A Language for Multi-Perspective Goal Modelling: Challenges, Requirements and Solutions

Sietse Overbeek, Ulrich Frank, Christian Köhling

PII: S0920-5489(14)00079-8
DOI: doi: 10.1016/j.csi.2014.08.001
Reference: CSI 2980

To appear in: Computer Standards & Interfaces

Accepted Manuscript

Received date: 1 October 2013
Revised date: 27 June 2014
Accepted date: 12 August 2014

Please cite this article as: Sietse Overbeek, Ulrich Frank, Christian Köhling, A Language for Multi-Perspective Goal Modelling: Challenges, Requirements and Solutions, Computer Standards & Interfaces (2014), doi: 10.1016/j.csi.2014.08.001

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.
A Language for Multi-Perspective Goal Modelling: Challenges, Requirements and Solutions

Sietse Overbeek*, Ulrich Frank, Christian Köhling

Institute for Computer Science and Business Information Systems, University of Duisburg-Essen, Reckhammerweg 2, D-45141 Essen, Germany, EU

Abstract

Successful implementation of an enterprise strategy, the reorganization of an enterprise, the successful enterprise-wide adoption of a new enterprise resource planning system, or simply being able to manage the daily operations at an enterprise in general are all common examples of organizational actions that are strongly interrelated with the achievement of goals related to these actions. From the research as presented in this paper, it becomes clear that it is not elementary to clearly formulate goals and to understand how to achieve them. In two use scenarios, it is described how the executive board of a mid-sized bank in Germany wants to achieve their overall goal to increase the bank appraisal. The first scenario deals with determining who is responsible for goal creation and accomplishment, while the second scenario deals with describing a concrete goal system. A domain-specific modelling language (DSML) for designing goal models is proposed that provides solutions for requirements that are derived from the described scenarios. This DSML is coined the ‘Goal Modelling Language’ (GoalML), which enables to develop goal models from multiple perspectives in order to relate goals with their context and vice versa.

Keywords: DSML, enterprise modelling, GoalML, goal modelling, MEMO

*Corresponding author. Tel: +49 201 1834150; fax: +49 201 1834011.
Email addresses: Sietse.Overbeek@uni-due.de (Sietse Overbeek), Ulrich.Frank@uni-due.de (Ulrich Frank), Christian.Koehling@uni-due.de (Christian Köhling)
1. Introduction

Organizations are defined as ‘goal-oriented social systems’ according to the predominant rational conception of the enterprise. Especially with respect to enterprises as specific kinds of organizations this conception is of particular importance. First, it emphasizes the idea of an enterprise as a “rationally constructed artifice” [1, p. 23]. Second, it stresses the pivotal relevance of economic action in enterprises, since the conception of economics is usually defined as a means-end relationship. It seems obvious that strategic planning, managerial decision making and every kind of professional action in a company requires the existence of explicit or at least implicit goals. Goals are the foundation of (rational) choice. At the same time, developing and using organizational goals is far from trivial. The problem starts with the very conception of an organizational goal. Before going into more detail, it has to be clarified first that goals have certain functions. A goal function is viewed as the achieved effect by accomplishing a goal and these effects can be different in nature. Such effects are, for example, reaching a decision (see, e.g. [2]), increasing focus (see, e.g. [3, 4]), or improving coordination (see, e.g. [5]). However, are such organizational goals simply the goals defined by top management or is “the organizational objective ... indirectly a personal objective of all the participants” [6, p. 17]? In any case, the definition of organizational goals needs to account for goals that are not compatible with each other or that are even contradictory.

In an extreme interpretation, the rational conception of the enterprise would recommend to precisely define goals for any kind of organizational action. However, there are reasons to challenge such a recommendation. First, research in Cognitive Psychology indicates a limited ability of most humans to make decisions purely on a rational basis [7]. These findings correspond to Simon’s conception of ‘bounded rationality’ [8]. Second, actual organizational behaviour indicates that organizations often operate without explicit goals. Instead, they may be “saturated with subjectivity, abstraction, guesses, ... and arbitrariness” [9, p. 5]. In addition to that, there are doubts that it is appropriate to develop elaborate plans: The complexity and contingency of organizations and their environment may rather recommend piecemeal engineering or even ‘muddling through’ [10]. Further studies stress the limitations of the rational conception of the organization. Pfeffer [11, p. 5], for instance, demands that in addition to pursue a rational path of action, managers should also account for ‘symbolic’ actions which are aimed
at fostering motivation and commitment. In line with that, various authors have stressed the relevance of organizational culture, i.e., of specific values, sense-making, and rituals (see, e.g. [12, 13]). Although these critical and well-thought statements object against a pure rational view of the organization, they do not refute the need for creating goals. Instead, they warn against a naive conception of organizational goals. It seems that in managerial practice goals have become even more important during the last years as there is an increasing use of performance indicators which are aimed at measuring a degree of goal achievement (see, e.g. [4, 14]).

Against this background, our investigation aims at supporting organizations with developing, using, and maintaining goal models. Our work is based on the following assumptions. While a pure rational or even mechanistic view of organizations is inappropriate, the modelling of goals, their interrelationships and possible consequences support an analytical approach to decision making and planning. Goal models also support coherence and consistency of organizational goals, thereby avoiding friction and waste of resources. Furthermore, explicit goals promote clarity and help employees to develop a better appreciation of organizational action. Moreover, goal models may help to prevent hidden agendas and the misuse of ambiguous goals, they are suited to foster sense-making and motivation, and may help to avoid dysfunctional effects of performance indicators. Goal models also support the assessment of organisational actions and serve as a common orientation for organisational development. At the same time they may be a core subject of organisational change themselves. With respect to building advanced enterprise systems, (semi-)formal goal models enable automated analyses, e.g., to check for consistency with business process models or the use of resources. Furthermore, they may be transformed into code of corresponding goal-oriented control systems. Finally, different stakeholders of an organization are interested in different goals and prefer different levels of detail for describing goals. Goal models that enable different views on an organizational goal system support this need. A goal system is understood to be the structure that is formed after generating relationships between a collection of organizational goals (see, e.g. [15]).

With respect to supporting the creation of goal models our work is based on two key decisions. First, a goal model represents relevant goals as well as relationships between goals. However, analysing goals without accounting for measures and constraints does not make sense. To satisfy this demand, we suggest goal models to be an integral part of enterprise models, which pro-
vide relevant contexts, such as: Descriptions of resources, business process models or models of the IT infrastructure (see, e.g. [16]). Second, while it is possible to model goals with a general purpose modelling language (GPML) like the Unified Modelling Language (UML) or the Entity-Relationship Modelling (ERM) language, we decided for a domain-specific modelling language (DSML). This is for three reasons: Using a GPML would require a modeller to reconstruct relevant concepts such as various kinds of goals from scratch, which would compromise modelling productivity. Furthermore, a DSML would include specific constraints that prevent modellers to a certain degree from creating fallacious models. Finally, a DSML enables the use of a specific concrete syntax, which fosters comprehensibility. The presented DSML, named Goal Modelling Language (GoalML), has been introduced first in [17], on which this paper further builds.

