

We used a balanced between-subject design, i.e., a participant was part of either the experimental group or the control group. Table 6 shows the experiment design. The experiment consisted of two runs. The first of these was conducted with the experimental group while the second was conducted

with the control group. Each run consisted of using one of the languages (i* or value@GRL) to model two different systems (i.e., Green Route and Lattes Scholar). We employed two systems in the hope that this would minimize the domain/system effect. We, therefore, had four treatments, owing to the combinations of language and system. We then randomly assigned one of the four treatments to each participant. The chosen design mitigated possible learning effects, since none of the participants repeated any language or system while carrying out the experiments.

Table 6. Experiment design.

| | **Run 1 (experimental group)** | **Run 2 (control group)** |
|---|---|---|
| Treatments | value@GRL, Green Route | i*, Lattes Scholar |
| | value@GRL, Lattes Scholar | i*, Green Route |

Various extraneous factors (also denominated as cofactors) may have had an undesirable effect on the effectiveness of the goal modeling language, and this effect might have been confused with the effect of Language. In this family of experiments, we analyzed the effect of the System (i.e., Object) cofactor, since the domain of the goal models used and the participants' familiarity with the application domain of the systems could have affected the effectiveness of the goal modeling language. For the baseline experiment, we also analyze the effect of the Profile cofactor in order to assess whether the participants' profile (novice software engineer or novice business analyst) influence the results.

#### 3.3.1.6 Operation

A pilot experiment with four PhD students was conducted to assess the experimental planning, as a result of which several improvements were made to the materials. Prior to the experiment, the participants attended a training session concerning the use of the languages and performed an exercise regarding the modeling of a Meeting Scheduler system. The participants created their own models using the instructions provided, after which the experimenter constructed the model solution interactively with the participants. The entire training session took four hours.

All the participants attended an introductory lesson in which detailed instructions on the experimental task were presented. A pre-questionnaire was administered to the participants in order to assess their experience. The results showed that they had no previous knowledge of goal modeling.

The experiment was performed under controlled conditions in a laboratory at the UPV according to the balanced between-subjects design outlined in Table 6. As explained above, we had four combinations of treatments, but each participant was randomly assigned to only one of these treatments. The experiment was executed in two runs: one at the Department of Computer Science (CS) involving the CS students and another at the Department of Business Organisation involving the business analyst students (see Section 3.2.3). In the first run at the CS department, 14 participants applied value@GRL, and the other 14 participants applied i*, with 7 participants randomly assigned to each experimental object (O1 and O2). The participants attended only the training and experimental session of the language to which they had been assigned, signifying that those who applied value@GRL had no knowledge of i* (and vice versa).

During the experiment, the participants were asked to carry out the experimental task and no time limit was imposed. They were allowed to consult the training materials. After specifying the goal model, the participants were asked to fill in the post-experiment survey.

### 3.3.2 Second experiment (UPV2)

The second experiment in our family was a strict internal replication of UPV1. The same experimental protocol was applied but to a different population, signifying that we varied only the participants, while the site, experimenters, design, variables and instrumentation remained the same. The purpose was to test the extent to which the study results could be generalized to other populations. The participants were 67 3rd–year students. They attended the Software Quality course in the Fall of 2017. The participants also attended a course on Model-Driven Engineering, where they acquired knowledge of software modeling. A pre-questionnaire was administered to the participants in order to assess their experience and the results showed that they had no previous knowledge of goal modeling.

As in the baseline experiment, it took place in a single room and no interaction was allowed among the participants. With regard to the data validation, in order to maintain a balanced design, we discarded the data concerning three participants (who did not represent outliers, as they were selected randomly) to obtain a total of 64 participants, i.e., 16 samples in each group.

### 3.3.3 Third experiment (UNA)

The third experiment in our family was the second internal replication of UPV1. The participants were 20 Master's degree students, all professional software engineers and managers from multiple companies who were enrolled on a professional MSc in SE at the National University of Asunción in Paraguay. The participants had an average of 6 years of experience in software engineering. However, despite having an average of four years of experience in software modeling with UML, they had no previous knowledge of goal modeling.

The experiment was organized as part of the Software Quality and Metrics course with a special focus on assessing modeling approaches. A different experimenter was involved in this study but the same experimental design and materials were used. We, therefore, varied the site, the participants and the experimenter, although the design, variables, and instrumentation remained the same.

### 3.3.4 Fourth experiment (UPV3)

The fourth experiment in our family was the third internal replication of UPV1. The participants were 60 undergraduate CS students enrolled on the Fall 2018 course on Software Quality at the Universitat Politècnica de València. This was the same context as the UPV2 experiment and in this replication, we varied only the participants. The site, design, variables, instrumentation, and experimenters remained the same and the results of the pre-questionnaire also showed that they had no previous knowledge of goal modeling.

## 3.4 Experimental tasks and materials

The experimental tasks consisted of specifying a goal model using one of the selected languages on two experimental objects. These tasks were structured to allow the comparison of both languages.

We provided the participants with the mission and description of the system and then asked them to build the corresponding goal model by following the steps and guidelines of the language. In the case of i*, the experimental task consisted of: defining and drawing the actors; modeling the intentional elements (i.e., goals, soft-goals, tasks, and resources), and defining links between the intentional elements within the boundary of the actors and the links between the intentional elements of different actors. Similarly, for value@GRL, the experimental task consisted of: defining and

drawing the actors (i.e., main actor, external actors, and the system actor); modeling the intentional elements (i.e., goals, soft-goals, and tasks) of the main and external actors; defining links between the intentional elements of the main and external actors, and modeling the system actor and its links. The participants were subsequently asked to fill in a questionnaire regarding their perceptions.

The experimental material (available at https://goo.gl/wMPH1e) included the system missions and a set of documents to support the training sessions and the experimental tasks, along with the questionnaire.

The training materials included: (i) a set of slides containing an introduction to goal modeling; (ii) a set of slides describing value@GRL along with an example of its use, and (iii) a set of slides describing i* along with an example of its use. They also included two booklets describing the requirements of a system (Meeting Scheduler) to be modeled and the experimental task to be performed with i* and value@GRL. These booklets helped us to gather the data concerning the experimental task.

The documents supporting the training and experimental tasks also included:

- Four kinds of booklets covering the four possible combinations of both goal languages and experimental objects (value@GRL-O1, value@GRL-O2, i*-O1, i*-O2). These booklets described the experimental tasks to be performed, described the requirements of each system and gathered the data appertaining to each experimental task.
- Two appendices containing a detailed explanation of each goal modeling language.
- A post-task experimental questionnaire with closed and open questions that allowed the participants to express their opinion of the language's ease of use and usefulness.

### 3.5 Family data analysis and meta-analysis

The results of each individual experiment were collected using the booklets and the online questionnaire, and were then analyzed. We used descriptive statistics, violin plots, and statistical tests to analyze the data collected from each experiment. As is usual, in all the tests, we accepted a probability of 5% of committing a Type-I Error [52], i.e., rejecting the null hypothesis when it is actually true.

The data analysis was carried out by considering the following steps:

1. We first carried out a descriptive study of the measures for the dependent variables.
2. We analyzed the characteristics of the data in order to determine which test would be most appropriate to test our hypotheses. Since the sample size of most of the experiments was less than 50, we applied the Shapiro-Wilk so as to test the normality of data and the Brown-Forsythe Levene-type test to determine the homogeneity of variances.
3. The results of the aforementioned tests were then employed as a basis on which to test the null hypotheses formulated. When the data were normally distributed and the variances were homogeneous, we used two-way Analysis of Variance (ANOVA) with interactions to analyze the data from each experiment by considering the Language (i.e., value@GRL vs. i*) and System (i.e., O1 vs. O2) factors and their interaction [40]. When the ANOVA assumptions could not be satisfied, we used the Mann-Whitney test to analyze the Language factor, as well as the System factor, and the Kruskal-Wallis test to analyze the means of the four treatments.
4. When the test results suggested that there was a significant interaction between the factors or a significant difference in means, we performed a *post-hoc* analysis to determine which pairs were significantly different. For this purpose, we used the non-parametric Mann–Whitney or t-test, depending on the normality of the data distribution.

5. Furthermore, the statistical significances of the experiments were complemented with the magnitude of their effects. For this purpose, Cliff's δ estimates [53] were obtained with a confidence interval of 95%.
6. We analyzed the interaction of the Profile cofactor with the Language (main factor) in the UPV1 experiment. This analysis is described in Section 4.3. We again used two-way ANOVA with interactions to analyze the data by comparing the four treatments and their interactions [40]. When the ANOVA assumptions could not be satisfied, we used the Mann-Whitney test for independent samples.
7. In order to strengthen the results of each individual experiment, we decided to aggregate them using a meta-analysis. We specifically performed an Aggregated Data (AD) meta-analysis based on Cliff's δ, as the experimental conditions were similar for all the experiments. The findings of a recent study indicated that AD is suitable to analyze a family of experiments [14]. This analysis, which is detailed in Section 5.1, enabled us to obtain more robust results and to extract more general conclusions when considering the set of experiments in the family.

The results of the individual experiments are outlined in Section 4, whereas the meta-analysis is presented in Section 5.

# 4. Results

In this section, we discuss the results of each experiment by quantitatively analyzing the data according to the hypotheses stated. The results were obtained by using SPSS v20 and R v3.5.0. A qualitative analysis based on the feedback obtained from the open questions of the post-task questionnaire is also provided.

## 4.1 Descriptive statistics and exploratory data analysis

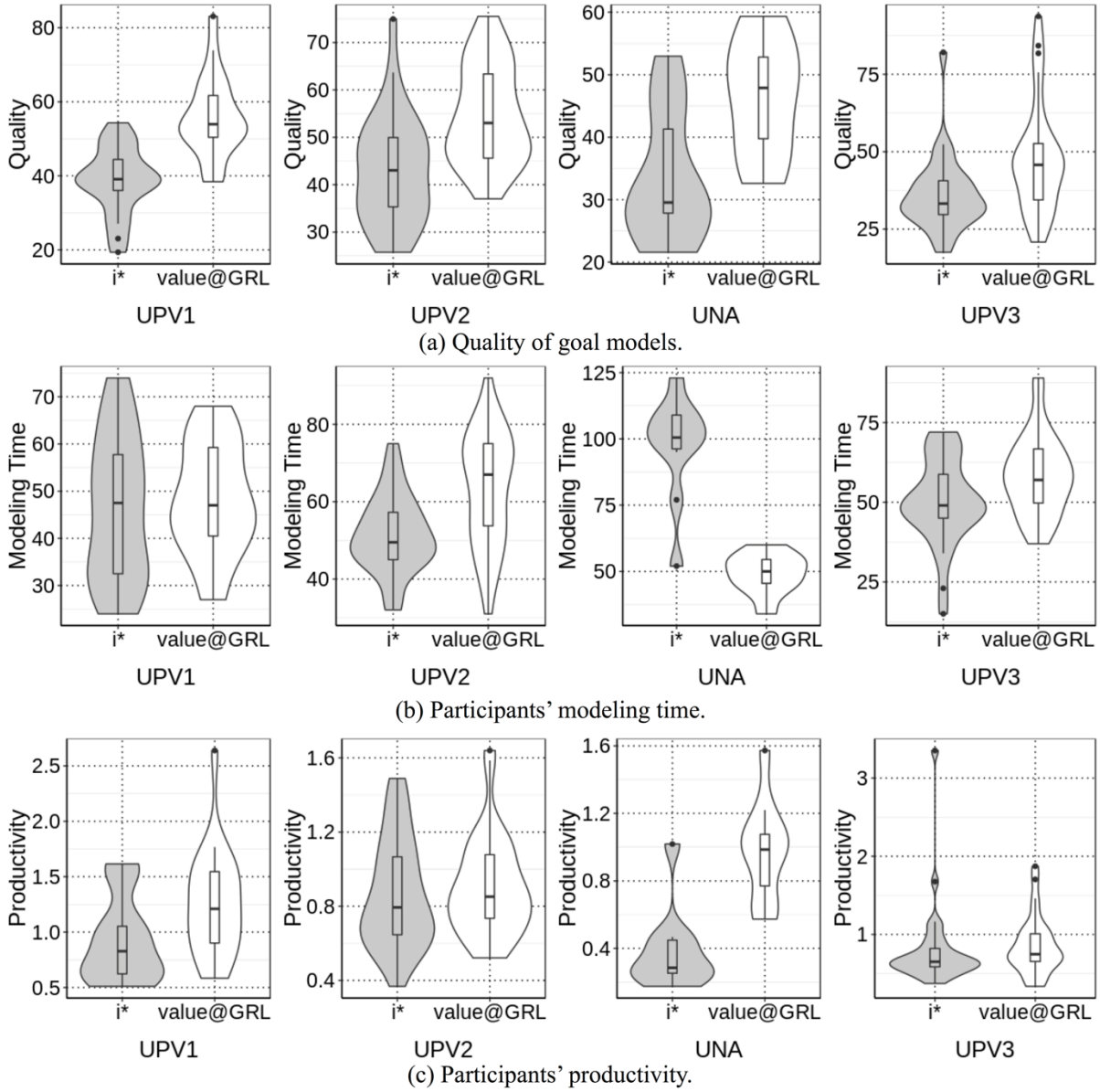
Table A-1 shows a summary of the results of the goal modeling task performed in each individual experiment, divided by Language and System. At a glance, we can observe that the participants performed best and also achieved the best perceptions on ease of use and usefulness when using value@GRL, with the exception of the modeling time for the UPV1, UPV2 and UPV3 experiments, and PU for the UPV3 experiment. The overall comparison of the two languages without splitting by System is visually presented in Fig. 4 and Fig. 5 by means of violin plots.

