



A technique for evaluating and improving the semantic transparency of modeling language notations

Dominik Bork¹ · Ben Roelens²

Received: 30 September 2020 / Revised: 5 March 2021 / Accepted: 14 May 2021 / Published online: 15 June 2021
© The Author(s) 2021

Abstract

The notation of a modeling language is of paramount importance for its efficient use and the correct comprehension of created models. A graphical notation, especially for domain-specific modeling languages, should therefore be aligned to the knowledge, beliefs, and expectations of the targeted model users. One quality attributed to notations is their semantic transparency, indicating the extent to which a notation intuitively suggests its meaning to untrained users. Method engineers should thus aim at semantic transparency for realizing intuitively understandable notations. However, notation design is often treated poorly—if at all—in method engineering methodologies. This paper proposes a technique that, based on iterative evaluation and improvement tasks, steers the notation toward semantic transparency. The approach can be efficiently applied to arbitrary modeling languages and allows easy integration into existing modeling language engineering methodologies. We show the feasibility of the technique by reporting on two cycles of Action Design Research including the evaluation and improvement of the semantic transparency of the Process-Goal Alignment modeling language notation. An empirical evaluation comparing the new notation against the initial one shows the effectiveness of the technique.

Keywords Modeling language · Notation · Concrete syntax · Semantic transparency · Empirical evaluation

1 Introduction

Due to their abstracting power, conceptual models are excellent in decreasing the complexity of a system under study, thereby highlighting its relevant aspects for means of understanding and communication by human beings [47]. With the growing complexity of today's enterprise ecosystems, conceptual modeling plays an increasingly important role to facilitate the understanding of and communication between relevant stakeholders. Consequently, new domain-specific modeling languages are continuously developed and existing languages revised. In both cases, method engineers are not only challenged with the specification of metamodels

but also with the design of adequate graphical notations. In the context of overcoming the notation design challenge, the demand for intuitively understandable graphical notations advances [23], consequently asking to fill a research gap of specialized design techniques [16,22]. This affects both general-purpose (see [5,13]) and domain-specific modeling languages (DSMLs) [31].

A conceptual model should be *a conscious construction to achieve a goal that goes beyond the making of the model itself*, ranging from *intermediate artifacts in systems analysis and design to sense-making and communication, model simulation, quality assurance, and requirements specification* [34, p. 186]. An intuitive modeling language notation is important to achieve the goal of efficient communication [8,18,47] as it establishes the “first contact of the users with the modeling language” [13, p. 123]. A well-designed graphical notation is the first precondition for adoption and correct use of a modeling method [5]. A graphical notation should thus support the modeler in creating and the user in interpreting a model. An *intuitive* notation should moreover account for *computational offloading*, i.e., shifting some of the *cognitive tasks to perceptual tasks* [44], which ultimately leads to an efficient and intuitive understanding of a modeling

Communicated by Jelena Zdravkovic and Iris Reinhartz-Berger.

✉ Dominik Bork
dominik.bork@tuwien.ac.at

Ben Roelens
ben.roelens@ou.nl

¹ Business Informatics Group, TU Wien, Favoritenstr. 9-11, 1040 Vienna, Austria

² Faculty of Science, Open University, Postbus 2960, 6401 DL Heerlen, The Netherlands

language [43]. Intuitiveness is formalized by the concept of *Semantic Transparency* as proposed in [44], i.e., the extent to which the graphical representation encodes the meaning of a modeling language concept. Other researchers refer to the intuitiveness of a notation by *readability*—models are represented “in a natural way and can be easily understood without the need for further explanations” [2, p. 214], *pragmatic quality*—“correspondence between the model and the audience’s interpretation of the model” [35, p. 94], or *understandability*—“the ease with which the concepts and structures in the [...] model can be understood by the users of the model” [46]. The understanding of a model thus establishes a fundamental prerequisite of realizing further modeling goals such as model analysis, model transformation, and code generation.

The design of modeling language notations should be guided by iterative cycles of development and evaluation [16, 30]. However, this design is difficult to formalize [40,45] as a “conceptual model can only be evaluated against people’s (tacit) needs, desires, and expectations” [45, p. 245]. The situation is even more problematic when focusing on intuitiveness, which can only be evaluated when the user’s knowledge, beliefs, and aptitudes are known. In contrast to general-purpose modeling languages (GPMLs), these aspects are concretely known to the method engineer when developing a DSML. Furthermore, *Semantic Transparency* formalizes the notion of intuitiveness in the context of an empirical evaluation [44]. Consequently, our research question entails: “How can a technique support a DSML method engineer in evaluating and improving the semantic transparency of an initial modeling language notation?”

This paper builds upon previous work [6,55] and extends it by proposing an extended notation evaluation and improvement technique and by demonstrating its use through an application [50]. As a result, we report on the lessons that were learnt during two cycles of Action Design Research [60]. The core of the technique is composed of two phases, which are further decomposed into several tasks. The first phase concentrates on *evaluating* an initial version of the notation, whereas the second phase concentrates on developing and evaluating an *improved* notation with respect to semantic transparency. The approach is cyclic and meant to be applied in an iterative manner. It continuously involves practitioners throughout the phases as stressed, e.g., by [54]. Moreover, the technique is modularly structured to enable flexibility in its application.

This paper is structured as follows. Section 2 establishes the foundations necessary to understand the research contribution. The Action Design Research methodology is presented in Sect. 3, while the extended technique is introduced in Sect. 4. The application of the technique to the initial notation of the Process-Goal Alignment (PGA) modeling language is covered in Sect. 5, after which concrete improvements are proposed and comparatively evaluated in

Sect. 6. Related work is presented in Sect. 7, while Sect. 8 discusses validity, applicability, scalability, and the extent to which the method requirements have been met. Section 9 concludes this paper and points to future research directions.

2 Foundations

Section 2 discusses the theoretical foundations of this research. First, domain-specific conceptual modeling is introduced. Afterward, the focus narrows to the visualization of modeling languages and to semantic transparency.

2.1 Domain-specific conceptual modeling

A conceptual modeling language is composed of three components [7]: an *abstract syntax*, commonly referred to and specified by means of a *metamodel* (i.e., the specification of the abstract concepts that can be used to create valid models), a *concrete syntax*, commonly referred to as a *notation* (i.e., the specification of the graphical and/or textual representations of the metamodel elements), and *semantics* (i.e., the specification of the meaning attached to the abstract syntax elements).

Based on their scope, GPMLs, such as the Unified Modeling Language (UML), can be differentiated from DSMLs [4]. Evaluating the semantic transparency of GPML notations is problematic because test persons who reflect the various stakeholders and purposes addressed by these languages are not available. Moreover, GPMLs tend to be positioned on very abstract levels, which hampers the identification of semantically transparent notations. Evaluating the semantic transparency of a new DSML, on the other hand, is feasible because the potential users and their purposes of using the DSML are more crystallized. As such, DSML engineers are challenged to reflect domain-specificity not only in the language’s abstract syntax but also in its graphical representation.

2.2 Visual aspects in conceptual modeling

Newell and Simon [48] developed a decoding theory considering humans as information processing entities. Information processing can be divided into two steps: *perceptual processing* (i.e., *seeing*), which is fast and automatic, and *cognitive processing* (i.e., *understanding*), which is slow and resource-intensive (see Fig. 1). Diagrams lead to *computational offloading*: they replace some cognitive tasks by perceptual ones. Consequently, the objective of designing good (i.e., cognitively effective) notations needs to be the reduction in cognitive processing. Similarly, Moody states that “designing cognitively effective visual notations can [...]”

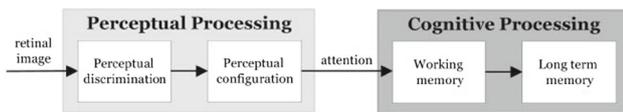


Fig. 1 Perceptual and cognitive processing [44, p. 761]

be seen as a problem of optimizing them for processing by the human mind” [44, p. 761].

In conceptual modeling, an appealing and semantically transparent visual representation is vital for acceptance and adoption of the modeling method [13, p. 123]. “The extent to which diagrams exploit perceptual processing largely explains differences in their effectiveness” [44, p. 761] (see also [36,51]). Comprehensive design principles for conceptual modeling notations were proposed by Daniel Moody’s Physics of Notation [44]. Moody not only established an empirical foundation for conducting research in this field, he also proposed nine principles to foster the design of cognitive effective notations. The motivation for his research was that “cognitive effectiveness of visual notations is one of the most widely held (and infrequently challenged) assumptions in the IT field. However, cognitive effectiveness is not an intrinsic property of visual representations but something that must be designed into them” [44, p. 757].

2.3 Semantic transparency

Semantic transparency is one of the design principles for efficient visual notations [44]. It is defined as “the extent to which a novice reader can infer the meaning of a symbol from its appearance alone” [44, p. 765]. Semantic transparency is the relevant design principle for our research as it formalizes the notion of intuitiveness, i.e., novice users having no training on a modeling language are capable of intuitively deriving the meaning of the language elements from looking at their notation [44]. A notation with a high semantic transparency moves cognitive processing toward perceptual processing as users can infer the meaning of a symbol/model from their working and/or long-term memory. Semantic transparency therefore “plays a crucial role in [the] acceptance” of modeling languages [13, p. 123].

The extent to which a visual notation exploits semantic transparency can be interpreted as a spectrum (see Fig. 2). In the worst case, a notation whose appearance suggests an incorrect meaning is considered *semantically perverse (false mnemonic)*. The neutral case refers to notations having an arbitrary relationship between their appearance and meaning (i.e., *semantically opaque (conventional)*). In the best case, a notation’s appearance suggests the correct meaning. This latter case is referred to as *semantically transparent (mnemonic)*.

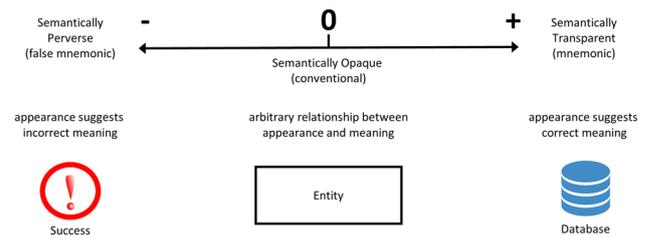


Fig. 2 Semantic transparency spectrum (cf., [8, p. 115], [44, p. 765])

Semantic transparency improves the empirical quality of conceptual models, which *deals with comprehensibility when a visual model is read by different social actors* [34, p. 196]. In the SEQUEL quality framework [34], guidelines are provided for graph aesthetics, but those are not explicitly oriented toward the intuitiveness of graphical representations. As such, the proposed technique could act as a refinement of the SEQUEL framework to improve the quality of DSMLs.

3 Research methodology and background

The following section describes the Action Design Research methodology and its chronological application process that resulted in the notation evaluation and improvement technique we are presenting in this paper.

3.1 Action design research

This research follows the Design Science Research [25] paradigm, more specifically the Action Design Research [60] (ADR) methodology. In ADR, iterative developments of the artifact are continuously supported by reflection and learning stages that steer the designed artifact toward its most effective utilization in its intended organizational context. The ADR methodology is structured by four stages [60], which will be described in the remainder of this paragraph.

3.1.1 Problem formulation

In the problem formulation stage, a research opportunity needs to be identified and conceptualized. While the importance of a modeling language notation has been stressed in Sects. 1 and 2, we now analyze existing modeling method design approaches with respect to the extent to which they consider notation aspects.

Several approaches have been proposed to support the development of new DSMLs. Table 1 summarizes the findings of our analysis. This shows the relevance of a notation for modeling method development as only one approach [12] does not include this aspect. Although the remaining seven approaches consider notation, development and evaluation support is mediocre.

