Institute of Architecture of Application Systems

Applicability of Process Viewing Patterns in Business Process Management

David Schumm, Tobias Anstett, Frank Leymann, Daniel Schleicher

Institute of Architecture of Application Systems,
University of Stuttgart, Germany
{schumm, anstett, leymann, schleicher}@iaas.uni-stuttgart.de

BibTeX:

```
@inproceedings{SchummALS10,
  author    = {David Schumm and Tobias Anstett and Frank Leymann
               and Daniel Schleicher},
  title     = {Applicability of Process Viewing Patterns
               in Business Process Management},
  booktitle = {Proceedings of the International Workshop on Models and
               Model-driven Methods for Service Engineering (3M4SE 2010),
               in conjunction with the 14th IEEE International
               EDOC Conference (EDOC 2010),
               25-29 October 2010, Vitória, Brazil},
  year      = {2010},
  pages     = {79--88},
  doi       = {10.1109/EDOCW.2010.13},
  publisher = {IEEE Computer Society}
}
```


Universität Stuttgart
Germany

Applicability of Process Viewing Patterns in Business Process Management

David Schumm, Tobias Anstett, Frank Leymann, Daniel Schleicher
University of Stuttgart,
Institute of Architecture of Application Systems,
70569 Stuttgart, Germany
{ Schumm, Anstett, Leymann, Schleicher }@iaas.uni-stuttgart.de

Abstract— **Process views are an emerging concept for coping with the increasing complexity of process models. We understand a process view as the result of specific transformations applied to a process model. In this paper we discuss concrete scenarios of process view transformations for providing assistance in business process management. We show how elementary patterns of view transformations can be combined to support the design, deployment, monitoring and analysis of business processes. The process views proposed in this paper are technology independent and can be applied to any process language which can be represented by a process graph, such as the Business Process Modeling Notation (BPMN) and Event-driven Process Chains (EPC).**

Keywords: Process Analysis, Process View, Model Transformation, Business Process Management

I. Introduction

The increasing demand for flexibility of businesses regarding the inner organization on the one hand and integration with other organizations on the other hand motivates more and more companies to express their business operations in terms of business processes. The management of business processes consists of process design, technical refinement by IT experts, execution, monitoring and analysis. These phases are facilitated by a set of tools which are integrated into a business process management (BPM) software suite. However, increasing complexity of processes is a problem. Complex business processes consist of hundreds of activities [1] which make them hard to manage without techniques for abstraction and corresponding tools. A business process can typically be represented by a process graph where nodes stand for the activities of a business process and edges represent control or data dependencies between them. Abstraction in the context of process graphs denotes the omission of nodes as well as the aggregation of nodes into higher level structures. Some algorithms for graph abstraction already exist. For instance, the approach in [35] can master graph abstraction up to 200.000 nodes. However, for process graphs particular semantics apply and therefore there is a need for specific abstraction methods. We argue that such methods can be defined by using process view concepts.

Process views are an emerging concept addressing the problem of increasing complexity in business processes. A basic principle is separation of concerns and abstraction from details which are not required in particular situations. We understand a process view as a presentation of the result of specific transformations applied to a process. Process view transformations typically comprise omission and aggregation of structures. Also visualization techniques are applied in order to provide a flexible instrument for process designers, process analysts and other stakeholders of a process. In [2] we introduced a metamodel for process views as well as process viewing patterns which specify elementary transformations to alter existing processes. We use these elementary transformation patterns and the viewing metamodel to define process views for usage in business scenarios. Process viewing applications represent tools that are based on process viewing concepts. They can either be self-contained, stand-alone applications and serve just a particular purpose, or be part of a larger framework. The viewing scenarios defined in this paper can be realized by implementing the required subset of process viewing patterns.

Process views have several purposes. One purpose is information filtering. Particular artifacts, activities, or whole structures in a process are not essential during particular tasks related to process management. They can therefore be neglected in those situations. For example, activities in a process which run fully automated can be faded out during the performance of staff related tasks. Filtering information reduces the overall complexity of a process. Another purpose of process viewing is information summarization. A filter removes information. In contrast to that, a summarization makes it more compact by aggregating structures. Besides, process views can also support the translation of information. The appearance of a process concerning a graphical notation can translate information for different stakeholders. Also alteration and rearrangement of particular structures or a change of their naming can make a process easier to understand. Another function of process viewing is information linkage. By augmenting and linking information from the outside to a process, interrelations can be recognized easier. In this paper we discuss application scenarios of process views that serve the purposes described above. We also demonstrate how they contribute to the tasks related to the management life cycle of a business process.

In general, process views provide a perspective on a process which is personalized for specific needs of a user and situation. The views we discuss in this paper represent solutions to problems which frequently occur in the management of business processes. The transformation steps can be either pre-defined or generated automatically by an algorithm. Transformations have a degree of freedom in terms of parameters or configuration and they may require user input during transformation for decisions.

The purpose and scope of this paper is to provide examples of process views which can be applied in business scenarios. To ease understanding we illustrate each viewing scenario. The concepts are described in a generic way. For implementation they need to be transferred to a concrete process language that is based on a process graph, for instance to Event-driven Process Chains (EPCs) or to the Business Process Model and Notation (BPMN).

As proof of concept we are currently developing a view transformation framework [2], based on the Business Process Execution Language (BPEL) [9]. In [15] we discuss initial results on how to use and extend this framework in the context of compliance management. Managing compliance requires performing profound and traceable changes on processes and a suitable visualization for process management and auditing reasons [4]. However, the concepts and principles presented here are also relevant in many other business scenarios, as increasing complexity is a fundamental problem of business processes and therefore approaches for mastering this complexity are crucial.

This paper is organized as follows: Section II describes the background of process view transformations and explains process viewing patterns. In Section III an exemplary viewing scenario is elaborated in detail to make the basics of process view transformations more tangible. In Section IV the viewing scenarios for business process management are discussed. This section also contains references to work related to the particular application scenarios. Section V summarizes the paper and identifies issues for further research.