The paper is organized as follows. At first, we will outline a method for designing the GoalML in section 2. Second, we will illustrate the benefit of goal models by presenting two use scenarios in section 3. The scenarios also serve to prepare for the subsequent analysis of requirements a goal modelling language should satisfy. A respective analysis is conducted in section 4. Against this background, the design of the DSML is presented in section 5. It extends an existing method for multi-perspective enterprise modelling and comprises a meta model as well as a concrete syntax. Subsequently, the proposed DSML is compared against related work in section 6. Finally, the conclusions and an overview of future research are presented in section 7.

2. Outline of a Method for Designing the GoalML

Modellers are supposed to create domain-specific models more efficiently than it would be possible with a GPML. It is based on reconstructing language concepts from technical terms used in the respective domain of discourse [18, p. 28]. In this sense, the GoalML is a DSML that enables the design of goal models as well as of some aspects to be accounted for during the process of defining goals.

Designing a DSML itself is a challenging task. At first, analysing the technical language characteristics for a targeted domain may be demanding, because it may lack coherence and consistency, which is definitely the case for the terms used to describe goals. On the one hand, this relates to the existence of various terms that are somehow adjacent to the term ‘goal’, such as: ‘Target’, ‘motive’, ‘objective’, ‘norm’, ‘destination’, ‘aim’, and ‘purpose’.
On the other hand, it will usually remain unclear, whether such a term is intended to represent a particular instance, a type or even a metatype. This is a distinction that is of crucial relevance for designing a DSML. Second, analysing the requirements a DSML should satisfy is particularly challenging as oftentimes prospective users do not have a complete picture of what such an artefact should include. To cope with these challenges, a method for designing DSMLs [19] that has been successfully used in different projects (see, e.g. [20, 21, 17]) was applied for developing the GoalML. The method proposes to use the process model shown in figure 1 as a guideline.

Figure 1: A macro-level process model for designing a DSML [19].

In this paper, we mainly focus on the clarification of scope and purpose, on the analysis of requirements that are specific for a goal modelling language, and on the language specification and design of the graphical notation. The overview of the state of the art in goal modelling presented in section 6 is seen as an evaluation activity as other approaches presented in that section are evaluated against a set of characteristics that are typical for goal modelling languages. The analysis of requirements for DSMLs in general and the development of a modelling tool are not part of this paper. More information on general requirements that would concern DSMLs in general are found
in [19]. The result of the first phase should “outline a convincing motivation and rationale for designing a DSML” [19, p. 140]. This motivation and rationale is in fact presented in the introduction of this paper. As the process model shown in figure 1 is a macro-level model, it is possible to zoom in on some phase to achieve the micro-level process model of that phase. This would further clarify the results as described in this paper with respect to the phase ‘analysis of specific requirements’ and zooming in on that phase results in the micro-level process model for that phase as shown in figure 2.

![Figure 2: A micro-level process model for analyzing specific requirements [19].](image)

To address the challenge that prospective users might not be able to imagine what a DSML could do for them, the method emphasizes developing and analyzing use scenarios. They are based on illustrating a DSML by presenting prospective users with diagrams that are created with a first draft of concepts and a preliminary concrete syntax. They allow conveying to users how diagrams of the targeted DSML could look like and what kind of tasks they might support. With that kind of support users can develop questions or problems to be addressed with respective diagrams together with analysts, thereby suggesting further requirements the DSML should satisfy. The extent of the requirements analysis that had been conducted to develop the GoalML makes it impossible to present it in this paper in detail. Therefore, we choose a different approach. The use scenarios presented in the following section are presented using the final concepts and notation of the GoalML. They give the reader an idea of the DSML to be developed. At the same time, they should illustrate how diagrams with a preliminary notation were used to elicitate
the specific requirements presented in section 4.

Scenarios that illustrate the use of a previously developed DSML support the evaluation of that DSML, since they allow to analyze whether and how corresponding requirements are satisfied by the DSML. Such an evaluation based on the introduced use scenarios should include stakeholders of the proposed language, where users of course play a key role.

3. Use Scenarios: Goal Modelling and Analysis for a Bank

In this section, two use scenarios are presented that illustrate the need for a multi-perspective goal modelling approach as a result of applying the first three micro-level phases that are abstracted by the ‘analyzing specific requirements’ macro-level phase as part of the process model for designing a DSML. The diagrams that are shown as part of the use scenarios serve as a medium to discuss the potential use of the goal modelling approach. Therefore, illustrating the purpose and key concepts of each of the diagrams and showing in what ways the diagrams are interrelated paves the way to commonly develop the specific requirements that are introduced in section 4 [19]. Common development of these requirements means the development of those requirements by discussing the possible set of requirements with those involved in the scenarios, while using the diagrams as a communication medium. Other requirements engineering approaches such as the Inquiry-Based Requirements Analysis approach [22] also underline the usefulness of introducing scenarios to give users insight and refine requirements by answering what-if questions while using the scenarios as a source of inspiration. The first scenario deals with goal planning and authorization relationships between goals and employees. The second scenario deals with the planning of an overall goal system that follows from the first scenario.

3.1. Scenario I: Goal Planning and Authorization

In this scenario, inspired by an online software demo application from the company ‘iteratec’ [23], a German middle-sized bank has just completed a reorganization. The bank serves private individuals and business clients varying from small-sized companies to big firms. The offered services stem from the product groups credit cards, check accounts, loans, construction loans, and savings accounts. Internally, the bank is divided in the following organizational units: Sales and marketing, capital and risk, investment, finance, controlling, HR, IT, and procurement. A total of 500 employees work
at the bank. The overall goal for the bank would be to increase the bank appraisal. The executive board of the bank wants to develop an organizational goal system to realize this overall goal. The executive board wants to determine how employees should deal with goals as part of a goal system and that the employees support this approach to prevent a limited effect and life expectancy of the formulated goals. On the one hand, the executive board wants to clarify at least broadly which organizational units create goals in what way and for what reason. On the other hand, instead of only involving top management in the process of goal creation they also want to involve team leaders and where possible team members in this process. This decision is based on the assumption that involvement of this kind promotes the identification of employees with the bank and the commitment of employees to the goal system. A complicating factor in this scenario is that a goal system has also been used as a management instrument before the reorganization of the bank. The results of using this goal system did not meet the expectations of managers and other employees and they experienced the usage of a goal system as frustrating. Possible reasons for this are the ambiguous formulation of goals and incorrect analysis of the relationships between goals and other enterprise-related concepts, leading to improper communication about the goals. Some employees had the impression that ambivalent goals were used on purpose in order to serve certain political interests. If an employee has the authority to create goals, then it depends among other aspects on his or her power to what extent the creation of a goal could be pushed in a certain direction to make sure the political interests of the goal creator are taken into account. It also became clear that employees had completely different understandings of the goal concept and of the functioning of the goals themselves. This may lead to differences in opinions on what is necessary to reach a goal, which may lead to situations the goal creator(s) never wanted as someone responsible for goal achievement may try to achieve something that was not intended by the goal creator(s).