In order to measure the quality of the goal models created by the participants, the assessment group (composed of one independent expert on goal modeling and two of the authors of this paper) developed five additional i* Oracles for O1 and four for O2, five additional value@GRL Oracles for O1, and five additional value@GRL Oracles for O2. The assessment group measured the quality of the participants' models against all the Oracles, and the highest quality score was selected. The quality scores for all the oracles and the raw data are available at https://goo.gl/wMPH1e.

Fig. 4(a) and the *Quality* rows in Table A-1 show the median and mean values of *F-measure* respectively. For all the experiments in the family, these measures of central tendency are higher for value@GRL than for i* (the mean values range from 46.39 to 55.86 for value@GRL and from 33.51 to 42.96 for i*).

The practical meaning of this is that the participants obtained goal models with a higher quality when using value@GRL than when using i*. Otherwise, as can be observed in Fig. 4(b) and the *Modeling Time* rows in Table A-1, the participants obtained better central tendency results as regards

the modeling time when using i* in all the experiments, with the exception of UNA, for which the modeling time results are better when using value@GRL.




(a) Quality of goal models.

(b) Participants' modeling time.

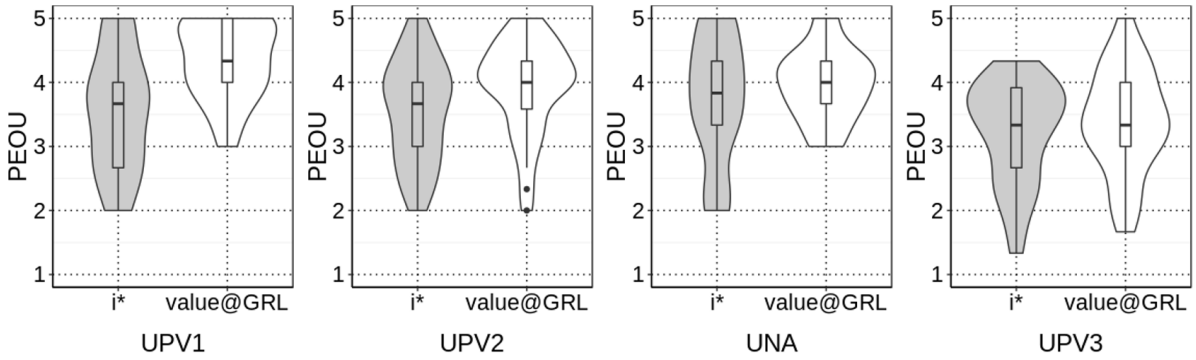
(c) Participants' productivity.

**Fig. 4.** Violin plots for the results related to the participants' performance.

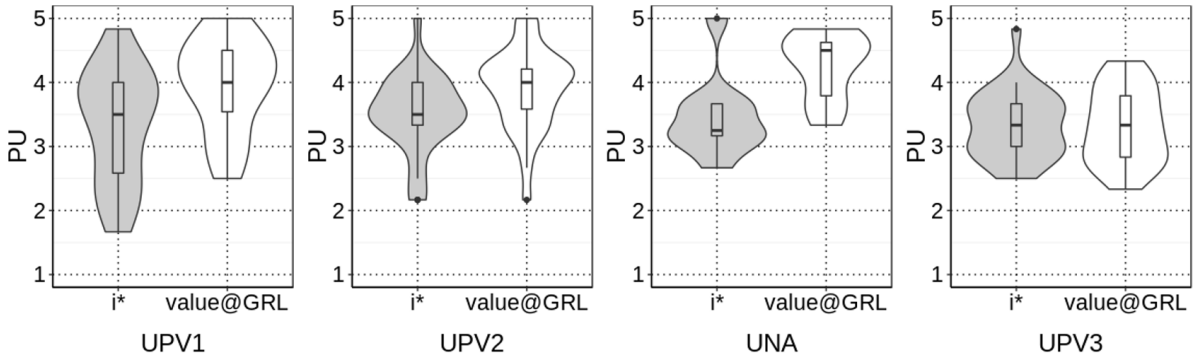
With regard to the participants' productivity, Table A-1 and Fig. 4(c) show that the participants' productivity was somewhat greater when using value@GRL than when using i* in all the experiments.

Table A-1 also presents a summary of the statistics regarding each of the perceived-based variables, according to each language and system. We measured these variables using a five-point Likert scale as an interval scale. The mean values indicate that value@GRL attained the participant's best perceptions as regards ease of use and usefulness in all the experiments, with the exception of PU for UPV3, which are similar. Fig. 5 shows the distribution of the perceived-based variables per language as violin plots. The median for each method is shown as the horizontal segment in the box plot inside each violin plot. Fig. 5(a) shows that the participants perceived value@GRL to be easier to use than i* in UPV1, UPV2 and UNA, although their perceptions as regards ease of use in UPV3 are

similar. Likewise, Fig. 5(b) shows that the participants perceived value@GRL to be more useful than i* in UPV1, UPV2 and UNA, although their perceptions as regards usefulness in UPV3 are similar.




(a) Perceived Ease of Use (PEOU).




(b) Perceived Usefulness (PU).

**Fig. 5.** Violin plots showing the distribution of the participants' perceptions.

## 4.2 Individual data analyses

In order to test the formulated hypotheses, when the data were normally distributed and the variances were homogeneous, we analyzed the effect of the main factor (i.e., Language), the effect of System (cofactor), and their interactions on the measures of the dependent variables considered (i.e., Quality, Modeling Time, Productivity, PEOU, and PU) using the two-way ANOVA with interactions. When the ANOVA assumptions could not be satisfied, we used the Kruskal-Wallis test to compare the means of the four treatments (i.e., value@GRL_O1, value@GRL_O2, i*_O1, i*_O2), as explained in Section 3.5.

Furthermore, the statistical significances of the experiments were complemented with the magnitude of their effects. The magnitude of an experiment effect can be expressed by the effect size measured as the standardized difference between two groups (*d* family) or as the correlation between two or more variables (*r* family) [54]. In the first family, Cohen's *d* is the most common Standardized Mean Difference statistic, while Hedges' *g* provides a measure of effect size weighted according to the relative size of each sample. The pooling of weighted standard deviations is used in the calculation of Hedges' *g* [55]. However, when dealing with ordinal scale data, non-parametric measures such as Cliff's δ are recommended [56]. Moreover, the non-parametric nature of Cliff's δ reduces the influence of distribution shape, differences in dispersion, and extreme values [52]. In our study, two of the five variables studied were ordinal (i.e., PEOU and PU), while among the continuous variables, only Modeling Time was normally distributed in the four experiments.

Cliff's δ can be defined as the difference between the probability that a random observation from group one is greater than a random observation from group two and the probability that a random observation from group one is less than a random observation from group two [53]. In the four experiments in our family, the Cliff's δ estimates were obtained with a confidence interval of 95% for each of the variables. Cliff's δ is bounded, signifying that an effect size of 1 or -1 indicates the absence of an overlap between the two groups, whereas 0 indicates that the group distributions are equivalent. Furthermore, the sign indicates the direction of the effect. A positive sign means that the direction of the effect is in favor of the value@GRL language.

The magnitude of the effect was assessed using the thresholds provided by Kraemer and Kupfer [57], i.e., |d| < 0.112 "negligible" (shown in gray), |d| < 0.276 "small" (red), |d| < 0.428 "medium" (yellow), otherwise "large" (green). These thresholds were taken into account by Kitchenham et al. [56] in their extended guidelines for effect size magnitude interpretation. With regard to the effect size, in the present study we considered medium and large effect sizes as practically significant, as suggested by the Cohen's benchmarks [54], [58].

In the following subsections, the results obtained for each dependent variable are shown in Tables 7 to 14 in which the "Experiment" column describes the baseline experiment and the three replications, "*Language p*-value" is the statistical significance obtained for the Language effect (main factor), "*System p*-value" is the statistical significance obtained for the System effect, and *"Interaction p*-value*"* is the statistical significance obtained for the interaction between Language and System. The tendency of the data if the null hypotheses regarding the effect of language and system are rejected is, meanwhile, shown in the "In favor of" columns. There is an additional column for the effect size for the main factor (Cliff's δ estimates). The results obtained are also discussed.

### 4.2.1 Testing quality of goal models

Table 7 shows the results obtained after testing the effects of the language, system, and their interactions for the Quality variable.

**Table 7.** Test results for Quality ($H1_0$).

| Experiment | Language | | | System | | Interaction |
|---|---|---|---|---|---|---|
| | ***p*-value** | **Cliff's δ** | **In favor of** | ***p*-value** | **In favor of** | ***p*-value** |
| UPV1 | **0.000** $ | 0.81 [0.54, 0.93] | value@GRL | **0.018** $ | Green Route | 0.353 $ |
| UPV2 | **0.000** $ | 0.55 [0.28, 0.74] | value@GRL | 0.213 $ | - | **0.023** $ |
| UNA | **0.007** $ | 0.70 [0.19, 0.91] | value@GRL | 0.172 $ | - | 0.159 $ |
| UPV3 | **0.002** # | 0.44 [0.15, 0.67] | value@GRL | 0.224 # | - | **0.010** * |

*$ ANOVA; * Kruskal-Wallis; #Mann-Whitney*

The ANOVA test indicates that there is an interaction effect between language and system for UPV2. We then performed a *post-hoc* analysis using t-tests to detect which pairs of treatments are significantly different. The results suggest that there are two significant interactions, as shown in Table 8. Although there is a significant difference between the languages for the quality variable, this difference only occurs with the Lattes-Scholar system, where the quality of the goal models specified with value@GRL is higher. There is also a significant difference for the system factor only in the case of the i* language, where the quality of the goal models for the Green Route system is higher than the quality of the models for the Lattes Scholar system.

**Table 8**. Test results for the *post-hoc* analysis for Quality.

| | **Treatment pairs** | **UPV2 *p*-value** | **UPV3 *p*-value** |
|---|---|---|---|
| Effect of language on system | i*-Green Route vs. value@GRL-Green Route | 0.181 & | **0.003** # |
| | i*-Lattes-Scholar vs. value@GRL-Lattes-Scholar | **0.000** & | 0.115 # |
| Effect of system on language | i*-Green Route vs. i*-Lattes-Scholar | **0.016** & | 0.774 # |
| | value@GRL-Green Route vs. value@GRL-Lattes-Scholar | 0.459 & | 0.081 # |

*& t-test; # Mann-Whitney*

On the other hand, the Kruskal-Wallis suggests that there is a significant difference in the means between language and system for UPV3. The *post-hoc* analysis using Mann-Whitney suggests that there is one significant difference for the language (see Table 8) when the Green Route system is used in favor of Value@GRL.

With regard to the system cofactor, the ANOVA test suggests that there is a significant difference in favor of Green Route for UPV1 in terms of the quality of the goal models created.

With regard to the language factor, the test results shown in Table 7 suggest that the null hypothesis $H1_0$ can be rejected for all the experiments in the family, meaning that a significant statistical difference exists between the two languages in terms of the quality of the goal models created. Furthermore, this difference is significant in practice, since all the experiments have a "large" effect size (> 0.428) in favor of value@GRL. This means that the goal models specified by the participants using value@GRL were of higher quality than the goal models specified using i*. For this variable, the lowest effect size estimate was obtained for the experiment UPV3 and the highest for UPV1.

