Towards a Viewpoint-based Modeling Method to Foster Collaborative Modeling — Conceptual Design and Implementation

Julian Krumeich
*German Research Center for Artificial Intelligence (DFKI)*, inselkampfseller@web.de

Dirk Werth
*German Research Center for Artificial Intelligence (DFKI)*, dirk.werth@dfki.de

Peter Loos
*German Research Center for Artificial Intelligence (DFKI)*, loos@iwi.uni-sb.de

Follow this and additional works at: [http://aisel.aisnet.org/pacis2013](http://aisel.aisnet.org/pacis2013)

Recommended Citation


[http://aisel.aisnet.org/pacis2013/249](http://aisel.aisnet.org/pacis2013/249)
TOWARDS A VIEWPOINT-BASED MODELING METHOD TO FOSTER COLLABORATIVE MODELING—CONCEPTUAL DESIGN AND IMPLEMENTATION

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

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

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

Abstract

This paper proposes a modeling method that links the so-called viewpoint concept with collaborative modeling. In collaborative modeling settings, problems like low model acceptance among involved stakeholders are typical because of their limited understanding of the overall model. The viewpoint-based modeling method aims at solving such problems by introducing and using stakeholder-specific viewpoints on collaboratively created models. As a consequence, the viewpoint concept facilitates and improves the involvement of multiple stakeholders from different domains into the collaborative modeling process. In addition to the conceptual design of the viewpoint-based modeling method, the paper—following design science research—presents a proof-of-concept by means of a prototypical implementation.

Keywords: Collaborative Modeling, Viewpoint Concept, View, Stakeholder, Design Science.
1 INTRODUCTION

1.1 Motivation

In the course of developing systems, 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. According to the General Model Theory (Stachowiak 1973), models represent or illustrate natural or artificial originals, which can again be models by themselves.

Even though an important inherent feature of models is to provide a reduction of complexity of the underlying system by means of shortening properties, they often tend to have a high level of complexity. This is in particular the case when developing highly complex systems. Consequently, the perceived value of models by their users is considerably limited compared to its original purpose. This effect is even magnified if several individuals, e.g. coming from different domains, are involved in the model creation process as well as in its usage. If in such settings only the conventional single and rigid perspective is provided on specific model aspects, their overall value remains fairly low for those who are working with them (Fischer et al. 2012).

To overcome this shortcoming, it is essential to put each involved stakeholders more into focus. Since all of them typically have different expertise, abilities, knowledge as well as responsibilities, this will effectively foster the perceived value of models (Finkelstein et al. 1992). Hence, they all need different perspectives on the development of the underlying system. Consequently, it is promising to provide means for viewing and manipulating models depending on the specific objectives of the participating stakeholders (Dijman et al. 2008).

1.2 Research Contribution

To support modeling with regard to these requirements, the viewpoint concept appears as a promising approach. In doing so, it offers a complexity reduction by allowing the underlying models to be broken down into stakeholder-specific fragments (Goldschmidt et al. 2012). Thus, the focus is moved to the involved individuals resulting in an increased understanding and productivity (Finkelstein et al. 1992). Concurrently, the viewpoint concept contributes to the development of better conceptual models, i.e. models depicting the underlying system more adequately. This effect has already been proved in studies (Easterbrook et al. 2005).

However, the growing need for collaboration, when multiple stakeholders are involved, makes an appropriate coordination necessary, which also results in higher complexity. For this purpose, the concept of collaborative modeling can be applied (Frost & Sullivan 2007; Renger et al. 2008). Collaborative modeling is defined as a “process where a number of people actively contribute to the creation of a model” (Rittgen 2010b).

The realization of this approach cannot be considered trivial. Rittgen (2010b) conducted an expert survey among IT consultants to identify major problems with collaborative modeling. The resulting answers included “low model acceptance”, “eliciting individual views”, “participants feel misunderstood”, “slow progress, facilitator overload”, “making different views convergent” and “limited model comprehension”.

