

Visualizing Large Business Process Models: Challenges, Techniques, Applications

Manfred Reichert

Institute of Databases and Information Systems, Ulm University, Germany
`manfred.reichert@uni-ulm.de`

Abstract. Large process models may comprise hundreds or thousands of process elements, like activities, gateways, and data objects. Presenting such process models to users and enabling them to interact with these models constitute crucial tasks of any process-aware information systems (PAISs). Existing PAISs, however, neither provide adequate techniques for visualizing and abstracting process models nor for interacting with them. In particular, PAISs do not provide tailored process visualizations as needed in complex application environments. This paper presents examples of large process models and discusses some of the challenges to be tackled when visualizing and abstracting respective models. Further, it presents a comprehensive framework that allows for personalized process model visualizations, which can be tailored to the specific needs of the different user groups. First, process model complexity can be reduced by abstracting the models, i.e., by eliminating or aggregating process elements not relevant in the given visualization context. Second, the appearance of process elements can be customized independent of the process modeling language used. Third, different visualization formats (e.g., process diagrams, process forms, and process trees) are supported. Finally, it will be discussed how tailored visualizations of process models may serve as basis for changing and evolving process models at a high level of abstraction.

1 Introduction

Many companies have to deal with a large number of business processes involving numerous tasks, data objects, organizational entities, and resources. Usually, these processes are captured in process models, which are stored in large process repositories comprising hundreds or even thousands of process models [1]. In turn, each of these process models may comprise a large number of activities and involve a multitude of stakeholders. In practice, each stakeholder may require a different perspective on the processes he or she is involved in, providing a customized visualization and information granularity. For example, managers rather prefer an abstract overview, whereas process actors need a detailed view of those process parts they are involved in. Hence, a personalized process visualization is a much needed feature to be provided by any process-aware information system (PAIS).

Despite its practical importance, current PAISs do not offer adequate process visualization support. Usually, process models are displayed to the user in the same way as drawn by the process designer. However, these process models are often too complex (see Fig. 1 for an example) and, hence, are not comprehensible to end-users (e.g., when containing data transformation steps or other kinds of technical activities). Some tools allow altering the graphical appearance of a process and hiding selected process aspects (e.g., data flow). However, more sophisticated and flexible process visualization concepts are still missing in most PAISs.

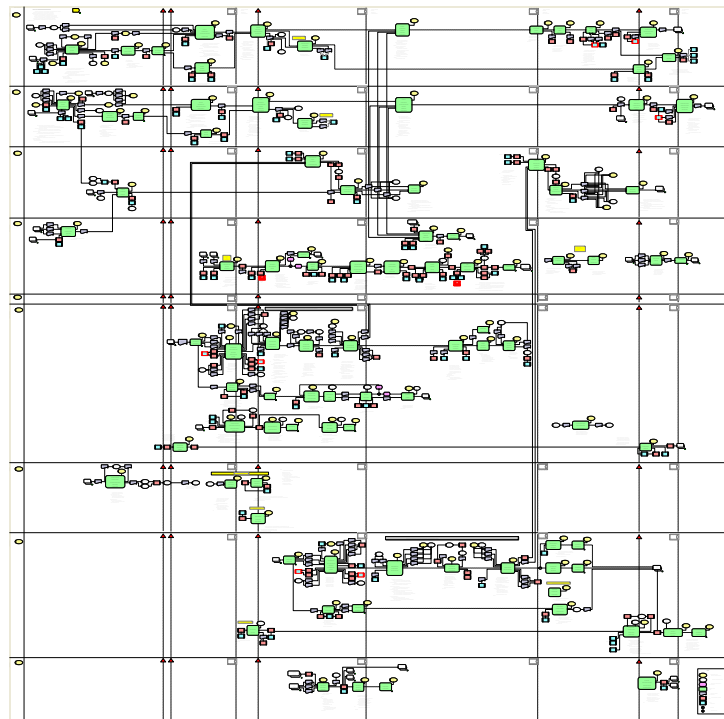


Fig. 1. Example of an engineering process model (partial view)

To elaborate basic visualization requirements we conducted several case studies [2], in which we identified three fundamental process visualization dimensions. First, it must be possible to reduce complexity by hiding or aggregating process information not relevant in the given context. Second, the notation and appearance of process nodes (e.g., activities and data objects) shall be customizable. Third, different visualization formats (e.g., process graph, table) need to be supported. This paper summarizes the Proviado framework that addresses all three visualization dimensions. Section 2 summarizes basic considerations required for

understanding this paper. Section 3 presents the Proviado visualization framework, while Section 4 illustrates its use by means of an example. Finally, next steps in our research are discussed in Section 5.