These findings can probably be explained by the process provided by value@GRL, which guides the users on how to specify the goal models. In addition, this language contains a reduced number of intentional elements and links when compared to i*. The *recall* (completeness) of the participants modeling with value@GRL was, therefore, higher than with i*, thus affecting the overall quality in favor of value@GRL. As future work, we plan to verify whether using other software metrics to measure the quality of goal models may affect this result.

### 4.2.2 Testing modeling time

Table 9 shows the results obtained after testing the effects of the language, system, and their interactions for the Modeling Time variable.

**Table 9.** Test results for Modeling Time ($H2_0$).

| **Experiment** | **Language** | | | **System** | | **Interaction** |
|---|---|---|---|---|---|---|
| | ***p*-value** | **Cliff's δ** | **In favor of** | ***p*-value** | **In favor of** | ***p*-value** |
| UPV1 | 0.612 $ | 0.08 [-0.29, 0.43] | - | **0.016** $ | Lattes Scholar | 0.198 $ |
| UPV2 | **0.000** $ | 0.48 [0.20, 0.69] | value@GRL | 0.508 $ | - | 0.391 $ |
| UNA | **0.000** $ | -0.91 [-0.99, -0.52] | i* | 0.110 $ | - | **0.009** $ |
| UPV3 | **0.015** $ | 0.32 [0.02, 0.57] | value@GRL | 0.144 $ | - | 0.661 $ |

*$ ANOVA*

As mentioned before, a positive sign for the effect size means that the direction of the effect, depicted in the column "In favor of", is on the side of the value@GRL language (and vice versa). Specifically, in the case of Modeling Time a positive sign means the worst scenario because lower modeling times are more suitable.

The ANOVA test indicates that there is an interaction between language and system for UNA. We then performed a *post-hoc* analysis using t-tests to determine which pairs of treatments are significantly different. The results shown in Table 10 indicate that there is a significant difference between the time taken to model both systems with value@GRL and the time taken to model the same systems with i*. As it can be seen in Table A-1, value@GRL (mean = 52.6 min.) implies shorter modeling times than i* (mean = 84.8 min.) when Green Route system is used. There is also a significant difference for the system cofactor only in the case of the i* language, where the time needed to model Green Route (mean = 84.8 min.) is lower than the time needed to model Lattes Scholar (mean = 110 min.).

**Table 10**. Test results for the *post-hoc* analysis for Modeling Time.

| | **Treatment pairs** | **UNA *p*-value** |
|---|---|---|
| Effect of language on system | i*-Green Route vs. value@GRL-Green Route | **0.022** & |
| | i*-Lattes-Scholar vs. value@GRL-Lattes-Scholar | **0.000** & |
| Effect of system on the language | i*-Green Route vs. i*-Lattes-Scholar | **0.049** & |
| | value@GRL-Green Route vs. value@GRL-Lattes-Scholar | 0.178 & |

*& t-test*

With regard to the system cofactor, the results suggest that there is a significant difference only for the UPV1 experiment, where the time taken to model the Green Route system is significantly lower than the time taken to model the Lattes Scholar system.

With regard to the language factor, the test results shown in Table 10 suggest that the null hypothesis $H2_0$ can be rejected for all the experiments in the family with the exception of UPV1, since the *p*-value is greater than 0.05. In the other experiments, a difference exists between the two languages in terms of the time spent by the participants when applying the languages. In the case of UPV2 and UPV3, the effect sizes are "large" and "medium" respectively. Since the difference between value@GRL and i* regarding modeling time is positive, we can assume that the participants spent less time when using i*. Conversely, the UNA experiment have a negative difference, which means that the time needed to specify value@GRL models was lower than the time needed to specify i* models. In this case, the effect size can be assessed as "large". A practical significance is, therefore, also confirmed in all these cases.

### 4.2.3 Testing productivity

Table 11 shows the results obtained after testing the effects of the language, system, and their interactions for the Productivity variable. For UPV1, the Kruskal-Wallis test indicates that there is a significant difference between the treatments employed in this experiment.

**Table 11.** Test results for Productivity ($H3_0$).

| **Experiment** | **Language** | | | **System** | | **Interaction** |
|---|---|---|---|---|---|---|
| | ***p*-value** | **Cliff's δ** | **In favor of** | ***p*-value** | **In favor of** | ***p*-value** |
| UPV1 | **0.025** # | 0.41 [0.04, 0.69] | value@GRL | **0.003** # | Green Route | **0.002** * |
| UPV2 | 0.502 # | 0.10 [-0.19, 0.37] | - | 0.115 # | - | 0.237 * |
| UNA | **0.000** $ | 0.88 [0.39, 0.98] | value@GRL | 0.662 $ | - | 0.086 $ |
| UPV3 | 0.138 # | 0.22 [-0.08, 0.49] | - | **0.037** # | Green Route | 0.076 * |

*$ ANOVA; * Kruskal-Wallis; #Mann-Whitney*

We then performed a *post-hoc* analysis using a Mann-Whitney test to identify which pairs of treatments are significantly different. The results presented in Table 12 indicate that although there is

a significant difference for the language factor, this difference can only be observed when comparing the Lattes Scholar system, where the participants were more productive when they modeled this system using value@GRL. In addition, it can also be observed that, although there is also a significant difference between the system factor, this can only be appreciated for the i* language, where the productivity of the participants modeling the Green Route system is significantly higher than the productivity of the participants modeling the Lattes Scholar system.

**Table 12**. Test results for the *post-hoc* analysis for Productivity.

| | **Treatment pairs** | **UPV1 *p*-value** |
|---|---|---|
| Effect of language on the system | i*-Green Route vs. value@GRL-Green Route | 0.314 # |
| | i*-Lattes-Scholar vs. value@GRL-Lattes-Scholar | **0.014** # |
| Effect of system on the language | i*-Green Route vs. i*-Lattes-Scholar | **0.002** # |
| | value@GRL-Green Route vs. value@GRL-Lattes-Scholar | 0.190 # |

*& t-test; # Mann-Whitney*

With regard to the system cofactor, the results suggest that there is also a significant difference for the UPV3 experiment, in which the participants' productivity when modeling the Green Route system is significantly higher than the participants' productivity when modeling the Lattes Scholar system.

With regard to the language factor, the test results shown in Table 11 indicate that the difference between the two languages in terms of productivity is statistically significant for UPV1 and UNA (*p*-value = 0.025 and *p*-value = 0.000, respectively), while it is not statistically significant for UPV2 and UPV3. The null hypothesis $H3_0$ can consequently be rejected for two out of the four experiments in the family: UNA and UPV1. In fact, the UNA experiment have a "large" effect size while the effect size of UPV1 is "medium", both in favor of value@GRL.

Although the modeling time when specifying goal models with i* was superior in two out of the four experiments, what in fact led to the difference in mean was the quality of the goal models (PROD = Quality/Modeling Time). The result obtained may suggest that the Master's degree students and professional participants' productivity was greater with value@GRL. The undergraduate students had a similar productivity when using the two languages. These results may indicate that more experienced participants benefit more from value@GRL, but this assumption should be validated in further experiments.

### 4.2.4 Testing perceived ease of use

Table 13 shows the results obtained after testing the effects of the language, system, and their interactions for the PEOU variable. The results show that there is no interaction between language and system in any of the experiments.

**Table 13.** Test results for perceived ease of use ($H4_0$).

| **Experiment** | **Language** | | | **System** | | **Interaction** |
|---|---|---|---|---|---|---|
| | **p-value** | **Cliff's δ** | **In favor of** | **p-value** | **In favor of** | **p-value** |
| UPV1 | **0.003** $ | 0.50 [0.14, 0.74] | value@GRL | 0.171 $ | - | 0.645 $ |
| UPV2 | **0.046** # | 0.29 [0.00, 0.53] | value@GRL | 0.496 # | - | 0.214 * |
| UNA | 0.514 $ | 0.10 [-0.42, 0.57] | - | 0.089 $ | - | 0.404 $ |
| UPV3 | 0.596 # | 0.08 [-0.21, 0.36] | - | 0.570 # | - | 0.852 * |

*$ ANOVA; * Kruskal-Wallis; #Mann-Whitney*

With regard to the system cofactor, no significant difference was found in any of the experiments for the ease of use perceived by the participants. Only with regard to the language factor did some significant differences arise. Indeed, the test results shown in Table 13 suggest that the difference between the two languages in terms of perceived ease of use is not statistically significant for UNA and UPV3 ($p$-value > 0.05), while we can assume that this difference is statistically significant for UPV1 and UPV2. For these two last experiments, UPV2 does not have a practical significance since the effect size is "small", while UPV1 has a "medium" effect size in favor of value@GRL.

In fact, the violin plots for UNA and UPV3 shown in Fig. 5(a) illustrate that the participants' perceived ease of use was quite similar for both languages, as confirmed by the values of the mean of the data (ranging from 3.22 to 3.97) from Table A-1. This may suggest that the participants are neutral regarding the ease of use of the two languages. The analysis of the answers to the open questions in the post-experiment questionnaire revealed that the participants had some difficulties when using both languages. For example, participant ID 2GR4 said that "*modeling with value@GRL is easy. However, it requires more training*", and participant ID 1GR2 said "*to make i* easier to understand, dependencies should be drawn directly between intentional elements of different actors (without an intermediate intentional element)*".

### 4.2.5 Testing perceived usefulness

Table 14 shows the results obtained after testing the effects of the language, system, and their interactions for the PU variable.

The ANOVA test indicates that there is an interaction effect between language and system for UPV1. We then performed a *post-hoc* analysis using t-tests to detect which pairs of treatments are significantly different. The results suggest that there are two significant interactions, as shown in Table 15.

**Table 14.** Test results for perceived usefulness ($H5_0$).

| Experiment | Language | | | System | | Interaction |
|---|---|---|---|---|---|---|
| | ***p*-value** | **Cliff's δ** | **In favor of** | ***p*-value** | **In favor of** | ***p*-value** |
| UPV1 | **0.017** $ | 0.40 [0.04,0.67] | value@GRL | 0.389 $ | - | **0.038** $ |
| UPV2 | **0.019** $ | 0.37 [0.08, 0.60] | value@GRL | 0.133 $ | - | 0.761 $ |
| UNA | **0.011** # | 0.67 [0.09, 0.91] | value@GRL | 0.168 # | - | **0.035** * |
| UPV3 | 0.907 $ | -0.02 [-0.30, 0.28] | - | 0.787 $ | - | 0.418 $ |

*$ ANOVA; * Kruskal-Wallis; #Mann-Whitney*

Although there is a significant difference between the languages for the PU variable, this difference only occurs with the Green Route system, where the usefulness perceived by the participants when using value@GRL is higher. There is also a significant difference for the system cofactor only in the case of the value@GRL language, where the perceived usefulness when modeling the Green Route system is significantly higher than the perceived usefulness when modeling the Lattes Scholar system.

On the other hand, the Kruskal-Wallis suggests that there is a significant difference in the means between language and system for UNA. However, the *post-hoc* analysis using the Mann-Whitney test does not suggest any significant difference between the treatment pairs.

**Table 15**. Test results for the *post-hoc* analysis for perceived usefulness variable.

| | Treatment pairs | UPV1 *p*-value | UNA *p*-value |
|---|---|---|---|
| Effect of language on the system | i*-Green Route vs. value@GRL-Green Route | **0.005** & | 0.056 # |
| | i*-Lattes-Scholar vs. value@GRL-Lattes-Scholar | 0.816 & | 0.170 # |
| Effect of system on the language | i*-Green Route vs. i*-Lattes-Scholar | 0.445 & | 0.391 # |
| | value@GRL-Green Route vs. value@GRL-Lattes-Scholar | **0.018** & | 0.110 # |

*& t-test; # Mann-Whitney*

With regard to the system cofactor, no significant difference was found in any of the experiments for the usefulness perceived by the participants. With regard to the language factor, the test results shown in Table 14 suggest that the null hypothesis $H5_0$ can be rejected for all the experiments in the family with the exception of UPV3, since the *p*-value is greater than 0.05. In UPV1, UPV2, and UNA, a statistically significant difference exists between the two languages in terms of the usefulness perceived by the participants when applying the languages. A practical significance is also confirmed in favor of value@GRL. The effect size is "large" in the case of UNA, while UPV1 and UPV2 have a "medium" effect size.