Table 1 Overview of notation consideration in modeling method design approaches

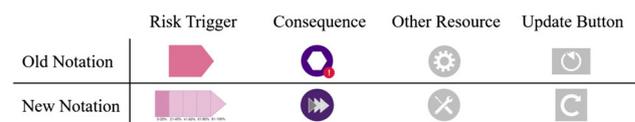
Approach	Aims & scope	Notation consideration	Development & evaluation support
[30]	Agile development of modeling methods with emphasis on evolving modeling method requirements	☉	Generic requirements and sources for notation are described. Concrete support is not given
[16]	A procedural approach comprising macro- and micro-processes that steer the design of DSMLs	☉	Generic requirements and design guidelines for notation are described in great detail. Concrete support is out of scope
[42]	A procedural approach aiming to support the creation of new DSMLs	☉	Graphical notation is emphasized, but concrete support is not provided
[12]	Goal- and value-oriented modeling language development approach	☉	Not in focus
[3]	Model-driven development of multi-view modeling methods and tools	☉	Graphical notation (even viewpoint-specific ones) is emphasized, but concrete support is not provided
[32]	Guidance in the design of DSMLs	☉	A few guidelines for textual and graphical concrete notation are given
[28]	A framework guiding the development of domain-specific process modeling languages	☉	Graphical notation is emphasized, but concrete support is not provided
[62]	A procedural approach that focuses the development of DSMLs in the software engineering domain	☉	The approach puts emphasis on concrete notation design, but merely on textual notations like XML. Concrete support for graphical notations is not given

Three approaches [3,28,42] stress the importance of a graphical notation during the development of modeling methods, but they do not support design and evaluation of a modeling language notation. The approaches described in [16,30,32] further informally support notation design by providing guidelines or by referring to notation design literature. However, these guidelines are too generic to be directly applicable, e.g., “the adoption of ‘customized notations’” [30, p. 8], or “make elements distinguishable” [32, p. 4]. Although such generic guidelines are helpful, their application is challenging and measuring their fulfilment is not possible due to a lack of metrics. The work in [62] provides some concrete guidelines for notation design but only consider a textual concrete syntax and therefore lack support for graphical notation design.

The preceding analysis points to the lack of specific support for modeling method engineers. We see a gap in hands-on tasks to be performed by method engineers when evaluating and improving a given DSML notation. This gap will be addressed by the notation evaluation and improvement technique.

3.1.2 Building, intervention, and evaluation

In the second ADR stage, the artifact is initially designed and “further shaped by organizational use and subsequent design

**Fig. 3** Revised notation for four BCM concepts

cycles [60, p. 41]. Considering the existing body of knowledge on notation design, an initial artifact was created and presented in [6]. It initially comprised three tasks: notation association, term association, and case study. The technique has been applied in the context of a modeling method for Business Continuity Management (BCM). Figure 3 exemplifies how the application yielded a revised notation for the BCM modeling language (for full details, we refer to [6]).

3.1.3 Reflection and learning

The application of the initial technique led to a revision of the BCM modeling language with respect to its semantic transparency. By involving the participants in co-creating and evaluating the notation, it was very efficient to identify weaknesses and derive ideas for improvements. The evaluation confirmed a good semantic transparency of the initial BCM notation, thus leading to marginal revisions. What was missing at that point was an empirical evaluation of the impact of the notation revisions on the semantic transparency.

3.1.4 Formalization of learning

Although we were able to show the applicability of the technique, only 15 participants took part in the initial evaluation and only the initial notation was considered. This needed to be tackled in a next application. In order to mitigate this shortcoming, the approach has been extended by a further phase, which involves the revision of the notation based on the gained feedback and the use of experts to compare the semantic transparency of the initial against the revised notation. This enables to empirically test the hypothesis that the notation has actually improved with respect to semantic transparency. Section 4 comprehensively introduces the extended technique.

This section further formalizes the requirements for this technique that have been derived while considering three sources: (i) the strengths of existing specific notation design approaches (cf., Sect. 7), (ii) the agile nature of method engineering in general [24,52] and domain-specific modeling methods in particular [30,64], and (iii) the experience of the authors in supporting the conceptualization of more than ten modeling methods [4]. Table 2 provides an overview of the high-level requirements with a corresponding description. The technique aims for easy customization (i.e., RQ1), a modular structure (i.e., RQ7), and an efficient use by method engineers (i.e., RQ3) without requiring any technical infrastructure (i.e., RQ6). Moreover, it combines the best parts of existing approaches (see Sect. 7) by incorporating participant's suggestions on notations (i.e., RQ4) as well as by testing the semantic transparency of an initial notation (i.e., RQ5). Consequently, the overall technique is limited to a context where an initial notation is given, whereas isolated tasks like the term association (see Sect. 4.1.2) can also be performed independently to derive an initial notation (i.e., RQ2).

4 The extended notation evaluation and improvement technique

The technique presented in this paragraph is generally applicable to any modeling language with an initial notation and follows an iterative approach. Each iteration consists of two phases, each of which comprises a set of tasks. While the first phase *evaluates* the initial notation, the second phase aims at *improving* the initial notation by considering the evaluation results. Eventually, phase two is concluded by an empirical evaluation that tests whether the revised notation actually improves the initial one with respect to semantic transparency. The method is modeling language agnostic, meaning that its constituting parts map to the abstract components of modeling languages like abstract and concrete syntax. In an application of the method, the concepts of the

specific modeling language under investigation then need to be used.

Naturally, a modeling notation is never perfect as, e.g., the domain under investigation evolves, new visualization techniques emerge, new purposes are supported by the method, etc. Consequently, the approach enables—but does not entail—continuous improvement of a modeling language notation. Based on the evaluation results of one iteration, the conductor decides whether a new iteration is meaningful. One particular reason for another iteration could be that the results of the initial notation assessment are equally bad as the results of the comparative evaluation for some language elements. Thus, the participants in that iteration did not yield suggestions for significant improvements. In this case, conducting another iteration with other participants might do so.

Figure 4 visualizes the notation evaluation and improvement technique. Each task description is decomposed into tasks to be performed by a participant and by a conductor. The following subsections moreover detail the *experimental design* and the approaches for the *analysis* of the experiment's results for each task. After detailing the individual tasks of phase 1 (Sect. 4.1) and phase 2 (Sect. 4.2), Sect. 4.3 eventually presents a summary of the technique and describes the prerequisites and required skills of a conductor.

4.1 Phase I: evaluation of the initial notation

After the initiation, four core tasks are proposed to evaluate the initial notation (see Fig. 4): *term association*, *notation association*, *case study*, and *feedback*. Note that these tasks need to be executed sequentially by participants in order to avoid interference between the tasks. Thus, when using a technical infrastructure, switching between the tasks should be disabled. Similarly, when using pen and paper, the individual tasks shall be printed on separate pages and distributed one by one. During the preparations, modeling language concepts need to be assigned to either the term association or the notation association task to prevent participants from identifying a concept during the notation association only based on their experience during the term association. To ensure that all concepts are eventually considered in all tasks, two groups can be created with interchanged modeling language concepts between these two tasks.

4.1.1 Initiation

Participants are briefly introduced to the domain and the building blocks of the modeling method to be evaluated. This primarily concerns the definition of the relevant domain and an introduction to the individual model types/viewpoints of the modeling method (if applicable). This introduction

Table 2 Requirements for the notation evaluation and improvement technique

ID	Name	Description
RQ1	Efficient customization	The technique shall enable efficient customization to be applicable with the modeling language at hand
RQ2	Notation improvement	The technique shall be used with an initial notation, aiming to evaluate it, derive improvement recommendations, and empirically test the impact of these recommendations
RQ3	Efficient use	The technique shall not require an extensive amount of time from the method engineer
RQ4	Involve participant suggestions	The technique shall enable participants to be involved in the improvement of the notation by enabling them to propose what they believe is the most semantically transparent notation for a given modeling language concept
RQ5	Semantic transparency	The technique shall focus on the semantic transparency of a modeling notation
RQ6	Technical independence	It shall be possible to apply the technique without any technical infrastructure
RQ7	Modular structure	If possible, the individual tasks shall be designed in a modular structure that allows to flexibly adjust the technique to the current situation

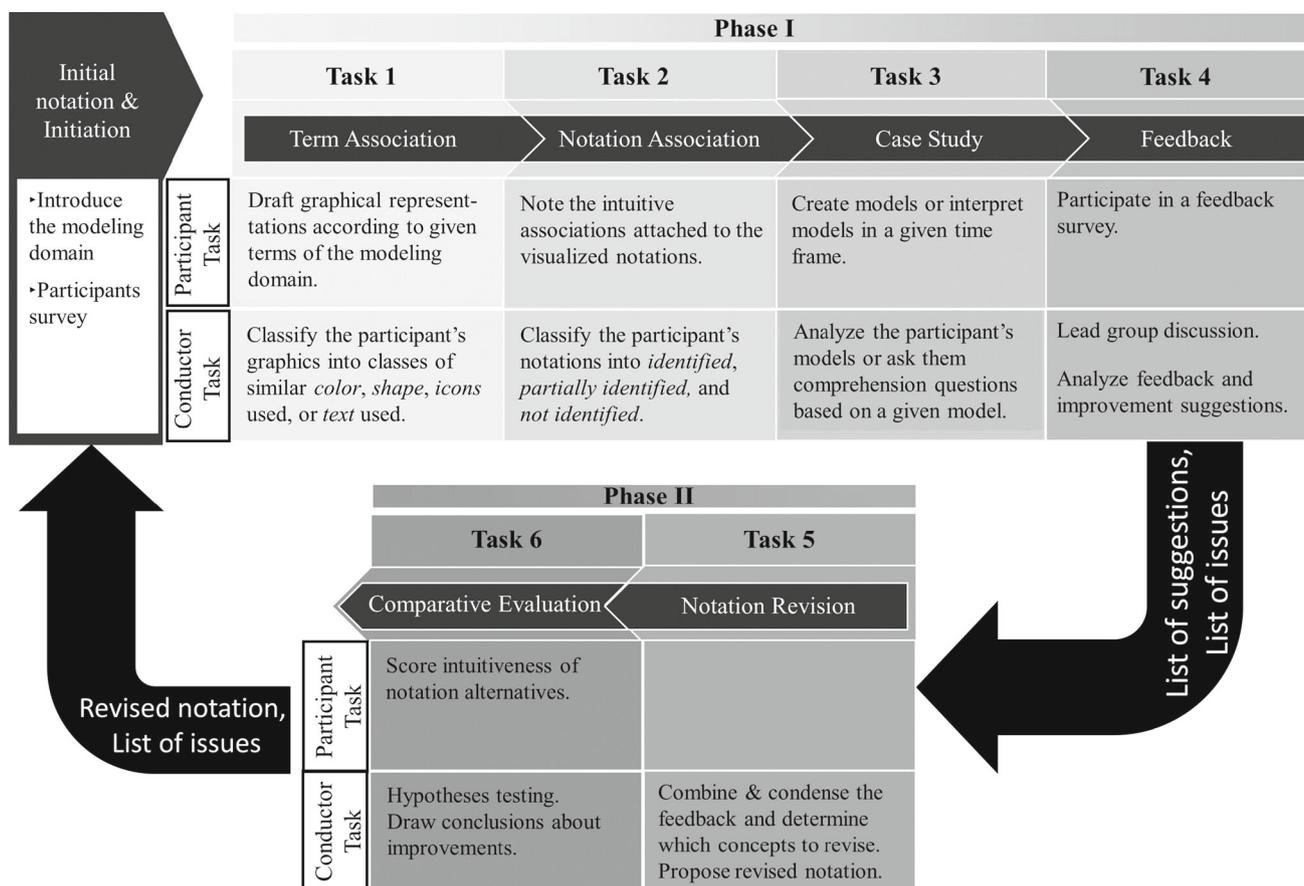


Fig. 4 Procedure of applying the notation evaluation and improvement technique

needs to be textually or orally, i.e., without showing any visual aspects like language concepts or sample models. As a last step, background information about the participants needs to be collected. This information comprises classical demographic aspects as well as questions regarding the participant's experience with modeling languages, modeling tools, and the relevant domain.

4.1.2 Term association task

Experimental design Participants are shown terms of modeling language concepts (i.e., the name of the metamodel concept). Each participant then sketches up to three graphical representations for each term, which she/he deems most intuitive to represent the term's semantics. For the execution

of this task, participants are provided with a paper that lists the terms and colored pencils for the sketches.

Analysis The conductor first classifies the returned notation drafts into groups of similar graphics with respect to the used visual variables: *horizontal/vertical position, shape, size, color, brightness, orientation, and texture* [44]. Afterward, the frequency (i.e., the number of occurrences) of a specific variable instance like ‘red color’ or ‘circular shape’ can be computed. Comparing the analysis with the initial notation might point to inadequacies. Moreover, the most frequently drafted notations provide pointers to potential improvements and serve as an indicator whether or not a dominant notation can be crystallized.

A potential automation of the conductor’s task can be achieved by digitizing the participant’s sketches (if they have been prepared with pen and paper) and by applying graphical classification algorithms as, e.g., provided within the OpenCV¹ library. While this automated analysis increases the efficiency, it also requires knowledge in algorithmic image processing.