2 Basic Considerations

Generally, any process visualization tool should distinguish between the model and instance level (cf Fig. 2). The former gathers various enterprise models, including organization, function, data, IT system, and process models. Thereby, a process model refers to elements of the other models and comprises a set of inter-connected activities, collectively realizing a certain business objective [3]; i.e., the activities are executed in a coordinated manner by different entities (e.g., humans and software agents) to reach process goals such as changing the design of a product, delivering merchandise, or treating a patient [4]. Furthermore, user- as well as pre-defined attributes may be associated with process models or activities (e.g. costs, needed resources), and hence become relevant in the context of process visualizations [5]. Examples of frameworks supporting the integrated modeling of the different enterprise aspects include ArchiMate [6], ADONIS [7], and ARIS [8].

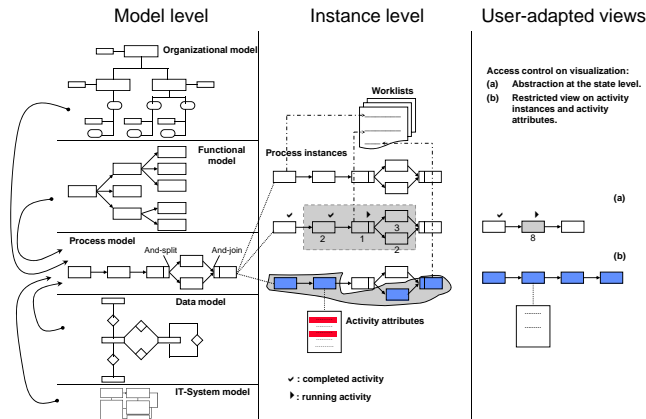


Fig. 2. Basic considerations

At the *model level*, a secure visualization of data related to a particular process model is required. An example of such a process model is depicted in Fig. 3. It shows a simplified model of a *change request process* as it can be found in the automotive domain. More precisely, this process model comprises five phases with 20 different activities in total. Furthermore, the control and data flow between activities, exceptional paths, role assignments, and IT system resources are shown.—Note that more complex variants of this change request process exist in practice (see Fig. 4 for an example from the automotive domain).