The analysis of the answers to the open questions in the post-experiment questionnaire revealed that the participants found value@GRL to be useful. For example, participant ID 2GR3 said "*although representing the hierarchy of intentional elements inside an actor seems to be complex, I liked value@GRL*". It is worth mentioning that the participants with the business analyst profile in UPV1 highlighted that they found the two methods useful for communication purposes. However, $H5_0$ could not be rejected for UPV3. This contradicts the results obtained for UPV2, in which the participants with the same profile found value@GRL to be easy to use and useful. It is, therefore, necessary to investigate this issue in further experiments.

## 4.3 Influence of profile

Since several stakeholders may be involved in goal modeling, we wished to test whether the participants' profile influenced the results. The influence of the Profile cofactor on the main factor (Language) was assessed only for UPV1, as this was the only experiment that involved participants with two different profiles: software engineers and business analysts.

Table 16 presents descriptive statistics (minimum, maximum, mean, median, and standard deviation) for the dependent variables by language and profile. The cells in bold type indicate the participants' values for each variable with the highest median and the lowest standard deviation. In the case of Modeling Time, the lowest median has been embellished since it implies the best situation. Upon comparing the language medians of these variables, it will be noted that the goal models specified by software engineers using i* have a higher quality than those specified by business analysts, while the opposite holds in relation to value@GRL. The standard deviation is higher for software engineers, which indicates that business analysts present a more uniform behavior when using value@GRL. In contrast, the standard deviation is higher for business analysts than software engineers when using i*.

We can also observe that, on average, business analysts specified the goal models in less time than the software engineers. Nevertheless, the business analysts were, on average, more productive than software engineers when using value@GRL, while the software engineers were, on average, more productive than the business analysts when using i*. Again, the standard deviation in modeling time and productivity is higher for software engineers.

With regard to the perception-based variables, the business analysts expressed, on average, a higher perception of ease of use and usefulness than the software engineers for both languages. One possible reason for this may be the fact that, since they modeled the systems in less time, this might have positively influenced their perceptions of these languages. Furthermore, business analysts may find these languages closer to their usual work practices and domain since the languages deal with the modeling of organizational objectives and stakeholders.

**Table 16.** Descriptive analysis for the variables of the experiment by the profile.

| Variable | | i* | | | value@GRL | | |
|---|---|---|---|---|---|---|---|
| | | **All** | **Business Analysts** | **Software Engineers** | **All** | **Business Analysts** | **Software Engineers** |
| Quality | Min | 19.37 | 19.37 | 29.40 | 38.41 | 49.18 | 38.41 |
| | Max | 54.32 | 45.14 | 54.32 | 83.07 | 60.68 | 83.07 |
| | Mean | 38.69 | 31.23 | 41.89 | 55.86 | 55.00 | 56.22 |
| | Mdn | 39.08 | 29.95 | **40.13** | 53.94 | **55.50** | 52.98 |
| | SD | 9.11 | 9.96 | **6.81** | 10.97 | **4.52** | 12.95 |
| Modeling Time | Min | 24.00 | 30.00 | 24.00 | 27.00 | 37.00 | 27.00 |
| | Max | 74.00 | 63.00 | 74.00 | 68.00 | 47.00 | 68.00 |
| | Mean | 46.15 | 41.33 | 48.21 | 48.20 | 42.50 | 50.64 |
| | Mdn | 47.50 | **34.50** | 49.00 | 47.00 | **43.00** | 52.50 |
| | SD | 15.27 | **13.40** | 16.01 | 11.79 | **4.97** | 13.13 |
| Productivity | Min | 0.51 | 0.51 | 0.52 | 0.58 | 1.05 | 0.58 |
| | Max | 1.62 | 1.50 | 1.62 | 2.64 | 1.56 | 2.64 |
| | Mean | 0.92 | 0.81 | 0.97 | 1.25 | 1.32 | 1.23 |
| | Mdn | 0.82 | 0.66 | **0.95** | 1.21 | **1.33** | 1.12 |
| | SD | 0.38 | **0.37** | 0.38 | 0.48 | **0.23** | 0.56 |
| PEOU | Min | 2.00 | 3.67 | 2.00 | 3.00 | 3.33 | 3.00 |
| | Max | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 |
| | Mean | 3.55 | 4.00 | 3.36 | 4.33 | 4.44 | 4.29 |
| | Mdn | 3.66 | **3.83** | 3.00 | 4.33 | **4.66** | 4.33 |
| | SD | 0.94 | **0.52** | 1.03 | 0.61 | 0.66 | **0.61** |
| PU | Min | 1.67 | 3.33 | 1.67 | 2.50 | 3.17 | 2.50 |
| | Max | 4.83 | 4.17 | 4.83 | 5.00 | 5.00 | 5.00 |
| | Mean | 3.34 | 3.78 | 3.15 | 3.96 | 4.25 | 3.83 |
| | Mdn | 3.50 | **3.91** | 3.08 | 4.00 | **4.50** | 4.00 |
| | SD | 0.90 | **0.36** | 1.00 | 0.74 | **0.70** | 0.74 |

In order to determine the significance of the interaction effects between the factors, we used two-way ANOVA with interactions when possible. Otherwise, we used the Kruskal-Wallis test to assess the difference of means between the treatments. The results summarized in Table 17 show that the effect of profile could not be confirmed in most of the cases ($p$-value > 0.05), i.e., the participants' profiles had no statistical influence on the results. Only in the case of the variable PEOU, a significant difference between the means of the treatments was confirmed.

**Table 17.** Interactions between language and profile.

| **Variable** | **Profile** | | **Interaction** |
|---|---|---|---|
| | ***p*-value** | **In favor of** | ***p*-value** |
| Quality | 0.084 $ | - | 0.166 $ |
| Modeling Time | 0.107 # | - | 0.410 * |
| Productivity | 0.800 $ | - | 0.409 $ |
| PEOU | 0.331 # | - | **0.034** * |
| PU | 0.068 $ | - | 0.711 $ |

*$ ANOVA; # Wilcoxon; * Kruskal-Wallis*

We then performed a *post-hoc* analysis using a Mann-Whitney test to identify which pairs of treatments are significantly different. The results shown in Table 18 indicate that a significant difference can only be observed for the software engineers, who expressed a higher perception of ease of use for the value@GRL language when compared to the i* language. This could be owing to the fact that these types of participants were quicker to grasp goal modeling concepts with value@GRL than with i*, specifically with regard to the understanding and use of intentional links (e.g., when they had to connect different intentional elements). In particular, we observed that their initial draft of a goal model for their application system under development was of good technical quality and that they adequately represented the intention of the system to be built.

**Table 18**. Test results for the *post-hoc* analysis for perceived usefulness variable.

| | **Treatment pairs** | **UPV1 *p*-value** |
|---|---|---|
| Effect of language on the profile | i*-Business Analyst vs. value@GRL-Business Analyst | 0.286 # |
| | i*-Software Engineer vs. value@GRL-Software Engineer | **0.015 #** |
| Effect of profile on the language | i*- Business Analyst vs. i*-Software Engineer | 0.180 # |
| | value@GRL-Business Analyst vs. value@GRL-Software Engineer | 0.613 # |

# *Mann-Whitney*

# 5. Family data analysis

In this section, we present a meta-analysis that aggregates the empirical findings obtained in the individual experiments. We then answer the stated research questions for the family of experiments as a whole by considering the results obtained in each individual experiment and the meta-analysis.

## 5.1 Meta-analysis

This section provides the results of a meta-analysis carried out to aggregate the empirical findings obtained in the individual experiments. Of the various existing statistical methods used to aggregate results from interrelated experiments [55], the Aggregated Data (AD) meta-analysis allows more general conclusions to be obtained [14] and was, therefore, chosen for this study. A meta-analysis consists of a set of statistical techniques that can be used to combine and contrast the results of multiple studies. Effect size estimates are commonly used in meta-analysis studies to summarize the findings.

Meta-analysis results are commonly displayed graphically as forest plots. In this regard, we used R [59], and specifically, the "effsize" package [60], to calculate the Cliff's δ effect size, and the "metafor" package [61] to conduct the meta-analysis and create the forest plots or blobbograms.

Fig. 6 and Fig. 7 show the forest plots obtained for the continuous and ordinal variables respectively considered in this study. The number of participants in the experiments is presented in the "Total n" column. The study results are visually displayed in the central column, in which the vertical line depicts that there is no difference between the outcomes for each language. The horizontal lines through the boxes depict the length of Cliff's δ effect sizes with 95% confidence intervals. The size of the box is directly related to the influence of the study on the meta-analysis ("Weight" column). This experiment's influence is determined by the study's sample size and the precision of the study results provided as CI [62]. The blobbograms are, therefore, a graphical summary of the results also presented in Table 19, complemented with the aggregated meta-analysis perspective.

There are two statistical models for meta-analysis, the fixed-effects (FE) model and the random-effects (RE) model [63]. The diamond in the last row of the graph illustrates the overall result of the RE model meta-analysis. The overall estimate effect is the central line of the diamond, while the lateral tips of the diamond confine the associated CI. When the diamond crosses over the central vertical line of the graph, this means that there is no significant difference between the aggregated results of the two methods.

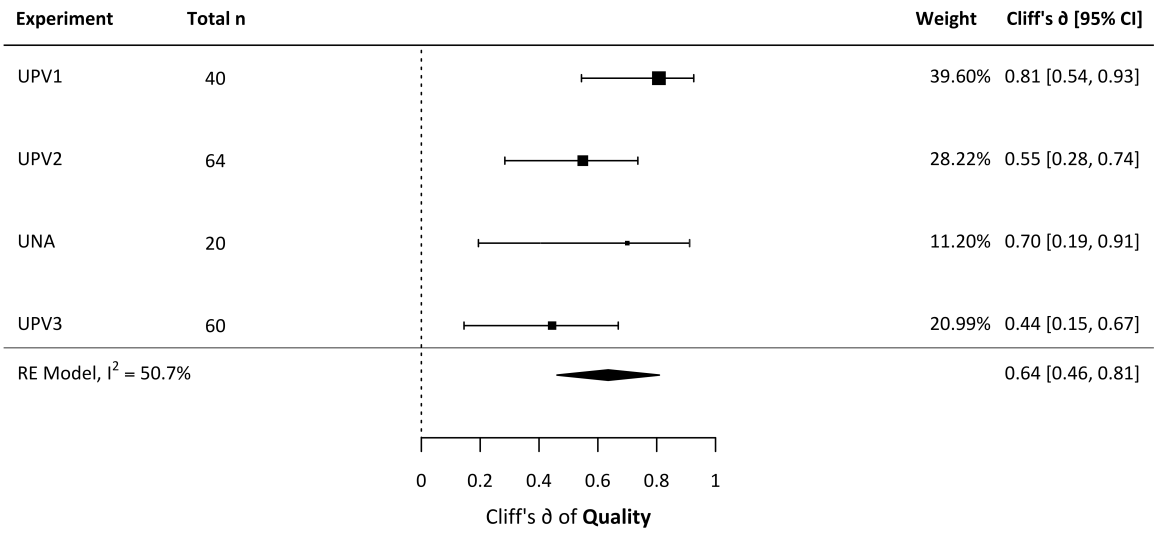
Assessing the heterogeneity in a meta-analysis is a crucial issue because the presence of true heterogeneity can affect the statistical model for the meta-analysis [64]. Heterogeneity measures the variability between studies, i.e., it gives an indication of how comparable the studies in the meta-analysis are and how consistent the overall meta-analysis is. Graphically, this can be checked when assessing the overlapping of the horizontal lines or whiskers in Fig. 6 and Fig. 7. Studies can, therefore, be considered as homogeneous if the CIs in all the studies overlap. Table 19 also shows heterogeneity statistics ($Q$ statistic and $I^2$) and the overall effect sizes (Cliff's δ 95% CI) obtained in our meta-analysis for the RE models for each of the variables.