Interestingly, regarding these answers the viewpoint concept appears especially suited to solve these problems and provides a promising contribution to foster collaborative modeling. Thus, this paper develops a viewpoint-based modeling method which applies the viewpoint concept into the collaborative modeling process to effectively incorporate multiple stakeholders. Additionally, the paper presents a prototypical implementation of this modeling method as a proof-of-concept.
1.3 Research Approach and Paper Structure

By following the design science research paradigm, in which innovative IT artifacts are developed to solve identified practical problems that have motivated the research, this paper develops a conceptual modeling method as a solution to the initial problems from organizational practice as mentioned in section 1.1. Building upon that, a prototype as a specific instance of this artifact is additionally presented.

According to Hevner et al. (2004), design science has to comply with seven guidelines to maintain a stringent and relevant research. In concordance with that, the introduction and motivation point out the relevance of the research problem (Guideline 2). The conceptual modeling method developed in section 3 represents the artifact (Guideline 1), whose prototypical implementation in section 5 extends it to a comprehensible and verifiable artifact (Guideline 4). According to Guideline 3 it is also necessary to perform evaluations on the utility, quality and efficacy of the designed artifact (cf. section 5). To build upon existing knowledge (Guideline 6), the developed approach is based on previous work, which also refers to existing concepts. Guideline 5 is accomplished by outlining the applied research methodology. Finally, the submission of this paper aims at fulfilling Guideline 7, the dissemination of research results. A first step towards this direction was taken by publishing initial results at the German information systems conference Wirtschaftsinformatik (cf. Krumeich et al. (2013)). Building upon this foundation, the present paper significantly extends the initial one and moreover expands its reach to the international information systems community, since the former one was written in German.

The structure of this paper is as follows. After introducing the motivation and the applied research method in section 1, section 2 discusses related works in the area of the viewpoint concept and collaborative modeling. In the succeeding section 3, the viewpoint-based modeling method is developed. Whereas that section rather abstractly introduces the conceptual method, section 4 illustrates it in a concrete application scenario. Section 5 instantiates the conceptual artifact with a prototypical implementation. The paper closes with a brief summary and an outlook on future work in section 6.

2 RELATED WORK

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 reference in the area of viewpoint research is the frequently cited work 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.

Moreover, IEEE 1471:2000 represents a standard definition of the viewpoint concept, which has also been included by ISO/IEC as ISO/IEC 42010:2007 standard in 2007 (Software Engineering Standards Committee of the IEEE Computer Society 2007). Following that standard, the viewpoint concept resp. a viewpoint can be defined as “a specification of the conventions for constructing and using a view”. Apart from the actual definition of the viewpoint concept as well as different viewpoints in various application domains—e.g. in the context of Enterprise Architecture Management it is referred to the TOGAF framework and its viewpoints (The Open Group 2012)—related work focusing more the real application should be mentioned as well.

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.

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. A previous work to the present paper also considered such requirements with regard to modeling tools (Fischer et al. 2012).

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.

Considering collaborative modeling, the COMA-tool—as described in Rittgen (2010a)—is one example for a concrete tool supporting the realization of collaborative modeling. COMA is an acronym for “COllaborative Modeling Architecture” and allows for the coordination of groups when modeling UML models. COMA however focuses synchronous collaboration in which all participants negotiate the same activity at the same time.

An earlier proposal for a collaborative modeling solution 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.

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.

The basic idea of different views for different roles during modeling can be found in a paper by Cera et al. (2004)—even though they consider CAD modeling instead of process modeling. 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 the viewpoint-based modeling approach tries to facilitate each modelers work to reach overall models of higher quality.

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 like in the viewpoint-based concept 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 as well as rationales from the field of belief theory are included.

Resulting from the literature review it can be attested that no approach in scientific literature 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.

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). As a consequence, a stronger foundational research, especially design-oriented research, is needed to understand the special characteristics of collaborative modeling and to provide new ideas for practice.