4.1.3 Notation association task

Experimental design Participants are shown the initial notation of individual modeling language concepts and are asked to record up to three associations that pop out when looking at them. It is important to note that participants are only presented the notation without any hint at, e.g., the name or the semantics of this concept. Providing such information would hamper the evaluation of the semantic transparency.

Analysis The conductor categorizes all responses to measure the percentage of participants that associated the correct semantics to a provided notation. If one of the named terms of a participant matches with the true name or semantics of the concept, the notation is classified *identified*. For instance, if one of the participant associations for the concept ‘Recovery activity’ is ‘Rollback activity,’ the notation is correctly identified because the essence of the concept’s semantics is correctly conceived. In the case that one of the named associations nearly fits the semantics, it is categorized as *partially identified*. In the ‘Recovery activity’ example, the terms ‘Task’ or ‘Recovery measure’ nearly fit the true semantics. If none of the provided associations expresses the true semantics or if no term is provided, the notation is *not identified*.

This analysis enables to calculate for each element the percentage of participants giving a matching association and the relative rank of this association. The relative rank is an indicator of the extent to which an element is outperformed by

other modeling language concepts, i.e., pointing to a problem with the perceptual discriminability between different notations [44].

Automation of the analysis can be achieved by using natural language processing techniques like taxonomic or ontological text analysis. Such approaches automatically classify text according to a given ontological structure and thereby explicitly define relationships between concepts (i.e., the terms responded by the participants). When using such an approach for the analysis, conductors can not only quantify the identified concepts, they can also analyze the distance between the terms provided by the participants and the concepts. Moreover, all responses can be automatically clustered to efficiently identify recurring terms.

4.1.4 Case study task

Experimental design The preceding tasks concentrated on the evaluation of the semantic transparency of individual modeling language concepts. In some cases, the semantics only become clear when considering the modeling concepts in their context. Therefore, the case study task addresses this aspect by testing whether or not participants are able to combine the language concepts to *model a presented case*. The case study should be textually introduced. If possible, participants shall be supported by a modeling tool for the modeling task. Alternatively, the case study could also include a *sample model with comprehension questions*. Questions should be designed such that they can be answered either by comprehending individual modeling language concepts, by comprehending a modeling language concept within its context, or by comprehending parts of a model comprising multiple concepts and relationships. This alternative is more suitable if a huge number of participants simultaneously takes part in the evaluation.

Analysis The conductor needs to categorize the provided models according to their semantic and syntactic correctness. Three types of errors might occur. A *semantic error* considers mistakes such as a wrong application of a concept, a wrong denotation of a concept or a wrong definition of a property of a concept. A *syntactic error* covers a wrong sequence of concepts, a wrong application of a relation or a missing relation. The error category *incomplete model* covers missing properties of a concept, missing concepts, or missing denotations of a concept.

In case comprehension questions are asked, the responses of the participants need to be categorized as *fully correct*, *partially correct*, or *wrong*. This enables to calculate the percentage of respondents who provided a fully or partially correct answer for each modeling language concept. In order to enable a fully automated analysis, e.g., for very large groups of participants, comprehension questions can also

¹ OpenCV framework [online], <https://opencv.org>, last visited: 22.02.2021.

be realized as multiple choice questions where participants select the answer they think is correct.

4.1.5 Feedback task

Experimental design The conductor discusses the evaluation tasks and shows the solution of the case study before the participants are asked to fill out a feedback survey. The survey shall cover questions about the perceived quality of the modeling language, the perceived usefulness of the modeling tool (if applicable), and general feedback and improvement suggestions.

Analysis During the analysis, it is important to identify feedback and improvement suggestions that are related to individual modeling language concepts as they provide input to identify the elements for which a notation revision is needed and how this revision could look like (see Sect. 4.2 for more details).

4.2 Phase II: notation revision and comparative evaluation

4.2.1 Notation revision task

The first phase of the technique focuses on an evaluation of the semantic transparency of the initial notation. In case of a low score of an element for the notation association and/or case study tasks combined with recurring negative feedback, the conductor is advised to revise the notation of this concept. For this revision, the conductor can make use of the visual deconstruction of the participants' graphical representations in the term association task and/or concrete improvement suggestions that are given in the feedback task. In this respect, it is important to not solely base the revision on the obtained results, but also to ensure homogeneity across the modeling language concepts.

4.2.2 Comparative evaluation task

Variables and measures The proposed language revisions need to be empirically evaluated and compared against the initial notation. These two notations will be used as the independent variable in this task. The semantic transparency of the DSML notation, defined as *the extent to which the meaning of a symbol can be inferred from its appearance* [44, p.765], is used as dependent variable. To measure this variable, we introduce the semantic transparency score (STS) and make use of a comparative scale that has been proposed by Figl and Recker [14]. Participants are presented the initial and a proposed revised notation for each modeling language concept and asked to define their preference. STS ranges from 0 to 100 with interval values of 5. 0 indicates that alternative A

is most semantically transparent, whereas 100 indicates that alternative B is most semantically transparent. The value 50 indicates no preference with respect to the semantic transparency of the given notations.

Besides this, it is important to collect demographic variables, including gender, birth year, education level, working sector, modeling knowledge, and modeling expertise. For the latter, respondents shall indicate (i) their expertise on a 5-point Likert scale as well as the number of conceptual models they have (ii) *read* and (iii) *made* during the last five years. These demographics serve a twofold purpose as they will be used to assess whether the participants have a suitable profile and to check whether these variables do not have confounding effects.

Hypotheses As the revision of the notation is based on the results of phase I (see Sect. 4.1), it is expected that the revised notation will outperform the initial notation with respect to its semantic transparency. As the STS is measured on a comparative scale ranging between 0 and 100 and the original and new notation are alternately divided between options A and B (see Experimental Design), the scores need to be recoded such that a score between 0 and 45 means that the original notation is considered as more semantically transparent, while a score between 55 and 100 shows a higher semantic transparency of the revised notation. As such, the following null and alternative hypothesis can be formulated for each revised concept.

$H_{0,concept}$: The mean/median STS of the element's notations is equal to 50.

$H_{1,concept}$: The mean/median STS of the element's notations is higher than 50.

Experimental design By using a comparative preference scale, the respondents are confronted with the two alternative notations for a selection of the DSML elements (see Table 7). As such, the task can be considered as having a within-subjects design. The measurement scale for the STS is implemented as a graphical rating scale using a slider. In comparison with traditional Likert-type scales, the chosen scale helps to better communicate to the participants that semantic transparency can be expressed on an interval continuum [10]. During the experimental tasks, the old and new notation need to be alternately divided between options A and B, to prevent that respondents would be able to guess which notation alternatives belong together. To obtain an insight in the rationale for the score given by the participants, they can provide a textual clarification. This clarification is important in case of an indifference between the concepts to analyze whether the semantic transparency of the notations is considered equally good or bad. As a conductor, one needs to provide a survey—either printed or online—to collect the respondents' scores.

Required number of participants To determine the intended sample size (SS), the expected effect size (ES) needs to be calculated first. Given the novelty of the experimental design, the conductor should rely on estimates of a pilot study with a few respondents to calculate the mean (i.e., μ_1) and standard deviation (i.e., σ) of the dependent variable. Given formula 1, the effect size can be calculated. Furthermore, the general convention is that the power of a statistical test should be at least 0.80 [9,20]. Given this power and a standard level of significance (i.e., 0.05), formula 2 enables to calculate the minimal sample size that is needed for a one-sided test.

$$ES = \frac{|\mu_1 - \mu_0|}{\sigma} \quad (1)$$

$$SS = \left(\frac{Z_{0.975} + Z_{0.80}}{ES} \right)^2 \quad (2)$$

The conductor could make use of different sampling techniques to attract the required number of participants, including both probability (i.e., random) and non-probability (e.g., quota, purposive, snowball, volunteer) sampling. The specific choice should be made based on the characteristics of the target population and the availability of prospective users [59].

Analysis After the collection of the evaluation data, an analysis is needed to identify the elements for which an improvement in semantic transparency can be observed. To this end, an analysis of the descriptive statistics for the demographic variables is needed first. This enables to get insights in the respondents' profile and exclude responses of participants that do not have the suitable profile of the intended end-users (e.g., based on modeling skills, working experience, etc.). Afterward, it needs to be determined which statistical analyses are suited to test the proposed hypotheses. In particular, the normality of the independent variables must be analyzed to determine whether parametric or nonparametric tests must be used for the analysis of the hypotheses and the post-tests. This can be implemented by determining the skewness and kurtosis of the dependent variables, together with the results of a suitable normality test (e.g., the Shapiro–Wilk or Kolmogorov–Smirnov test).

The Z_{skewness} and Z_{kurtosis} should be between $[-1.96, 1.96]$ to safely assume normally distributed data. Furthermore, a visual check of the histograms and Q-Q plots should supplement the normality analysis. To test the hypotheses, the median value of one sample is compared against a hypothesized value (i.e., 50). In case of a normally distributed STS score, the parametric one-sample t -test should be used. Otherwise, the nonparametric one-sample Wilcoxon signed rank test is the suitable alternative. These tests enable to determine for which elements the

revised notation provides a statistically significant improvement in semantic transparency. If this improvement is not present for a certain element, it is advised to analyze the textual clarifications given by the respondents. If it becomes apparent that both notations are considered as being inappropriate, a new iteration of the proposed technique should be started. Finally, post-tests should be performed to check whether demographic and time variables have confounding effects on the dependent variables. The suitable statistical test is based on the normality of the independent variables and the measurement scale of the demographic and time variables. For the nominal/ordinal demographic variables with two categories, the parametric independent-samples t -test or the nonparametric independent-samples Mann–Whitney U test is suited. In case of multiple categories, the parametric ANOVA analysis or the nonparametric independent-samples Kruskal–Wallis test should be performed. Finally, the parametric Pearson correlation test or the nonparametric Spearman's correlation analysis is useful to test whether there is a significant correlation between the interval demographic and time variables on the one hand and the STS scores on the other hand.

4.3 Summary

Following the requirements of a modular structure (RQ7) and easy customization (RQ1) (see Table 2), the notation evaluation and improvement technique provides a clear structure along with the two phases that are further decomposed into individual tasks while remaining flexible in the concrete methods being applied. Consequently, the technique allows to use many methods within the overall structure it provides (see Table 3 for an overview and a description of prerequisites and mandatory skills). Table 3 shall support method engineers in customizing the technique to the modeling method at hand and the context in which they execute it. The selection of methods should consider the impact on validity and the reliability of the results. The table also describes methods that can be applied manually in cases with only few participants and automated methods that make the application of the technique also feasible in cases with many participants.

5 Evaluating the semantic transparency of the initial PGA modeling language notation

In the following, a comprehensive application of the extended notation evaluation and improvement technique is reported. The technique has been applied to evaluate the initial PGA notation (phase 1 of the technique, covered in this section),

Table 3 Applicable methods for the notation evaluation and improvement tasks

Task	Technique	Description	Prerequisites & Skills
Term Association	Manual Classification	Classify the participant's sketches according to the used visual variables proposed in [44]	Few participants. Awareness of the visual variables
	Automated Classification	Apply image processing and classification algorithms on the participant's sketches	Expertise in using existing libraries. Preparation and pre-processing (e.g., size, format) of sketches
Notation Association	Manual Classification	Classify the participant's responses into <i>identified</i> , <i>partly identified</i> , and <i>not identified</i>	Few participants. Domain knowledge (of the DSML)
	Automated Classification	Apply natural language process techniques to automatically classify the responses into an ontological structure	Expertise in using existing libraries. Preparation (e.g., format) of responses
Case Study	Model creation	Participants create a model to solve a textual case study	Few participants. Adequate time for experiment. Optional tool support
	Model comprehension	Participants are presented sample models and asked comprehension questions	Domain knowledge (of the DSML)
Feedback	Survey	Participants fill out a survey comprising open feedback and, if applicable, usability questions of the used tool [33]	None
Notation Revision	Manual revision	Revision of the notation based on the analysis results of all tasks and the creativity of the method engineer	Creativity
Comparative Evaluation	Comparative Preference	Comparatively evaluate the initial with the revised notation	Knowledge in statistics and statistical software like SPSS

and in the second phase to develop and evaluate an improved PGA language notation (covered in Sect. 6).