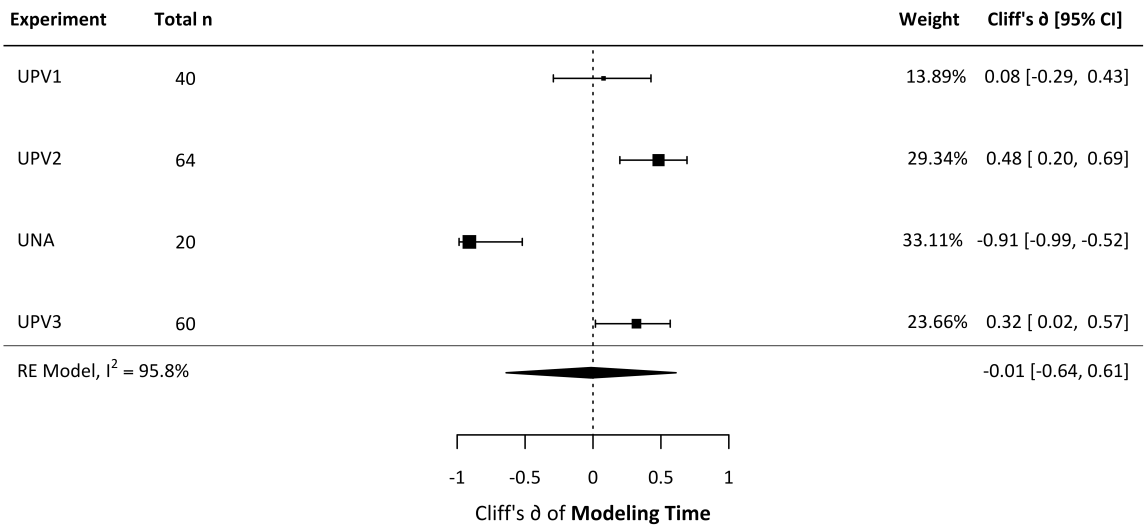
**Table 19.** Heterogeneity statistics ($Q$ statistic, $I^2$) and overall effect sizes for the random-effects models for all the variables

| | ***Q* test** | ***Q* test *p*-value** | $I^2$ | **Cliff's δ RE model** |
|---|---|---|---|---|
| Quality | 6.11 | 0.1064 | 50.7% | 0.64 [0.46, 0.81] |
| Modeling Time | 105.75 | < 0.0001 | 95.8% | -0.01 [-0.64, 0.61] |
| Productivity | 22.92 | 0.0001 | 83.4% | 0.41 [0.06, 0.76] |
| PEOU | 4.13 | 0.2481 | 30.9% | 0.26 [0.07, 0.46] |
| PU | 8.18 | 0.0425 | 64.3% | 0.34 [0.07, 0,60] |

When considering heterogeneity, the $p$-value of the $Q$ test is frequently used as an indication of the extent of between studies variability [65]. A shortcoming of the $Q$ statistic is that it has poor power to detect true heterogeneity among studies when the meta-analysis includes a small number of studies. This means that a non-significant result must not be taken as evidence of no heterogeneity. In fact, a $p$-value of 0.10 is sometimes used to determine statistical significance rather than the conventional level of 0.05. Additionally, it has excessive power to detect negligible variability with a high number of studies [66], [67].

Since the $Q$ statistic informs us only the statistical significance of true heterogeneity, it should not be reported alone. Higgins et al. [68] proposed the descriptive statistic $I^2$ to reflect the ratio of true heterogeneity to total variance for the observed effect estimates, providing information on the percentage of variability that cannot be explained by random sampling or chance [63].



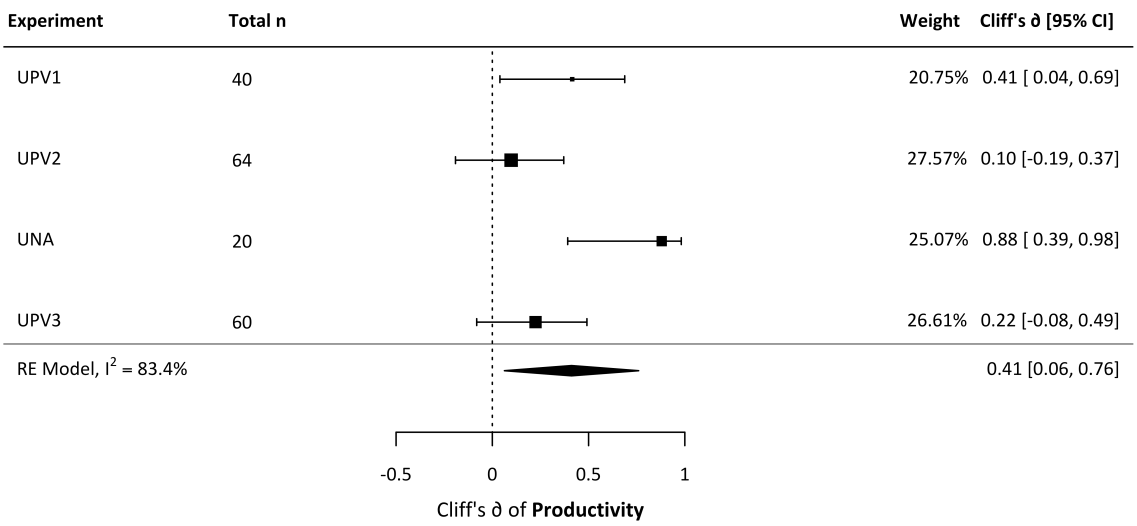




**Fig.** 6. Meta-analysis blobbogram for Quality, Modeling Time and Productivity.

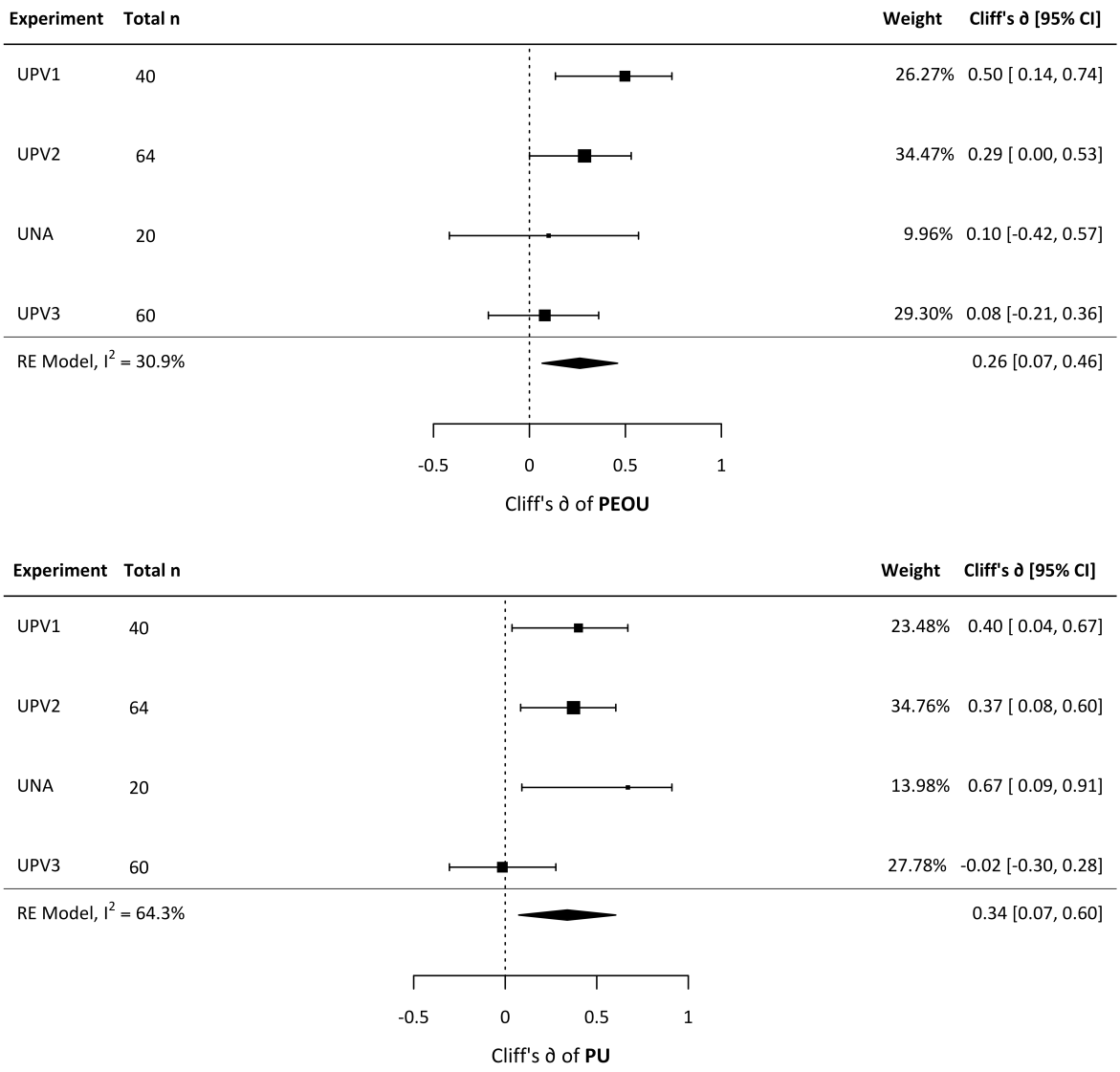



**Fig.** 7**.** Meta-analysis blobbogram for PEOU and PU.

Taking into account both heterogeneity statistics reported in Table 19, along with the forest plots, the studies of the variables Modeling Time, Productivity, and PU can be considered as heterogeneous. Meta-regressions have been used to suggest reasons for observed heterogeneity [69]. As in any regression analysis, meta-regressions attempt to identify significant relations between the dependent variable and covariates of interest. Such an analysis is beyond the scope of this work. However, the heterogeneity observed for Modeling Time and Productivity is probably owing to the type of participants: the participants in UNA (professionals) are faster with i*, while the other participants are faster with value@GRL.

With regard to the heterogeneity observed for PU, we could not find any plausible explanation for the difference observed in UPV3 in contrast to the other experiments that had a similar behavior, since the type of participant (3rd year Computer Science Bachelor's Degree students), location and experimenter were the same as in the UPV2 experiment. Indeed, we plan to replicate this study to explore possible causes.

In fact, only the RE values have been considered in Table 19 since the values of Cliff's δ RE models are quite similar to those of the FE models when there is low heterogeneity. Three out of five variables have a practically significant overall effect size: "large" for Quality, and "medium" for Productivity and PU. In the case of the remaining variables, PEOU has a "low" effect size and the overall effect size for Modeling Time is "negligible".

In summary, the meta-analysis strengthens the overall results obtained in the individual experiments. The effect size estimates may also indicate that it will be necessary to perform further replications with a larger sample of participants.

## 5.2. Answering the research questions

A summary of the results obtained for each individual experiment and the meta-analysis is provided in Table 20. The most prominent result is that value@GRL proved to be more effective than i* as regards creating good-quality goal models.

Although there are some interaction effects between language and system that affect the quality of goal models, these interactions only occur in the UPV2 and UPV3 experiments and three pairs of treatments, as shown in Section 4.2.1. With regard to the system cofactor, only in UPV is there a significant difference in terms of the quality of the goal models created, in favor of Green Route.

**Table 20.** Summary of results of the family of experiments.

| | **Individual Experiments** | | | **Cliff's δ RE model** |
|---|---|---|---|---|
| **Hypotheses** | **$H_i$ (value@GRL) > $H_i$ (i*)** | **$H_i$ (value@GRL) < $H_i$ (i*)** | **$H_i$ (value@GRL) = $H_i$ (i*)** | |
| H1: Quality | UPV1 (large), UPV2 (large), UNA (large), UPV3 (large) | - | - | Large |
| H2: Modeling time | UPV2 (large), UPV3 (medium) | UNA (large) | UPV1 (negligible) | Negligible |
| H3: Productivity | UPV1 (medium), UNA (large) | | UPV2 (negligible), UPV3 (small) | Medium |
| H4: PEOU | UPV1 (medium), UPV2 (small) | | UNA (negligible), UPV3 (negligible) | Small |
| H5: PU | UPV1 (medium), UPV2 (medium), UNA (large) | | UPV3 (negligible) | Medium |

Nevertheless, we found statistical and practical significance for hypothesis $H1_a$ in all the experiments in the family with a large effect, regardless of the participants' experience or background. The meta-analysis also confirmed that the Cliff's δ RE model for the quality variable has a practical significance, with a large effect size. These results are promising, because participants from two different profiles (novice software engineers and business analysts) with relatively low training were able to create goal models with a good level of correctness and completeness. However, we plan to verify whether the use of other metrics whose purpose is to assess the quality of goal models may affect this result.

With regard to modeling time, we found some interaction effects between language and system that affect this variable, but these interactions only occurred in the UNA experiment and three pairs of

treatments. With regard to the system cofactor, there is a significant difference in UPV1, where the time taken to model Green Route was significantly lower than that required to model Lattes Scholar.