Those who are involved in this scenario are the executive board, consisting of the chief executive officer (CEO), chief financial officer (CFO), the chief operating officer (COO), the chief information officer (CIO), and the respective organizational unit heads. Team leaders and where possible team members are involved in the goal creation process in order to involve as many employees as possible. To promote collaboration and communication, goals are represented with graphical diagrams. Therefore, the employees involved in the goal creation process require a diagram which integrates what the re-
sponsibilities are of those who are concerned with the planned goals. In such a diagram it could be depicted, for example, who or which organizational unit is responsible for goal creation and who or which unit is responsible for the accomplishment of goals. Figure 3 shows a GoalML goal planning diagram of the bank. A goal planning diagram is a model that integrates a representation of a goal system with those of other parts of an enterprise model, like business process models and an organizational chart. The shown organizational chart is modelled by means of those elements from the organizational modelling language (OrgML) that allow modelling organizational structures [24]. In the figure, it is shown that the executive board is the initiator and has the authorization for creating goals related to the business process types ‘customer strategy’, ‘process improvement & change management’, and ‘infrastructure management’. The key symbol together with the symbol showing an opened padlock combined with a target express that the executive board initiates and has the authorization for creating goals. What can be further interpreted from the diagram is that the CEO is the one who is held accountable for the accomplishment of the customer strategy goals created by the executive board. This is indicated by the symbol showing an outstretched hand. Subsequently, it can be interpreted that the COO and the CIO are the ones held accountable for the accomplishment of goals related to process improvement & change management respectively infrastructure management. It is illustrated that goals that relate to the infrastructure management process type also relate to the customer management information system. The model also shows that the CEO is not only held accountable for goal accomplishment as the CEO is also authorized to initiate creation of goals related to the five previously mentioned products that are offered by the bank. These products are visualized by means of the shopping cart symbol. The heads of the units ‘business customers’, ‘corporate customers’, and ‘retail customers’ are held accountable for accomplishing these goals.

The goal planning diagram provides support for the executive board and for executive committees on lower hierarchical levels to determine which employees or organizational units have the authority to create goals and which employees or units are appointed to accomplish created goals. Although details of the goals themselves are not shown, it is depicted to which business process types or products the created goals are related. This information should support the executive board in clarifying which employees or units create goals in what way and for what reason and to involve those employees that have to create and / or accomplish goals.
Figure 3: An example GoalML goal planning diagram of a mid-sized bank, based on [23].
3.2. Scenario II: Goal System Planning

After dealing with goal planning and goal authorization relationships, the creation of a goal system is considered in this scenario. As a first step, those who have the authority to create goals discuss with those who are appointed to accomplish goals exactly which goals should be included in a goal system. The executive board desires that at least one goal is created for each of the departments as is shown in figure 3. As a second step, a group of employees needs to relate each of the goals with each other once they are created and integrate them in one goal system. There are different kinds of relationships between goals that are possible, such as: Means-end relationships, logical interdependencies, decompositions and causal relationships. Next to relating goals, the relationships between goals and organizational units or other contextual elements need to be clarified. All possible relationships between the goals themselves can be illustrated by means of a GoalML goal system diagram.

Figure 4 is an example of such a goal system diagram. The diagram includes both so-called engagement goals and symbolic goals. An engagement goal is a goal of which the desired result is quantifiable, for example, the goal ‘decrease risk costs’ is an engagement goal, as it is quantifiable whether these costs have indeed decreased. Motivations and the performance of employees responsible for goal achievement can also be made more explicit by means of such engagement goals. An engagement goal is visualized as a target. A symbolic goal is a goal of which the desired result is not directly quantifiable and includes a qualitative aspect. A symbolic goal is visualized as a lighthouse. An example of a symbolic goal is ‘increase customer satisfaction’, as the increase of customer satisfaction is not directly quantifiable. A symbolic goal ‘increase bank appraisal’ is shown on top of the diagram. The star symbol with the number one inside the star shows this specific goal has the highest priority. The circles shown on the top right of each of the goals depict specific goal matter, which further specifies the goal content. A yellow hexagon with a plus symbol in it is part of the goal matter of the mentioned symbolic goal. This shows that something needs to increase upon achievement of the goal, in this case the bank appraisal. On the contrary, a yellow hexagon with a minus symbol indicates that something needs to decrease. The symbol of an eye looking at a diamond as part of the goal matter of the symbolic goal shows that the goal content is qualitative in nature. When further interpreting the diagram, it can be determined that two other symbols can be part of the goal matter, which are the indicator symbol and the ‘object’ symbol.
Figure 4: An example GoalML goal system diagram of a mid-sized bank, based on [23].
For example, the goal matter of the ‘decrease IT costs’ contains an indicator symbol, expressing that the goal content is of a quantitative nature. The ‘object’ symbol is illustrated by means of a combination of a circle, a triangle, and a rectangle. This symbol is used to indicate that an explicit ‘object’ is part of the goal content. For example, the ‘object’ symbol in the goal matter of the ‘decrease cumulated overtime hours’ goal indicates that an ‘overtime hour’ is a specific object to take into account as part of the goal content.

The different relationships that are found in the diagram need to be explained as well, next to the two different kinds of goals themselves and the goal content. Four kinds of relationships are found: A causal relationship, a means-end relationship, a mathematical relationship, and a decomposition relationship. The causal relationships are indicated by means of domino pieces, together with an arrow that points in the upward direction indicating a positive causal relationship. For example, the goal ‘increase quality of banking talent’ has a positive causal effect on the ‘increase bank appraisal’ goal. The means-end relationships are combined with the causal (and other) relationships shown in the diagram. For example, the goals ‘increase profit’, ‘increase customer value’, and ‘increase quality of banking event’ are a means to support in reaching another final goal, which is the ‘increase bank appraisal’ goal in this case. The logical AND-decomposition relationship in the diagram shows that the ‘decrease operating costs’ goal is actually built up of the two sub goals ‘keep labor costs stable’ and ‘decrease IT costs’. Finally, the positive mathematical relationship between, for example, the goals ‘increase revenue’ and ‘increase profit’ indicates that increasing the revenue has a positive mathematical effect on the actual profit itself, thus positively influencing the profit that is made by the bank.

In addition to the goal system diagram, a goal organizational structure diagram is shown in figure 5 that indicates the coupling between organizational units and organizational roles, and the respective goal instances. The shown goal instances show additional instance-level information. The handshaking symbols show that, for example, the heads of the different organizational units are held accountable for goal accomplishment and will most probably successfully achieve goal accomplishment. On the contrary, an outstretched hand indicates that someone who is held accountable for goal accomplishment will probably not achieve goal accomplishment in the current situation. This is seen, for example, in the case of the CFO who has to satisfy the risk cost goal of €40M. The ‘under construction’ symbol indicated on top of this goal instance expresses that more work has to be done in order to reach a sit-
Figure 5: An example GoalML goal organizational structure diagram of a mid-sized bank, based on [23].
uation where the accountability of the CFO accomplishing the goal will be high. Note that the outstretched hand symbol has a different meaning in the goal planning diagram of figure 3, where this symbol is used to indicate who is held accountable for goal accomplishment. The mentioned €40M is a concrete value that has to be satisfied in the case of the ‘risk cost’ goal instance. What exactly the form of these values is, is also supported by a small illustration in the form of a white arrow crossing a vertical line. This indicates that the shown value has to be satisfied in order to accomplish the goal. A circle with a vertical line such as is shown at the ‘keep labor costs stable’ goal instance indicates that the value of €90M should remain the same during a certain time period. For this goal instance, this time period is indicated as the reference period Q2 2012. A due date is also shown, which indicates when a goal instance should be accomplished. This is the date 2012-06-30 in this case. When analyzing the issues and resulting illustrative goal models as part of the described scenarios, there are five groups of specific requirements for a multi-perspective goal modelling language that can be identified.

