CONCEIVING A METHOD FOR VIEWPOINT-BASED MODELING USING RECOMMENDER SYSTEMS IN A MULTIPLE-USER ENVIRONMENT – CONCEPTUAL APPROACH AND PROOF-OF-CONCEPT

Complete Research

Krumeich, Julian, Institute for Information Systems (IWi), German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany, Julian.Krumeich@iwi.dfki.de
Werth, Dirk, Institute for Information Systems (IWi), German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany, Dirk.Werth@iwi.dfki.de
Loos, Peter, Institute for Information Systems (IWi), German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany, Peter.Loos@iwi.dfki.de

Abstract

This paper conceives a viewpoint-based modeling method that applies the concept of viewpoints to collaborative modeling to foster the incorporation of multiple stakeholders. In collaborative modeling settings, problems like low model acceptance among involved stakeholders are typical due to their limited understanding of the overall model or system to be developed. The conceived viewpoint-based modeling method aims at solving such problems by introducing and using stakeholder-specific viewpoints on collaboratively created models. In doing so, the viewpoint concept facilitates and improves the involvement of multiple stakeholders from different domains into the collaborative modeling process. To effectively distribute and coordinate modeling activities among all participants, the method utilizes the concept of recommender systems with the eventual goal to end up with a consolidated, conflict-free model that has been collectively constructed. Besides the development of the viewpoint-based modeling method, the paper at hand—following design science research—additionally presents its Proof-of-Concept by means of a prototypical implementation and an evaluation of the proposed recommender algorithm.

Keywords: Collaborative Modeling, Viewpoint Concept, Recommender Systems, Design Science.

1 Introduction

1.1 Motivation

In the course of developing any kind of system—e.g. engineering constructs or software products—models are a common means for gaining a better understanding of the underlying artifact to be developed. Within the Information Systems domain such models are typically information models like data models or process models. They can either represent natural originals or illustrate artificial ones, which can again be models by themselves (Stachowiak, 1973). Even though an important inherent feature of models is to provide a reduction of complexity of the underlying system, they still often tend to have a high level of complexity—both in term of creating and using them (Frank, 1999).
Consequently, the perceived value out of models is considerably limited compared to its original purpose. This effect is magnified if several individuals, e.g. stemming from different domains, are involved in the model creation process as well as in its usage (Kurpjuweit, 2007).

In the age of collective intelligence, this involvement of multiple stakeholders within the model creation process is increasingly demanded. Leading vendors of business process management suites started to incorporate collaborative and social aspects into their software tools, so that they “now provide social work interfaces that allow customers and employees to easily ‘follow’ a given process and quickly identify experts for a given business process or task” (Richardson and Miers, 2013). However, these features are foremost dedicated to publish processes to employees. Yet, the actual process of modeling is not adequately supported by the idea of collaborative interaction. To adequately integrate the literally crowd of stakeholders into modeling activities, all of them have to be considered individually.

1.2 Research contribution

Providing only a conventional single and rigid perspective on models hinders an effective working with them (Fischer et al., 2012). Hence, different perspectives on the development of the underlying system are essential calling for means to view and manipulate models depending on the specific objectives of the participating stakeholders (Dijkman et al., 2008). To support modeling with regard to these requirements, the viewpoint concept generally appears as a promising approach. It offers complexity reduction by breaking down underlying models to stakeholder-specific fragments (Goldschmidt et al., 2012). Hence, each individual actor is put into focus resulting in an “increased understanding and productivity (Finkelstein et al., 1992). This contributes to the development of better conceptual models, i.e. models depicting the underlying system more adequately, which has already been proven in studies (Easterbrook et al., 2005).

The growing need for collaboration, when multiple stakeholders are involved, requires appropriate coordination mechanisms, which are yet missing in research on viewpoint-based modeling (Fischer et al., 2012). Thus, this paper conceives a viewpoint-based modeling method which applies the concept of viewpoints to a collaborative modeling environment to foster the incorporation of multiple stakeholders. To effectively distribute and coordinate modeling activities among them, the method utilizes the concept of recommender systems with the eventual objective to end up with a consolidated, conflict-free model outcome that has been collectively constructed.

1.3 Research approach and paper structure

In this paper, a design science research approach is applied. According to Hevner et al. (2004), design science has to comply with seven guidelines to maintain rigorous and relevant research: The introduction section points out the relevance of the research problem (Guideline 2). The viewpoint-based modeling method developed in section 4 represents the design science artifact (Guideline 1), whose prototypical implementation in section 5 proofs its implementability (Guideline 4). To comply with guideline 3, evaluations on the designed artifact, in particular the recommender system, has been conducted (cf. section 5). To build upon existing knowledge (Guideline 6), the developed approach is based on previous related work, which again refers existing concepts (cf. sections 2 and 3). Guideline 5 is accomplished by outlining the applied research methodology. Last but not least, the submission of this paper aims at fulfilling Guideline 7, the dissemination of research results. First steps in this direction were taken by publishing initial results at the International Conference on Wirtschaftsinformatik and the Pacific Asia Conference on Information Systems (Krumeich et al., 2013a and 2013b).
Towards Viewpoint-based Modeling

Typically, two general groups of persons come into contact with a model: one creating it and another using it (cf. Werth, 2005). In the simplest form, both groups are represented by one single individual or multiple individuals working as a collective unit (cf. Figure 1, left). In reality, the creation and usage of models comprise multiple individuals not acting as a collective unit, since they all have different expertise, abilities, knowledge as well as responsibilities. That is why a collaborative model formation can be assumed in practice (cf. Figure 1, middle).