3 METHOD FOR APPLYING THE VIEWPOINT CONCEPT TO FOSTER COLLABORATIVE MODELING

3.1 Different Principles of Model Creation and Usage

Typically, two general groups of persons come into contact with a model: persons creating a model and persons using a model. 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).

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 towards a common objective (Mangenot & 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 commonly 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.

To effectively support collaborative work resp. collaborative modeling, the modeling method developed in this paper follows the so-called viewpoint-based approach. Hereby it is assumed that multiple individuals construct and use a system in question.

![Figure 1](image-url)  
**Figure 1.** Different principles of model creation and usage.

Such a scenario typically exists if several domain experts aiming at developing an integrated system are involved in the system modeling process. Consequently, each of the participating individuals—in the context of the viewpoint concept often called stakeholder (cf. Goldschmidt et al. 2012)—require a different focus and possibly different representations for the model usage and creation. By applying specific viewpoints onto the model, such stakeholder-specific representation as well as adjustment can be realized allowing an adequate adaptation of the modeling process to the needs of the involved
stakeholder. Thus, the connection of both paradigms is called viewpoint-based modeling (cf. Figure 1, right). Hereby, from a static point of view the viewpoint-specific presentation and from a dynamic point of view the dynamic coordination of the modeling process with its multitude of related viewpoints have to be considered.

3.2 Static View on 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, a cross-domain definition has been compiled in a preliminary paper (Fischer et al. 2012) serving as foundation for the static view on the viewpoint-based modeling method, which is presented in this section. Following that, a dynamic consideration is performed in the subsequent section 3.3.

For introducing the static view, basic terms and definitions to define viewpoint-specific concepts are clarified first. These terms are stated in the lists below and are visualized in Figure 2.

- A **Metamodel** 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 a 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 represents either a domain object or a relationship between two or more objects. These elements are a 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 and thus targets 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 thus 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.
- 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.
- 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 viewpoint’s metamodel. 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 of a model instance, 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.
An intuitive reading of the metamodel depicted in Figure 2 is that viewpoints are either directly defined on metamodels or on top of profiles. It is stressed that this metamodel is not meant to define concepts based on which models and model instances will be created. It describes the terms which are important when it comes to viewpoint-based modeling and how these concepts relate to each other. Hence, it aims at providing a guideline to tool developers regarding how the viewpoint concept should be used and implemented in tools supporting viewpoint-based modeling.

**Figure 2.** Static view on view-based modeling based on Fischer et al. (2012).

Based on the general terms and an analysis of existing definitions, a harmonized viewpoint definition could be derived that serves as a basis for the further progress of the paper.

**Definition:** A viewpoint is defined in relation to one or more metamodels. For each viewpoint a non-empty set of view types is defined. In a viewpoint instance any number of views for each of the view types can be dynamically created. These views allow the stakeholder for whom the corresponding viewpoint was defined to access and manipulate the model instances.

### 3.3 Dynamic View on Viewpoint-based Modeling

The static view on the viewpoint-based modeling method primarily addresses the breakdown of models according to stakeholder-specific requirements to yield domain-adequate representations. Still, the dynamic components in the sense of creating and manipulating model instances need to be conceived as well.

To realize viewpoint-based modeling, it is necessary to distribute the individual modeling activities appropriately to achieve the primary goal of the collaborative creation of a consistent model instance. All required modeling activities thus are broken down into more or less atomic modeling tasks by the person in charge of the modeling project. Yet, this task division is not sufficient, it is especially necessary to allocate the tasks to the corresponding stakeholders. To enable the allocation of modeling tasks to different stakeholders and their viewpoints in an efficient way—i.e. largely automated or presented to the person in charge as recommendations—, a suitable database needs to exist. This has to include information regarding the experience of stakeholders in the context of viewpoints, the experience with similar tasks and the current modeling workload. On this basis, the generation of recommendations can be executed. Since a detailed description of each conceptual aspect of
viewpoint-based modeling, e.g. the actual recommender mechanism, is not the objective of this paper, the development of a general understanding how different viewpoints interact in the sense of an integrated overall functionality is the focus and will be clarified in the following.