Nevertheless, we found statistical and practical significance for hypothesis $H2_a$ in three out of the four experiments. In UPV2 and UPV3, the time needed to specify value@GRL models was higher than the time needed to specify i* models, while the participants in UNA spent less time when using value@GRL. As a consequence, the Cliff's δ RE model for modeling time was found to be negligible. This suggests that the time spent by the participants when applying both languages was similar.

With regard to the participants' productivity, we found some interaction effects between language and system that affect this variable, but these interactions occurred only in the UPV1 experiment and two pairs of treatments. With regard to the system cofactor, there is a significant difference in UPV1 and UPV3, where the participants' productivity when modeling Green Route was significantly higher than the participants' productivity when modeling Lattes Scholar.

Nevertheless, we found statistical and practical significance for hypothesis $H3_a$ in the UPV1 and UNA experiments. The meta-analysis confirmed that the Cliff's δ RE model for the productivity variable had a practical significance, with a medium effect size.

In general, it would appear that the MSc students and practitioners were more productive than the undergraduate students. Since our sample size for this type of participants is limited when compared to the undergraduate students, we cannot draw any conclusions regarding the possible influence of experience in the modeling languages. In future experiments, we plan to investigate how the participants' experience and ability in goal modeling influence modeling time, productivity and the quality of the goal models when using the languages selected. We should, therefore, involve participants with experience in goal modeling. Moreover, the participants' productivity could be enhanced by providing tool support to assist the users when specifying and validating their goal models. Tool support is an important factor that impacts on the usage and acceptance of a goal modeling language.

With regard to the participants' perception of ease of use, the results show that there is neither an interaction effect nor a difference in means between language and system for this variable in any of the experiments. In addition, no effect of system was found in any of the experiments for this variable.

With respect to the language factor, we found a statistical significance for hypothesis $H4_a$ in two experiments of the family (UPV1 and UPV2), but this difference had a practical significance only in the case of UPV1. The meta-analysis results show that the Cliff's δ RE model for the PEOU variable did not have a practical significance (i.e., small effect size). These results suggest that both languages should be improved to make them easier to use. In fact, some participants highlighted that they experienced difficulties when using both languages owing to their qualitative nature, i.e., expressing the stakeholders' intention using the constructs of the language. This was mentioned in their responses to the questionnaire. The main issues are related to difficulties in distinguishing the meanings of some constructs (i.e., goals and the means used to achieve them), representing dependencies among goals, determining the granularity of goal decomposition, solving conflicts among goals and stakeholders' conflicts for a goal. Although this is an inherent problem of goal modeling languages in general, in order to make them easier for users, these languages need: i) formal semantics to ensure that the language elements cannot be misunderstood, or ii) well-defined yet informal constructs with practical guidelines on how to specify a goal model. Unfortunately, many goal-oriented languages are specified only through their abstract syntax and concrete syntax and lack guidelines and/or precise semantics beyond informal explanations.

With regard to the participants' perception of usefulness, we found an interaction effect in UPV1, but this interaction occurred only in two pairs of treatments. We also found a difference in means in UNA, but the *post-hoc* analysis does not suggest any significant difference between the treatment pairs. In addition, no effect of system was found in any of the experiments for this variable.

Nevertheless, we found statistical and practical significance for hypothesis $H5_a$ in three (UPV1, UPV2 and UNA) out of the four experiments in the family. The meta-analysis confirmed that the Cliff's δ RE model for the PU variable has a practical significance, with a medium effect size. An analysis of the individual responses to the questionnaire revealed that the participants scored low on the PU1 item for both languages (i.e., "I believe that the goal models obtained by this language are clear, concise, and unambiguous"). The overall level of agreement with this question was 35%, while for the other items it was over 50%. Further efforts are, therefore, required as regards assessing the understandability of goal modeling languages and improving their clarity. The Physics of Notations [70] could be exploited for this purpose, as it aspires to provide a theory with which to assess and design effective visual notations.

We also tested whether the participants' profile influenced the results, but the effect of this co-factor was confirmed only in the case of software engineers who expressed a higher perception of use for the value@GRL language when compared to the i* language.

All in all, the results are promising, as we obtained empirical evidence regarding in which contexts i* and value@GRL are more effective. We identified some interaction effects or significant differences in mean between the languages and systems for some dependent variables being studied. The results suggest that the effect of value@GRL or i* can vary from a system to another one. Besides, the effect of Green Route or Lattes-Scholar systems can vary from a language to another one. However, these differences occurred only in certain experiments and pairs of treatments. The variation may be caused by the reduced number of observations when considering treatment pairs or other factors such as the system domain or complexity. Hence, further experiments are needed to study the cause of variation. In general, the differences observed do not dramatically impact on the effects of i* and value@GRL as regards the quality, modeling time, productivity, and the perceived ease of use and usefulness of the participants when using these languages.

# 6. Threats to validity

In this section, we follow the recommendations of Wohlin et al. [40] to discuss some of the issues that might have threatened the validity of this family of experiments.

## 6.1 Internal validity

The main threats related to internal validity are: learning effect, participant experience, information exchange among participants, understandability of the materials, and instrumentation validity.

The learning effect was mitigated by using two experimental objects for each experiment in the family. There were no differences on the participants' experience since none of them had previous experience in creating goal models. We were able to prevent information exchange by using different experimental objects in the two runs and monitoring the participants during the experiments. The understandability of the materials was assessed by conducting a pilot study. The analysis of the interaction of the System cofactor with the Language (main factor) on the dependent variables presented in Section 4.2 shows that there are some interactions in certain experiments and pairs of treatments, but the selection of the experimental objects does not severely affect the instrumentation

validity and the experimental results. We mitigated this threat by assessing the complexity of the experimental objects in the pilot study, and several mistakes were identified and corrected. Finally, in order to avoid a possible source of bias, the experimental materials were evaluated by an independent experienced Empirical Software Engineering researcher.

### 6.2 External validity

Threats related to external validity are: representativeness of the results, and the size and complexity of tasks that might affect the generalization of the results.

The representativeness of the results could have been affected by the software systems used and the context of the participants selected. We selected two software systems from different domains. The experimental task can be considered realistic for small-sized projects and they are not trivial. The size and complexity of the tasks may also affect the external validity. We decided to use relatively small tasks since a controlled experiment requires the participants to complete the assigned tasks in a limited amount of time. However, we plan to conduct case studies with larger and more complex tasks in order to confirm or contradict the results obtained.

With regard to the participants' experience, the random heterogeneity of subjects is always present when experimenting with students and practitioners, and we are also conscious that they had no previous knowledge of the goal languages being compared. Although the knowledge of the students involved in our family could be assumed to be comparable to the knowledge of junior industry professionals, the working pressure and the overall environment within industry are different. Experiments in industrial contexts involving participants with experience in goal modeling are, therefore, necessary in order to increase our awareness as regards these results.

### 6.3 Construct validity

The construct validity of our family might have been influenced by both the measures that were applied during the quantitative analysis and the reliability of the questionnaire.

We mitigated this by using measures that are commonly applied in other empirical software engineering studies, including controlled experiments [46] and a meta-analysis [45], [14]. In particular, Quality was measured using an information retrieval-based approach to avoid any subjective evaluation; Modeling Time was measure in minutes; Productivity was measured as a function of Quality and modeling time to create the goal models. The subjective variables (PEOU and PU) were based on TAM [49].

The reliability of the questionnaire as regards assessing the subjective variables was tested using the Cronbach's alpha test. For the UPV1 experiment, questions related to PEOU and PU obtained a Cronbach's α coefficient of 0.805 and 0.867, and the result for the whole questionnaire was 0.741; for the UPV2 experiment, the result was 0.800 and 0.818, and 0.546 for the whole questionnaire; for the UNA experiment, the result was 0.789 and 0.632, and 0.624 for the whole questionnaire; finally, for the UPV3 experiment, the result was 0.850 and 0.592, and 0.567 for the whole questionnaire. Most of the results were higher than the threshold level (0.70) [71]. In addition, as indicated by Loewenthal [72], the α coefficient of 0.6 could be acceptable if the objective is scale development.

Other threats to construct validity that might exist are the participants' apprehension about being evaluated, and hypothesis guessing on their part. Evaluation apprehension has been avoided, since the students were not graded on the results obtained. In order to avoid hypothesis guessing, the students were not made aware that they were part of a study (they were invited to attend a workshop on goal

modeling methods). The participants were volunteers and were aware of the practical and pedagogical purpose of the workshop, but the research questions were not disclosed to them. In addition, bias introduced into the study by expectancies on the part of the experimenter was mitigated while interacting with the participants. We followed the same protocol for each language.

### 6.4 Conclusion validity

With regard to the conclusion validity, the main threats are: the data collection and the validity of the statistical tests applied.

In order to decrease the data collection threat, we applied the same data-extraction procedures in each individual experiment and ensured that each dependent variable was calculated consistently. With regard to the validity of the statistical tests proposed, we considered the recommendations of Maxwell [71]. The statistical tests were selected by considering the type and nature of the variables and were selected by checking that they followed the specific assumptions related to their use.

## 7. Conclusions

In this family of experiments, we have gained empirical evidence on how a recently proposed specialization of a goal-oriented modeling language (value@GRL) may help novice modelers when specifying goal models in comparison to a well-known language (i*).

This evidence is a contribution to the body of knowledge on goal-oriented languages, since it provides factual data about which language is more suitable under certain conditions. In particular, we found evidence supporting the claim that the quality of the goal models created with value@GRL is significantly higher than that of i*. These results are promising, because participants from two different profiles (novice software engineers and business analysts) with relatively low training were able to create models with a good level of correctness and completeness. The results also showed that the participants judged value@GRL to be more useful than i*, although their perceptions on the ease of use of the two languages were similar. Moreover, the results show that neither the profile of the participants nor the system used greatly influenced their performance and perceptions when using i* and value@GRL. Nevertheless, more replications are needed to confirm or refute these results.

From a research perspective, these results may be of interest to the requirements engineering community in general and to novice software engineers and business analysis in particular (since we have tested the usefulness of value@GRL for guiding novice modelers when performing goal modeling). It may, however, also be useful for researchers in the area who wish to replicate the experiments (the research package has been made available online). The evaluation strategy could also be relevant (and reused) by other researchers to evaluate other existing goal modeling languages.

Our findings also have practical implications. We found that the modeling time required to create goal models with value@GRL is somewhat greater than the modeling time required to create goal models with i*. However, the participant's productivity, which takes into account both the quality of the models created and the modeling time, is similar with both languages. Note that the effort required to create these models may decrease after the intensive adoption of the language by an organization. However, this should be assessed empirically. Indeed, we plan to carry out an empirical study to assess the effort involved when modeling with i* and value@GRL. As suggested by Jolak et al. [73], the creation of models consists of different cognitive activities: (i) *designing*, *i.e.*, thinking about the design (ideation, key-design decision making), (ii) *notation expression*, i.e., expressing a design in a modeling notation and (iii) *layouting*, i.e., the spatial organization of model elements in a diagram. In

order to better understand the effort needed to create goal models, we should run experiments to measure how much effort each of these cognitive activities requires.

Other implications are related to education in the field of requirements engineering. Educators confront the need to choose between different modeling languages when teaching requirements analysis and specification. Understanding the strengths and weaknesses of each language, based on the results of studies like ours, may provide the basis required to select the language that is most appropriate for the teaching objectives. In particular, the results could guide educators to focus on certain aspects of a given language so as to better support students in overcoming related modeling difficulties. For instance, in our experiments, the participants had difficulties deciding which type of link to use when applying the two goal modeling languages.

Nevertheless, we are aware that this study provides preliminary results on the effectiveness of value@GRL as a goal modeling language. Although the findings are promising, these results need to be interpreted with caution since they are only valid within the context established in this family of experiments. It is necessary to verify whether the same results hold if more complex experimental objects and practitioners experienced in goal modeling are used. Nevertheless, this study has value as a first family of experiments used to evaluate the goal modeling languages selected with the objective of providing evidence of their usefulness for modeling small and mid-sized software systems.

In terms of future work, we plan to extend the GRL tool to support our process with regard to the modeling and prioritization of intentional elements. Owing to the release of a new version of i* [36], we believe that an interesting research direction will be to compare value@GRL with i* 2.0. This will allow us to extend the findings of our family of experiments by assessing whether the differences observed between value@GRL and i* still hold with the i* 2.0. Furthermore, it may be interesting to appraise whether the treatments have different effects when varying the type of participants (Computer Sciences vs. Business Administration and Management students, graduate vs. undergraduate students). Gathering new data and performing a subgroup analysis may consequently help explain differences regarding these additional factors. Further experiments are also needed to evaluate the other activities in the value@GRL approach.