Figure 1. Different principles of model creation and usage

When talking about a set of people working together, the terminology regarding the type of participation has to be defined. In this context, especially the terms collective, cooperative and collaborative need to be delimited as they are frequently used interchangeably. Hence, the present paper considers collective as a generic term describing all processes where multiple people work together toward a common objective (Mangenot and Nissen, 2006). It thereby encompasses both the adjectives cooperative and collaborative, which represent two different degrees of structure. Cooperative work often consists of informal relationships and lacks a common mission and structure (Mattessich et al., 2001). Authority and resources are usually not shared. Collaborative work however is based on shared resources and shared responsibilities (Mattessich et al., 2001). The participants thereby “act as individual experts addressing design issues from their perspectives” (Kvan, 2000). A stronger commitment to a common goal and a better organization of the participation are also characteristics of collaborative work.

The involvement of multiple stakeholders entails several social and technical challenges. From a technical perspective, especially concurrency needs to be supported in modeling, i.e. the concurrent access and manipulation of models by different stakeholders. This requires synchronization mechanisms to propagate changes within models among involved stakeholders (Cicchetti et al., 2011). Besides technical issues, collaborative modeling should be considered particularly from a social perspective, since several individuals with different roles, knowledge etc. taking part in modeling the same artifact (for a detailed examination on social influences on collaborative modeling it is referred to Renger et al. (2008) and Mendling et al. (2012)). Particularly in synchronous group-modeling processes, the composition and role allocation of involved modelers have a significant effect on the modeling result (Renger et al., 2008). In literature, there are several roles discussed to facilitate the synchronous process of modeling. Richardson and Andersen (1995) propose five essential roles: the facilitator, the modeler/reflector, the process coach, the recorder and the gatekeeper. It has shown to be useful to externalize them for a better performance of the group, especially for large and complex modeling projects (Renger et al., 2008). Examining asynchronous group-modeling projects, other important social implications appear. Here, it is characteristic that conflicts—potentially resulting when stakeholder contribute their knowledge on a model instance—cannot be discussed or even resolved instantaneously. However, temporally tolerating them should not generally be considered as negative, but as a “necessary means for identifying aspects of systems which need further analysis or which need to reflect different viewpoints of different stakeholders” (Wieland et al., 2013). Hence, it is vital to involve the “right” stakeholders, which means those that are able to resolve conflicts by incorporating their specific
knowledge. “If wrong people are involved at the wrong moments in time”, a poor involvement of participants will be the result potentially risking a failure of the overall project (den Hengst and de Vreede, 2004).

To effectively support collaborative modeling in this regard, the modeling method developed in this paper follows viewpoint-based modeling and extends it through a recommender systems approach fostering the involvement of the “right” participants. In general, viewpoint-based modeling provides, for each of the participants involved, dedicated viewpoints onto models (cf. Figure 1, right). In the context of the viewpoint concept, participants are often called stakeholder (cf. Goldschmidt et al., 2012). Such a stakeholder-specific representation as well as means for adjustment allow an adequate adaptation of the modeling process to the particular needs of the involved stakeholder.

3 Related Work

3.1 Viewpoint concept and domain-specific model visualization

The origin of the viewpoint concept can be traced back to a work of Wood-Harper et al. in 1985 (Lankhorst, 2009). The MultiView approach presented therein aims at supporting the development process of computer-based information systems by dividing this complex process into five different perspectives resp. viewpoints. Another early paper in the area of viewpoint research is the frequently cited one of Finkelstein et al. (1992). Whereas Wood-Harper et al. (1985) primarily address the application of viewpoints within software engineering, the approach of Finkelstein et al. (1992) is potentially applicable to the development of any artifact that exceeds a trivial engineering process. From a standardization point of view, efforts have been done by IEEE as well as ISO/IEC standard organizations. In the course of this, the IEEE 1471 standard definition of the viewpoint concept has been incorporated into ISO/IEC 42010 standard in 2007 (Software Engineering Standards Committee of the IEEE Computer Society, 2007). The basic idea of different views for different roles and stakeholders during modeling can be found in a paper by Cera et al. (2004), who consider in particular CAD models as their information models. Each modeler sees parts of the model in varying resolutions, i.e. different levels of detail, to only highlight the important parts for his activities. Still, the motivation behind this approach is different from the one presented in this paper. Cera et al. only address information security to protect critical intellectual property while viewpoint-based modeling in general aims at facilitating each modeler’s work to reach overall models of higher quality. However, while the objectives differ, the techniques to achieve them are basically identical. One approach that mainly addresses the graphical aspect of modeling is proposed by Poppe et al. (2013). They claim that visual cues like avatars in a virtual environment would benefit synchronous process modeling and present a corresponding prototype based on 3D virtual world technology. In doing so, they especially focus communication-intense modeling, while the approach presented in this paper favors asynchronous modeling where each participant models those aspects s/he is most familiar with.