When a modeling task is assigned to a stakeholder $S_1$, the underlying model instance $M$ is displayed in the modeling tool based on his or her viewpoint $VP_1$ in the view $v_{VP_1;1}$ (cf. Figure 3, 1). After stakeholder $S_1$ has applied his or her domain knowledge to modify the underlying model instance $M$, either a conflict-free or a conflicting adjustment result. A conflict-free adjustment (cf. Figure 3, 2a) means that all performed changes in the view $v_{VP_1;1}$ bear no conflicts to other views $v_{VP_1;2-n}$ of the same viewpoint, no conflicts to other viewpoints and their views $v_{VP_2-n;1-m}$ and no conflicts to the complete underlying metamodels $MM_{1-n}$ themselves (Fischer et al. 2012). In this case, the model adjustment can be stored as a conflict-free model instance $M''$ without any syntactical changes needed (cf. Figure 3, 3a). Syntactical problems can be identified from the underlying metamodels that each contain the valid relations between the model concepts (cf. section 3.2). Several algorithms for this purpose have been published in scientific literature (for more technical details it is referred to Paige et al. (2007)).

While the conflict-free modification can be judged as a trivial case, it is necessary to consider the conflicting case for a purposeful collaborative modeling (cf. Figure 3, 2b). At this point it is explicitly stated that inconsistent intermediate states are permissible within the viewpoint-based modeling (cf. Jarke et al. (1996)). If stakeholder $S_1$ modifies model instance $M$ in his or her view $v_{VP_1;1}$ in such a way that the adaption constitutes a conflict $k$ in the modified model instance $M'$ (cf. Figure 3, 3b) with regard to another viewpoint $VP_x$ or with regard to an underlying metamodel $MM_{global}$ (global inconsistency), a syntactic adjustment has to be performed. This adjustment can again be defined as a modeling task. If the conflicting model $M'$ however appears as consistent in the chosen viewpoint $VP_1$, stakeholder $S_1$ is not able to restore global consistency (unless he possesses additional viewpoints). Yet, if the model is already in conflict with the chosen viewpoint $VP_1$ (local inconsistency), stakeholder $S_1$ should be able to solve the conflict from a different view $v_{VP_1;2}$ within the view type $VT_x$. In case of local consistency, but global inconsistency stakeholder $S_1$ cannot solve the conflict by themself. The conflict resolution has to be performed by a stakeholder $S_2$ who has a viewpoint $VP_2$ with the model concepts and rights to solve the conflicting state $k$ in $M'$ (cf. Figure 3, 4b). For this purpose, the corresponding modeling task has to be allocated to him or her. The adaption of model instance $M'$ by stakeholder $S_2$ again either yields a conflicting state or a conflict-free model instance $M''$ (cf. Figure 3, 5b).

---

**Figure 3.** Model adjustments during viewpoint-based modeling.
4 VIEWPOINT-BASED MODELING IN THE BUSINESS PROCESS MODELING DOMAIN

The following exemplary scenario will illustrate the interaction of the static and dynamic component of the viewpoint-based modeling method.

The project manager starts the modeling project by creating a task structure covering all necessary modeling tasks. Supported by a recommender system s/he then allocates the individual tasks to appropriate stakeholders. At this, the recommender system suggests suitable viewpoints for the realization of the modeling task. After assigning the tasks to the stakeholders, they receive messages within their modeling environment notifying them about the new modeling tasks. Again, the recommender system comes into effect by determining the most convenient sequence of tasks for the stakeholder to perform. By choosing a viewpoint resp. one of its views, the stakeholder implicitly receives a suitable user environment to efficiently solve the modeling task. After completion of the task, the stakeholder stores the created or adapted model instance into the central model repository. The consistency check, which is performed in the background, then creates new tasks for conflict resolution and assigns them to the corresponding stakeholders if necessary.