II. BACKGROUND

In this section we explain our understanding of the fundamentals of process views which we described in [2]. The conceptual model for process viewing described in the following is illustrated in Fig. 1. The upper part of the conceptual model shows four groups of transformations that either concern the structure of a process, the augmentation of a process with additional information, relations between different views (inter-view) or the presentation of a process. In this sense we understand a process viewing pattern as an elementary form of such a model transformation. In this paper we focus on process views that make use of transformations concerning structure, augmentation and presentation. For viewing scenarios concerning inter-view relations please refer to [2].

To create a process view several steps (so-called transformation items) that control the transformation need to be defined. Each of these transformation items refers to one of the elementary transformation patterns. The transformation items also contain targeting information to declare to which artifacts of the input process the transformation should be applied. For execution of the transformation a corresponding implementation needs to evaluate all transformation items and then apply them to an input process. After all transformations that alter the process have been completed, the outcome (i.e. the transformed process) is displayed. The graphical functions for visualization display the outcome to the user according to further instructions for presentation.

There is a certain difference to Model-driven Architecture (MDA) terminology, in which a source model is transformed to a target model. The source model in MDA terminology corresponds to the original process model in our conceptual model (see Fig. 1), and the target model in MDA corresponds to the process view. However, we use the term "target" to indicate the structures in the original process which are to be affected by a transformation.

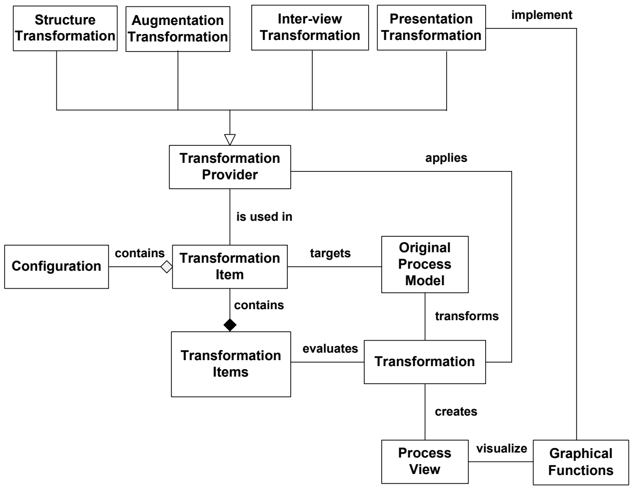

Figure 1. Conceptual Model for Process Viewing

To describe the viewing transformations in a manner easy to understand we have proposed in [2] a set of icons for the metamodel constructs of process graphs and process views (shown in Fig. 2). We use typed nodes to represent regular activities of a process. An abstract node stands for an activity that is abstracted from all properties. In other words it states that something happens, but what happens stays unclear. An aggregate node is a summarization of multiple nodes and edges into an atomic unit. An inserted node represents an activity, just like a typed node. The difference is that an inserted node does not exist in the original process that has been taken as input for the transformation. This means that an inserted node has been added to the process during the transformation. Analogously an inserted edge is a control edge which has been added during the transformation. Control edges represent control dependencies between activities. In some views also data dependency is of importance, therefore data edges are also contained in the metamodel. The execution of the transformation, see "Transformation" in Fig. 2, is denoted by a large arrow. It targets a specific set of nodes and edges of the input process, see "Target set" in Fig. 2. This is not meant to be a definitive graphical notation for process views, we use these icons here informally as an aid to ease understanding.

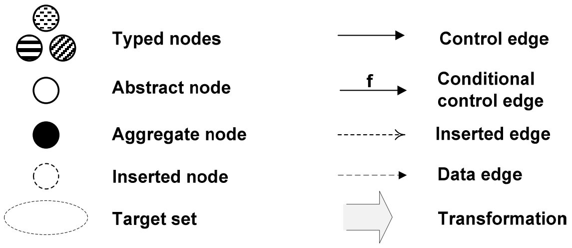

Figure 2. Process View Metamodel Constructs [2]

The viewing scenarios presented in this paper follow a particular order in which the different kinds of transformations (structure, augmentation and presentation) are applied. As illustrated in Fig. 3, in the beginning the process is augmented with additional information. Next, structure transformations are applied. Afterwards, the result of these transformations is presented to the user, i.e. visualized.

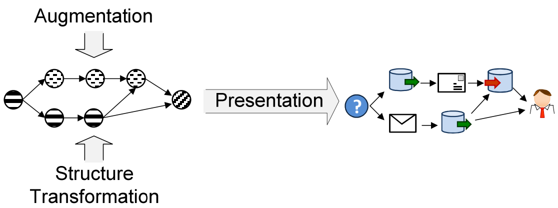

Figure 3. Execution Order of Transformations

The transformation patterns concerning process structure which have been used to define the views are shown in Fig. 4. These patterns are omission of structures (a), aggregation of structures (b) and abstraction of nodes (c). We have discussed the elementary patterns of process viewing in former work, for a complete description of process viewing patterns please refer to [2].

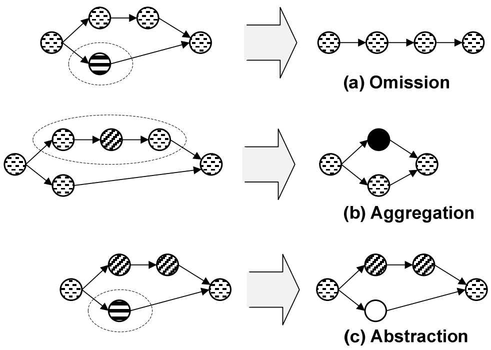

Figure 4. Structure Transformation Patterns [2]

The transformation patterns concerning augmentation that have been used in this work are illustrated in Fig. 5. These patterns are runtime augmentation patterns (a) which describe the augmentation with information related to current or former executions. Another augmentation pattern (b) considers automatic techniques such as pattern recognition, deadlock detection and particular heuristics. Human-assisted augmentation (c) addresses human knowledge about the process.

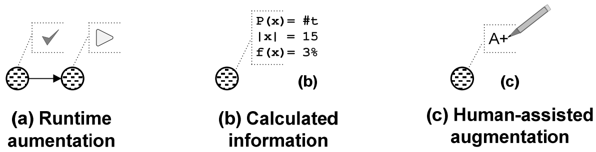

Figure 5. Augmentation Transformation Patterns [2]

The viewing scenarios discussed here also make use of presentation patterns, see Fig. 6. Appearance (a) refers to size, color, shape and other display properties of nodes and width, length, color and other display properties of edges. Furthermore, organizational information (b) concerning data flow and control flow which is contained in a process can explicitly be made visible. Finally, custom categories (c) address other information that could be used as node positioning or a grouping criterion, for instance the location of a participant in a process that is spread across multiple locations.