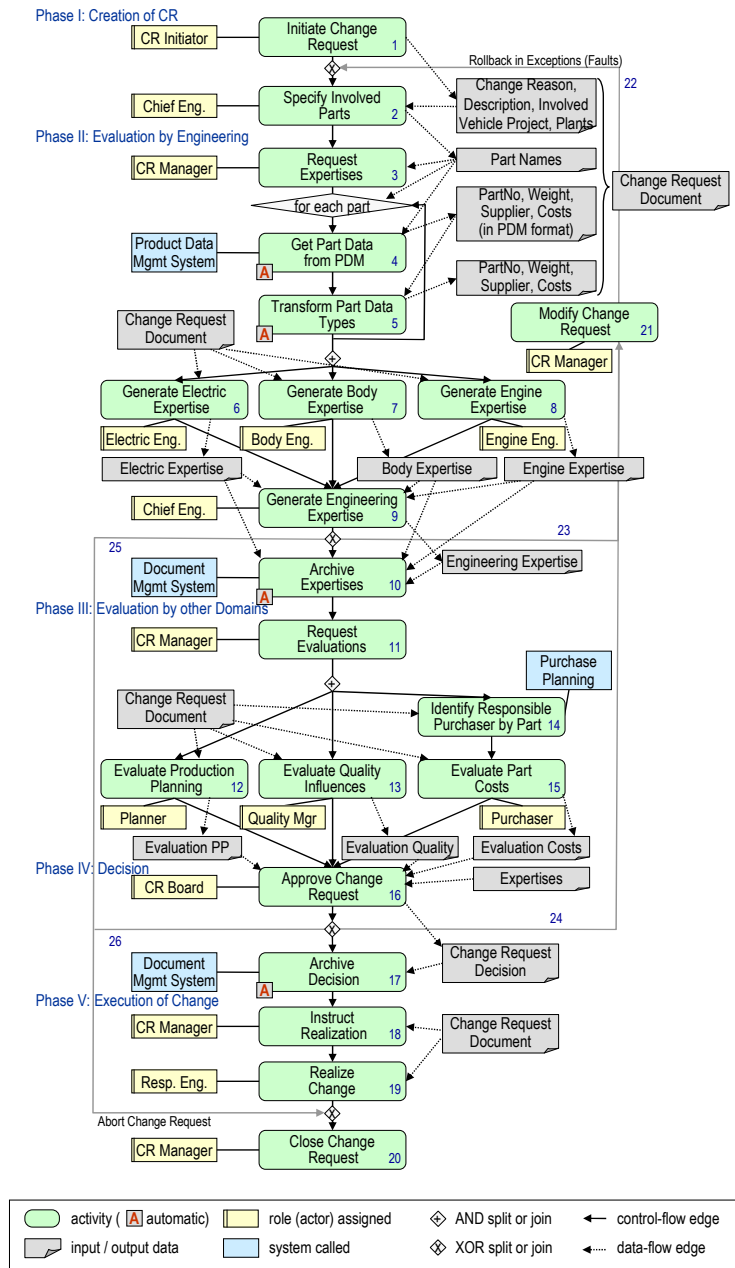


Fig. 3. Process model of a change request (CR)

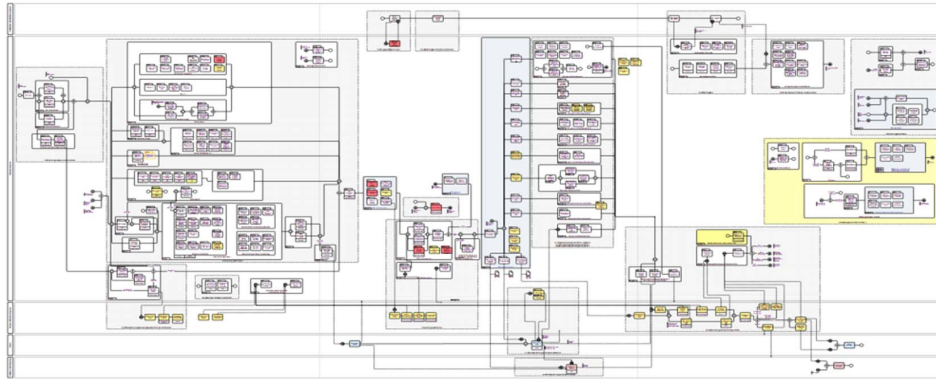


Fig. 4. A more complex variant of a change request process

In this paper, we will use the process model from Fig. 3 to illustrate basic process visualization concepts. Furthermore, we will show how a process model may be enriched with run-time data and then be displayed to authorized users [9]. Overall, flexible *configuration* and *personalization* of large process models are fundamental for the user-friendly visualization of these models.

At the *instance level*, a secure monitoring of running process instances is needed. More precisely, a process instance represents a particular enactment of a business process model (i.e., a single business case). Artifacts such as user work lists, activity execution states (e.g. **Running**), and invoked application services are characteristic for the instance level and hence need to be considered [3].

3 The Proviado Framework

The Proviado framework targets at a flexible and configurable visualization of large business process models and related process instances [2, 10, 11, 9]. In particular, respective process visualizations must be customizable to the specific needs of the different stakeholders involved the process [2]. In this context, three dimensions need to be considered: First, it must be possible to reduce process model complexity by eliminating or aggregating information not relevant in the given context or for which the user does not have sufficient access rights. Second, the appearance of process elements (e.g., activities, data objects, control and data edges) should be customizable to user preferences, independent from the way the source process model is represented. Third, different visualization formats (e.g., process diagrams, Gantt charts, tables, or forms) need to be supported to cope with different user preferences.

For realizing a particular visualization of a process model and process instance, respectively, Proviado allows specifying related visualization models separately from the given process model. Such a visualization model comprises a number of configuration settings, which determine the process elements to be

displayed and the graphical notation to be used. In particular, respective configuration settings can be specified at a high level of abstraction based on a sophisticated process view concept as well as on a flexible template mechanism.