5.1 Introduction to the PGA modeling language

PGA has been introduced in [56] and further developed in [57] as a project within the Open Models Laboratory [4]. A PGA modeling tool has been realized with ADOxx [1]. To achieve strategic fit in the business architecture, PGA aims at the development of a business architecture heat map following a modeling procedure that consists of three activities: (i) developing a prioritized business architecture hierarchy, (ii) executing the performance measurement, and (iii) performing the strategic fit improvement analysis.

In the first step, the value creation throughout a hierarchical business architecture is modeled. Based on existing Strategic Management frameworks, the PGA meta-model incorporates the following elements (i.e., capitalized in the remainder of the text): Activity, Process, Competence, Value Proposition, Financial Structure, Internal Goal, Customer Goal, and Financial Goal. Icons were designed to provide

a notation suited for business-oriented users. Afterward, valueStream relations are added between these elements to show the hierarchical value structure. Each valueStream is prioritized by using the AHP mechanism [58] and a color coding with accompanying line texture is used to differentiate between a high (i.e., solid red color), medium (i.e., dashed orange color), or low priority (i.e., dotted green color) with respect to their strategic Importance.

The performance measurement mechanism is applied to each business architecture element to identify an appropriate performance indicator, set a performance target and an allowed deviation percentage, and to analyze the actual outcome for each indicator. This enables the differentiation between an excellent (i.e., dotted green color), expected (i.e., dashed orange color), or bad Performance (i.e., solid red color).

The resulting business architecture heat map (see Fig. 5) can be further analyzed during the strategic fit improvement analysis to identify operational changes that potentially improve the value creation throughout the business architecture. To support this, a critical path can be identified starting

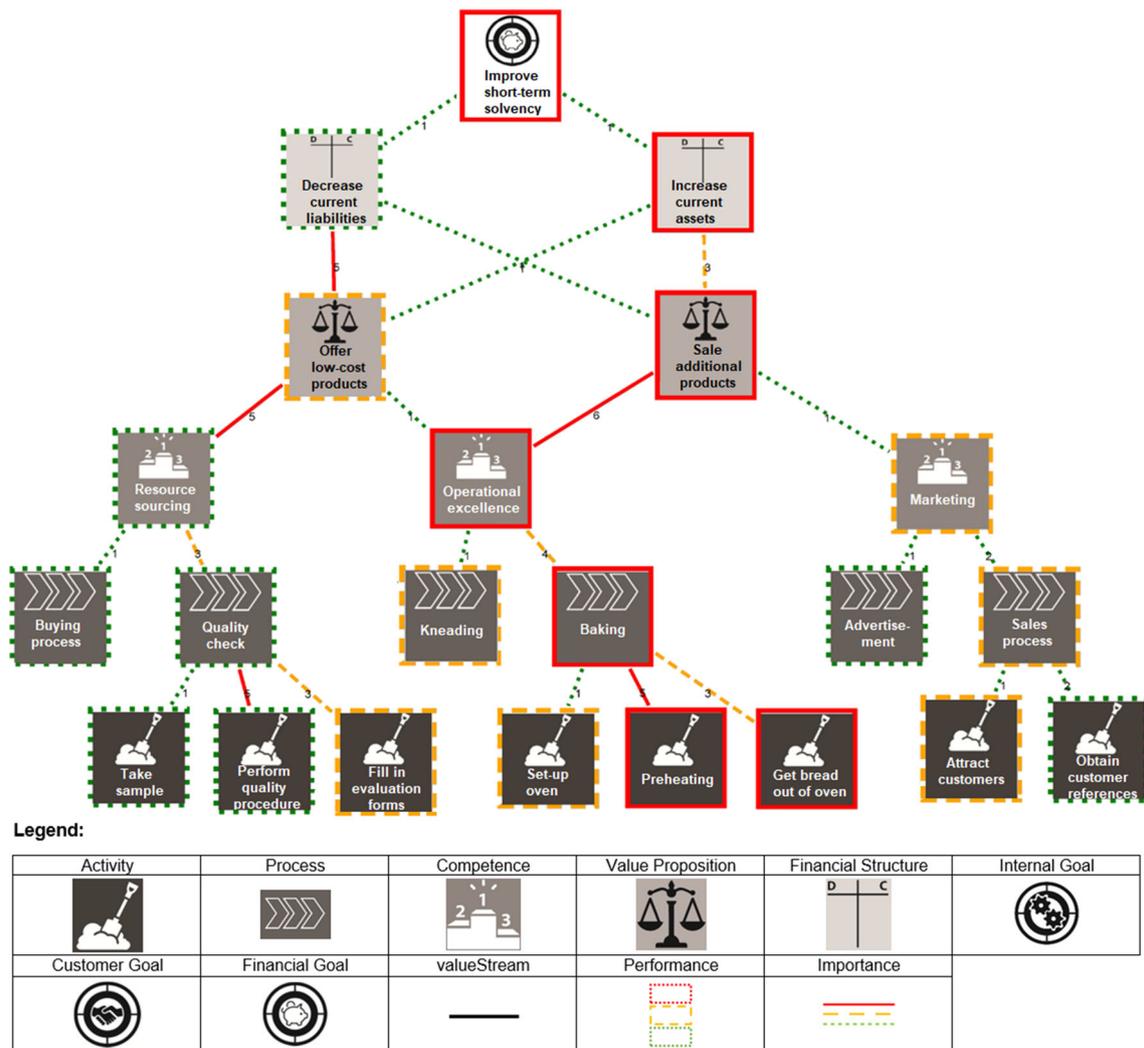


Fig. 5 Example of a business architecture heat map in the initial PGA notation [56]

from a Goal with a bad Performance (e.g., improve short-term solvency in Fig. 5) as a chain of valueStream relations that have a high or medium Importance and that connect business architecture elements on downstream hierarchical levels of which the Performance can be improved (e.g., Increase current assets, Sale additional products) [57]. A critical path ends at the Activity or Process level (e.g., Preheating), pointing to concrete changes that can be applied to the organizational processes.

5.2 Evaluation results for the initial PGA notation

A total of 139 students following a Master level class on IT Management of Ghent University participated in the user study. The participants were randomly assigned to two groups: 70 participants for group A and 69 for group B. Their average age was 22 years, and 41% of them were female. Although the participants were not familiar with the PGA

method, 86% had prior modeling knowledge in Requirements Engineering, 90% in Business Process Management, and 34% in Enterprise Architecture.

All responses to the questionnaire² were digitized and stored in a shared cloud infrastructure. All authors performed a pretest to analyze the results. Afterward, the gained experience was exchanged to streamline the structure of the analysis, e.g., the visual variables to be applied during the classification of the term associations. Next, the authors independently analyzed all responses, after which the analysis results were condensed and harmonized.

² An overview of the experimental tasks is available via <https://doi.org/10.13140/RG.2.2.27473.48489>.

Table 4 Results of the term association task

PGA concept	Color	Shape	Icons	Text
Activity	1. Blue (71%)	1. (Rounded) rectangle (75%)	1. Person (18%)	1. Activity (46%) 2. A (23%)
Process	1. Blue (75%) 2. Rectangle (28%)	1. Arrow (28%)	–	1. Act. nr. (71%)
Competence	1. Blue (76%)	1. Rectangle (31%) 2. Triangle (23%)	1. Thinking balloon (21%) 2. Person (14%) 3. Light bulb (14%) 4. Brain (10%)	1. C (57%)
Value Proposition	1. Blue (78%)	1. Ellipse (27%) 2. Rectangle (24%)	1. Dollar/Euro (29%) 2. + sign (10%) 3. Light bulb (10%) 4. People (6%)	1. V (27%) 2. VP (27%)
Financial Structure	1. Blue (63%)	1. Ellipse (35%) 2. Rectangle (33%)	1. Dollar/Euro (80%)	1. Cost & revenues (40%) 2. C & R (20%)
Internal Goal	1. Blue (67%)	1. Ellipse (54%)	1. Bull's-eye/arrow (64%)	1. I (29%) 2. x (21%)
Customer Goal	1. Blue (65%)	1. Ellipse (34%) 2. Cloud (16%)	1. Bull's-eye/arrow (48%) 2. Person (33%)	1. C (44%) 2. Customer (goal) (22%)
Financial Goal	1. Blue (65%)	1. Ellipse (30%) 2. Rectangle (30%)	1. Dollar/Euro (67%)	–
valueStream	1. Blue (73%)	1. Arrow (69%)	1. Dollar/Euro (47%) 2. Stream (18%)	1. V (67%)
Performance	1. Blue (72%)	1. Rectangle (32%) 2. Ellipse (21%)	1. Graph (18%) 2. V checkbox (18%) 3. Muscle (13%) 4. Trophy (10%)	–
Importance	1. Blue (66%)	1. Rectangle (26%) 2. Triangle (26%)	1. Exclamation mark (75%)	–

5.2.1 Results of the term association task

Table 4 shows the deconstruction of the participants' graphical representations according to different visual variables. The relevant variables that were used by participants include *color*, *shape*, *icons*, and *text*. Icons are symbols that perceptually resemble the concepts they represent [44], while shapes refer to geometric figures (e.g., square, line). Due to limited space, Table 4 only covers those visual variables with a cumulative frequency of at least 50 % if the individual absolute frequency is at least two.

The analysis shows that the participants dominantly used a blue color to design a graphical representation for a given PGA concept, although they were instructed to bring and use a variety of colors. This unexpected result is a consequence of the flexibility to set up the evaluation technique for simultaneous evaluation by a large number of participants in a classroom using pen and paper. Concerning shape,

some recurring proposals can be seen such as a rectangle, triangle, ellipse, or arrow. The origin of these proposals can be explained by the modeling experience of the participants, which are familiar with Requirements Engineering and Business Process modeling. Participants added corresponding icons to the basic shapes to further encode the semantics of the concepts (e.g., thinking balloon, bull's-eye). These icons seem to provide an important instrument to design a semantically transparent notation.

In line with the principle of Dual Coding [44], participants used text to complement the proposed graphics, which enables to depict the meaning of an element both visually and verbally. In most cases, the text is equal to the first letter of the PGA concept (e.g., 'C' for Competence) or its complete name (e.g., 'Activity'). In other cases, text refers to the content of a PGA concept. This applies to Financial Structure, which is a representation of the cost and revenue structure

Table 5 Results of the notation association task

PGA concept	Correct associations	Relative rank
Activity	24.29%	1
Process	36.23%	1
Competence	2.90%	8
Value Proposition	2.83%	9
Financial Structure	12.75%	3
Internal Goal	4.90%	5
Customer Goal	0%	–
Financial Goal	20.29%	2
Performance	0%	–
Importance	0%	–

(e.g., as coded by C & R) that is implemented by an organization.

Finally, *number* and *graphical position* were used for specific PGA concepts. The participants employed spatial enclosure to represent activities as a subset of the overarching process. In this case, we see a dominant proposal of three rectangles connected by arrows. Graphical position is also proposed for the representation of a valueStream as participants use a hierarchy of arrows to depict the value creation throughout the business architecture.

5.2.2 Results of the notation association task

Table 5 shows the results of the notation association task. For each element, the percentage of participants giving a matching association and the relative rank of this association is listed. Important to note here is that the visualization of the valueStream relation (i.e., a non-directed line) was not explicitly tested as the meaning of this relation only becomes clear when included in a hierarchical business architecture heat map.

The percentage of correct associations ranges between 0% and 36.23%. The concepts Activity (24.29%—rank 1), Process (36.23%—rank 1), Financial Structure (12.75%—rank 3), and Financial Goal (20.29%—rank 2) performed the best considering both the percentage and the relative rank of the correct associations. The notation of the other PGA elements is less semantically transparent, as the percentages are below 5%. Moreover, some of them are outperformed by other PGA concepts. More specifically, the Competence notation (i.e., a stage icon) is confused with Performance by 52.17% of the participants and the icon of Internal Goal (i.e., a cogwheel) is associated with a Process by 35.29% of the participants. Such results clearly point to a weak perceptual discriminability between those notations in the initial PGA notation.

5.2.3 Results of the case study task

In Table 6, the results of the case study are given. To keep this example manageable for participants in the given time, only one type of Goal (i.e., Financial Goal) was included in the sample model. This is the reason why no results are available for Customer and Internal Goal in Table 6. Actually, the PGA model depicted by Fig. 5 was used.