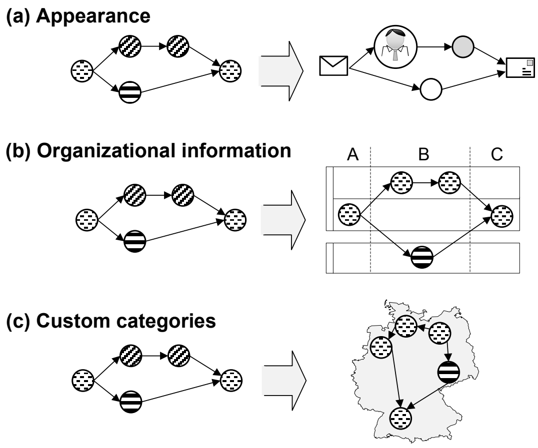

Figure 6. Presentation Patterns [2]

III. From a Process to a Process View

In this section we discuss a concrete process viewing scenario in detail and we exemplify basic concepts of process view transformations. Basically, we can distinguish between two different kinds of process views. On the one hand, there are pre-defined views. These views are good solutions to common problems, with pre-defined transformation items that specify how the process view has to be created from a given input process. For flexibility reasons these views may have a particular degree of freedom in terms of parameters and configuration though. On the other hand, there are custom views which are specified in an ad-hoc manner for quite specific problems. In the following we discuss such an ad-hoc view.

Before we can specify any ad-hoc view transformation we need a problem that should be solved with the aid of a process view. Let us assume that in some running instance of a loan approval process (depicted in Fig. 7) there is a problem in the data structure of the object *customerRecord* which needs to be fixed. In other words, the task is to pursue this object in order to locate the problem.

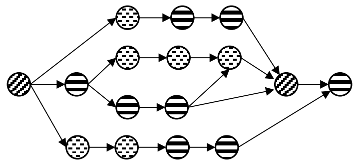

Figure 7. Input for View Transformations: Loan Approval Process

In order to create a process view for pursuing an object we need to specify particular transformation items. These items declare which parts of the process have to be transformed and how. A transformation component evaluates these items and applies the requested transformations to the input process. There exist various technologies which can be used for this purpose. The Object Management Group for instance defines the Model Driven Architecture (MDA), an architecture for model transformations which is based on the Unified Modeling Language (UML). Together with the specification Query/View/Transformation (QVT), which describes transformation languages, powerful instruments are available to be built on. Or otherwise, a new language can be created from scratch. Based on the chosen language, we can specify transformation step items which declare something like "highlight all activities which either read from, or write to the object *customerRecord*". Process views refer to transformation of the process structure, and also to the visualization of a process. This kind of transformation item refers to the visualization component. As a result, the process view shown in Fig. 8 is displayed. It highlights the activities that access the *customerRecord* object.

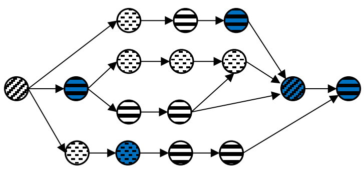

Figure 8. Process View: Particular Activities are Highlighted

Process view transformations can be applied in an iterative manner to find a solution. This means that we can gradually refine this view to find an answer to the initial problem. For example, if the process is very complex (which is not the case here) then a useful refinement would be to omit all activities that do not access the *customerRecord* object. We are then aware which activities are accessing the object. In a next step we can augment the view with runtime information from the instance in which the problem occurred. We can bind this runtime information to the visualization, e.g. by setting the transparency of activities which have not yet been executed to a higher level. The resulting process view (illustrated in Fig. 9) indicates that there are three candidate activities, which might have caused the problem. These activities then need to be checked manually.

The presented view is limited in its power and might just be needed once, but its principle - providing an emphasis on particular parts of a process - might be useful in multiple situations.

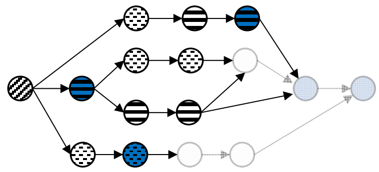

Figure 9. Refined Process View: Augmented with Runtime Information

IV. PROCESS VIEWING SCENARIOS

In this section, we discuss several application scenarios of process views to support the management life cycle of a business process which is illustrated in Fig. 10. Building on the conceptual model and the viewing patterns described in Section II, we propose viewing scenarios to support process design, management of deployment information and monitoring of process instances. Afterwards we discuss process views for visual process analysis and process intelligence. We conclude this section with the description of general purpose views. These views provide common abstraction and viewing support and are not bound to a particular phase in the process life cycle. Process views to support the execution phase of a process can be defined using structure patterns and augmentation patterns. However, these views are language dependent and little business-oriented, thus they are not discussed here. As mentioned in the introduction, the viewing scenarios shown in the following are technology independent and can be applied to common graph-based process languages. For each scenario we describe the problem the view intends to solve. The solution to the problem is described in terms of view transformations that need to be applied. We exemplify each scenario with an illustration using the process view metamodel constructs shown in Fig. 2.

Our approach to derive the application scenarios was to go along the process-related tasks which have to be performed along the life cycle. For each task we have analyzed if there is a useful combination of the elementary viewing transformation patterns, i.e. if there is a process view to support the task. Literature study provided further application scenarios, for instance an application scenario related to views on security.

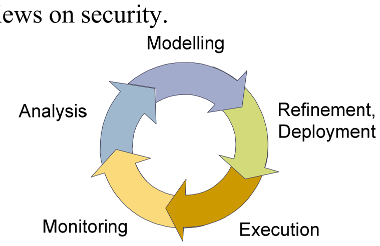

Figure 10. Business Process Management Life Cycle

A. *Process Design*

View transformations are well applicable for information filtering and information summarization in order to simplify process design. However, for process design the views need to be very restrictive to assure that the transformations have a well-formed and consistent outcome. This includes the preservation of order consistency [6], structural consistency (e.g., absence of control cycles) and preservation of executability [7]. Through our current experience concerning the implementation of viewing scenarios, we think that providing support for process design is much harder to achieve than support for analysis due to "ambiguities" which can occur during transformations [2], i.e. a transformation can have multiple possible outcomes.