3.2 Interaction between viewpoints and contribution to system design

An early proposal for collaborative modeling can be found in the work of Pinkwart (2003). He developed a set of plug-ins for the modeling framework Cool Modes. In doing so, he aims at “bridging the gap between a communication means and a system with AI functionality” (Pinkwart, 2003). The framework is restricted to processes modeled as graphs. A number of rules for a model is defined and serves as the basis for the collaborative modeling. This approach does not seem suitable for complex modeling tasks and rather focuses on communication and feedback. At the same time, the claimed AI functionalities can be considered as rather limited. Andrade et al. (2004) propose a methodical framework that targets an improvement of the cooperation between individual viewpoints. In this context, the Hybrid Multi-View Modeling concept of Cicchetti et al. (2011) has to be included as well. This approach pursues the goal of achieving an efficient synchronization of different views in a modeling process out of a technical
perspective. That means, any change performed in a particular view of a viewpoint is recognized by certain techniques and propagated to the remaining views in an incremental way. In doing so, they aim at reducing the overhead of recalculations of the affected views—especially for larger models. Another approach is proposed by Kim et al. (2005). Their Fragment-Driven Process Modeling Methodology enables the collaborative development of a process model by multiple actors. Each actor models his or her own activities, which are subsequently merged into one model by the system. However, only a bottom-up approach is pursued herein, which is also termed fragment-driven approach. Accordingly, the process fragments are merely assembled in a process-oriented way. A collaborative modeling on the same process sequence resp. fragment is not addressed. One approach that focuses more on connecting distributed designed model parts is Bagheri and Ghorbani (2008). Its aim is to support modelers in merging and integrating different conceptual models. To regard the inevitable degree of uncertainty, specific uncertainty formalisms from the field of belief theory are included.

3.3 Tool support for viewpoint-based modeling

A stronger focus on modeling tools supporting the viewpoint concept can be found in the recent work of Goldschmidt et al. (2012). Therein, specifications for the actual realization of the viewpoint concept for the development of modeling tools are covered. Also Fischer et al. (2012) considered such requirements with regard to modeling tools. Besides scientific studies and tools, it is also important to consider commercial tools which are used in practice. A detailed study of process modeling tools and their support for collaborative modeling was performed by Riemer et al. (2011). They apply a set of collaboration criteria to the twelve most common process modeling tools including e.g. ARIS Design Platform, CA ERwin Process Modeler and IBM WebSphere Business Modeller. Their study reveals “surprisingly little support in existing modelling tools and only fragmented support for the various aspects of joint process modelling” (Riemer et al., 2011).

4 Developing a Viewpoint-based Modeling Method

4.1 Requirement analysis

Prior to this paper, an in-depth analysis of different viewpoint methods revealed drawbacks in utilizing the viewpoint-based modeling approach in modeling projects (cf. Fischer et al., 2012). It can be attested that no approach exists that combines the viewpoint concept—which supports the involvement of different stakeholders—with the field of collaborative modeling—which actually pursues the coordination of modeling among multiple stakeholders. Even though ideas for combining models from different viewpoints exist in research, collaborative aspects of the actual modeling activity cannot be found in the viewpoint research stream so far (Krumeich et al., 2013b).

4.1.1 Requirement 1: Domain-adequate representation

To achieve the inherent objectives of viewpoint-based modeling, a viewpoint definition that allows for a dynamic creation of viewpoints in modeling project has to be constructed. This should form the cornerstone of the approach. Existing methods or definitions are too rigid in this respect, i.e. they hinder the application of existing methods for an ad-hoc creation of viewpoints during a modeling project (Fischer et al., 2012). Hence, the viewpoint concept developed in this method should directly access the model instance without intermediate layers, which must be implemented or configured beforehand. In this regard, viewpoints should be developed as models that are self-contained and map onto metamodels. The realization of domain-specific representation through viewpoints foster the awareness of participants, which is essential for social interactions in collaborative modeling (Mendling et al., 2012).
4.1.2 Requirement 2: Modeling task recommendation

The first requirement directly calls for the second one in order to provide means for a dynamic distribution and coordination of modeling tasks among existing viewpoints. This means, the method must allow to set up and to integrate new viewpoints. While most of existing methods agree on the aim of assigning viewpoints to specific stakeholders, the actual assignment process, especially among large groups of stakeholders, has not been systematically examined (Krumeich et al., 2013b). However, the overall concept can only be successful if the numerous stakeholders can be effectively managed. Hence, the approach under development should provide recommendations regarding the assignment and distribution of modeling task to stakeholders and their viewpoints. This contributes to an improved coordination in collaborative modeling that is vital for social interactions (Mendling et al., 2012).