4. Specific Requirements

The requirements that are described in this section are the result of applying the ‘derive requirements’ micro-level phase that is abstracted by the ‘analysis of specific requirements’ macro-level phase for designing a DSML as shown in figures 1 and 2. The fifth and final micro-level phase is omitted as priorities for implementing the mentioned requirements are not specifically assigned. The requirements presented in this section have been categorized in order to structure the list of requirements and to improve the presentation of the list. To order requirements within a category, those requirements are mentioned first which surfaced after studying the application domain itself and, subsequently, those requirements are mentioned that surfaced after interpreting the diagrams as part of the use scenarios that have been discussed in section 3.

Firstly, goals need to be related to a context as the goal planning diagram and the goal organizational structure diagrams illustrate. In the goal planning scenario for the bank, for example, the goal planning diagram includes goal-related concepts and concepts related to other organizational elements such as business process types and elements from the organizational chart, such as roles enacted by employees in the company. The questions of the executive board, such as determining how to involve employees in goal creation
and accomplishment or to which specific products the goals are related demand a view on goal modelling from multiple perspectives. Therefore, a goal modelling language should support the ability to relate goals to, for example, contextual elements like stakeholders (including employees), organizational units, products, processes, IT concepts, measures, projects, performance indicators, costs, and risks. The following requirements are formulated that are related to this need for context:

Integration with other EM languages (RC1) – A goal modelling language should be integrated with other modelling languages or it should at least be able to integrate with other languages. Of particular importance is the integration with modelling languages for the modelling of organizational structures, business processes, and strategy assessments.

Goal creatorship clarity (RC2) – It should be possible to express who is / are the creator(s) of a goal. These can be employees or other stakeholders.

Clarity of goal accomplishment responsibility (RC3) – It should be possible to express which organizational role (that is enacted by an employee) is responsible for accomplishment of a goal. Goal accomplishment is considered to be related to the commitment of a responsible employee to accomplish a goal, where a low commitment will increase the risk that an employee is unable to accomplish a goal while a high commitment decreases this risk.

Possibility to relate contextual elements to goals (RC4) – It should not only be possible to relate typical organizational elements like organizational units, processes, or resources to goals, but also other contextual elements like measures, performance indicators, costs, and risks.

Secondly, those who are involved in goal creation must be supported in the design and analysis of the goal models themselves. For this purpose, an adaptable goal modelling process model should be developed. On the highest abstraction level, this process model should contain phases that after completion lead to: A modelled goal planning diagram, modelled individual goals, a modelled goal system, an analysis of the created goal system, and eventually the application of the goal system. Each of the phases that lead to these results will then need to be detailed in sub phases for completeness.
sake. Although creation of a process model would be in the scope of our overall research in the context of multi-perspective goal modelling, it is out of the scope of this paper. One overall requirement can be distilled that would be satisfied by developing such a process model:

**Adaptable process model (RP1)** – An adaptable process model should provide support for designing and analyzing goal models. This process model should provide support in goal planning, modelling individual goals, modelling and analyzing a goal system, and applying this system in an organization.

A solution to satisfy this particular requirement is part of future research and is mentioned in section 7.

Thirdly, a conceptual reconstruction of the goal concept and adjacent concepts is called for. In the scenarios, it has been mentioned that ambiguous formulation of goals have led to improper communication of goals, completely different understandings of the goal concept and of the functioning of the goals. Besides this issue, there are concepts other than the goal concept itself that can be thought of which are actually tightly related to the goal concept. This requires to shape the conceptual foundation of goal modelling to introduce and reconstruct key goal modelling concepts. The following requirements are verbalized to address this issue:

**Unambiguous representation of goal concept (RR1)** – A goal modelling language should contain an unambiguous representation of the goal concept itself and of those concepts that are closely related to this central concept, including synonyms of the goal concept.

**Unambiguous representation of goal functions (RR2)** – A goal modelling language should provide support to make the functioning of goals explicit. As has been mentioned in section 1 already, a goal function is viewed as the achieved effect by accomplishing a goal and these effects can be different in nature. The following functions that goals may have are distinguished: Reaching a decision (see, e.g. [2]), increasing focus (see, e.g. [3, 4]), improve coordination (see, e.g. [5]), acquiring information (see, e.g. [4, 14]), increasing performance (see, e.g. [25]), evaluating performance (see, e.g. [4, 14]), achieving legitimation (see, e.g. [4]), and the ability to steer the enterprise (see, e.g. [25]).
Fourthly, various levels of abstraction can be applied when designing a goal model. This also relates to the question what concepts should be defined in the language itself and what concepts should be defined with the language. There can be goal types and goal instances of which examples have been shown in the diagrams of both use scenarios. Next to these two levels of abstraction, the meta level can be found one level above the type level that includes the goal meta types. This consideration, however, leads to thinking about how a goal instance, type, and meta type can be properly modelled and how goals that range from various levels of abstraction can be used together in certain goal models where possible. The following requirements surface that are related to the support for different levels of abstraction:

**Multiple abstraction levels (RA1)**—A goal modelling language should enable the modelling of goals on different levels of abstraction. This will make it possible to model goals on a level of abstraction that is required by the modeller without being restricted by a limited number of abstraction levels. This will make it possible to model goals as instances, types, meta types, or even as meta meta types and the modelled goals themselves can be combined in goal models on different levels of abstraction. The possibility to model goals on different levels of abstractions will also promote reuse of goal models.

**Enrichment of type-level models with instance information (RA2)**—It should be possible to enrich models that contain goal types with goal instance information. For example, on the type level a goal can have a deadline before it should be accomplished. Instance-level information will then show how many instances of such a goal type were indeed accomplished before this deadline.

Fifthly, goals are abstract in nature and, therefore, creating a concrete syntax of a goal modelling language is challenging. A concrete syntax contains the visual symbols used for constructing models, which are in this case goal models. The conceptual reconstruction of key goal modelling concepts probably requires the addition of new complex constructs to the concrete syntax of goal modelling languages. Composition of these constructs from widely independent icons is then called for, in line with principles applied to other modelling languages within an enterprise modelling framework. This topic is elaborated in section 5.4. A particular complex construct that should be part of the concrete syntax is a construct with which it is possible to express
goal achievement satisfaction. The accomplishment of a goal is considered to be closely related to the level of satisfaction of the stakeholders of a goal (see, e.g. [26]). These include employees in general and more specifically project managers, project sponsors, team leaders and many more that should somehow be supported to express their level of satisfaction with respect to goal achievement. Stakeholders might have completely different opinions whether a goal is considered to be accomplished. Take, for instance, the goal ‘reduction of overall costs’ and the sub goal ‘reduction of labour costs’. Achieving the latter by cutting salaries may delight stockholders, however, employees will probably come to a clearly different assessment. Goal achievement satisfaction of goal stakeholders can be determined once a scale of goal achievement satisfaction and degrees of satisfaction are determined. The following requirements make explicit what at least should be included in a concrete syntax for a goal modelling language:

Goal achievement satisfaction (RV1) – Stakeholders of a goal should be supported in expressing to what extent they are satisfied with goal achievement. Therefore, degrees of goal achievement satisfaction should be mapped on a scale of satisfaction.