Abstract process modeling. One issue in the business-IT gap is that processes are modeled and dealt with at different levels of abstraction [25]. It is therefore desirable to be able to model one and the same process on multiple levels of abstraction, for instance a process analyst models a process in high level using BPMN [8], while it is refined and executed using BPEL [9]. Layering concepts can be applied to provide multiple levels of abstraction for process modelling. A shift between different layers can be expressed by view transformations as the illustration in Fig. 11 shows. The aggregation and omission can be applied to all constructs in the process, also to constructs in the metamodel of a view, i.e. to aggregate nodes, inserted nodes etc. A technical difficulty of this scenario is updating the models in lower layers. In other words, the problem is how changes during modeling in the abstract form can be propagated downwards to the more concrete form, because ambiguities may arise [2], [7]. The layering approach was also discussed in [11], in which a methodology to build business models on layers of increasing complexity was proposed. Another work related to this viewing scenario is FlexView [10], a framework to support process abstraction and concretization. The work in [10] also contains an algorithm for realization of process abstraction and concretization in compliance with defined consistency rules.

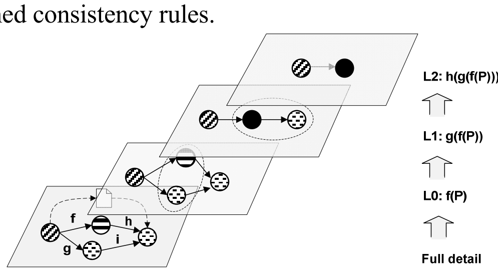

Figure 11. Abstract Process Modeling

Public views for process outsourcing. For integration with other business partners it is necessary to disclose particular parts of the internal process. Abstract processes (i.e. particular views) can be published by an organization to show the business protocol of a process, in other words for showing its public behavior, see Fig. 12. Research on business protocols investigates in particular the notions of compatibility, equivalence and replaceability of processes with respect to the public behavior [13]. The basic problem in this scenario is how to generate the public view for a particular business partner while hiding internal information. Process view transformations are well applicable in this scenario. At first, all parts which are classified as 'confidential' (see Section II, human-assisted augmentation) are omitted. Then, the basic idea is to provide each business partner with a different view by omitting all activities and artifacts that match or do not match particular criteria which are specific for each business partner. For instance, the public view of a travel agency process generated for an airline partner omits the activities related to the hotel booking and car rental. Several works have already been published on this viewing scenario, the authors of [14] for example apply view transformations to generate business protocols for block-structured process languages (similar to UML Activity Diagrams). In [2] we present a view transformation framework based on the process language BPEL [9], which also implements the generation of public views. An important aspect in this scenario is the assessment of consistency between the process and the different views. The conceptual framework for service modelling and refinement discussed in [24] addresses the checking of consistency between service models at different levels of abstraction and is therefore related to this aspect. An open research question is how process views can be transformed to a counterpart that represents a process template which can be used by a business partner. In this sense, process view counterparts might represent an alternative to operating guidelines [12] that can be used for this purpose.

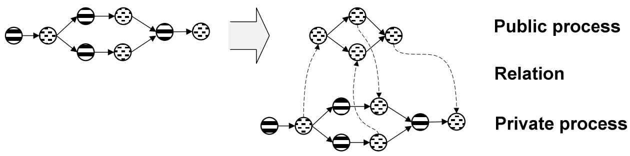

Figure 12. Public Views for Process Outsourcing

Extraction of process logic for reuse. Some parts or even whole structures of a process (so-called fragments) might be reusable in other processes. For efficient reuse an extraction mechanism is needed. Process viewing patterns provide the elementary function blocks to build an application for this task. A view can be defined that reduces a process to the desired structures while preserving control dependency, see Fig. 13. At first, those activities need to be specified which shall be extracted for later reuse. This can either be done by manual selection, or automatically by selecting all constructs that match particular characteristics. In the next step all other activities are omitted while preserving consistency. An inherent problem in this scenario is the occurrence of ambiguities during the transformation. If there are multiple outcomes of the omission steps possible, then either an automatic or manual disambiguation is needed [2]. In [16] the need for reuse in e-Science applications using workflow fragments is discussed. The authors expound the problem of missing tools for efficient extraction and reintegration into service compositions. The presented viewing scenario addresses a part of this need. In [15] we discuss a concrete implementation of this scenario based on the process language BPEL [9]. In that work this viewing scenario is used as a technique to extract process fragments which are recognized as being related to compliance [4] in order to make them reusable.

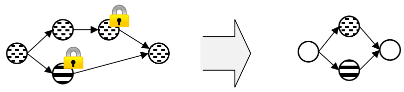

Figure 13. Extraction of Process Logic for Reuse

B. Process Deployment

The viewing scenarios discussed in the following mainly perform information linking. Deployment information that is related to security or the target runtime environment is typically separated from the process. A consolidated view that links this information to a process allows managing this information easier and faster.

View on security. Aspects related to security are typically non-functional requirements which are annotated to a process, stored in a so-called deployment descriptor. For managing this information it is handy to define a view that combines this information with the process. At first, the process has to be augmented with runtime information contained in security-related annotations. Another possibility is human-assisted augmentation regarding information about the required security level. This augmented information can be treated as additional node attributes which can be accessed by a modeling tool. The current security level can be shown in many different ways, for instance by drawing border colors in red, orange and green. In addition, the nodes can be arranged in swimlanes, as shown in Fig. 14. Decorators can also be used to show particular security-related aspects, e.g., if a secured connection is used. Depending on the implementation, this approach can be used to change or define security settings by placing nodes on the particular swimlanes. Some BPM vendors already implement this viewing scenario in their products. For instance, in [20] it is realized by showing a process in a swimlane layout that is bound to a custom classifier. Also related to this scenario is [17] in which view transformations are used to manage access control on the process. Furthermore, in [18] security views for outsourced business processes are described.