4.1.3 Requirement 3: Automatic conflict detection and distribution

To commonly achieve a consistent model instance in which all stakeholders have contributed with their knowledge, an automatic conflict detection mechanism is required. Since viewpoints can be dynamically defined based on metamodels, they can cause conflicts to the overall metamodel as only a subset of its model concepts and constraints are selected (Jarke et al., 1996). In current methods, consistency rules between viewpoints and views are only considered insufficiently, especially in terms of dynamic viewpoint creation and the distribution of conflicts to stakeholders (Fischer et al., 2012). Hence, the viewpoint-based modeling method developed in this paper should provide conflict detection features and combines them with a recommender system to identify those stakeholders that can resolve these conflicts the best. This promotes group decision-making to some degree, which is crucial for social interaction in collaborative modeling (Mendling et al., 2012).

4.2 Core Concept 1: Conceptual basis of viewpoint-based modeling

Even though the viewpoint concept has been applied in miscellaneous domains like Software Engineering or Enterprise Architecture Management, no common cross-domain understanding of the concept can be attested. For that reason, an overarching definition has been compiled in a preliminary paper (Fischer et al., 2012) serving as foundation for the conceptual basis of the viewpoint-based modeling method, which is introduced in this section. First of all, basic terms and definitions to define viewpoint-specific concepts will be clarified and visualized in Figure 2.

A Metamodel $MM := \{MC_{1}, ..., MC_{n}\}$ defines a frame and a set of rules for creating models by introducing concepts and their relationships as well as constraints that should be applied to them. A metamodel serves as a basis for models instantiating it. Related modeling concepts usually belong to a certain metamodel. A Model Concept (or modeling concept) is part of a metamodel and the basis for model elements in model instances. A Model Element is a concrete instance of a modeling concept, and thus it either represents a domain object or a relationship between two or more objects. These elements are part of a model instance and are being exposed in certain views belonging to certain viewpoint instances. A Profile is an extension of a metamodel, which uses the metamodel as a reference for redefining existing modeling concepts in order to target a metamodel towards a given application domain. It refers to a certain metamodel and can be applied to various models for domain alignment. A Model Instance contains a concrete set of model elements, which adhere to the rules defined in the corresponding metamodel. Models can apply certain profiles and represent model elements accordingly.

These basic terms provide a foundation for creating models for specific application domains. To be able to support viewpoint-based modeling, additional terms have to be defined on top of this foundation. A Viewpoint supports a stakeholder in contributing to system design from a specific perspective. In doing so, a viewpoint defines which model concepts and relations can be used to define, view, or manipulate model instances within this viewpoint; thus, it separates modeling concerns and guarantees consistency regarding the information in the model instance. It is therefore related to a (set of) metamodel(s), a (set
of) profile(s) or a part of them. The viewpoint in this sense can restrict the original metamodel(s), but it can also correspond to a metamodel in a 1:1 relationship. Formally, a viewpoint \( VP_1 \) is defined as 
\[ VP_1 = \{ VT_{VP_1;1}, \ldots, VT_{VP_1;m} \} \], \( VP_1 \neq \emptyset \).

A View Type serves as a basis for view instantiation and offers a specific slice of system perspective to the stakeholders. A collection of view types is defined for each viewpoint. In doing so, a view type can again relate to a subset of the chosen model concepts of its corresponding viewpoint. Formally, a View Type \( VT_{VP_1;1} \) is defined as 
\[ VT_{VP_1;1} = \{ MC_1, \ldots, MC_k \} \Rightarrow VT_{VP_1;1} \subseteq MM. \]

A View is an instance of a view type and defines the presentation of model elements to a stakeholder to whom the viewpoint belongs and the way(s) how model elements can be modified (this is usually achieved by diagram types together with a tool box for manipulating model elements). It enables its users to interact with particular aspects of one or more models that adhere to the metamodel of a viewpoint. Consistency between views is dealt with at the level of the model instances, i.e. when changes in a view are stored, the model instance is checked and in case of inconsistencies these are alerted to the stakeholder. Whereas a view realizes the representation of model elements, a Projection—as a specialization of a view—also allows for a stakeholder-specific visualization of individual model elements. Consequently, not only a specific selection and de-selection of model concepts is possible, but also their domain-adequate visualization.