Although all questions were oriented toward the identification of PGA concepts, partially correct answers could also be identified. These include naming elements at the instance level (e.g., Take sample instead of Activity) or using close synonyms for the PGA concept (i.e., Task as a synonym for Activity). Besides, there was not a question that directly targeted the identification of a valueStream, but problems with this relation can be derived from incorrect answers to the questions about the Activity and Value Proposition concept. More specifically, some incorrect answers indicate that the valueStream relation was interpreted in the wrong direction.

Although the mean score of complete correct answers for this task is 41.32%, Table 6 shows that the meaning of the Value Proposition (i.e., 5.04% correct answers) and Importance (i.e., 5.76% correct answers) notation cannot easily be derived from the business architecture heat map. Even if partially correct answers are included, these elements are the two least performing of all PGA concepts with total scores of 21.59% for Value Proposition and 14.39% for Importance. Besides, there seems to be a problem with the semantic transparency of the valueStream notation, which was read in the wrong direction in the Activity and Value Proposition question by, respectively, 18.71% and 27.34% of the participants. As one can notice, the scores for Financial Structure and Financial Goal are the same, as the identification of these PGA concepts was covered by one question during the case study task.

5.2.4 Results of the feedback task

During the feedback task, we obtained 104 remarks from 58 unique participants (i.e., a response rate of 41.73%). Of these responses, 45 could be specifically traced back to the PGA concepts, distributed among the aspects color and line style (24 remarks), Importance (12 remarks), valueStream (5 remarks), and Activity (4 remarks). Illustrative feedback includes the following aspects:

- Color & line style: “Using colors is a good idea, it gives a nice and quick overview.” “The meaning of the different colors & line styles is not clear.”
- Importance: “It is not clear what the numbers next to the relations mean.”
- valueStream: “It is difficult to see where certain value streams go to.”

Table 6 Results of the case study task

PGA concept	Correct answers	Partially correct answers
Activity	23.02%	Synonym (task): 9.35% Instance level elements: 15.83%
Process	74.10%	–
Competence	42.45%	–
Value Proposition	5.04%	Instance level elements: 16.55%
Financial Structure	57.55%	–
Financial Goal	57.55%	–
valueStream	–	Incorrect question Activity: 18.71% Incorrect question Value Proposition: 27.34%
Performance	81.29%	–
Importance	5.76%	Partial answer: 8.63%

- Activity: “The model would improve if the total process of how the organization operates was represented.”

6 Development and comparative evaluation of the improved PGA notation

The application is continued by reporting on the second phase of the technique in the context of the PGA notation. In Sect. 6.1, we first propose revisions to the PGA notation. Section 6.2 then reports on the empirical comparative evaluation of the initial and revised PGA notation, thereby concluding whether the revisions can be considered as actual improvements.

6.1 Notation revision of the PGA notation

The main argument to propose a new notation for **Competence** is the confusion the initial one caused for end-users. Indeed, during the notation association task, it became clear that people naturally attach the meaning of Performance to the visualization. Based on the suggestions of the participants during the term association task, we propose a combination of a person and light bulb icon as the new notation. This notation should refer to the cognitive abilities that are associated with the definition of a Competence as the internal knowledge, skills, and abilities of an organization.

A new notation for **Value Proposition** is also proposed as the initial notation was one of the least performing PGA elements during the notation association (i.e., 2.83%) and case study (i.e., 5.04%) tasks. However, the suggested icons by participants do not show a clear preference as they are closely related to financial elements (i.e., dollar/euro or + sign) or cognitive abilities (i.e., light bulb). Therefore, the new notation is a gift that is exchanged between two people. We believe this provides a more semantically transparent notation for the products and services that are exchanged

between a company and its customers. This proposal is in line with the notation of a Value Proposition in the Business Model Canvas (i.e., a gift icon) [49].

The notation of **Internal Goal** needed improvement as respondents confused it with processes during the notation association. As the PGA notation already includes a bull’s-eye to represent the goal aspect, the analysis of the term association task did not provide further concrete suggestions. To stress the internal characteristic of the term, it was decided to graphically enclose cog wheel icons (i.e., the initial notation) inside a factory icon (see Table 7).

A similar argument can be provided for the new notation of a **Customer Goal**. This element scored low (i.e., 0%) in the notation association task, which clearly shows that the semantic transparency of the initial notation needs improvement. Based on the suggestions of the term association, it was decided to clearly represent the customer by an iconic person who is shaking hands with another person, holding a briefcase (see Table 7).

Problems with the semantic transparency of the **valueStream** relation became apparent during the case study, in which many participants applied it in the wrong direction. Furthermore, some of the qualitative feedback confirms that the direction of the valueStream is not semantically transparent. In line with the term association task, this issue is solved by using an arrow in the newly proposed PGA notation. This arrow points toward the element in the business architecture heat map, of which the value creation is supported by.

A last change that is proposed concerns the **Importance** element. In the initial notation, this attribute was visualized by a colored valueStream accompanied by a certain texture. Furthermore, a number indicated its relative Importance as a form of dual coding. However, the results showed that this notation was poorly understood by participants. This was also confirmed by the qualitative feedback about the confusing color coding and numbers of the valueStream relations. A first improvement could be identified based on the term asso-

Table 7 Suggested changes to the initial PGA notation

PGA concept	Initial notation	Consolidated Suggestion by Participants	Revised notation
Competence		Blue rectangle/triangle with a thinking cloud, person, or light bulb icon	
Value Proposition		Blue ellipse/rectangle with a euro/dollar icon	
Internal Goal		Blue ellipse with a bull's-eye/arrow icon and text I	
Customer Goal		Blue ellipse with a bull's-eye/arrow and person icon and text C	
valueStream + Importance		Blue arrow in a hierarchical structure with an exclamation mark icon	

ciations, of which the results show that an exclamation mark is a semantically transparent way of representing Importance. We combine this suggestion by replacing the color coding by a different thickness of the valueStream relation. As a result, a valueStream with a high Importance will be depicted by a thick arrow, combined with three exclamation marks. The thickness and number of exclamation marks decrease for a valueStream with a medium or low Importance.

The proposed revisions are implemented in a new version of the PGA modeling tool. The new tool is openly available through the PGA project space³ within the Open Models Laboratory (OMiLAB) [4]. A sample model with the new PGA notation can be found in Fig. 6.

6.2 Comparative evaluation of the PGA notations

The comparative evaluation is set up according the design of Sect. 4.2.2 to ensure reproducibility of the evaluation and to limit possible validity threats. While Sect. 6.2.1 presents some customized details of the empirical design, Sect. 6.2.2 reports on the evaluation results.

6.2.1 Evaluation design

The proposed PGA language improvements are empirically evaluated against the original notation. These two notations will be used as the independent variables. As dependent variable, the semantic transparency of the PGA modeling notation is used [44]. The STS is measured on the comparative scale as proposed by Figl and Recker [14]. To

avoid unnecessary repetition, only the alternative **hypotheses** are listed below for the revised PGA concepts (i.e., Competence, Value Proposition, Internal Goal, Customer Goal, valueStream, and Importance). As Importance is an attribute of a valueStream, which are visualized together in the PGA modeling language (e.g., see Fig. 5), these concepts will be tested together during the evaluation.

- $H_{1,Competence}$: The mean/median STS of the Competence notations is higher than 50.
- $H_{1,ValueProposition}$: The mean/median STS of the Value Proposition notations is higher than 50.
- $H_{1,InternalGoal}$: The mean/median STS of the Internal Goal notations is higher than 50.
- $H_{1,CustomerGoal}$: The mean/median STS of the Customer Goal notations is higher than 50.
- $H_{1,valueStream+Importance}$: The mean/median STS of the valueStream and Importance notations is higher than 50.

We calculated the intended sample size using formulae 1 and 2. We relied on a pilot study with five respondents to estimate the mean (i.e., $\mu_1 = 15$) and standard deviation (i.e., $\sigma = 35$) over all five dependent variables. Given formula 3, this results in an effect size of 0.429. Using standard levels for the power of the test (i.e., 0.80) and the level of significance (i.e., 0.05), formula 4 results in a minimal sample size of at

³ PGA project space [online], <https://austria.omilab.org/psm/content/PGA/>, last accessed: 21.09.2020.

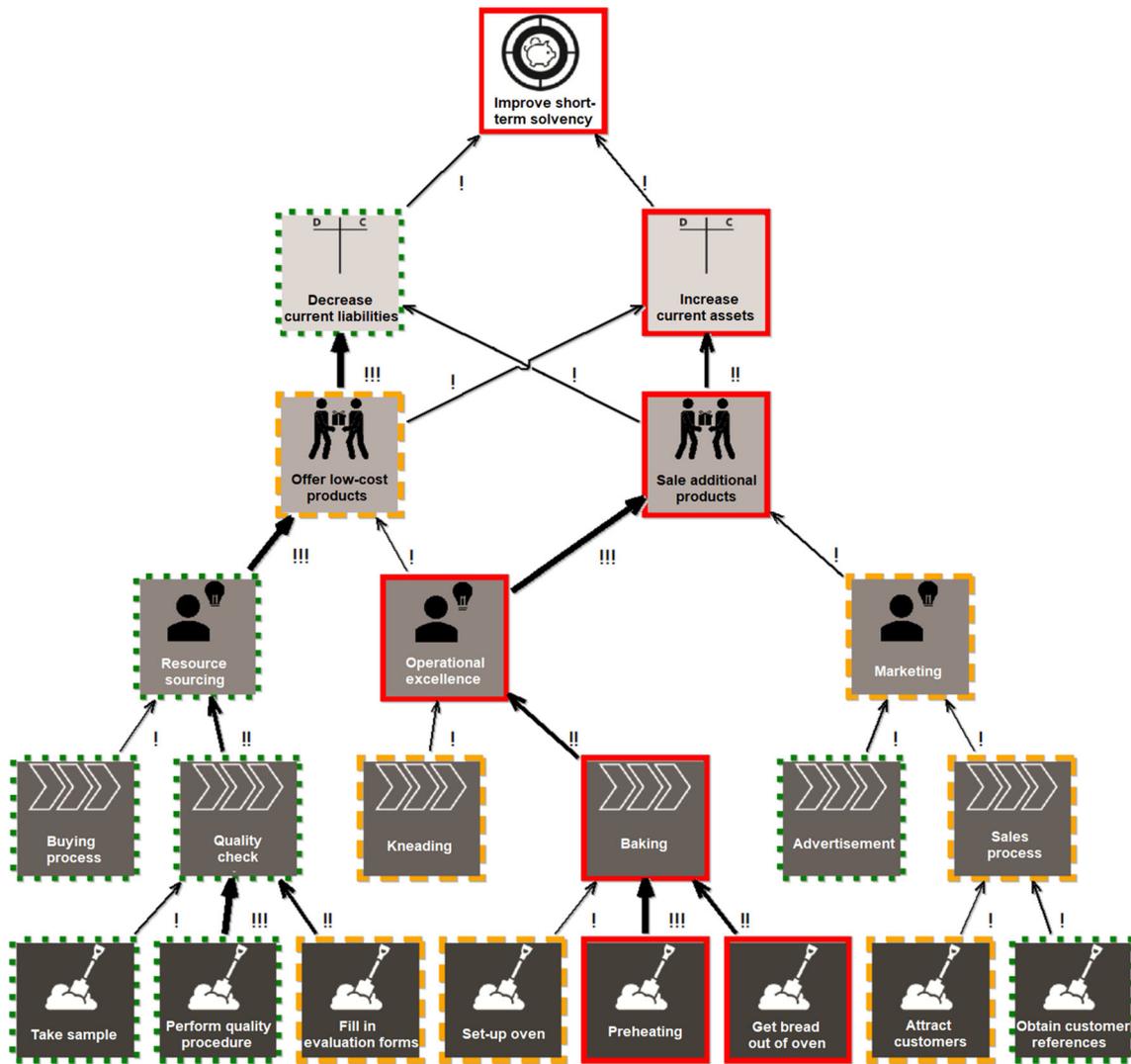


Fig. 6 Example of a business architecture heat map in the revised PGA notation

least 34.

$$ES = \frac{|\mu_1 - \mu_0|}{\sigma} = \frac{|65 - 50|}{35} = 0.429 \tag{3}$$

$$SS = \left(\frac{Z_{0.975} + Z_{0.80}}{ES} \right)^2 = \left(\frac{1.645 + 0.842}{0.429} \right)^2 = 33.608 \tag{4}$$

Besides this quantitative threshold, prospective participants should have working experience (i.e., private, public, or academic sector) and have basic notions about either Requirements Engineering, Business Process Management, or Enterprise Architecture modeling languages. Participants were targeted via different communication means, including direct email invitations to colleagues in the professional network of the authors and the dissemination of the questionnaire via social networking channels (i.e., Twitter, LinkedIn). As

such, the selection of participants can be considered as self-selection (i.e., a specific form of volunteer sampling), in which participants voluntarily participate in the study based on a certain invitation [59]. Therefore, demographic questions are used to check whether the participants have the intended user profile. An overview of all experimental tasks and the implementation of the questionnaire is available online.⁴

6.2.2 Evaluation results

Between July 19 and September 14, 2020, we received 56 full responses to the online questionnaire. Two participants were excluded from the analysis as they had no working experience, which means that they did not fit the intended

⁴ <https://doi.org/10.13140/RG.2.2.11559.37285>.