Goal achievement specification (RV2) – Expressing whether a goal has been accomplished is not identical for each and every goal. Therefore, a goal modelling language should support whether the accomplishment of a goal is expressed by means of a final result that should be reached, whether a certain final result should be maintained, or whether a gradual improvement has been realized.

Visualization of priorities (RV3) – It should be possible to visualize the priorities of goals in a goal model.

Other requirements related to a concrete syntax can be explained in a similar way as the latter one, however, these relate to other symbols: Visualization of goal decompositions (RV4), visualization of goal interdependencies (RV5), visualization of causalities between goals (RV6), visualization of means-end relationships (RV7), and visualization of goal similarities (RV8). The explanations of these requirements is of a form that is identical to the explanation of requirement RV3 and are, therefore, omitted. With the descriptions of specific requirements for a multi-perspective goal modelling language, an outline of a solution is called for.
5. Language Design

The possible solutions that are presented are differentiated for the sets of requirements as introduced in section 4. The presented solutions in this section are the result of applying the phases ‘language specification’ and ‘design of graphical notation’ as part of the process model for designing a DSML.

5.1. Background: Multi-Perspective Enterprise Modelling

The integration of a goal modelling language with other enterprise modelling languages (requirement RC1) is possible when an enterprise modelling language architecture is available that contains a meta meta model. This meta meta model is required to specify the abstract syntax and semantics of a meta modelling language (MML) [27]. The method for Multi-perspective Enterprise Modelling (MEMO) [18, 16] is a suitable approach to enterprise modelling that serves this need. Figure 6 shows the MEMO language architecture. With the MEMO-MML, an extensible set of DSMLs can be specified of which the GoalML is one. In concrete, this means that a meta meta modelling language serves to develop new modelling languages. A DSML
includes a meta model which specifies the abstract syntax and semantics of the domain-specific language. Eventually, models can be created by instantiating this meta model. A goal modelling language and a business process modelling language that can be used to design goal models or business process models for specific domains are examples of DSMLs. The GoalML and OrgML are two examples of concrete DSMLs that are part of the MEMO approach. When interpreting the goal planning diagram shown in figure 3, elements from different DSMLs are combined in one diagram. An example of mutual enrichment that follows from this integration is the possible solution for satisfying requirements RC2 and RC3. In a goal planning diagram (a GoalML diagram) it can be shown which role (an OrgML concept) as part of an organizational chart (an OrgML diagram) is the creator of a goal (a GoalML concept), and which role is responsible for goal accomplishment (also a GoalML concept). The fact that all MEMO-DSMLs are based on one MML enables the integration of these languages which leads to possibilities in creating models that contain elements from multiple DSMLs. This implies contextual elements as part of other DSMLs can be related to goals and goal-related concepts as part of the GoalML when designing goal models (requirement RC4).

5.2. Conceptual Reconstruction

Requirements RR1 and RR2 were verbalized to address the challenge to unambiguously represent the goal concept itself and goal functions in a goal modelling language. The central aspects of what the goal concept encompasses are summarized as follows: A goal is a description of a state (see, e.g. [28, 29]) or it is a description of a future state which is compared to a current state [29]. This future state should reflect a positive valence [15]. A goal is an indication to perform certain actions, i.e., it is a call for action (see, e.g. [28, 29]). Lastly, a goal reflects the value premises of its creator [28]. These central aspects are explicitly incorporated in the GoalML meta model shown in figure 7. Although many other conceptualizations of the term goal are possible, these central aspects seem most appropriate for the discussed use scenarios. The ‘AbstractGoal’ meta type includes an attribute ‘state-OfType’ and an attribute ‘stateOfInstance’ to reflect the steps in the life cycle of a goal. The attribute ‘resultDescription’ reflects what result should be reached. The meta type ‘TargetSituationRelation’ shows that goal accomplishment should reflect a positive valence. The ‘UnitOfWork’ meta type and directly related meta types ‘GoalAuthorizationRelation’ and ‘InitiationRela-
tion’ indicate the creator(s) of a goal (requirement RC2). The entity or entities responsible for goal accomplishment are indicated by the meta type ‘AccountabilityRelation’ (requirement RC3). Full descriptions of the specific goal functions that can be distinguished (requirement RR2) are found in [17]. In summary, the functions of goals vary from: Decision making (see, e.g. [2]), focussing (see, e.g. [3, 4]), coordination (see, e.g. [5]), performance evaluation (see, e.g. [4, 14]), realization of increased performance (see, e.g. [25]), provision of information (see, e.g. [4, 14]), showing legitimation (see, e.g. [4]), and the steering of individual workers or enterprises (see, e.g. [25]).

5.3. Support for Different Levels of Abstraction

Besides the integration of elements from DSMLs such as has been mentioned at the beginning of this section, it is also possible to integrate elements that are specified on different levels of abstraction (requirement RA1). I.e., the integration of elements from a model and a meta model in one final model. A clear example of integration of elements on different abstraction levels is found in the GoalML meta model. This meta model is partially shown in figure 7, the full version can be found in [17]. Some classes in the meta model are taken from other MEMO-DSMLs. The classes ‘UnitOfWork’ and ‘BusinessProcess’ are borrowed from the OrgML [24, 30], the classes ‘Resource’ and ‘ResourceAllocation’ are from the ResourceML [20], and the class ‘InformationSystem’ is from the ITML [21]. Logically, instances of the meta types in the meta model are types. Sometimes, this results in the problem that the essence of a concept includes features that do not apply directly to the type level but apply to the instance level instead [27]. To overcome this problem, the meta model incorporates intrinsic features [27]. Intrinsic features are elements in the meta model that can be instantiated only with the instances of its instances. I.e., an intrinsic feature in a meta model is not instantiated on the type level but on the instance level. In the context of goal modelling, this implies that an instance of an intrinsic goal meta type is not a goal type but a goal instance. For example, the GoalML meta model includes an abstract super type ‘AbstractGoal’ that has relationships with two subtypes, which are ‘EngagementGoal’ and ‘SymbolicGoal’.

The use of intrinsic features in the GoalML meta model is exemplified as follows. The engagement goal meta type as part of the GoalML meta model contains four attributes and they are all intrinsic in nature. These attributes are: Due date, reference period, fulfillment period, and evaluation date. The due date and reference period have already been explained in
Figure 7: Partial GoalML meta model, based on [17].
section 3.2. The fulfillment period indicates between which start date and end date a goal should be accomplished. The evaluation date indicates on what date the accomplishment of the goal can be evaluated. Instantiating the engagement goal meta type will result in an engagement goal type. Such a goal type can then be applied in a goal model. As the four mentioned attributes are intrinsic, they are not instantiated for this goal type. This implies that the due date, reference period, fulfillment period, and evaluation date are not yet known on the type level. This is indeed desired, as this will only need to become clear once the goal type is instantiated. An example of such an instance might be ‘keep labour costs stable for the department in North Rhine-Westphalia, Germany with €12M’, which then might include a concrete due date, reference period, fulfillment period, and evaluation date. In one goal model, it is possible to include both goal types and instances (requirement RA2). For example, instances of sustainable sales goals are included next to a goal type that just indicates to keep the labour costs stable.