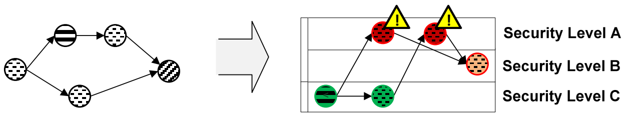

Figure 14. View on Security

Process distribution. For analyzing and improving a process it might be helpful to see how the process is distributed. Distribution refers to where the participants are geographically located and where the programs and services that execute the actual tasks are deployed. This can also be valuable when analyzing a process whose services are distributed over multiple cloud providers (see Fig. 15), or for viewing different time zones of the involved participants. An augmentation step needs to provide information about the location of the programs and the processes participants. Those coordinates are mapped to a graphical entity that represents the theme to be analyzed. In the same manner as in the view on security, this viewing scenario is even more powerful when deployment information can actually be changed by moving the nodes from one place to another.

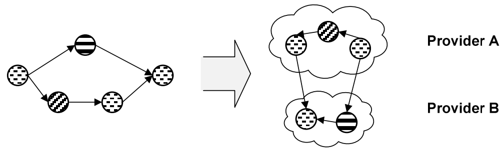

Figure 15. View on Process Distribution

C. Process Monitoring and Instance Management

Process monitoring from a business perspective beyond dashboards means providing an overview of the current state of the instances of a business process. Process views in this context are mainly linking information about an instance of a process to the model of the process. As prerequisite for these views, the runtime information from monitoring tools, from audit trails or from ETL flows (Extract, Transform, Load) needs to be available.

Status of an instance. Audit trails of process engines as well as monitoring components implementing Complex Event Processing (CEP) provide simple or aggregated runtime information about which activities have been executed, which ones are running and so on. Having a clear view on the current status of one or more instances of a process is not only important for technical supervision, but also for business people. Runtime augmentation (see Section II) is a basic prerequisite to link the current status of an instance to its process model. Building on this, the graphical representation (see Section II) can use decorators to show the current execution state, while lowering the contrast for activities that are not yet ready for execution. This solution reveals ambiguities when it is applied on a view, i.e., on an abstracted form of the process. Therefore, the representation of the status of aggregated and omitted nodes and edges needs to be defined, for instance based on number and status of contained nodes as illustrated in Fig. 16. Basically, the principle of this viewing scenario can also be used to visualize an execution simulation. Furthermore, this viewing scenario can be helpful in the task of instance migration. Process views concerning the status of an instance are frequently discussed in research, for instance in [21] and [22]. Besides, Online-Analytical-Processing (OLAP) systems are related to this scenario as they can be used for determination of the status of process instances.

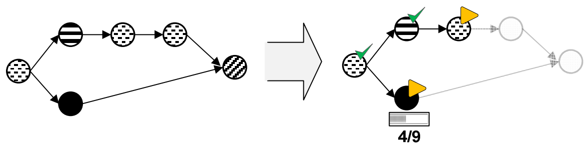

Figure 16. Status of an Instance

Custom business process monitoring. Modeling and execution of processes are often performed at various different levels of abstraction. While users near to business use more high level notations, the technical staff responsible for refinement and execution is more familiar with programming languages. This is just one scenario in which business process monitoring should show something different than the process which is actually executed. An illustration of this scenario is given in Fig. 17. In principle, defining custom views for monitoring is about defining projections of activities and states. For example it is useful to have a custom view on a process instance which is hiding technical details like activities for variable assignment. It might also be feasible to aggregate particular steps or to remove paths which have not been taken, or cannot be reached anymore. Some frameworks already provide support for this viewing scenario, most notably the custom monitoring tools in Java CAPS [26].

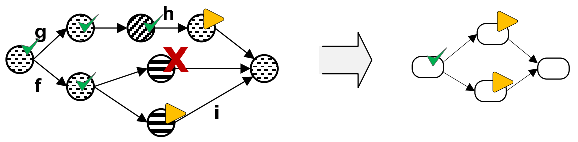

Figure 17. Custom Business Process Monitoring

D. *Visual Process Analysis and Process Intelligence*

Optimization of efficiency in terms of costs, time and automation are essential goals of process analysis and intelligence. The viewing scenarios for process analysis support the achievement of these goals by linking runtime data to a process model and presenting this linked information in a manner easy to understand. The runtime data used in these scenarios needs to be calculated by analysis algorithms in advance.

Bottlenecks. Bottlenecks are hot spots in a process because they require more resources than currently available. Bottlenecks mean costs as they increase the overall duration of the process. In a workflow system a bottleneck can either be rooted in the resources which execute tasks (i.e., services or humans), or in data transmission. A viewing scenario that addresses the analysis of bottlenecks needs to augment the process with runtime information concerning the average or current execution in terms of activity duration, size of transferred data and network bandwidth. This is applicable to a set of process instances (for viewing the average) or just to one single instance. For example, the width of nodes can be bound to the duration of an activity (or other dimensions for costs), see Fig. 18. The transmission time is bound to the thickness of control edges. In addition, also a linear time bar at the lower border is shown. Such viewing scenarios are frequently used for process analysis and process mining, such as in ProM [23]. Another possible viewing solution would be to apply the mechanism known from geographical maps, where red indicates hot regions (i.e. bottlenecks) and blue stands for unproblematic parts.

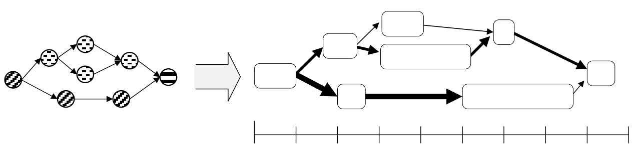
Figure 18. Bottlenecks

Probable execution path. When it comes to human decision points in a business process, the person who has to decide might like to see the probable execution path as a help for making the decisions. A similar problem is how to make recommendations how to proceed, for instance in a manner like the shopping proposals made in some online shops (customers who liked this item also liked…). In order to define a view that addresses this need, runtime augmentation is required in a first step for enrichment of the process with probabilities of further execution, based on the status of a particular instance. Presentation shows the current status of the instance using decorators. Next, it binds the size of nodes to the calculated probability value. For clarity a maximum and minimum size for 100% and 0 % is set, and for nodes with 0% probability the contrast is decreased. In addition, showing a probability decorator can visualize the probability of occurrence of events like faults. The resulting view highlights the probable execution part for the rest of the processing, see Fig. 19. Optionally, improbable steps could even be omitted.