In order to put it in more concrete terms, a simple, yet intuitive example from the business process modeling domain will be considered. In doing so, the chosen modeling project has the objective of depicting one of a company’s key business processes. The project manager sets up the following task structure:

- **Task 1:** Model the business process on a high level.
- **Task 2:** Refine and detail this high level representation.
  - **Subtask 2.1:** Add organizational units.
  - **Subtask 2.2:** Add supporting information systems.

As the underlying modeling language, the event-driven process chain (EPC) (cf. Keller et al. (1992)) is chosen. To apply the viewpoint-based modeling method to this domain, a simplified version of the EPC metamodel will be initially introduced (cf. Figure 4; for an elaborate discussion on the EPC syntax it is referred to Mendling (2007)). Based on the metamodel, it can be elucidated how viewpoints can be defined from a static view on viewpoint-based modeling and how does the dynamic component of the viewpoint-based modeling method work.

![Figure 4. Simplified metamodel of the Event-Driven Process Chain (EPC).](image)

In general, EPCs describe business processes by depicting events and functions in an alternating manner. Whereas functions—as active elements—define which activities are carried out in a business process, events—as passive elements—represent invoking and resulting states of a process resp. function. In doing so, each EPC consists of at least two events—i.e. the initial and the resulting state of a process—and one function—which describes by which activities the state change have been accomplished. Consequently, each function has exactly one event as a successor element and one as a predecessor element. Moreover, in EPCs, elements are linked by directional flow edges. In addition,
functions can be assigned with executing entities, e.g. an organizational unit(s) or an information system(s) that conduct the process function.

To cope with the tasks in the modeling project, the project manager creates different viewpoints. One viewpoint, for example, abstracts from all executing entities helping to depict the business process on a high level, which will be particularly suitable for Task 1. Thus, this viewpoint VP_1 only consists of the model concepts: Function, Event and Flow Edge; consequently, the concept Executing Entity is omitted from the viewpoint’s selection (cf. section 3.2).

After Task 1 is completed by Stakeholder S_1 using viewpoint VP_1 and its view v_{VP1;1}, there will be a conflict-free process instance, which is consistent to the underlying EPC metamodel (cf. Figure 5, upper left). Then, Subtask 2.1 will be accomplished by Stakeholder S_2 who has been assigned with viewpoint VP_2 that provides an ideal means to “add organizational units” to the high level process model. Since organizational units can only be connected with functions, the chosen view v_{VP2;1} of viewpoint VP_2 depicts the business events as grey elements in order not to sidetrack the stakeholder from the more important functions; hence, reducing complexity of the underlying model instance. Consequently, the viewpoint VP_2 consists of the model concepts: Function, Event, Flow Edge and Organizational Unit. The resulting model instance after completing Subtask 2.1 is displayed in Figure 5, upper right.

In contrast to Subtask 2.1, in which non-important model concepts are visualized in the chosen view v_{VP2;1} of viewpoint VP_2 in discreet colors, in Subtask 2.2 the model complexity is reduced by omitting non-relevant model concepts (cf. section 3.2). In doing so, only the model concepts Function, Flow Edge and Information System are selected and can be modeled in the view v_{VP1;1} of viewpoint VP_3 helping the Stakeholder S_1 to “add supporting information systems” to the high level process model according to Subtask 2.2. Since Stakeholder S_3 is an information systems analyst, s/he is more aware of the actual process flow than Stakeholder S_1. As a result, s/he replaces function F2 by two functions F2a and F2b to explicitly express that F2 is successively progressed by using two different information systems. However from the chosen view v_{VP3;1} of viewpoint VP_3, the mandatory Event between F2a and F2b cannot be modeled; thus, resulting in a conflicting model instance (cf. Figure 5, lower left).