**Funding**

This work was supported by the Spanish Ministry of Science, Innovation and Universities (Adapt@Cloud project, grant number TIN2017-84550-R) and the "Programa de Ayudas de Investigación y Desarrollo" (PAID-01-17) from the Universitat Politècnica de València.

# References

[1] L. Liu, E. Yu, Designing information systems in social context: a goal and scenario modelling approach, Inf. Syst. 29 (2004) 187–203. doi:10.1016/S0306-4379(03)00052-8.

[2] E. Gonçalves, J. Castro, J. Araújo, T. Heineck, A Systematic Literature Review of iStar extensions, J. Syst. Softw. 137 (2018) 1–33. doi:10.1016/J.JSS.2017.11.023.

[3] E.S.K. Yu, Towards modelling and reasoning support for early-phase requirements engineering, in: Proc. ISRE '97 3rd IEEE Int. Symp. Requir. Eng., IEEE Comput. Soc. Press, n.d.: pp. 226–235. doi:10.1109/ISRE.1997.566873.

[4] A. Lapouchnian, Y. Yu, S. Liaskos, J. Mylopoulos, Requirements-driven Design of Autonomic Application Software, in: Proc. 2006 Conf. Cent. Adv. Stud. Collab. Res., IBM Corp., Riverton, NJ, USA, 2006. doi:10.1145/1188966.1188976.

[5] E. Insfran, S. Abrahao, R. de Oliveira, F. González-Ladrón-de-Guevara, M. Fernández-Diego, C. Cano-Genoves, Specifying Value in GRL for Guiding BPMN Activities Prioritization, Int. Conf. Inf. Syst. Dev. (2017). https://aisel.aisnet.org/isd2014/proceedings2017/ISDMethodologies/10.

[6] ITU-T – International Telecommunications Union, Recommendation Z.151 (11/08) User Requirements Notation (URN) – Language definition, (2008) 192. https://www.itu.int/rec/T-REC-Z.151-200811-S/en (accessed November 14, 2018).

[7] J. Horkoff, T. Li, F.-L. Li, M. Salnitri, E. Cardoso, P. Giorgini, J. Mylopoulos, Using goal models downstream: a systematic roadmap and literature review, Int. J. Inf. Syst. Model. Des. 6 (2015) 1–42.
[8] J. Horkoff, F.B. Aydemir, E. Cardoso, T. Li, A. Maté, E. Paja, M. Salnitri, L. Piras, J. Mylopoulos, P. Giorgini, Goal-oriented requirements engineering: an extended systematic mapping study, Requir. Eng. (2017) 1–28. doi:10.1007/s00766-017-0280-z.
[9] J.C. Carver, N. Juristo, M.T. Baldassarre, S. Vegas, Replications of software engineering experiments, Empir. Softw. Eng. 19 (2014) 267–276. doi:10.1007/s10664-013-9290-8.
[10] D.T. Campbell, J.C. Stanley, Experimental and quasi-experimental designs for research, Handb. Res. Teach. (1963) 171–246.
[11] S. Abrahão, E. Insfran, F.G.-L. de Guevara, M. Fernández-Diego, C. Cano-Genoves, R.P. de Oliveira, Comparing the effectiveness of goal-oriented languages: results from a controlled experiment, in: Proc. 12th ACM/IEEE Int. Symp. Empir. Softw. Eng. Meas. - ESEM '18, ACM Press, New York, New York, USA, 2018: pp. 1–4. doi:10.1145/3239235.3267433.
[12] R. Matulevičius, P. Heymans, Comparing Goal Modelling Languages: An Experiment, in: Requir. Eng. Found. Softw. Qual., Springer Berlin Heidelberg, Berlin, Heidelberg, 2007: pp. 18–32. doi:10.1007/978-3-540-73031-6_2.
[13] V.R. Basili, F. Shull, F. Lanubile, Building knowledge through families of experiments, IEEE Trans. Softw. Eng. 25 (1999) 456–473. doi:10.1109/32.799939.
[14] A. Santos, O.S. Gomez, N. Juristo, Analyzing Families of Experiments in SE: a Systematic Mapping Study, IEEE Trans. Softw. Eng. (2018). doi:10.1109/TSE.2018.2864633.
[15] A. Dardenne, A. Van Lamsweerde, S. Fickas, Goal-directed requirements acquisition, Sci. Comput. Program. 20 (1993) 3–50.
[16] A.I. Anton, Goal-based requirements analysis, in: Proc. Second Int. Conf. Requir. Eng., IEEE Comput. Soc. Press, n.d.: pp. 136–144. doi:10.1109/ICRE.1996.491438.
[17] L. Chung, B.A. Nixon, E. Yu, J. Mylopoulos, Non-Functional Requirements in Software Engineering, Springer US, 2000.
[18] J. Mylopoulos, M. Kolp, J. Castro, UML for Agent-Oriented Software Development: The Tropos Proposal, in: Springer, Berlin, Heidelberg, 2001: pp. 422–441. doi:10.1007/3-540-45441-1_31.
[19] E. Kavakli, P. Loucopoulos, Goal Modeling in Requirements Engineering: Analysis and Critique of Current Methods, in: Inf. Model. Methods Methodol., IGI Global, 2005: pp. 102–124. doi:10.4018/9781591403753.ch006.
[20] X. Franch, L. López, C. Cares, D. Colomer, The i* Framework for Goal-Oriented Modeling, in: Domain-Specific Concept. Model., Springer International Publishing, Cham, 2016: pp. 485–506. doi:10.1007/978-3-319-39417-6_22.
[21] H. Estrada, K. Najera, B. Vázquez, A. Martinez, J.C. Tellez, J.J. Hierro, Applying Tropos modeling for smart mobility applications based on the FIWARE platform, CEUR Workshop Proc. 1674 (2016) 85–90.
[22] Project Management Institute (PMI), Requirements Management: Core Competency for Project and Program Success, 2014.
[23] K. Siau, M. Rossi, Evaluation techniques for systems analysis and design modelling methods - a review and comparative analysis, Inf. Syst. J. 21 (2011) 249–268. doi:10.1111/j.1365-2575.2007.00255.x.
[24] G. Regev, A. Wegmann, Where do goals come from: the underlying principles of goal-oriented requirements engineering, in: 13th IEEE Int. Conf. Requir. Eng., IEEE, 2005: pp. 353–362. doi:10.1109/RE.2005.80.
[25] J. Horkoff, E. Yu, Analyzing goal models: different approaches and how to choose among them, in: Proc. 2011 ACM Symp. Appl. Comput. - SAC '11, ACM Press, New York, New York, USA, 2011: p. 675. doi:10.1145/1982185.1982334.
[26] M.A. Teruel, E. Navarro, V. López-Jaquero, F. Montero, P. González, Comparing Goal-Oriented Approaches to Model Requirements for CSCW, in: Springer, Berlin, Heidelberg, 2013: pp. 169–184. doi:10.1007/978-3-642-32341-6_12.
[27] C.P. Ayala Martínez, C. Cares, J.P. Carvallo Vega, G. Grau Colom, M. Haya, G. Salazar, J. Franch Gutiérrez, E. Mayol Sarroca, M.C. Quer Bosor, A comparative analisys of i*-based agent-oriented modeling languages, in: Proc. 17th Int. Conf. Softw. Eng. Knowl. Eng., 2005: pp. 43–50.
[28] H.S. F. Al-subaie, T.S. E. Maibaum, Evaluating the Effectiveness of a Goal-Oriented Requirements Engineering Method, in: Fourth Int. Work. Comp. Eval. Requir. Eng. (CERE'06 - RE'06 Work., IEEE, 2006: pp. 8–19. doi:10.1109/CERE.2006.3.
[29] R. Matuleviius, P. Heymans, A.L. Opdahl, R. Matulevičius, P. Heymans, A.L. Opdahl, Comparing GRL and KAOS using the UEML Approach, Springer London, London, n.d. doi:10.1007/978-1-84628-858-6_7.
[30] M.A. Teruel, E. Navarro, V. López-Jaquero, F. Montero, J. Jaen, P. González, Analyzing the understandability of Requirements Engineering languages for CSCW systems: A family of experiments, Inf. Softw. Technol. 54 (2012) 1215–1228. doi:10.1016/J.INFSOF.2012.06.001.
[31] J.M. Morales, E. Navarro, P. Sánchez, D. Alonso, A controlled experiment to evaluate the understandability of KAOS and i* for modeling Teleo-Reactive systems, J. Syst. Softw. 100 (2015) 1–14. doi:10.1016/J.JSS.2014.10.010.
[32] J.M. Morales, E. Navarro, P. Sánchez, D. Alonso, A family of experiments to evaluate the understandability of TRiStar and i∗ for modeling teleo-reactive systems, J. Syst. Softw. 114 (2016) 82–100. doi:10.1016/j.jss.2015.12.056.
[33] A. Maté, J. Trujillo, X. Franch, Adding semantic modules to improve goal-oriented analysis of data warehouses using I-star, J. Syst. Softw. 88 (2014) 102–111. doi:10.1016/J.JSS.2013.10.011.
[34] J. Krogstie, A Semiotic Approach to Quality in Requirements Specifications, in: Springer, Boston, MA, 2002: pp.

231–249. doi:10.1007/978-0-387-35611-2_14.
[35] V.R. Basili, H.D. Rombach, The TAME project: towards improvement-oriented software environments, IEEE Trans. Softw. Eng. 14 (1988) 758–773. doi:10.1109/32.6156.
[36] F. Dalpiaz, X. Franch, J. Horkoff, istar 2.0 language guide, ArXiv Prepr. ArXiv1605.07767. (2016). http://arxiv.org/abs/1605.07767 (accessed November 14, 2018).
[37] M. Serrano, J.C.S. do Prado Leite, Development of agent-driven systems: From i*; architectural models to intentional agents' code, CEUR Workshop Proc. 766 (2011) 55–60.
[38] N. Juristo, O.S. Gómez, Replication of Software Engineering Experiments, in: Springer, Berlin, Heidelberg, 2012: pp. 60–88. doi:10.1007/978-3-642-25231-0_2.
[39] O.S. Gómez, N. Juristo, S. Vegas, Understanding replication of experiments in software engineering: A classification, Inf. Softw. Technol. 56 (2014) 1033–1048. doi:10.1016/J.INFSOF.2014.04.004.
[40] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering, 2012. doi:10.1007/978-3-642-29044-2.
[41] N. Pippenger, Complexity Theory, Sci. Am. 238 (1978) 114–125. doi:10.2307/24955758.
[42] H.J. Nelson, G. Poels, M. Genero, M. Piattini, A conceptual modeling quality framework, Softw. Qual. J. 20 (2012) 201–228. doi:10.1007/s11219-011-9136-9.
[43] O.I. Lindland, G. Sindre, A. Solvberg, Understanding quality in conceptual modeling, IEEE Softw. 11 (1994) 42–49. doi:10.1109/52.268955.
[44] W.B. Frakes, R. Baeza-Yates, Information retrieval: data structures and algorithms, Prentice Hall PTR. (1992).
[45] S. Abrahão, C. Gravino, E. Insfran, G. Scanniello, G. Tortora, Assessing the effectiveness of sequence diagrams in the comprehension of functional requirements: Results from a family of five experiments, IEEE Trans. Softw. Eng. 39 (2013) 327–342. doi:10.1109/TSE.2012.27.
[46] G. Scanniello, U. Erra, Distributed modeling of use case diagrams with a method based on think-pair-square: Results from two controlled experiments, J. Vis. Lang. Comput. 25 (2014) 494–517. doi:10.1016/J.JVLC.2014.03.002.
[47] E. Souza, A. Moreira, J. Araújo, S. Abrahão, E. Insfran, D.S. da Silveira, Comparing business value modeling methods: A family of experiments, Inf. Softw. Technol. 104 (2018) 179–193. doi:10.1016/J.INFSOF.2018.08.001.
[48] K. Labunets, Katsiaryna, No search allowed: what risk modeling notation to choose?, in: Proc. 12th ACM/IEEE Int. Symp. Empir. Softw. Eng. Meas. - ESEM '18, ACM Press, New York, New York, USA, 2018: pp. 1–10. doi:10.1145/3239235.3239247.
[49] F.D. Davis, Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology, MIS Q. 13 (1989) 319. doi:10.2307/249008.
[50] W.R. King, J. He, A meta-analysis of the technology acceptance model, Inf. Manag. 43 (2006) 740–755. doi:10.1016/J.IM.2006.05.003.
[51] P.J. Hu, P.Y.K. Chau, O.R.L. Sheng, K.Y. Tam, Examining the Technology Acceptance Model Using Physician Acceptance of Telemedicine Technology, J. Manag. Inf. Syst. 16 (1999) 91–112. doi:10.1080/07421222.1999.11518247.
[52] M.R. Hess, J.D. Kromrey, Robust confidence intervals for effect sizes: A comparative study of Cohen'sd and Cliff's delta under non-normality and heterogeneous variances, in: Annu. Meet. Am. Educ. Res. Assoc., 2004: pp. 1–30.
[53] N. Cliff, Dominance statistics: Ordinal analyses to answer ordinal questions., Psychol. Bull. 114 (1993) 494–509. doi:10.1037/0033-2909.114.3.494.
[54] P.D. Ellis, The essential guide to effect sizes : statistical power, meta-analysis, and the interpretation of research results, Cambridge University Press, 2010.
[55] L. V Hedges, I. Olkin, Statistical methods for meta-analysis, Academic Press, 1985.
[56] B. Kitchenham, L. Madeyski, D. Budgen, J. Keung, P. Brereton, S. Charters, S. Gibbs, A. Pohthong, Robust Statistical Methods for Empirical Software Engineering, Empir. Softw. Eng. 22 (2017) 579–630. doi:10.1007/s10664-016-9437-5.
[57] H.C. Kraemer, D.J. Kupfer, Size of Treatment Effects and Their Importance to Clinical Research and Practice, Biol. Psychiatry. 59 (2006) 990–996. doi:10.1016/J.BIOPSYCH.2005.09.014.
[58] J. Cohen, Statistical power analysis for the behavioral sciences 2nd edn, (1988).
[59] R.C. Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Austria, 2015, (2018).
[60] M. Torchiano, Efficient effect size computation, (2017). https://cran.r-project.org/package=effsize (accessed October 10, 2018).
[61] W. Viechtbauer, Conducting meta-analyses in R with the metafor package, J. Stat. Softw. 36 (2010).
[62] K. Ried, Interpreting and understanding meta-analysis graphs: a practical guide, (2006).
[63] M. Borenstein, L. V Hedges, J.P.T. Higgins, H.R. Rothstein, A basic introduction to fixed-effect and random-effects models for meta-analysis, Res. Synth. Methods. 1 (2010) 97–111. doi:10.1002/jrsm.12.
[64] T.B. Huedo-Medina, J. Sanchez-Meca, F. Marin-Martinez, J. Botella, Assessing heterogeneity in meta-analysis: Q statistic or I2 index?, Psychol. Methods. 11 (2006) 193–206. doi:10.1037/1082-989X.11.2.193.
[65] W.G. Cochran, The combination of estimates from different experiments., Biometrics. 10 (1954) 101–129. doi:10.2307/3001666.
[66] R.J. Hardy, S.G. Thompson, Detecting and describing heterogeneity in meta-analysis., Stat. Med. 17 (1998) 841–856.
[67] J.P.T. Higgins, S. Green, Cochrane Handbook for Systematic Reviews of Interventions, 2011.
[68] J.P.T. Higgins, S.G. Thompson, J.J. Deeks, D.G. Altman, Measuring inconsistency in meta-analyses, BMJ Br. Med.