Having a broader look at viewpoint-based modeling, a Modeling Project consist of a set of Modeling Tasks \( t \in T \) specifying modeling activities that need to be collectively done to create a valid model instance representing the system under development. These tasks are accomplished by several stakeholders, all of which are supported in their modeling activities by specific viewpoints (commonly one per stakeholder). Both, the stakeholder and the task have each a profile describing them. A stakeholder is described by its viewpoints and their corresponding ViewTypes all of which define the available model concepts. In addition, each stakeholder possesses several (modeling) Competences ranging from functional to technical to social competencies—all at different levels of knowledge—that more or less correlate with their role and the domain of modeling projects. Take for instance business architects or information architects in the Enterprise Architecture Management (EAM) domain. Here a

---

**Figure 2. Conceptual basis of viewpoint-based modeling based on Fischer et al. (2012)**

---
stakeholder can be responsible for depicting organizational aspects based on different organizational modeling concepts. The modeling task on the other hand also describes by different concepts what is to be modeled requiring a stakeholder with similar properties (for detailed examination on competences within the EAM domain, it is referred to Wieringa et al. (2009)). Hence, competences are also inherent and required to solve modeling tasks; thus, they formalize a task and represent a valuable basis for matching modeling tasks with certain stakeholders. Based on previous project experience, each stakeholder possesses additionally a personal Task History, including already assigned and solved tasks as well as a computed probability of creating conflicts while solving them. In a modeling project, each stakeholder has a specific Workload due to currently assigned tasks. Formally, a stakeholder $s \in S$ has a profile $p_s := \{\text{task history}_s, \text{workload}_s, \text{competences}_s\}$ and is using one or more viewpoints.

To control and coordinate the modeling project, the Viewpoint-based Modeling Method consist of two further core concepts: one for modeling task recommendations (section 4.3) and one for conflict detection and resolution (section 4.2).

### 4.3 Core Concept 2: Modeling task recommendation

Having defined a conceptual basis on viewpoint-based modeling, this section introduces the modeling method in terms of how to distribute and coordinate tasks within a modeling project. This is one of the core concepts of viewpoint-based modeling, since an appropriate distribution is essential to achieve the primary goal of a collaborative creation: obtaining a consistent model instance in which all involved stakeholders have contributed to with their knowledge.

To point out different aspects of the introduced method, an abstract running example scenario will be used (for a concrete example stemming from the business process modeling domain, it is referred to Krumeich et al. (2013b)). The scenario in Figure 3 outlines some typical steps within a modeling project applying the viewpoint-based modeling method. In this scenario, but also for the overall recommender approaches, we make the assumption that there is always one stakeholder who can solve a modeling task and who can consequently be recommended by our method.

![Figure 3. Modeling steps in viewpoint-based modeling](image)

In a very first step of the scenario, a project manager divides a modeling project into several modeling tasks that should be allocated to corresponding stakeholders. To assist the project manager in assigning tasks to stakeholders, the viewpoint-based modeling method will make use of a recommendation approach, which in principle can also be used for an automatic allocation. In this regard, for each modeling task $t \in T$, the recommender approach will determine the stakeholder $s \in S$ that maximizes the utility function $u$:

$$\forall t \in T: s_{\text{recommend}} = \arg \max_{s \in S} u(t, s)$$
This utility function $u$ consists of four different rating criteria—each weighted with specific quotients $\alpha, \beta, \gamma$ and $\delta$ (where $\alpha + \beta + \gamma + \delta = 1$)—that are calculated to a recommendation value:

$$u(t, s) = \begin{cases} 
\text{similarity} \cdot \alpha + \text{experience} \cdot \beta + \text{conflict\_probability} \cdot \gamma + \text{current\_workload} \cdot \delta, & \text{current\_workload} + t \leq 1 \\
0, & \text{current\_workload} + t > 1
\end{cases}$$

This function assumes that a task $t$ can only be added to the work schedule of a stakeholder if his current workload plus the necessary time to complete task $t$ does not exceed the value 1.

**Matching between modeling task and stakeholder profile.** The first rating criterion represents the similarity between the modeling task $t$ and the profile of the stakeholder $s$, in particular regarding required competences and role descriptions. We have chosen the Jaccard similarity coefficient $f(t, s)$ to measure the similarity between both sets of properties. For more information on alternative methods to measure similarity, also with regard to user profiles, it is referred to Shardanand and Maes (1995).

$$\text{similarity}(t, s) = \frac{\text{sum}_t \cap \text{sum}_s}{\text{sum}_t \cup \text{sum}_s}$$

**Task experience.** The second ranking criterion measures the task experience a stakeholder has regarding the task to be assigned. We consider the experience of a stakeholder as a logarithmic function, i.e. first iterations in solving tasks lead to a stronger increase of the experience measure than later iterations. A maximum value of 1 is achieved when the number of successfully solved tasks that are similar to the current one is equal to a predefined experience factor $\exp$, which will be the basis of the logarithmic function. The experience value will progressively decrease if a stakeholder causes conflicting states in solving tasks (cf. section 3.2.3 for more details on conflicting model states).

$$\text{experience}(t, s) = \begin{cases} 
\log_{\exp}(\sum_{t' \in \text{task\_history}}\text{sum\_similar}(t, t')) - \sum_{t' \in \text{task\_history}}\text{sum\_conflict}(t, t'), & \sum_{t' \in \text{task\_history}}\text{sum\_similar}(t, t') \neq 0 \\
0, & \text{else}
\end{cases}$$