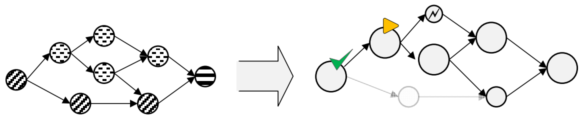
Figure 19. Probable Execution Path

Path analysis. Having a clear view on the longest or most expensive path in a process is essential in business reengineering. Furthermore, having an immediate visual feedback on whether a changed design will lead to a cheaper execution or not is also desirable. Defining a process view for this scenario requires an augmentation of the process with either runtime information or estimations about average costs of activities. Various different definitions for costs are applicable, for instance costs in terms of duration, costs of workforce, costs of used infrastructure etc. The augmented process can be taken as an input for graph analysis algorithms that calculate for example the shortest, the cheapest, the longest, the most expensive, the most frequently used and the least used path. These paths can for instance be presented with different colors for edges (see Fig. 20). Performing further view transformations can then help improving the process: Omission and Insertion of nodes allow analyzing variants of the process. If inserted nodes also contain estimated costs, then immediate feedback concerning the impact on the critical paths can be shown. Existing applications that implement similar functionality can be found in works on process monitoring like in [27] or in the field of business intelligence [28].

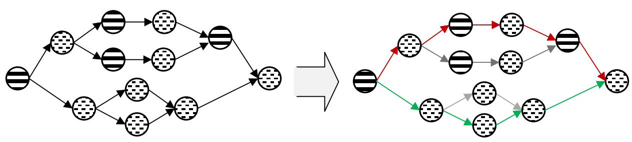
Figure 20. Path Analysis

Process automation. Nowadays, a business process often is a mixture of activities which are executed by either humans or automated with software, e.g., using Web services. Getting a clear insight which parts of a process are automated and which ones are not, is useful in many scenarios of process analysis, e.g., when improving overall performance. Often it can be derived from activity properties, which ones are executed by programs and which by humans. If it is not possible to determine this with the available information, then runtime augmentation can provide this information. A view can display this information, for instance using a distinct decorator for each involved role (see left part in Fig. 21). When also using omission it is even possible to show those aspects separately: The upper view transformation results in the human process graph and the lower one shows the automation graph. This scenario has also been proposed in an earlier work on use cases for process views [3].

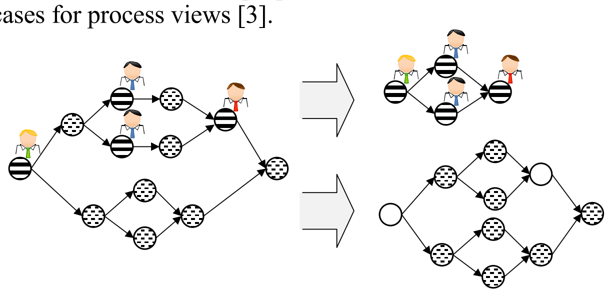

Figure 21. Process Automation

E. General Purpose

Some viewing scenarios cannot be clearly assigned to one particular phase in the process management life cycle. Besides, they lend themselves for composition with other view transformations. That means that they are applicable as views on other views.

Access control on the process model. In almost any kind of application related to business, role-based access control is common practice. It also applies for business processes and the corresponding models that people with particular roles should only have limited access. This implies limited visibility of the contained structures. For example, people concerned with technical refinement may not modify activities related to salary handling, while the technical parts should not be changeable by people from the human resource department. The basic idea is to show the granted parts, while the rest is either invisible or blurred and locked. This scenario is similar to business process outsourcing, but here the focus is set on modeling. The users should be able to make changes on the process via the view. In order to realize this viewing scenario, a human has to augment the process with access control information about the roles and access rights. A straightforward solution is then the abstraction of all "restricted" artifacts to show that something happens but it is not clear *what* happens, see Fig. 22. The works related to this scenario confirm that providing tool support for this scenario is straightforward for visualization, but providing modeling support is quite challenging. The problem is the disambiguation that has to be performed for cases where updates of the original process have to be performed, although a part of the process is hidden or locked. This scenario has also been proposed in [7] and [17], the technical issues to support modeling however are unsolved yet.

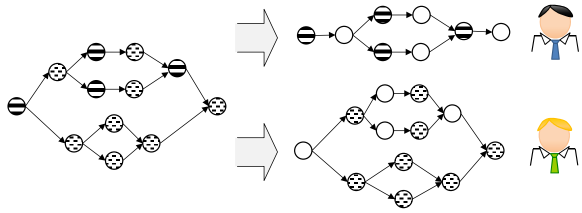

Figure 22. Access Control on a Process Model

Set a focus. Business processes are quickly becoming complex and "crowded" with activities and control structures. For many situations it is useful to focus on one or more "points" (i.e., activities) in the process. To set a focus on one or more particular areas, the nodes and artifacts which are "outside" the focus can be omitted as they are less interesting. A viewing scenario supporting this function needs user input. One or more activities need to be selected for defining the center of the focus. Furthermore, an offset for predecessor and ancestor nodes needs to be defined, i.e. the LookBack and LookAhead of the focus, see Fig. 23 top right. This view is applicable to be composed with other views and it can be implemented in a straightforward manner. All nodes surrounding the focus are removed and within the focus all nodes are preserved. No ambiguities occur when changes within the focus are made. This means that changes can directly be synchronized with the original process without requiring user interaction. For viewing a process instance also a point in time can be used as focus definition, and a time interval can be used as an offset to define the zoom factor of the focus, see Fig. 23 bottom right. Works related to this scenario can be found in the field of visualization techniques, like so-called fisheyeing [31], [32].