Table 8 Descriptive statistics of the demographic data

Variable	Mean ¹ /Median ²	Category	Frequency
Gender	–	Female	22.2%
	–	Male	77.8%
Year of Birth	1978 ¹	–	–
Highest degree	–	Doctoral	33.3%
	–	Master	51.9%
	–	Bachelor	11.1%
	–	Other	3.7%
Working sector	–	Academic	46.3%
	–	Private/public	53.7%
Requirements Engineering modeling	79.6% ¹	–	–
Business Process modeling	94.4% ¹	–	–
Enterprise Architecture modeling	59.3% ¹	–	–
Modeling expertise	3: Medium ²	–	–
Models read	[31–40] ²	–	–
Models made	[11–20] ²	–	–

Table 9 Normality tests for the dependent variables

Dependent variable	Skewness	Z _{skewness}	Kurtosis	Z _{kurtosis}	Shapiro–Wilk Statistic	Sig.
<i>STS</i> _{Competence}	−0.338	−1.040	−0.523	−0.818	0.955	0.041
<i>STS</i> _{ValueProposition}	−0.761	−2.342	0.098	1.534	0.912	0.001
<i>STS</i> _{InternalGoal}	−0.234	−0.720	−0.958	−1.499	0.941	0.010
<i>STS</i> _{CustomerGoal}	−0.568	−1.748	−0.339	−0.531	0.946	0.017
<i>STS</i> _{valueStream+Importance}	−0.868	−2.671	−0.255	−0.399	0.879	0.000

Table 10 Hypotheses tests

Hypothesis	Median	Test Statistic	Sig. (one-sided)
H _{0, Competence}	65.0	3.320	0.001
H _{0, ValueProposition}	75.0	5.129	0.000
H _{0, InternalGoal}	52.5	0.860	0.20
H _{0, CustomerGoal}	65.0	2.865	0.002
H _{0, valueStream+Importance}	80.0	4.167	0.000

participant profile. This results in 54 valid responses, which exceeds the minimal sample size by 20. An overview of the demographic data is given in Table 8. Depending on the type of variable (i.e., nominal, ordinal, or interval), corresponding descriptive statistics are provided. In our sample, 22.2% of the respondents are female and 77.8% are male. The average birth year of the participants is 1978. The birth year was also used to randomly assign participants to two groups, for which the order of the PGA elements was switched in the comparison tasks. As a result, 42.6% of the respondents having an even birth year were assigned to group A and 57.4% to group B. With respect to their education, 33.3% of the participants possessed a doctoral degree, 51.9% a Master degree, and 11.1% a Bachelor degree. If the current working sector of the respondents is analyzed, we see a compar-

able distribution between the academic sector (i.e., 46.3%) and the private/public sector (i.e., 53.7%). This illustrates that we were able to attract an evenly distributed mix of both academics and practitioners. With respect to modeling skills, the reported median is equal to a medium expertise. In more detail, 79.6% of the participants had basic modeling knowledge in Requirements Engineering, 94.4% in Business Process Management, and 59.3% in Enterprise Architecture. Eventually, we collected the number of models that were read (i.e., median value between 31 and 40) and made (i.e., median value between 11 and 20) during the last five years.

Normality tests To determine whether parametric tests can be used, the normality of the five independent variables was analyzed. Table 9 presents the skewness and kurtosis of the collected data, together with the results of the Shapiro–

Table 11 Post-tests with significant results

Dependent variable	Independent variable	Test	Test Statistic	Sig.
STS _{Competence}	Degree	Independent-samples Kruskal–Wallis test	8.018	0.046
STS _{Competence}	Requirements Engineering modeling	Independent-samples Mann–Whitney <i>U</i> test	−2.050	0.040
STS _{Competence}	Models read	Independent-samples Kruskal–Wallis test	12.623	0.049
STS _{valueStream+Importance}	Gender	Independent-samples Mann–Whitney <i>U</i> test	−1.966	0.049
STS _{valueStream+Importance}	Models read	Independent-samples Kruskal–Wallis test	15.324	0.018
STS _{valueStream+Importance}	Models made	Independent-samples Kruskal–Wallis test	18.584	0.002

Wilk test [63]. The Z_{skewness} of STS_{ValueProposition} (−2.342) and STS_{valueStream+Importance} (−2.761) are both smaller than −1.96 (i.e., $Z_{0.95}$), which indicates a negatively skewed distribution of these variables. We should thus reject the normal distribution (i.e., H_0) at the 0.05 level of significance for all dependent variables, based on the Shapiro–Wilk statistic. We supplemented this analysis by a visual check of the histograms and Q–Q plots. Consequently, we will employ nonparametric tests to analyze the hypotheses and post-tests.

Hypothesis tests In the hypotheses, the median value of one sample is tested against a hypothesized value (i.e., 50). As the independent variables are not normally distributed, the nonparametric one-sample Wilcoxon signed rank test is suited. Table 10 shows the results of this statistical test for each of the five independent variables. The median STSs range between 52.5 and 80. For the concept of an Internal Goal, the null hypotheses cannot be rejected as the significance (i.e., 0.200) exceeds 0.05. By analyzing the textual clarifications of the respondents, it becomes apparent that both notations are considered as being inappropriate. The main reason is the industrial character of the current notations, which still induces the meaning of a Process. Nevertheless, the alternative hypotheses of the Competence, Value Proposition, Customer Goal, and valueStream + Importance concepts can be accepted at a level of significance between 0.000 and 0.002. It can thus be concluded that the semantic transparency of the revised notations of these concepts is higher compared to the initial one (see Sect. 6).

Post-tests Several post-tests were performed to check whether demographic and time variables have any confounding effects on the five dependent variables. For the nominal/ordinal demographic variables with two categories (i.e., Gender, Requirements Engineering modeling, Business Process modeling, Enterprise Architecture modeling), the nonparametric independent-samples Mann–Whitney *U* test was used. In case of multiple categories (i.e., Highest degree, Working sector, Modeling expertise, Models read, and Models made), an independent-samples Kruskal–Wallis test was performed. Finally, a Spearman’s correlation analysis was employed to test whether there is a significant correlation between the interval variables (i.e., birth year and question

time) and the STSs. In Table 11, post-tests with significant results are listed.

For STS_{Competence}, we see confounding effects by the variables Degree (i.e., Sig. = 0.046), Models read (i.e., Sig. = 0.049), and Requirements Engineering modeling (i.e., Sig. = 0.040). Concerning the highest degree obtained, the highest median score (i.e., 80) belongs to the group with a Master’s degree, while the lowest median score (i.e., 52.5) can be found in the category of people educated at the Bachelor and Doctoral level. If we analyze the medians across the categories of Models read, the group with a number of models between 41 and 50 have the highest score (i.e., median = 95), while the lowest scores origin in the ranges [1–10] and 50+ (i.e., median = 50). With respect to Requirements Engineering modeling, respondents without knowledge have a significantly higher median (i.e., 85) than the group with basic notions (i.e., median = 60). Gender, Models read, and Models made have a significant effect on STS_{valueStream+Importance}. More specifically, the median STS_{valueStream+Importance} of the male respondents is equal 82.5, which is significantly higher than the median of 55 for the female participants. With respect to Models read and Models made, the highest median score (i.e., 100) can be found in the category of respondents, which have read or made more than 50 models during the last five years. For these variables, the lowest median (i.e., 40) origins in the range [21–30] for models read and [31–40] for models made.

7 Related work

Table 12 provides a summary of related work that focuses on the design and/or evaluation of modeling language notations. In the following, the related work will be introduced briefly and contrasted to our notation evaluation and improvement technique.

Already in 1968, Howell and Fuchs were concerned with the challenge of developing “efficient signs or symbols for use in visual communication” [26, p. 310]. Their *sign production method* is based on a series of experiments aiming to identify *population stereotypes*, a concept first described

Table 12 Comparative overview of notation design approaches

Approach	Prerequisites	Design	Evaluation	Experiments and techniques used
[26]	–	☑	☑	Each participant draws five sample notations (I) which were then classified and statistically analyzed (II) to identify <i>Stereotypes</i> . Eventually, the proposed notations were validated with new participants using blind matching (III)
[65]	Existing notation	☐	☑	This approach uses a simple matching technique where participants need to match notation with a term it represents
[8,18]	–	☑	☑	Each participant draws one sample notation (I), which were then classified and statistically analyzed to identify <i>Stereotypes</i> . Participants then chose the best stereotype, thereby defining a new <i>Prototype</i> notation (III). Eventually, participants chose the best fitting prototype for a given concept name (IV)
[13]	–	☑	☑	Similar to [8,18], participants first draw samples which are then analyzed for identifying <i>Stereotypes</i> and <i>Prototypes</i> . Other participants then interpreted the prototypes
[29]	Existing notation	☐	☑	Participants are shown sample models and asked to respond to comprehension questions
[43]	Existing notation	☐	☑	Participants are shown sample models and asked to respond to comprehension questions
[7]	Existing notation	☐	☑	The approach shows sample models with three or more alternative notations the participants need to choose
[38]	Existing models	☐	☑	Participants are shown small sample models and need to respond to comprehension questions
[61]	Existing notation	☑	☑	Participants needed to define the meaning they derive from a presented notation. Additionally, they were asked for potential improved notations

in [15]. The approach comprises two phases. First, participants are given a set of names (of modeling language concepts) with a brief description and are asked to produce five visual drawings, which are intuitive for them and minimal in complexity. Afterward, the experimenters categorize the drawings and identify the six most occurring signs for a concept. This list is then shown to a second group of participants, who validates the order of the signs according to their applicability for a given concept.

The *matching test* is introduced in [65] to evaluate the efficiency of graphical symbols by means of intuitive comprehensibility. Following this technique, a natural language introduction to a domain concept is provided first. Afterward, the participants are given a catalogue of all graphic elements belonging to that domain and asked to select the one they believed is most suitable to represent that concept.

An approach to empirically evaluate semantic transparency has been proposed as a set of experiments applied to the *i** modeling language in [8,18]. The approach comprises three phases. In phase one, participants are provided the name and a brief description of a syntactic element of *i**. They are then asked to sketch an intuitive notation for that element.

In phase two, the authors analyze the provided sketches and identify population stereotypes. In a concluding third phase, another group of participants were provided 160 sketches (among which are the population stereotypes identified in phase two), as well as the name and definition of *i** concepts. The participants were asked to select one sketch, which they deemed most intuitive. A slight extension of the technique of Genon et al. [8,18] has been proposed in [37]. The authors show that the time-consuming task of asking participants to sketch potential notations can—to some extent—be replaced by querying image databases.

The approach proposed in [8] extends the ideas in [13] by introducing a fourth phase called *semantic transparency experiment* and by an application to the UML. In this last phase, participants are not only given the population stereotypes but also the as-is language to enable a comparison between a given and an alternative notation that was created by participants.

A study aiming to compare the intuitiveness of UML Activity Diagrams, Event-driven Process Chains, and the Business Process Model and Notation is presented in [29]. The study operationalizes intuitiveness by “measuring how

many questions the participants answered correctly and how much time it took them to do so" [29, p. 94]. Therefore, participants were given a set of models of different process modeling languages, for which comprehension questions needed to be solved.

An evaluation technique for a DSML for ambient assistant living called Human Cognitive Modeling Language (HCM-L) [41] is presented in [43]. The technique is concerned with the intuitive understanding of a whole HCM-L model. Participants were first asked to describe what information they could extract by looking at the model. Afterward, they were given seven comprehension questions about this model.

An interesting approach to improve the notation of DSMLs was proposed in [7]. The approach involves the crowd in the language design process. A first version of the language and its notation is being developed by the language engineer and afterward evaluated by end users. This revised language is then presented to the crowd for fine-tuning purposes.