**Probability of causing conflicts in the context of a task.** The third ranking criterion qualifies the correlated second ranking criterion in terms of the probability of creating conflicts due to the constraints defined in the metamodel when dealing with a task. When computing it, the system makes the following assumption: a stakeholder who has comparably less experience with a task, but caused significantly less conflicts while solving this smaller number of tasks, the higher should be the absolute recommendation value for this stakeholder.

$$\text{conflict\_probability}(t, s) = 1 - \frac{\sum_{t' \in \text{task\_history}}\text{sum\_conflict}(t, t')}{\sum_{t' \in \text{task\_history}}\text{sum\_similar}(t, t')}$$

**Stakeholder workload.** The fourth ranking component reflects the current workload of stakeholders. This should increase the recommendation value for those stakeholders who have more available time to solve tasks than others. The workload of a stakeholder user is the ratio between the maximum number of tasks assignable to him and the current number of assigned tasks.

$$\text{current\_workload}(s) = 1 - \frac{\text{assigned\_tasks}(s)}{\text{max\_tasks}(s)}$$

As a result of the recommender system’s calculations, the current modeling task in the considered scenario is assigned to a stakeholder $S_l$. Consequently, the underlying model instance $M$ is displayed based on his or her viewpoint $VP_l$ in the ViewType $VT_{VP_l:1}$ (cf. Figure 3, 1). After stakeholder $S_l$ has processed the task by applying his or her domain knowledge to modify the underlying model instance
$M$, either a conflict-free or a conflicting adjustment result, which leads to the second core concept of the viewpoint-based modeling method.

4.4 Core Concept 3: Automatic conflict detection and distribution

A conflict-free adjustment (cf. Figure 3, 2a) requires all performed changes in ViewType $VT_{VP_i}$ to model instance $M$ to bear no conflicts to the underlying metamodel(s) $MM$. Formally spoken, let $M$ be a set of model elements in a model instance that should adhere to the set of model concepts forming a metamodel $MM$, i.e. $MM$ defines a set of constraints to which $M$ must be consistent. Let $satisfy$ be a function that verifies whether all constraints $c$ in a metamodel $MM$ are satisfied by a model instance $M$, i.e. $satisfy : M \times MM \rightarrow boolean$. Thus, each constraint $c \in MM$ must be met by the model instance $M$:

$$satisfy(M, MM) = \forall c \in MM : M \ satis c$$

Syntactical conformance can be verified by several algorithms published in scientific literature (for more algorithmic details cf. Paige et al. (2007)). In this case, the adjustment of the model instance can be stored as a conflict-free model instance $M''$ without any syntactical changes needed (cf. Figure 3, 3a). While such a conflict-free modification can be judged as a trivial case in a collaborative modeling setting, it is necessary to consider the conflicting case for a purposeful viewpoint-based modeling method (cf. Figure 3, 3b). At this point it is explicitly stated that inconsistent intermediate states are permissible within viewpoint-based modeling (Jarke et al., 1996).

If stakeholder $S_i$ modifies model instance $M$ in ViewType $VT_{VP_i}$ in such a way that the adjustment constitutes a conflict $k$ in the modified model instance $M'$ (cf. Figure 3, 3b) with regard to an underlying metamodel $MM$—global inconsistency—a model adjustment has to be performed in order to meet the constraints defined in the metamodel $MM$. This need for a model adjustment will be defined as another modeling task. If the conflicting model $M'$ however appears as consistent in the chosen viewpoint $VP_i$, stakeholder $S_i$ is not able to restore global consistency (unless s/he possesses additional viewpoints). In this regard, local consistency exists if all ViewTypes within a respective viewpoint satisfy the constraints defined within this viewpoint:

$$\forall VT_{VP_i} \in VP : satisfy(view(VT_{VP_i}, M), VP)$$

In this regard, the function $view : VT \times M \rightarrow M_{VT}$ provides a ViewType-adapted model instance $M_{VT}$ based on the underlying model instance $M$. Yet, if the model is already in conflict with the chosen viewpoint $VP_i$ (local inconsistency), stakeholder $S_i$ should be able to solve the conflict from a different ViewType $VT_{VP_j}$ in the chosen Viewpoint $VP_j$. This can be defined as:

$$\exists VT_{VP_j} \in VP : \neg satisfy(view(VT_{VP_j}, M), VP)$$

In case of local consistency, but global inconsistency stakeholder $S_i$ cannot solve the conflict by herself. The conflict resolution has to be performed by another stakeholder $S$ who has a viewpoint $VP$ with the model concepts and rights to solve the conflicting states $k$ in $M'$ (cf. Figure 3, 4b). This algorithm will be processed after a stakeholder has declared an allocated task as solved.