J. 327 (2003) 557. http://www.ncbi.nlm.nih.gov/pubmed/12958120.
[69] J.A. Berlin, E.M. Antman, Advantages and limitations of metaanalytic regressions of clinical trials data., Online J. Curr. Clin. Trials. Doc No 134 (1994) [8425 words; 84 paragraphs].
[70] D. Moody, The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering, IEEE Trans. Softw. Eng. 35 (2009) 756–779. doi:10.1109/TSE.2009.67.
[71] K.D. Maxwell, Applied statistics for software managers, Appl. Stat. Softw. Manag. (2002).
[72] K. Loewenthal, C.A. Lewis, C.A. Lewis, An Introduction to Psychological Tests and Scales, (2018). doi:10.4324/9781315782980.
[73] R. Jolak, E. Umuhoza, T. Ho-Quang, M.R. V Chaudron, M. Brambilla, Dissecting Design Effort and Drawing Effort in UML Modeling, in: 2017 43rd Euromicro Conf. Softw. Eng. Adv. Appl., IEEE, 2017: pp. 384–391. doi:10.1109/SEAA.2017.55.

# Appendix A. Descriptive statistics

In the following, we show descriptive statistics per language and system for each dependent variable and experiment in the family.

**Table A-1.** Descriptive statistics for the variables in the family of experiments.

| Exp | Language | System | Quality | | | | Modeling Time | | | | Productivity | | | | PEOU | | | | PU | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Mean | SD | Min | Max | Mean | SD | Min | Max | Mean | SD | Min | Max | Mean | SD | Min | Max | Mean | SD |
| UPV1 | i* | All | 19.37 | 54.32 | 38.69 | 9.11 | 24.00 | 74.00 | 46.15 | 15.27 | 0.51 | 1.62 | 0.92 | 0.38 | 2.00 | 5.00 | 3.55 | 0.94 | 1.67 | 4.83 | 3.34 | 0.90 |
| | | O1 | 23.05 | 54.32 | 40.99 | 9.62 | 24.00 | 63.00 | 38.50 | 12.78 | 0.63 | 1.62 | 1.15 | 0.39 | 2.00 | 5.00 | 3.67 | 1.02 | 1.67 | 4.50 | 3.18 | 0.95 |
| | | O2 | 19.37 | 50.27 | 36.39 | 8.42 | 34.00 | 74.00 | 53.80 | 14.09 | 0.51 | 1.01 | 0.70 | 0.19 | 2.00 | 5.00 | 3.43 | 0.90 | 2.00 | 4.83 | 3.50 | 0.86 |
| | value@GRL | All | 38.41 | 83.07 | 55.86 | 10.97 | 27.00 | 68.00 | 48.20 | 11.79 | 0.58 | 2.64 | 1.25 | 0.48 | 3.00 | 5.00 | 4.33 | 0.61 | 2.50 | 5.00 | 3.96 | 0.74 |
| | | O1 | 49.18 | 83.07 | 60.97 | 11.31 | 28.00 | 62.00 | 45.80 | 10.28 | 0.80 | 2.64 | 1.42 | 0.53 | 3.67 | 5.00 | 4.57 | 0.47 | 3.17 | 5.00 | 4.33 | 0.56 |
| | | O2 | 38.41 | 64.68 | 50.74 | 8.26 | 27.00 | 68.00 | 50.60 | 13.23 | 0.58 | 1.58 | 1.09 | 0.37 | 3.00 | 5.00 | 4.10 | 0.67 | 2.50 | 4.50 | 3.58 | 0.72 |
| UPV2 | i* | All | 25.76 | 74.96 | 42.96 | 11.38 | 32.00 | 75.00 | 51.56 | 9.91 | 0.37 | 1.49 | 0.87 | 0.29 | 2.00 | 5.00 | 3.55 | 0.80 | 2.17 | 5.00 | 3.53 | 0.60 |
| | | O1 | 29.14 | 74.96 | 47.67 | 11.24 | 32.00 | 75.00 | 51.88 | 11.37 | 0.58 | 1.49 | 0.96 | 0.30 | 2.67 | 4.67 | 3.69 | 0.58 | 2.17 | 4.17 | 3.39 | 0.49 |
| | | O2 | 25.76 | 58.64 | 38.26 | 9.69 | 36.00 | 70.00 | 51.25 | 8.57 | 0.37 | 1.33 | 0.77 | 0.26 | 2.00 | 5.00 | 3.42 | 0.97 | 2.17 | 5.00 | 3.67 | 0.68 |
| | value@GRL | All | 37.01 | 75.52 | 54.50 | 10.46 | 31.00 | 92.00 | 63.47 | 14.67 | 0.52 | 1.64 | 0.90 | 0.26 | 2.00 | 5.00 | 3.93 | 0.74 | 2.17 | 5.00 | 3.90 | 0.63 |
| | | O1 | 37.01 | 75.52 | 53.10 | 11.23 | 31.00 | 82.00 | 61.06 | 15.65 | 0.52 | 1.59 | 0.92 | 0.27 | 2.00 | 5.00 | 3.98 | 0.78 | 2.17 | 5.00 | 3.80 | 0.72 |
| | | O2 | 41.62 | 72.90 | 55.90 | 9.78 | 40.00 | 92.00 | 65.88 | 13.69 | 0.58 | 1.64 | 0.89 | 0.27 | 2.33 | 4.67 | 3.88 | 0.71 | 3.00 | 5.00 | 3.99 | 0.54 |
| UNA | i* | All | 21.59 | 52.96 | 33.51 | 10.59 | 52.00 | 123.00 | 97.40 | 19.88 | 0.18 | 1.02 | 0.38 | 0.24 | 2.00 | 5.00 | 3.73 | 1.02 | 2.67 | 5.00 | 3.45 | 0.63 |
| | | O1 | 22.73 | 52.96 | 39.62 | 12.16 | 52.00 | 100.00 | 84.80 | 20.63 | 0.23 | 1.02 | 0.52 | 0.30 | 3.33 | 5.00 | 4.20 | 0.61 | 2.67 | 3.67 | 3.23 | 0.43 |
| | | O2 | 21.59 | 30.22 | 27.41 | 3.39 | 101.00 | 123.00 | 110.00 | 8.15 | 0.18 | 0.29 | 0.25 | 0.05 | 2.00 | 5.00 | 3.27 | 1.19 | 3.17 | 5.00 | 3.67 | 0.77 |
| | value@GRL | All | 32.61 | 59.34 | 46.39 | 9.29 | 34.00 | 60.00 | 49.20 | 7.64 | 0.57 | 1.57 | 0.98 | 0.30 | 3.00 | 5.00 | 3.97 | 0.58 | 3.33 | 4.83 | 4.25 | 0.52 |
| | | O1 | 32.61 | 59.34 | 46.29 | 12.47 | 47.00 | 60.00 | 52.60 | 5.32 | 0.57 | 1.22 | 0.90 | 0.29 | 3.67 | 5.00 | 4.13 | 0.56 | 3.33 | 4.50 | 4.03 | 0.52 |
| | | O2 | 39.47 | 53.47 | 46.49 | 6.20 | 34.00 | 56.00 | 45.80 | 8.61 | 0.70 | 1.57 | 1.06 | 0.32 | 3.00 | 4.33 | 3.80 | 0.61 | 3.67 | 4.83 | 4.47 | 0.46 |
| UPV3 | i* | All | 17.53 | 82.10 | 36.08 | 12.02 | 15.00 | 72.00 | 50.13 | 13.36 | 0.37 | 3.35 | 0.81 | 0.55 | 1.33 | 4.33 | 3.22 | 0.79 | 2.50 | 4.83 | 3.33 | 0.52 |
| | | O1 | 20.49 | 52.36 | 35.55 | 9.20 | 15.00 | 72.00 | 46.93 | 12.53 | 0.37 | 3.35 | 0.89 | 0.71 | 1.67 | 4.00 | 3.33 | 0.71 | 2.50 | 4.83 | 3.41 | 0.62 |
| | | O2 | 17.53 | 82.10 | 36.61 | 14.63 | 23.00 | 70.00 | 53.33 | 13.82 | 0.45 | 1.68 | 0.72 | 0.32 | 1.33 | 4.33 | 3.11 | 0.87 | 2.50 | 4.00 | 3.26 | 0.41 |
| | value@GRL | All | 20.79 | 93.72 | 47.71 | 17.90 | 37.00 | 89.00 | 58.47 | 12.50 | 0.34 | 1.87 | 0.85 | 0.37 | 1.67 | 5.00 | 3.39 | 0.82 | 2.33 | 4.33 | 3.32 | 0.57 |
| | | O1 | 25.67 | 93.72 | 54.90 | 21.04 | 41.00 | 75.00 | 56.73 | 9.22 | 0.50 | 1.87 | 0.98 | 0.40 | 2.00 | 5.00 | 3.40 | 0.91 | 2.33 | 4.17 | 3.28 | 0.58 |
| | | O2 | 20.79 | 58.41 | 40.52 | 10.50 | 37.00 | 89.00 | 60.20 | 15.24 | 0.34 | 1.33 | 0.72 | 0.29 | 1.67 | 4.67 | 3.38 | 0.75 | 2.50 | 4.33 | 3.36 | 0.59 |

O1 = Green Route; O2 = Lattes Scholar