Since the project manager was aware of potential conflicts, he created another view v_{VP3;2} resp. view type for viewpoint VP_3, so that similar inconsistencies can be directly resolved without the help from a different stakeholder or viewpoint. This means, whereas the view v_{VP3;1} has excluded the model concept Event, the view v_{VP3;2} resp. its view type (cf. section 3.2) of the corresponding viewpoint VP_3 consists of this model concept. Consequently, the conflicting model instance consists of a local inconsistency and can be resolved by using view v_{VP3;2} of viewpoint VP_3, which actually caused the inconsistency. Hence, the modeling system immediately create a new Subtask 2.2.1, which is automatically allocated to Stakeholder S_3, who receives a message about this new task in the modeling environment. By selecting view v_{VP3;2}, s/he can resolve the local inconsistency by adding the required model element to satisfy the underlying metamodel (cf. Figure 5, lower right).

Figure 5. Viewpoint-specific model instances.
5 PROTOTYPE IMPLEMENTATION

This section introduces the prototype CIMFlex4CM as a realization of the previously developed conceptual artifact. The CIMFlex4CM modeling tool technically builds upon the Eclipse Modeling Framework (EMF) and Graphical Modeling Framework (GMF) as provided by Eclipse. These frameworks allow for developing modeling tools within the Eclipse environment 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 2006). On this basis GMF allows the generation of graphical editors on top of the EMF-based models. For a detailed consideration of Eclipse frameworks in this context it is referred to Rempp et al. (2011).

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 (cf. Figure 6). These have been implemented based on wizards known from the Eclipse IDE.

In a first step, the viewpoint generator loads a metamodel as Ecore-file and stores the selected model concepts of the viewpoint creator as an XML-based file. Besides the pure selection of concepts, it is also possible to set flags for defining model concepts as read-only or for enabling an underlying with additional model instances. In the latter case the appropriate metamodel also needs to be selected to define the model type to underlie. As a matter of course, the generation not only allows for creating new viewpoints, but also to load and adjust existing ones.

In addition to the created viewpoints, a second generator is used to create the corresponding view types. The generator loads the viewpoint as well as a default-view-file which defines the default drawing of the metamodel concepts. The view type generator enables a further restriction of the viewpoint-specific selection of model concept. It also supports the definition of view-specific presentation forms and its storage as an XML-based file.

![Diagram](image-url)

Figure 6. Creation of viewpoints and view types.
As soon as viewpoints and a set of view types are created, the corresponding Eclipse plugin can be generated and rolled out to the particular stakeholders. They can load the plugin into their Eclipse instances, open the desired model instances from the repository and view and edit them under their specific viewpoint resp. view.

To realize the dynamic components of viewpoint-based modeling (cf. section 3.3), the prototype utilizes the Eclipse Mylyn Plugin. 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 allocation of the individual modeling tasks to suitable stakeholders is achieved by including the recommender system Apache Mahout. This 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. In doing so, a reporting of the modeling project is achieved implicitly. For the recommendation allocation, stakeholders are ranked according to their experience in the context of the viewpoint, their experience with similar tasks and their current workload. It should however 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 additionally suggests suitable modeling environments to the users, which have been defined in the viewpoint resp. the view type. As a third feature the recommender system offers 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 7 shows the prototype’s viewpoint generator, in which the mapping of flags to the model concepts is conducted. In the background, one can see the EPC metamodel as an Ecore diagram, which can be dynamically changed and loaded into the prototype.
6 SUMMARY AND FUTURE WORK

This paper has presented a viewpoint-based modeling method which links the so-called viewpoint concept with collaborative modeling. By using stakeholder-specific viewpoints, typical problems of collaborative modeling—like low acceptance of models among stakeholders from different domains because of their limited understanding of the overall model—can be solved. Consequently, the viewpoint concept facilitates and improves the involvement of multiple stakeholders from different domains into the collaborative modeling process. Besides the development of the viewpoint-based modeling method, the paper additionally presented its proof-of-concept by means of a prototypical implementation.

How the developed viewpoint-based modeling method can be utilized in practice 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. This allows for satisfying Guideline 3 according to the principals of rigorous design science research in information systems (cf. section 1.3).

References