While there are several algorithms to detect conflicts, yet the challenge is to determine the most suitable stakeholder to resolve conflicts. For this purpose, the corresponding modeling task to solve the conflict has to be allocated a stakeholder respectively his or her ViewType that might solve the conflict the best. For determining this ViewType $vt$, we conceived an algorithm that weights all existing ViewTypes $vt$ and recommends the one with the highest weighting value based on the utility function $u(vt, c)$:

$$u(vt, c) = solvable conflicts _{vt} \cdot cm + conflict \_ probability _{vt} \cdot \theta + current \_ workload _{vt} \cdot \mu$$

In a first step, a list $C_M$ with all constraints of the metamodel $MM$ that are contradicted by the model instance $M$ is created. Already assigned conflicting constraints are omitted from this list in order not to reassign them. In case, an allocated constrained still exists after a stakeholder has completed a task to solve this conflict, this will be added to the list, since this particular ViewType might not be able to
solve the constraint successfully. Concurrently, the probability of causing a conflict is calculated for the current ViewType. Afterwards, in a consecutive manner, all conflicting constraints within this list are assigned to ViewTypes that fit to solve the conflicts the best. This is achieved by first determining how many conflicts can be resolved by each single ViewType. The first summand of $u(\text{vt}, c)$—concerning the possibly solvable conflicts per ViewType—will be normalized by dividing the absolute number by the number of that ViewType that is able to solve the most conflicts.

$$\text{solvable_conlicts}_{\text{vt}} = \frac{\text{solvable_conlicts}(\text{vt})}{\max_{\text{vt}' \in \text{vt}}(\text{solvable_conlicts})}, \quad \text{solvable_conlicts}_{\text{vt}} \in [0, 1]$$

In a second step, for each ViewType $\text{vt}$ the probability to yield new conflicts when a conflicts is assigned will be calculated based upon a global variable $VP_c$ that stores this probability for each ViewType and constraint. The second summand is already normalized due to its percentage value.

$$\text{conflict\_Propability}_{\text{vt}_c} = 1 - \prod_{k \in c}(1 - \text{conflict\_Propability}(c_k))$$

The third summand aims at weighting the current workload of viewpoints, i.e. a viewpoint with already a high workload should be less weighted than another one with less workload.

$$\text{current\_workload}_{\text{vp}} = 1 - \frac{\text{assignedTasks}(\text{vp})}{\max_{\text{vp}}\text{tasks}(\text{vp})}$$

Having done these calculations, for each ViewType the weighting value can be assigned. This is the sum of all three summands each weighted by adjustable variables $\alpha$, $\beta$ and $\gamma$. In order to have each summand comparable to each other all of them are normalized to a value in the interval $[0, 1]$. After each existing ViewType is weighted, the one with the highest value is selected and will be allocated with concerning tasks to solve the underlying conflicts $k$ to metamodel constraints $c$. At the same time, a function $\text{allocateTask(\text{vt})}$ updates the list $G_{\text{distributed}}$ and increases the workload $WL_{\text{vp}}$ of the concerning viewpoint due to the allocation of further tasks. In a last step, the list $G_o$ is reduced by the allocated conflicts so that the algorithm can eventually terminate. The adaption of model instance $M'$ by stakeholder $S_2$ again yields either a conflicting state or a conflict-free model instance $M''$ (cf. Figure 3, 5b), which is determined by reprocessing the algorithm.

5 Prototypical Implementation and Recommender Evaluation

5.1 Proof-of-concept

To proof the implementability of the viewpoint-based modeling method including its recommender functionality, we built the prototype CIMFlex4CM as a realization of the previously developed conceptual artifact. CIMFlex4CM technically builds upon the Eclipse Modeling Framework (EMF) and Graphical Modeling Framework (GMF) as provided by Eclipse. They allow for developing modeling tools without having to implement all modeling-specific functions from scratch. EMF enables an automatic generation of Java-based source code out of structured (meta-) models. The underlying metamodels are so-called Ecore-models that rest upon the EMOF-standard (Essential Meta-Object Facility) (Object Management Group, 2013). GMF allows the generation of graphical editors on top of the EMF-based models. To realize the viewpoint-based modeling method, the creation and adjustment of viewpoints and the corresponding view types has been implemented by two generators building upon EMF/GMF. These have been implemented based on wizards known from the Eclipse IDE.

To realize the dynamic coordination of tasks within viewpoint-based modeling, the prototype utilizes the Eclipse Plugin Mylyn. Mylyn is a “Task and Application Lifecycle Management” framework which implements a task-focused interface as well as issue tracking functionalities into the Eclipse IDE. An optimal distribution and coordination of the individual modeling tasks to suitable stakeholders is achieved by the developed recommender system on a basis of Apache Mahout. The system enables an optimal assignment of modeling tasks based on existing viewpoints and the related view types. To create
a data basis for the applied recommender system, the issue tracking system of Mylyn saves the allocation and the solving of tasks as well as problems that arose in the modeling process. It should be noted that the system—like any recommender system—requires a certain data basis to generate satisfying recommendations. Besides the task allocation, the recommender system offers further guidance for an optimal task order both from a project view and a stakeholder-specific view. Hence, the prototype guarantees an appropriate distribution of modeling tasks among participating stakeholders.