Finally, note that in the partial GoalML meta model as shown in figure 7 a boxed letter ‘d’ is found that is related to some attributes. This means that the resulting values of these attributes are found by deriving them from other elements in the model. For example, the attribute ‘avgAchievement-Expectation’ is a derivable attribute as part of the ‘AbstractGoal’ class. In this case, the resulting value for this attribute is found by taking the average of all ‘achievementExpectation’ outcomes of goal instances. This attribute is also part of the ‘AbstractGoal’ class.

5.4. Concrete Syntax

Finally, a possible solution needs to be found for the challenge concerning the creation of visual symbols to construct goal models for which requirements RV1 up to and including RV8 have been devised. A possible solution for this challenge also further contributes to satisfying requirements RC2, RC3, and RC4. Requirement RV1 concerns the mapping of degrees of goal achievement satisfaction on a scale of satisfaction. The concrete syntax of GoalML contains symbols that altogether contribute to satisfying requirement RV1. These specific symbols are visualized in figure 8 and are explained hereafter. In the concrete syntax, a refinement of the goal achievement satisfaction concept is made. On the level of a goal instance, the achievement expectation can be expressed as being bad, medium, or good. This is visualized by means of a thumbs-down, thumbs-neutral, respectively thumbs-up
Figure 8: Symbols used for illustrating goal achievement.

symbol. For a goal type, the average achievement expectation is indicated in the same way, based on the achievement expectation of its instances. Furthermore, the concrete syntax contains symbols to indicate whether the value development in order to reach a desired value is stable, if it is going up, or if it is going down. It can also be shown whether instances of a goal type have been achieved or not in the past, whether some goal instance itself has been achieved or not, and whether the result after trying to achieve a goal exceeded a desired value or was insufficient. Finally, the actual impression to what extent someone is satisfied with overall goal achievement can be formed after interpreting a goal model containing these kind of details. To satisfy requirement RV2, the concrete syntax of GoalML contains symbols to specify whether the accomplishment of a goal means that a certain value should be ‘en-bloc’ (e.g., ensure liquidity), satisfied (i.e., a value that is minimally reached which can be ‘reach a turnover of at least 5%’), exactly reached (e.g., reach a turnover of exactly 5%), maintained during a certain time period (e.g., maintain the current turnover), improved to a certain extent (e.g., increase turnover), or extremized (e.g., maximize turnover). These symbols are also shown in figure 8.

Moody [31] proposes guidelines for constructing visual notations in software engineering, which also applies to the creation of visual symbols for a goal modelling language. Moody describes nine principles for designing cognitively effective visual notations, which are: Semiotic clarity, perceptual discriminability, semantic clarity, complexity management, cognitive integration, visual expressiveness, dual coding, graphic economy, and cognitive fit. While Moody’s principles provide a valuable orientation, they are in part not convincing (see: [19]). Therefore, they were partially modified and extended
in [19]. Due to our experience with developing multiple DSMLs, it is a key success factor to collaborate with a graphics designer who is experienced in developing well-designed icons to visualize complex constructs that can be easily recognized [32]. Therefore, a graphics designer was involved in the process to create the concrete syntax for the GoalML. Nevertheless, the design of the graphical notation was a considerable challenge. First, visualizing abstract concepts such as ‘EngagementGoal’ or ‘SymbolicGoal’ has to deal with the obstacle that there are no physical objects that could serve as an orientation for creating respective symbols. To cope with this problem, the designer made use of metaphors in order to account for the principle of semantic clarity, which means that the appearance of the visual representations used suggest their meaning [31]. For example: The target symbol illustrates that in order to accomplish an engagement goal you need to ‘hit’ a target as an engagement goal can be quantitatively expressed. The symbolic goal that is depicted as a lighthouse illustrates that in order to accomplish such a goal you need to steer in a certain direction, as the accomplishment of a symbolic goal is qualitative in nature and cannot be measured by means of a hard numerical value. Second, the conceptual richness of the GoalML allows for defining very elaborate goal systems, i.e., that include a lot of detail. To make use of this conceptual richness, the concrete syntax has to provide a corresponding amount of symbols. As a consequence, diagrams can reach a level of confusing complexity. The GoalML includes mechanisms to reduce visual complexity, e.g., by presenting high-level goals, which can be decomposed into subgoals on demand. To take full advantage of the GoalML’s expressiveness without creating diagrams that lack comprehensibility, it is required to use modelling tools that allow fading certain aspects of a diagram in and out on demand. Furthermore, it is conceivable to create different ‘visual dialects’, e.g., for teaching purposes on the one hand and experienced modellers on the other hand. More details on this topic are discussed in section 7.

6. Related Work and Evaluation

The research fields of Requirements Engineering (RE) and Enterprise Modelling (EM) are particular fields that have brought forward various kinds of goal modelling languages. In this section, a selection of popular or pioneering goal modelling approaches that are part of these research fields are discussed. Subsequently, a comparative evaluation of the GoalML is presented.
6.1. State of the Art in Goal Modelling

In the context of RE in general, goal modelling approaches have been developed to derive requirements for information systems from the goals that stakeholders of such systems have with respect to the system that is to be built. As is described in [33], goal modelling approaches are also applied in sub disciplines of requirements engineering, such as the engineering of requirements for secure information systems, for example. In the context of EM, goal modelling languages serve the purpose to model organizational goals and integrate them with or relate them to organizational structures and processes. The following characteristics have been taken into account for every approach that is discussed in this section: The particularities for the modelling of the goals themselves as part of an approach, the elements and properties of goals that can be modelled, relationships of goals with other concepts that can be modelled, relationships between goals that can be modelled, and what functions the goals have once they are modelled. As such, this section is the result of applying the final phase of the process model for designing a DSML, which is called ‘evaluation and refinement’. In this case, the focus is on evaluating the state of the art in goal modelling against the aforementioned characteristics. Note that through this section it is intended to explicate which approach offers which particular strengths and solutions in the area of goal modelling. In case some approaches possess less of some characteristics it should not be interpreted as negative critique on the approaches that are discussed.

The Goal Question Metric (GQM) [34] approach exists for almost 30 years and is a pioneering goal modelling approach in the context of RE. GQM is a method for creating goal-oriented project evaluations in software engineering to improve the quality of future projects. The method consists of three phases [34, 35, 36]. First, goals have to be created for each project evaluation. Second, questions are derived from the goals in order to understand how to accomplish the goals. Third, metrics are created to answer the questions in a formal manner. The goals that are made in the first phase are operationalized by means of the questions and made quantifiable by means of the metrics [35]. The GQM approach does not consider different abstraction levels of goals, nor does it include a possibility to integrate with other modelling languages. With GQM, the goal content is further explicated by determining the purpose, focus, and issues a goal is related to [36]. Goal priorities are also mentioned [35], however, it is not mentioned how they can be illustrated. Products, processes and resources are considered as concepts
to which goals can be related [36], although it is formally not possible to distinguish between different kinds of resources. GQM includes the possibility to develop a ‘What-Planning’ and a ‘How-Planning’ and, therefore, enables to differentiate between a means and an end [35]. Illustrating these or other kinds of relationships is not possible with GQM. As GQM is oriented towards measurability [36], the functions of goals that are modelled with GQM are increase of focus and performance evaluation.