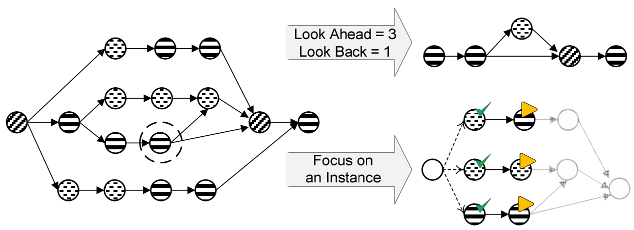

Figure 23. Set a Focus

Abstraction from details. Process abstraction is a viewing scenario to reduce complexity by removing details which are of minor importance for the understanding. For instance, business stakeholders are interested in business exception handling (e.g., out of stock) and not in technical exception handling (e.g., cast exception). Additionally, in many cases multiple activities are serving only one single purpose (e.g., charge a credit card) and can be aggregated into one activity as depicted in Fig. 24. In our implementation we have made good experience with human-assisted augmentation for providing a distinction of activities and artifacts into different confidentiality levels. This augmentation is in the same manner applicable for different levels of importance. Also technical attributes of artifacts can be taken into account to decide about its importance. This scenario is frequently discussed and most thoroughly investigated in current research. Often techniques for automated aggregation and omission of structures are addressed. In [1] a technique is presented that can be used for automatic aggregation of sequential structures. In [29] a slider approach for flexibly defining the level of abstraction is discussed. Similar approaches are also implemented in some products. All those approaches share the common ground of applying graph reduction techniques (like [30]) to process models. For more works related to this kind of abstraction please see our state of the art discussion in [2].

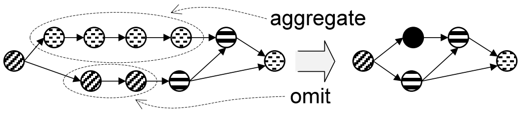

Figure 24. Abstraction from Details

We are currently investigating the benefits of semantics for this purpose. To be more precise, we recognize and aggregate particular structures related to compliance [4] as well as patterns which are known to have a particular meaning. An example for such a pattern is given in Fig. 25. Human-assisted augmentation using ontology languages are well supported by free software [19] and represent a solution in this context. The work in [34] is quite useful as well, as it proposes a concrete scheme for activity naming. This allows using techniques for automated graph clustering like [33] without requiring the usage of ontologies or similar concepts.

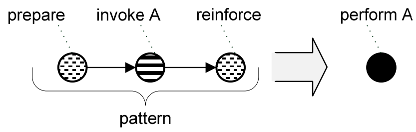

Figure 25. Labeling of Aggregate Nodes using Pattern Recognition

Custom appearance. Business process management is made more efficient when complying with the best practice on using labels and icons in business processes because many misunderstandings can be prevented [34]. In addition, using shapes that represent the meaning of the activities can ease understanding as well, as we showed in [5]. In [36] it is furthermore discussed that different groups of people may also have a different understanding of graphical elements and notations. Thus, for different stakeholders in process management different shapes might be better for understanding. Viewing applications can provide a solution to this by allowing to flexibly mapping activities to meaningful shapes. Many offerings implement this functionality already. Human-assisted augmentation can provide a classification of activities, possibly based on ontology. Depending on this classification and the target audience, particular shapes can be displayed. Activity names, attributes and other technical information (e.g., service properties) can be used as classifier as well. An example for this scenario is illustrated in Fig. 26. The shapes used in this illustration are basically irrelevant, they just serve to exemplify for the concept.

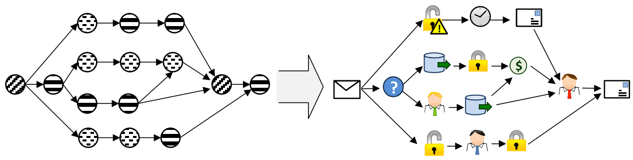

Figure 26. Custom Appearance

V. Conclusion and Outlook

In this paper we have shown a variety of application scenarios to support enterprises in the management of complex business processes. We believe that the process views presented here have the potential to increase the efficiency of process design and graphical process analysis dramatically. We also see significant benefit for usage of views in process monitoring and deployment. Besides, views can be combined to provide views on views. For example, one could be interested in (i.) the process distribution of (ii.) the current status of an instance, (iii.) limited to a particular focus area. Thereby the incremental sequence of viewing transformations results in new kinds of views. Overall, most of the views we presented in this paper have a technical focus. For business users other views might be more relevant, this is subject to further research.

In some of the presented application scenarios we have made use of human-assisted augmentation. We have come to the conclusion that knowledge about the semantics of activities has enormous impact on the power of process viewing. When nodes are annotated with their meaning, more advanced functionality can be provided. Abstract structures can be defined to describe higher level structures, for instance in [4] we defined a process fragment for an approval. When such structures are recognized in a process, they can be automatically highlighted, or transformed into meaningful aggregates with corresponding shapes or labels.

Along with increasing adoption of viewing concepts, the traditional role of process modeling tools is going to change. Support for both abstract and low-level process modeling as well as instance management will turn the current tools into a powerful and flexible platform moving towards Business Intelligence. Viewing concepts are not only in the business domain useful, they can also be applied in other domains in which processes are used. For instance, views can be helpful for analysis of software processes defined in UML Activity Diagrams or for medical treatment guidelines to name just a few examples.

Acknowledgment

The work published in this paper was partially funded by the COMPAS project [1] (FP7-215175) and the MASTER project [2] (FP7-216917), both projects under the EU 7th Framework Programme ICT Objective. Many thanks go to Ralph Mietzner for his constructive feedback.

[1] http://www.compas-ict.eu/
[2] http://www.master-fp7.eu/

REFERENCES

[1] J. Vanhatalo, H. Völzer, F. Leymann. Faster and more focused Control-flow Analysis for Business Process Models though SESE Decomposition. Proceedings of 5th International Conference on Service-Oriented Computing (ICSOC), Springer, 2007.

[2] D. Schumm, F. Leymann, A. Streule. Process Viewing Patterns. Proceedings of the 14th IEEE International EDOC Conference (EDOC 2010), IEEE Computer Society Press, 2010, in press.

[3] S. Rinderle, R. Bobrik, M. Reichert, T. Bauer. Business Process Visualization – Use Cases, Challenges, Solutions. Proceedings of the International Conference on Enterprise Information Systems (ICEIS), pages 204–211, INSTICC Press, 2006.

[4] D. Schumm, F. Leymann, Z. Ma, T. Scheibler, S. Strauch. Integrating Compliance into Business Processes: Process Fragments as Reusable Compliance Controls. Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI'10), 2010.