Figure 4. Screenshot of the CIMFlex4CM prototype

Figure 4 shows a screenshot of the prototype’s viewpoint generator, in which the mapping of flags to the model concepts is conducted. In the background, one can see a metamodel as an Ecore diagram, which can be dynamically changed and loaded into the prototype.

5.2 Recommender evaluation

We evaluated the processing of our recommender based on the underlying example scenario in Figure 3. In the first step, we simulated three potential viewpoints of which the recommender should identify the best one regarding the current situation as well as some defined variables. In this automatic evaluation we make some assumptions. We will assign a task t 40 times to see how the recommendations progress. The three available stakeholders respectively viewpoints have the following properties:

- Stakeholder S1: similarity between task t and the profile of stakeholder S1 = 0.40; initial experience with task t = 0.00; conflict probability for task t: 0.20; initial workload = 0.50
- Stakeholder S2: similarity between task t and the profile of stakeholder S2 = 0.80; initial experience with task t = 0.00; conflict probability for task t: 0.40; initial workload = 0.50
- Stakeholder S3: similarity between task t and the profile of stakeholder S3 = 0.60; initial experience with task t = 0.00; conflict probability for task t: 0.60; initial workload = 0.50

The value of the experience factor will be determined with 10. After each distribution, the stakeholder’s workload will be reduced by 0.1 if s/he will not be assigned with the task. We further assume that the number of allocated tasks t multiplied by the conflict probability determines if a current allocation leads to a conflict that reduces the viewpoint’s experience concurrently. We tested the recommender with four different variable values for α, β, γ and δ, to see how this influences the recommender’s calculations.
In Figure 5, a) stakeholder \( S_2 \) will be selected since s/he features the highest similarity to tasks \( t \) which corresponds with the high value of \( \alpha \). The oscillation can be explained due to the conflict value 0.40 which causes after a threefold task allocation to reduce the experience value. Since the experience is based on a logarithmic function this oscillation levels off. Increasing \( \beta \) to 0.55 and decreasing \( \alpha \) to 0.15 produces similar results according to Figure 5, b). Stakeholder \( S_2 \) will be recommended throughout all 40 iterations. Since the experience is weighted more strongly, all three graphs start with smaller values and more closely. As in a) the increase levels of after about 17 iterations. The initial increase of the utility values for Stakeholder \( S_1 \) and \( S_3 \) is due to the decrease of their workload. Figure 5, c) shows a different picture, while Stakeholder \( S_2 \) dominates the first three and Stakeholder \( S_3 \) the succeeding two first iterations, Stakeholder \( S_1 \) outperforms them after six iterations. This can be explained based on the different conflict values. While \( S_1 \) exhibits the highest similarity value, its utility value decreases promptly after causing the first conflict, this brings \( S_3 \) at the highest recommendation value; however, with a conflict value of 0.60 the utility value drastically decreases after two iterations with an even higher magnitude than for \( S_2 \). Resulting Stakeholder \( S_2 \) will shows the highest recommendation value, even after five allocations and the resulting conflict causing (conflict value 0.20) does not rate him lower than \( S_2 \) and \( S_3 \). Increasing \( \delta \) to 0.55 and decreasing \( \gamma \) to 0.15, as in Figure 5, d), provide similar results as in c). However, Stakeholder \( S_1 \) outperforms the other stakeholder one iteration later then in c).

Having a look at both performance indicators in the four scenarios, i.e. the workload mean among all 40 iterations as well as the caused conflicts, it can be concluded, that the weighting of \( \alpha \), \( \beta \), \( \gamma \) and \( \delta \) in c) and d) lead to better results than in a) and b). However, it cannot be concluded that \( \gamma \) and \( \delta \) should always be higher than \( \alpha \) and \( \beta \). This needs to be defined for each project leading to another requirement for a successful implementation of the approach in a practical modeling tool: a kind of dashboard for project managers is required based on which they can adjust weighting factors during the project.

6 Conclusion and Future Work

This paper conceived a viewpoint-based modeling method which applies the concept of viewpoints to collaborative modeling to incorporate multiple stakeholders. To effectively distribute and coordinate modeling activities among all participants, the method utilizes the concept of recommender systems to eventually end up with a consolidated, conflict-free model outcome that has been collectively constructed. Besides the development of the viewpoint-based modeling method, the paper at hand additionally presented its proof-of-concept by means of a prototypical implementation. How the conceived viewpoint-based modeling method can be utilized in specific application domains is described in detail in Fischer et al. (2013). Currently, we are in the process of implementing the developed concept into commercial modeling tools of industrial research partners. Afterwards, field studies with their customers will be conducted. Further research efforts will comprise the extension of the (syntactical) conflict detection algorithm by semantic features, so that also semantically conflicts can be detected, distributed among stakeholders and eventually resolved in a collaborative manner.
References