Another well-known approach is called ‘Keep All Objectives Satisfied’ (KAOS) [37], which has originally been coined as Knowledge Acquisition in autOmated Specification [38] and has been developed for acquisition of software system requirements [38]. In KAOS, hard, i.e., formalizable goals and soft goals are differentiated. The abstract syntax of KAOS is described by means of a meta model. The three different levels of abstraction are part of the approach, which are the instance, type, and meta type levels [38]. Integration with other modelling languages and the possibility to show instance-level information in goal models that include goals on the type level is not possible. The actual goal content can be described textually. Furthermore, temporal constraints are discussed but no possibilities are given how to concretely model this [38]. Goals can also be related to other concepts like an event, an agent, or a constraint, however, these concepts are limited to what is provided by the KAOS language. When it comes to the modelling of relationships between goals it can be concluded that goal priorities, decompositions of goals, and means-end relationships can be modelled. Four goal functions can be expressed with KAOS, which are: Focussing, legitimation, provisioning of information, and coordination [38].

Another widespread goal modelling approach in RE is ‘i*’, see e.g. [39, 40, 41]. i* is part of the ‘ITU-T Recommendation Z.151’ [42] international standard. This approach includes social aspects in the development of a software system [40]. Hard goals and soft goals can be modelled, as well as actors, resources, and tasks that need to be fulfilled in order to accomplish goals. Dependencies between actors (goal dependency, task dependency, and resource dependency) can be modelled in a so-called strategic dependency model (SDM) which reflects the level of autonomy that an actor has [40]. The strategic rationale model (SRM) is another kind of model that is offered by i*, with which it is possible to distinguish between tasks and goals by means of means-end relationships, to model decomposition relationships for tasks and goals, and to model cause-and-effect relationships between tasks and goals [40]. Goal priorities cannot be modelled, though. A big difference
compared to other approaches is that a goal model is modelled from the perspective of a single actor, where with other approaches usually the whole system is taken as perspective. Although i* differentiates between the three general levels of abstraction as well as a meta meta level, the semantics of modelling goals on these different levels are not yet further specified. Integration with other languages will be difficult as the meta meta level is described rudimentary [40]. Goals can be specified by means of brief textual fragments, however, other than the modelling of resources related concepts cannot be modelled. Goals modelled with i* can be used for purposes of legitimating the software requirements and decision making in software engineering. Other functions are not considered.

Tropos [43, 44] is a popular agent-oriented goal modelling approach related to all phases in software engineering which is closely related to i*. Concepts are borrowed from i* and applied not only for goal modelling in the context of RE but also in the context of the actual software design and implementation [44]. Tropos is specified by means of a meta model and includes the i* concepts of actor, hard goal, soft goal, resource, dependency, and belief. Non-i* concepts are ‘plan’ and ‘capability’ [44]. With Tropos, actors and their dependencies can be modelled next to goals. In addition, plans can be modelled that express how to accomplish goals and the capabilities of actors to accomplish goals can be modelled as well [44]. The SDM and SRM from i* are also included in Tropos. Tropos explicitly differentiates between different levels of abstraction and the semantics of concepts on these levels are described as well. Integration with other languages is possible by means of a Tropos meta model and meta meta model [44]. Furthermore, Tropos does not differ much from i* when it comes to elements and properties of goals that can be modelled, the relationships of goals with other concepts that can be modelled, the relationships between goals that can be modelled, and what functions the goals have once they are modelled. For example, Tropos includes the possibility to model means-end relationships, decomposition relationships, and cause-and-effect relationships between goals but the modelling of goal priorities is not possible.

The Architecture of Integrated Information Systems (ARIS) [45] is an enterprise modelling approach existing more than two decades which became a popular EM approach in practice [46]. The so-called ‘ARIS-House’ is at the core of this approach, which underlines the importance to view organizational processes from five different perspectives, which are the following: Organizational view, data view, control view, functional view, and product
or service view. Goals in ARIS serve the purpose to control functions and are, therefore, considered from a functional viewpoint and a control viewpoint. ARIS is specified by means of a meta model, which implies that the approach takes into account different levels of abstraction. The elements and properties of goals that can be modelled in ARIS can be described in an informal way only. Goals can be related to functions and organizational units although it is not possible to model goal creatorship. Although ARIS includes concepts to model human and computer-based resources, it is not possible to relate resources to goals yet. ARIS supports the modelling of means-end relationships between goals.

Enterprise Knowledge Development (EKD) [47, 48] is an enterprise modelling approach that includes a well-elaborated goal concept. For example, it is possible to model goal properties such as criticality, priority and individual importance, where it is possible to depict the first two properties on an ordinal scale that includes the values ‘low’, ‘medium’, and ‘high’ [48]. Other concepts that are offered by EKD includes the ‘problem’ concept, with sub types ‘weakness’ and ‘threat’ as well as concepts for ‘opportunity’, ‘cause’, and ‘constraint’ [48]. The process model that is part of the EKD approach describes two core activities for the creation of goal models, which are the identification of goals and the operationalization of goals. With EKD it is possible to integrate with other modelling languages, for example, with a language for the modelling of processes and organizational structures [48]. Potential future goals can be modelled as well next to the modelling of current goals in order to increase the understanding how to adapt the overall goal system to reach potential future goals. It is not possible to model goals on different levels of abstraction, though. Temporal parameters are not supported by EKD, however, it can be expressed whether the result of a goal should ‘introduce’, ‘maintain’, ‘cease’ or ‘improve’ something. Preconditions can be shown by means of the ‘constraint’ concept and the ‘assumption’ concept can be used to substantiate the creation of goals. With EKD it is possible to model who is the creator of a goal and who is responsible for goal accomplishment [48]. Goal decomposition relationships and goal interdependencies can be modelled as well. The interdependency relationships can be used to indicate whether something ‘supports’ a goal, ‘hinders’ a goal, or ‘conflicts’ with a goal. Finally, it is determined that the many different kinds of functions that goals might have are not clearly considered as the primary purpose of modelling goals in EKD is to legitimate other modelled objects in enterprise models [48].
6.2. Evaluation

The evaluation of a language in general is a challenging task. Blindly relying on assessments of all types of stakeholders would not be a good idea, as it is important to account for the differences in backgrounds, such as experiences with similar or other modeling languages, their formal education, the time they had to learn the language, and their attitude towards DSML in general, for example. However, the background of language designers might be different compared to other types of stakeholders and this should enable them to come up with more elaborate assessments. This might have as a disadvantage that subtle bias can creep in their judgment. Although the language we know is a necessary instrument, it limits our reasoning powers at the same time because we cannot go beyond our language completely. A different language that lies beyond our imagination would perhaps provide for a more sophisticated structuring of the problem, however, the problem itself might appear different at the same time if it is described in a different language. Although these thoughts can be considered to be of a philosophical nature, they are important for the reflective evaluation of a DSML. These considerations make us realize that the effect of existing languages skills is limited and they emphasize the contingency of language skills and language use. In other words, a particular DSML is a moving target and as a consequence its assessment should not be restricted to a certain point in time, but rather be organized as a permanent process [19]. Following the method presented in [19], an extensive evaluation of the GoalML has been performed in [17]. It is mainly based on a thorough analysis of how the GoalML is suited to satisfy the requirements. A respective assessment may be obvious in some cases. In other cases, the method recommends to clearly state the fallible hypotheses an assessment in based on, indicating the need for further research. With respect to the extent of the evaluation performed in [17], this paper includes a summary of the evaluation results only. To illustrate how the GoalML contributes to advancing the field, it is compared against the goal modelling approaches that have been discussed in this section.