A recent technique has been presented by Liaskos et al. [38] using the case of evaluating the intuitive comprehensibility of a goal modeling language. In contrast to the previous works, this research is concerned with the semantic transparency of model parts. Consequently, the experiments aim to provide insights to what extent participants can derive the meaning of a combination of interrelated modeling concepts. As such, the approach currently concentrates on the preparations of the experiments, i.e., the preparation of training material, the preparation of model samples that focus on a testable aspect of the language, and the design of the statistical analysis.

Siau and Tian [61] propose an approach that uses the notation association task. This is combined with a question that enables participants to propose a better notation in case they are not satisfied with the current one. The technique is evaluated by an application to a subset of the UML [61].

A lot of research is oriented toward the evaluation of single [19,21] or a group of modeling languages [5,29]. Most of these works focus on visual expressiveness as this is an objective measure, whereas other dimensions such as semantic transparency are subject to personal-, context-, and culture-specific influences (cf., [27, p. 17]). The challenge of objectively evaluating semantic transparency (cf., [39]) might be one indicator why this principle is scarcely considered in research and also in current modeling standards like Business Process Model and Notation [19] and Decision Model and Notation [11]. In contrast to GPMLs, this challenge can be overcome for DSMLs as the intended users are well known during the language design.

The previously presented research is mostly concerned with either the evaluation of complete models, e.g., testing whether participants derive the correct information when looking at a model, or with a comparison of alternative nota-

tions for a given (set of) language(s). Within this paper, we aim at an easily usable and customizable technique that supports the design of semantically transparent DSML notations. The scope of the proposed technique is on evaluating the language concepts individually (i.e., *concept-level intuitiveness*) first and only afterward combined within a model (i.e., *model-level intuitiveness*). Moreover, the proposed technique combines a phase for evaluating an initial notation with a phase for (i) revising the notation based on the tasks' feedback, and (ii) an empirical comparative evaluation of the revised against the initial notation. This latter evaluation thus provides a statistically sound response to whether improvements to the notation have been achieved and thereby concludes one design cycle.

The presented technique can be also positioned within the broader research field of (situational) method engineering [24,52], in which different assembly techniques are available to derive a situational method from relating or integrating existing *method chunks*. We believe our technique can support the identification and specification of mappings between different method chunks by looking at the terms (term association) and the notation (notation association) provided by individual chunks [53]. The explicit consideration of a notation during the composition of method chunks could enable a wider adoption of the resulting situational method. The research at hand might contribute to establishing another research line that also focuses on integrating or aligning the notation of method chunks toward a homogeneous situational method.

8 Discussion

Before concluding the paper, we want to discuss the validity and reliability of the obtained results. While Sect. 8.1 focuses on the proposed technique, the validity and reliability of the evaluation and improvement of the PGA modeling language with respect to semantic transparency are subsequently discussed in Sect. 8.2.

8.1 Reflection on the technique's requirements

In the following, we refer to our initial requirements of the notation evaluation and improvement technique (see Table 2 for an overview) and discuss to what extent and how they have been addressed in the proposed technique. As four out of the seven requirements can be considered being fulfilled completely and the remaining three partially (see Table 13), the presented approach still comes with some limitations, which are discussed in the remaining paragraphs

What concerns **applicability**, the technique generally requires an initial notation to start with. It is—in its entirety—thus no green-field approach that supports the initial develop-

Table 13 Requirements-based discussion of the proposed technique

Requirement	Fulfilment	Consideration in the technique
Efficient customization	●	The technique is generic and allows an efficient customization to DSMLs
Notation improvement	●	In both applications, BCM and PGA, improvements of the initial notation have been achieved
Efficient use	⓪	The approach requires some preparation, however, we believe the amount of effort is minimal when conducting manual experiments and analysis in small groups. For larger groups, technical support should be configured which increases the preparation efforts but at the same time facilitates efficient analysis
Involve participant suggestions	●	The approach enables participants to continuously provide their feedback on the initial notation as well as to provide suggestions for further improvements
Semantic transparency	●	The whole technique is focused on semantic transparency: first on a concept-level, later on a model-level
Technical independence	⓪	The technique can be used without any technology (cf., [6] and [55], and Table 3) However, technology-support might facilitate using the technique with many participants
Modular structure	⓪	Whenever possible, the individual tasks are designed in an independent manner. As such, term and notation association can be applied in isolation. However, the best and most comprehensive results will be derived when applying phase 1 and phase 2 as proposed in the design procedure (see Fig. 4)

ment of a modeling language notation. Considering the term association task though, this task can be executed as soon as the meta-model of the modeling language is developed. We further expect that the presented technique yields best results when being applied to DSMLs. This is because the abstraction level of GPMLs is mostly too high to properly address semantic transparency. Consider the UML concept of Class as an example. Although most computer scientists might comprehend the semantics of a Class, they would have troubles drafting semantically transparent notations during a term association task. Furthermore, if the Class notation is shown to laymen in a notation association task, it is very unlikely that they intuitively derive its semantics. In other words, the approach at hand might yield valuable insights only if semantic transparency is an explicit requirement of a GPML.

Concerning the **scalability** of the approach, we have presented the successful application in two completely independent and diverging cases. In the context of the BCM modeling method case [6], 15 students were performing the tasks in groups of up to five persons. In this case, participants heavily used the provided pens and thus proposed colorful suggestions in the term association task. Likewise, manual evaluation of the term and notation association responses was conducted. In the PGA case, 139 students participated in the experiments. For the analysis, technological support was used for digitizing the responses, but the analysis was

still conducted manually. Consequently, scalability can be considered as achieved for the evaluation and improvement technique aside from some weaknesses identified in the process.

Although the 139 students were instructed to use color to convey semantics during the evaluation of the PGA method [55], hardly any color besides (standard) blue was used (see Table 4). In this respect, the approach would benefit from tool support to efficiently collect and handle large data sets. In particular, a web environment should support the term association task with a WYSIWYG drawing editor and the comparative evaluation task with a score selected on a graphical scale using a slider. Such an environment should also automatically ensure randomness in allocating the concepts to the tasks, allocate participants into the different test groups and statistically analyze the responses.

With respect to the **generalizability** of the overall claim that the presented technique helps in improving the semantic transparency of domain-specific modeling languages, we need to constrain our positive results by the limitation of only two applications. In both applications, the initial notation was revised based on the feedback gained from different groups of respondents. Feedback from peers in the conceptual modeling and enterprise modeling communities moreover makes us confident to expect further applications of the technique in the near future. These future applications can take the

individual STS scores as a benchmark against which future notation alternatives can be evaluated.

8.2 Validity and reliability of the PGA evaluation and improvement

To preserve **construct validity**, it is important to ensure that the executed tasks are suited to evaluate the semantic transparency of a modeling language. Therefore, our technique builds upon and integrates existing evaluation approaches (see Sect. 7), for which the origin of the tasks is rigorously substantiated.

With respect to **internal validity**, factors that confound the results need to be avoided. In this respect, participants were chosen with the same educational background (i.e., Master's students in Business Engineering and Business Administration) for the first phase of the technique. Besides, the participants had similar foreknowledge in conceptual modeling and received a collective introduction to PGA. For the second phase, we explicitly approached modeling practitioners with prior modeling experience to empirically evaluate the semantic transparency of the initial and the revised notation. Participation in all experiments was voluntarily, and no compensation was provided. Finally, we used different experimental groups during each phase to prevent confounding effects. More specifically, we made sure that the terms given during the term association did not influence the results of the notation association task while in the second phase we could test whether there is an influence of the position of a concept on the given score. Participants were randomly assigned to the experimental groups to mitigate potential allocation biases. Moreover, the results of the comparative evaluation within the second phase of the technique are statistically analyzed to test the hypotheses and potential confounding effects.

The choice of participants also affects the **external validity** or generalizability of the results. The participants in the first phase have a strong economic orientation which enabled us to obtain a group of respondents with knowledge and skills that can act as a proxy for business-oriented stakeholders. These stakeholders are the targeted end-users of the PGA modeling method. Nevertheless, the choice for students in phase one is an inherent limitation. We addressed this threat by explicitly incorporating practitioners with modeling experience in the second phase. For this group, post-tests are performed to compare the STS scores given by academics with the ones given by participants working in the public and/or private sector. These tests did not yield significant differences between the two groups.

Reliability reflects the degree to which the results could be reproduced by the modeling community. To ensure this, the generic procedure to apply the technique is comprehensively explained in Sect. 4 and the URL of the evaluation questionnaires can be found online^{1,3}. In Sects. 5 and 6, we

report on the results of applying the complete technique in the context of the PGA modeling method. We believe that the level of detail in the description of the generic approach and its application to PGA enables the modeling community to replicate it to other modeling languages.

9 Conclusion

We introduced a technique that employs user participation and an empirical evaluation to help method engineers in the evaluation and improvement of semantically transparent modeling language notations. This technique was developed by two iterations of Action Design Research, in which it was applied to the BCM and PGA modeling methods. This paper particularly focuses on the generic description of the approach and its application in the context of improving the semantic transparency of the PGA modeling method. We showed that this technique is able to: (i) identify initial PGA notation concepts that require improvement, (ii) involve participants to derive suggestions for notation revisions, and (iii) empirically test the achieved improvements. Further statistical tests confirmed that the PGA notation has been improved. The revised notations are deployed in a new version of the openly available PGA tool. Still, further improvements to the PGA notation are necessary in the future, especially considering the Internal Goal notation.

We are currently investigating possibilities of automating the presented technique. We aim to set up a web environment that automatically generates the evaluation sheets once the concepts and sample notations are uploaded. Moreover, it shall provide a WYSIWYG web editor for drawing notations and storing them. The system shall use OpenCV or similar technologies to automatically analyze the created proposals for new notations. Besides, enabling text analysis could be useful for the results of the notation association task as well as implementing statistical analysis of the responses and the automated generation of evaluation reports. Ultimately, the web environment will enable an efficient setup, execution, and analysis of the technique. Consequently, it will mitigate issues related to the paper-and-pen application.

This paper reports on the iterative development of a notation evaluation and improvement technique through ADR. The development was guided by reflection and learning by the involved researchers during two applications, with the aim of realizing the proposed requirements. The tight coupling between development and evaluation activities is an inherent feature of ADR, which combines theory generation with researcher intervention while solving a particular problem [60]. A following research step includes the evaluation of the technique by method engineers. This could be realized by a replication in the context of other DSMLs, as presented in Sect. 4, followed by an evaluation of the requirements by

the method engineers (see Sect. 8). This will further improve the generalizability of this study's findings in other method engineering contexts that are oriented toward achieving an efficient model-based communication with end-users.

Future research is also needed to investigate if models with a higher semantic transparency correlate with a better understanding of the model content and the underlying domain by model users. Therefore, an empirical comparison could be set up, in which the relationship between the semantic transparency of the notation and the effectiveness and efficiency of script interpretation is investigated. Whereas effectiveness relates to how good a modeler is able to perform certain tasks using a certain modeling language, efficiency concerns the effort (i.e., time) required to interpret a script and develop domain understanding [17]. In particular, the comparative evaluation of the technique (i.e., task 6 in Fig. 4) could be extended by including notation association [6], recall, comprehension and problem-solving tasks [17]. This enables to investigate the strength of relationship between the collected data by a correlation analysis.

Acknowledgements This research has been partly funded through the Federal Ministry of Education, Science and Research (BMBWF) funded France / Austria Joint Scientific and Technological Cooperation program with the project number FR 01/2019.