[5] D. Schumm, D. Karastoyanova, F. Leymann, J. Nitzsche. On Visualizing and Modelling BPEL with BPMN. Grid and Pervasive Computing Workshops: 4th International Workshop on Workflow Management (ICWM), IEEE Press, 2009.

[6] D. Liu, M. Shen. Workflow Modeling for virtual Processes: an Order-preserving Process-view Approach. In: Information Systems, 28(6):505–532, Elsevier, 2003.

[7] M. Chinosi. Representing Business Processes: Conceptual Model and Design Methodology. PhD Thesis, Università degli studi dell'Insubria, Varese, Italy, 2008.

[8] Object Management Group (OMG). Business Process Modeling Notation Version 1.2. OMG Standard, 2009.

[9] Organization for the Advancement of Structured Information Standards (OASIS). Web Services Business Process Execution Language Version 2.0. OASIS Standard, 2007.

[10] X. Zhao, C. Liu, W. Sadiq, M. Kowalkiewicz, S. Yongchareon. WS-BPEL Business Process Abstraction and Concretisation. Proceedings of the 14th International Conference on Database Systems for Advanced Applications, Springer, 2009.

[11] I. Salomie, T. Cioara, I. Anghel, M. Dinsoreanu, T. Salomie. A Layered Workflow Model Enhanced with Process Algebra Verification for Industrial Processes. Proceedings of the 3rd International Conference on Intelligent Computer Communication and Processing, IEEE, 2007.

[12] N. Lohmann, P. Massuthe, K. Wolf. Operating Guidelines for Finite-State Services. Proceedings of the 28th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency (ICATPN), Springer, 2007.

[13] B. Benatallah, F. Casati, F. Toumani. Analysis and Management of Web Service Protocols. Proceedings of the 23rd International Conference on Conceptual Modeling, Springer, 2004.

[14] R. Eshuis, P. Grefen. Constructing Customized Process Views. In: Data & Knowledge Engineering, 64(2):419– 438, Elsevier, 2008.

[15] D. Schumm, F. Leymann, A. Streule. Process Views to Support Compliance Management in Business Processes. In: Proceedings of the 11th International Conference on Electronic Commerce and Web Technologies (EC-Web), Springer, 2010, in press.

[16] A. Goderis, U. Sattler, P. Lord, C. Goble. Seven Bottlenecks to Workflow Reuse and Repurposing. Proceedings of the 4th International Semantic Web Conference, Springer, 2005.

[17] S. Bassil, M. Reichert, R. Bobrik, T. Bauer. Access Control for Monitoring System-Spanning Business Processes in Proviado. Proceedings of the 3rd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'09), Koellen-Verlag, 2009.

[18] A. Benameur, F. Massacci, N. Rassadko. Security Views for Outsourced Business Processes. Proceedings of the 2008 ACM Workshop on Secure Web Services, ACM NY, 2008.

[19] Stanford Center for Biomedical Informatics Research. Protégé, 2010. http://protege.stanford.edu/

[20] International Business Machines Corp. WebSphere Business Modeler, 2009. http://www-01.ibm.com/software/integration/wbimodeler/

[21] R. Bobrik, T. Bauer, M. Reichert. Proviado - Personalized and Configurable Visualizations of Business Processes. Proceedings of the 7th International Conference on Electronic Commerce and Web Technologies (EC-Web), Springer, 2006.

[22] S. Rinderle, U. Kreher, M. Lauer, P. Dadam, M. Reichert. On Representing Instance Changes in Adaptive Process Management Systems. Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE Computer Society, 2006.

[23] Process Mining Group. ProM, 2009. http://prom.win.tue.nl/tools/

[24] D.A.C. Quartel, M.W.A. Steen, S. Pokraev, M.J. van Sinderen. COSMO: a Conceptual Framework for Service Modelling and Refinement.In: Information Systems Frontiers, 9(2):225-244, Springer, 2007.

[25] M. Hepp, F. Leymann, J. Domingue, A. Wahler, D. Fensel. Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE), IEEE Computer Society, 2005.

[26] Java CAPS Business Process Monitor. Custom Monitoring tools with Java CAPS, 2009. http://wikis.sun.com/display/JavaCAPS/Graphical+Business+Process+Monitoring

[27] M. Zur Muehlen, M. Rosemann. Workflow-based Process Monitoring and Controlling-technical and Organizational Issues. Proceedings of the 33rd Hawaii International Conference on System Science (HICSS-33), IEEE Computer Society, 2000.

[28] D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, M. Shan. Business Process Intelligence. In: Computers in Industry, 53(3):321-343, Elsevier, 2004.

[29] A. Polyvyanyy, S. Smirnov, M. Weske. Process Model Abstraction: A Slider Approach. Proceedings of the 12th IEEE Enterprise Distributed Object Conference (EDOC), IEEE Computer Society, 2008.

[30] W. Sadiq, M. Orlowska. Analyzing Process Models Using Graph Reduction Techniques. In: Information systems 25(2):117–134, Elsevier, 2000.

[31] W. Citrin, R. Hall, C. Santiago, B. Zorn. Addressing the Scalability Problem in Visual Programming through Containment, Zooming and Fisheyeing. Proceedings of the IEEE Aerospace Conference, IEEE, 1998.

[32] C. Tominski, J. Abello, F. van Ham, H. Schumann. Fisheye Tree Views and Lenses for Graph Visualization. Proceedings of the 10th International Conference on Information Visualization (IV), IEEE, 2006.

[33] O. Maqbool, H. Babri. Automated Software Clustering: An Insight using Cluster Labels. In: The Journal of Systems and Software, 79 (2006):1632–1648, Elsevier, 2006.

[34] J. Mendling, J. Recker. Towards Systematic Usage of Labels and Icons in Business Process Models. Proceedings of the 13th International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD), CEUR-WS, 2008.

[35] J. Abello, F. van Ham, N. Krishnan. ASK-GraphView: A Large Scale Graph Visualization System. In: IEEE Transactions on Visualization and Computer Graphics, 12(5), IEEE, 2006.

[36] M. Petre. Why Looking Isn't Always Seeing: Readership Skills and Graphical Programming. In: Communications of the ACM, Vol. 38, No. 6, June 1995.