Table 1 shows an overview of the goal modelling approaches that have been discussed in this section. The outcomes of the evaluation against the five criteria are visualized in table 1. Based on the solution as presented in section 5 it can be derived from the table that the GoalML possesses all but the last characteristic convincingly.
Table 1: Overview of evaluated goal modelling approaches.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Modelling particularities</th>
<th>Goal elements and properties</th>
<th>Relationships between goals and other concepts</th>
<th>Relationships between goals</th>
<th>Goal functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>GQM</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>KAOS</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>i*</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Tropos</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>ARIS</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>EKD</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>GoalML</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

Legend: ○ = unsatisfactory ● = partly ○ = very satisfactory.

7. Conclusions and Future Research

Developing a multi-perspective goal modelling language with the purpose to support the personnel of an enterprise in goal creation, goal achievement and, subsequently, in a successful implementation and realization of an enterprise-wide strategy comes with specific requirements that call for concrete solutions in order to be satisfied. By means of two use scenarios regarding goal modelling for a mid-sized bank it has been illustrated what these specific challenges are. In the first scenario, the executive board of the bank prepared the modelling of the actual goals themselves by planning who in the enterprise is responsible for creating and accomplishing goals, related to which organizational elements. In this first scenario, the actual goals themselves were not described, instead only the responsibilities and relationships of the to-be modelled goals with elements like business process types, company products, and departments became clear. In the second scenario, the executive board concretely modelled types of goals in order to form an overall goal system, which implies that relationships between the goals become clear and the content of the individual goals themselves becomes clear. Furthermore, the concrete instances of the previously modelled goal types were modelled and related to those roles enacted by employees that were identified in the first scenario as responsible for goal achievement. Five sets of requirements have been distilled from the two use scenarios. The first set of requirements concerns the consideration that goals need to be context-related, otherwise they do not make much sense. This does not only include clarifying relationships with other goals, but also positioning goals with re-
spect to related business process types, organizational roles, products, and so on. Therefore, it has been claimed that a goal modelling language should be *multi-perspective*. The second requirement has been considered to be out of scope for this paper, however, it became clear that employees should be supported in goal modelling and analysis and, therefore, an adaptable goal modelling process model should be at their disposal next to a goal modelling language. The third set of requirements expresses that a clear and unambiguous understanding of the goal concept and similar concepts is needed to foster clear communication between employees. Fourth, a goal modelling language should provide support for modelling on different levels of abstraction when modelling goals and a goal model. Fifth, and finally, it is considered challenging to create visual symbols used for constructing goal models, i.e., a concrete syntax as goals are abstract in nature.

An outline of a solution to satisfy the sets of requirements has been presented as well. The goal modelling language that we have developed and that would be able to offer a solution has been coined GoalML and is part of the Multi-perspective Enterprise Modelling (MEMO) approach. MEMO provides a meta meta model with which it is possible to specify an extensible set of domain-specific modelling languages (DSMLs), which includes GoalML. This enables the integration of GoalML with other enterprise modelling languages and the possibility to relate contextual elements that are part of other enterprise modelling languages to modelled goals. Besides the integration with other DSMLs, integration of GoalML constructs that are modelled on different levels of abstraction is possible. For example, goal models might contain goal types and goal instances, depending on how much detail should be included in the model. The GoalML meta model describes the language itself, including core goal-related aspects. All GoalML symbols that can be used to create goal models are part of the concrete syntax of GoalML. When it comes to illustrating goal achievement, the GoalML concrete syntax includes symbols to show the goal achievement expectation, what the status is of the value development, whether a result is achieved or not, whether previous goal instances that are related to some instance have been successfully achieved or not, and what exactly goal accomplishment means for a certain goal as this might differ for each goal type. When designing symbols for GoalML, Moody’s principles [31] have been taken into account and augmented with own principles (see [19]). Next to adhering to these principles, a graphics designer who is experienced in developing icons to visualize complex constructs that can be easily recognized has provided

33
support in creating the GoalML concrete syntax.

At least four paths for future research can be considered. The first one is related to the development of an adaptable goal modelling process model. Together with the GoalML, it will build the foundation of a method for multi-perspective goal modelling when considering that a modelling method consists of at least a modelling language and a process model. Most probably, this method will contain phases that should lead to: A modelled goal planning diagram, modelled individual goals, a modelled goal system, an analysis of the created goal system, and eventually the application of the goal system. For completeness sake, each of these phases will need to be detailed in sub phases. The second future research path of which the first results are described in [49] is related to Moody’s principle of cognitive fit [31]. This means that different visual dialects should be used as part of a modelling language for different audiences. An important task to perform in this context is to make GoalML suitable for use in different contexts such as teaching, instead of having a language that is very rich in detail which might be suitable for expert modellers but less suitable for more novice modellers. The third path for future research is related to the question how goal models can be integrated into software with the intention to provide different forms of computer-based support for, e.g., strategy formulation, goal achievement, and enterprise management in general. A possible form of computer-based support is to use the information presented in goal models for deductive purposes. For example, information from goal models provide a foundation for the generation of rules that need to be adhered to when conducting tasks or processes that are related to a goal. By adhering to these rules, it is possible to steer in a direction that would lead to goal achievement. Moreover, rules for goal achievement could be seen to be part of the proposed goal modelling process model. This third path for future research also relates to creating models at runtime and self-adaptive systems, which implies that a system adapts its structure, functions, or processes to a (manually) modified goal system. A self-adaptive system might also modify the goal system itself to better cope with a changing environment, or at least creates alternative goal systems that need to be approved by human experts. The fourth path of future research is motivated by the need for multiple abstraction levels in general (RA1), and multiple classification levels in particular. The existing MEMO language architecture (see figure 6) which follows the structure of the MOF, does not allow for metaclasses above $M_2$, which is too restrictive in some cases. To overcome this limitation, a new meta modelling language
has been developed. It is based on a recursive architecture and enables an arbitrary number of classification levels [50]. We plan to respify the GoalML as well as all other MEMO-DSMLs with the new meta modelling language as soon as it is mature enough.

References


[23] iteratec, iteraplan enterprise architecture management einfach und effektiv - online-demo, 2013. In German.


and Business Information Systems, University of Duisburg-Essen, Essen, Germany, EU, 2011.


39


Highlights

- Formulating and understanding how to achieve organizational goals is not elementary.
- Two use scenarios at a bank reflect goal modelling and analysis challenges.
- A language for designing goal models is proposed that is called GoalML.
- GoalML provides solutions for requirements that are derived from the use scenarios.
- Goals are related to their context by modelling them from multiple perspectives.