Funding Open access funding provided by TU Wien (TUW).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. ADOxx.org: ADOxx Metamodeling Platform (2020). <https://www.adoxx.org/live/home>. Accessed 21 Sept 2020
2. Batini, C., Ceri, S., Navathe, S.B., et al.: *Conceptual Database Design: An Entity-Relationship Approach*, vol. 116. Benjamin/Cummings, Redwood City (1992)
3. Bork, D.: A development method for the conceptual design of multi-view modeling tools with an emphasis on consistency requirements. Ph.D. thesis, University of Bamberg (2015). <https://d-nb.info/1079840273/34>
4. Bork, D., Buchmann, R.A., Karagiannis, D., Lee, M., Miron, E.T.: An open platform for modeling method conceptualization: the OMiLAB digital ecosystem. *Commun. Assoc. Inf. Syst.* **44**, 673–697 (2019)
5. Bork, D., Karagiannis, D., Pittl, B.: Systematic analysis and evaluation of visual conceptual modeling language notations. In: 2018 12th International Conference on Research Challenges in Information Science (RCIS), pp. 1–11. IEEE (2018)
6. Bork, D., Schriffer, C., Karagiannis, D.: Intuitive understanding of domain-specific modeling languages: proposition and application of an evaluation technique. In: *International Conference on Conceptual Modeling*, pp. 311–319. Springer (2019)
7. Brambilla, M., Cabot, J., Cánovas Izquierdo, J.L., Mauri, A.: Better call the crowd: using crowdsourcing to shape the notation of domain-specific languages. In: 10th ACM SIGPLAN International Conference on Software Language Engineering, pp. 129–138. ACM (2017)
8. Caire, P., Genon, N., Heymans, P., Moody, D.L.: Visual notation design 2.0: towards user comprehensible requirements engineering notations. In: 21st IEEE International Requirements Engineering Conference (RE), pp. 115–124. IEEE (2013)
9. Cohen, J.: *Statistical Power Analysis for the Behavioral Sciences*, revised. Academic Press, New York (1977)
10. Cook, C., Heath, F., Thompson, R.B.T.: Score reliability in Webor internet-based surveys: unnumbered graphic rating scales versus Likert-type scales. *Educ. Psychol. Meas.* **61**(4), 697–706 (2001)
11. Dangarska, Z., Figl, K., Mendling, J.: An explorative analysis of the notational characteristics of the decision model and notation (dmn). In: *IEEE 20th International Enterprise Distributed Object Computing Workshop*, pp. 1–9. IEEE (2016)
12. De Kinderen, S., Ma, Q.: Requirements engineering for the design of conceptual modeling languages. *Appl. Ontol.* **10**(1), 7–24 (2015)
13. El Kouhen, A., Gherbi, A., Dumoulin, C., Khendek, F.: On the semantic transparency of visual notations: experiments with uml. In: *International SDL Forum*, pp. 122–137. Springer (2015)
14. Figl, K., Recker, J.: Exploring cognitive style and task-specific preferences for process representations. *Requir. Eng.* **21**(1), 63–85 (2016)
15. Fitts, P.M.: *Human information handling in speeded tasks* (Report No. RC-109). Tech. rep., IBM Research Center, Yorktown Heights, NY (1959)
16. Frank, U.: Domain-specific modeling languages: requirements analysis and design guidelines. In: *Domain Engineering*, pp. 133–157. Springer (2013)
17. Gemino, A., Wand, Y.: A framework for empirical evaluation of conceptual modeling techniques. *Requir. Eng.* **9**(4), 248–260 (2004). <https://doi.org/10.1007/s00766-004-0204-6>
18. Genon, N., Caire, P., Toussaint, H., Heymans, P., Moody, D.: Towards a more semantically transparent i* visual syntax. In: *International Conference on Requirements Engineering: Foundation for Software Quality*, pp. 140–146. Springer (2012)
19. Genon, N., Heymans, P., Amyot, D.: Analysing the cognitive effectiveness of the BPMN 2.0 visual notation. In: *International Conference on Software Language Engineering*, pp. 377–396. Springer (2010)
20. Goodhue, D., Lewis, W., Thompson, R.: Statistical power in analyzing interaction effects: questioning the advantage of pls with product indicators. *Inf. Syst. Res.* **18**(2), 221–227 (2007)

21. Granada, D., Vara, J.M., Brambilla, M., Bollati, V., Marcos, E.: Analysing the cognitive effectiveness of the webml visual notation. *Softw. Syst. Model.* **16**(1), 195–227 (2017)
22. Gulden, J., van der Linden, D., Aysolmaz, B.: A research agenda on visualizations in information systems engineering. In: 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering, pp. 234–240 (2016)
23. Gulden, J., Yu, E.: Toward requirements-driven design of visual modeling languages. In: IFIP Working Conference on The Practice of Enterprise Modeling, pp. 21–36. Springer (2018)
24. Henderson-Sellers, B., Ralyté, J.: Situational method engineering: state-of-the-art review. *J. Univ. Comput. Sci.* **16**(3), 424–478 (2010)
25. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *MIS Q.* **28**(1), 75–105 (2004)
26. Howell, W.C., Fuchs, A.H.: Population stereotypy in code design. *Organ. Behav. Hum. Perform.* **3**(3), 310–339 (1968)
27. Izquierdo, J.L.C., Cabot, J.: Collaboro: a collaborative (meta) modeling tool. *PeerJ Comput. Sci.* **2**, e84 (2016)
28. Jannaber, S., Riehle, D.M., Delfmann, P., Thomas, O., Becker, J.: Designing a framework for the development of domain-specific process modelling languages. In: International Conference on Design Science Research in Information System and Technology, pp. 39–54. Springer (2017)
29. Jošt, G., Huber, J., Heričko, M., Polančič, G.: An empirical investigation of intuitive understandability of process diagrams. *Comput. Stand. Interfaces* **48**, 90–111 (2016)
30. Karagiannis, D.: Agile modeling method engineering. In: Karanikolas, N.N., Akoumianakis, D., Nikolaidou, M., Vergados, D.D., Xenos, M., Giaglis, G.M., Gritzalis, S., Merakos, L.F., Tsanakas, P., Sgouropoulou, C. (eds.) Proceedings of the 19th Panhellenic Conference on Informatics, PCI 2015, Athens, Greece, pp. 5–10. ACM (2015)
31. Karagiannis, D., Mayr, H.C., Mylopoulos, J. (eds.): Domain-Specific Conceptual Modeling, Concepts, Methods and Tools. Springer, Berlin (2016)
32. Karsai, G., Krahn, H., Pinkernell, C., Rumpe, B., Schindler, M., Völkel, S.: Design guidelines for domain specific languages. In: Proceedings of the 9th OOPSLA Workshop on Domain-Specific Modeling (2009)
33. Kirakowski, J., Corbett, M.: Sumi: the software usability measurement inventory. *Br. J. Educ. Technol.* **24**(3), 210–212 (1993)
34. Krogstie, J.: Quality of conceptual models in model driven software engineering. In: Cabot, J., Gómez, C., Pastor, O., Sancho, M.R., Teniente, E. (eds.) Conceptual Modeling Perspectives, pp. 185–198. Springer (2017)
35. Krogstie, J., Sindre, G., Jørgensen, H.: Process models representing knowledge for action: a revised quality framework. *Eur. J. Inf. Syst.* **15**(1), 91–102 (2006)
36. Larkin, J.H., Simon, H.A.: Why a diagram is (sometimes) worth ten thousand words. *Cogn. Sci.* **11**(1), 65–100 (1987)
37. Laue, R.: Nutzung von bilddatenbanken zur erstellung von symbolen für graphische modellierungssprachen. *Modellierung* **2018** (2018)
38. Liaskos, S., Zhian, M., Jaouhar, I.: Experimental practices for measuring the intuitive comprehensibility of modeling constructs: an example design. In: Grossmann, G., Ram, S. (eds.) Proceedings of the 2020 ER Workshops (in press, 2020)
39. van der Linden, D., Zamansky, A., Hadar, I.: How cognitively effective is a visual notation? On the inherent difficulty of operationalizing the physics of notations. In: Enterprise, Business-Process and Information Systems Modeling, pp. 448–462. Springer (2016)
40. Lindland, O.I., Sindre, G., Solvberg, A.: Understanding quality in conceptual modeling. *IEEE Softw.* **11**(2), 42–49 (1994)
41. Mayr, H.C., Al Machot, F., Michael, J., Morak, G., Ranasinghe, S., Shekhovtsov, V., Steinberger, C.: Hcm-l: domain-specific modeling for active and assisted living. In: Domain-Specific Conceptual Modeling, pp. 527–552. Springer (2016)
42. Michael, J., Mayr, H.C.: Creating a domain specific modelling method for ambient assistance. In: 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 119–124. IEEE (2015)
43. Michael, J., Mayr, H.C.: Intuitive understanding of a modeling language. In: Australasian Computer Science Week Multiconference, p. 35. ACM (2017)
44. Moody, D.: The ‘physics’ of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Trans. Softw. Eng.* **35**(6), 756–779 (2009)
45. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data Knowl. Eng.* **55**(3), 243–276 (2005)
46. Moody, D.L., Shanks, G.G.: What makes a good data model? Evaluating the quality of entity relationship models. In: International Conference on Conceptual Modeling, pp. 94–111. Springer (1994)
47. Mylopoulos, J.: Conceptual Modelling and Telos. Conceptual Modelling, Databases, and CASE: An Integrated View of Information System Development, pp. 49–68. Wiley, New York (1992)
48. Newell, A., Simon, H.A.: Human Problem Solving. Prentice-Hall, Englewood Cliffs (1972)
49. Osterwalder, A., Pigneur, Y., Tucci, C.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. Wiley, Hoboken (2010)
50. Peffers, K., Rothenberger, M., Tuunanen, T., Vaezi, R.: Design science research evaluation. In: International Conference on Design Science Research in Information Systems, pp. 398–410. Springer (2012)
51. Petre, M.: Why looking isn’t always seeing: readership skills and graphical programming. *Communications of the ACM* **38**(6), 33–44 (1995)
52. Ralyté, J., Deneckère, R., Rolland, C.: Towards a generic model for situational method engineering. In: International Conference on Advanced Information Systems Engineering, pp. 95–110. Springer (2003)
53. Ralyté, J., Rolland, C.: An assembly process model for method engineering. In: International Conference on Advanced Information Systems Engineering, pp. 267–283. Springer (2001)
54. Recker, J.: Opportunities and constraints: the current struggle with BPMN. *Bus Process Manag J* **16**(1), 181–201 (2010)
55. Roelens, B., Bork, D.: An evaluation of the intuitiveness of the pga modeling language notation. In: Enterprise, Business-Process and Information Systems Modeling, pp. 395–410. Springer (2020)
56. Roelens, B., Poels, G.: The creation of business architecture heat maps to support strategy-aligned organizational decisions. In: 8th European Conference on IS Management and Evaluation (ECIME), pp. 388–392. Acad. Conferences Ltd. (2014)
57. Roelens, B., Steenacker, W., Poels, G.: Realizing strategic fit within the business architecture: the design of a process-goal alignment modeling and analysis technique. *Softw. Syst. Model.* **18**(1), 631–662 (2019)
58. Saaty, T.: How to make a decision: The analytic hierarchy process. *Eur. J. Oper. Res.* **48**(1), 9–26 (1990)
59. Saunders, M., Lewis, P., Thornhill, A.: Research Methods for Business Students, 7th edn. Pearson Education Ltd, Essex (2016)
60. Sein, M.K., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R.: Action design research. *MIS Q.* **2011**, 37–56 (2011)
61. Siau, K., Tian, Y.: A semiotic analysis of unified modeling language graphical notations. *Requir. Eng.* **14**(1), 15–26 (2009)
62. Strembeck, M., Zdun, U.: An approach for the systematic development of domain-specific languages. *Softw. Pract. Exp.* **39**(15), 1253–1292 (2009)
63. Thode, H.: Testing for Normality. Taylor and Francis, Hoboken (2002)

64. Utz, W., Buchmann, R., Bork, D., Karagiannis, D.: A bpm lifecycle plug-in for modeling methods agility. In: Americas Conference on Information Systems (AMCIS) (2020)
65. Zwaga, H., Boersema, T.: Evaluation of a set of graphic symbols. *Appl. Ergon.* **14**(1), 43–54 (1983)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Dominik Bork is Assistant Professor for Business Systems Engineering at the Faculty of Informatics, Institute of Information Systems Engineering, Business Informatics Group at TU Wien. His research interests comprise conceptual modeling and model engineering as well as their application in domains such as modeling tool development, web modeling. A primary focus of ongoing research is on empirical aspects of conceptual modeling and the mutual benefits of conceptual modeling and Artificial Intelligence. For more information, please visit <https://www.model-engineering.info/>.



Ben Roelens is assistant professor at the Open University of the Netherlands, Faculty of Science, at which he is member of the department of Information Science. He obtained his doctoral degree in 2015 at Ghent University, which dealt with the use of conceptual models to realize strategic fit. He (co-)authored 21 publications, of which 11 are listed in Web of Science in Business & Information Systems Engineering, Software and Systems Modeling, Journal of Database Management, among others. His main research interests lie in the fields of Enterprise Modelling, Business Ontology, and Service Science. For more information, please visit <https://www.researchgate.net/profile/Ben-Roelens>.