3.1 Process View Concept

The view generation approach provided by Proviado [11, 12] allows reducing the complexity of business process models through abstracting them. This model abstraction, in turn, is accomplished by applying well-defined transformation rules, which rely on model reduction as well as model aggregation techniques. As a result one obtains an abstracted *process view* of the original process model. *Model reduction operations* may be applied in order to hide (i.e., delete) selected elements of a process model. For example, consider Fig. 5: activities E, F and G are removed from the process model and a new control edge is inserted instead. Further, Fig. 5 illustrates the use of *aggregation operations*. For example, **Aggregate(B,C,H,K)** aggregates four activities and replaces them by one abstract node in the process model. Depending on the concrete structure of the sub-graph, induced by the set of activities to be aggregated, different model transformations become necessary. While in some cases simple model transformations are sufficient, in other scenarios a more complex restructuring of the process model to be visualized is required. Generally, realizing model abstractions based on aggregation operations is more difficult compared to the use of reduction operations. In particular, the relations the activities show in respect to their satellite objects (e.g., data elements, organizational entities) need to be preserved (cf. Fig. 5). Furthermore, for an abstracted node, its attribute values must be determined based on the attributes of all activities aggregated. Finally, aggregation operations are provided for all process aspects including data flow and actor assignments [12].

It is noteworthy that the Proviado view-building operations maintain the soundness of a process model if required. However, to introduce additional flexibility for process visualization, operations may violate certain structural model constraints if favorable (see [12] for a detailed discussion on this). Furthermore, complex view-building operations based on elementary aggregation and reduction operations are provided; e.g., "Show a process view containing all activities performed by a particular user role." Overall, these high-level view-building operations ease the definition and creation of meaningful process views significantly [12]).

3.2 Proviado Template Mechanism

While the described view-building approach allows us to define which process elements shall be displayed, the *Proviado Template Mechanism* [10] enables a flexible configuration of the graphical appearance of the different elements of a process model. For this purpose, Proviado provides a sophisticated template mechanisms. More precisely, a *visualization template* describes the concrete notation (i.e. the symbols) to be used for visualizing a particular process element

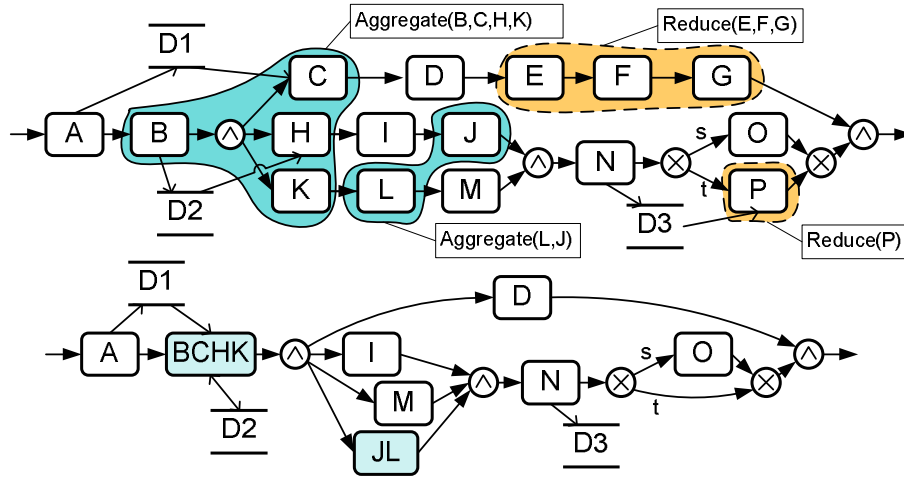


Fig. 5. Proviado view concept

(e.g. an activity or a data object). Its graphical appearance (e.g. shape, arrow), in turn, is described using SVG (Scalable Vector Graphics). Based on this XML format, to a large degree, templates may be graphically defined using a standard SVG Editor.

Each template comprises a set of data fields (i.e. parameters) that may be filled with concrete process attribute values (e.g. activity name or activity state) at visualization time. Proviado uses XPath expressions to establish the relationship between symbol definition and data fields. Required data transformations (e.g. date format conversion) can be realized via ECMA-Script expressions. Altogether, a complete notation for process visualization comprises a set of templates. More precisely, each process element must be linked to a template. This link can be established statically (i.e. remain unchanged) or dynamically based on selected process data (e.g. depending on the runtime status of the process element). For example, the latter enables the use of different symbols for activities, e.g. depending on their state or the actor working on them. Finally, *Cascading Style Sheets* are used to vary the look of process drawings. Fig. 6 shows an example of a respective template.

Overall, the Proviado template mechanism allows for the use of a tailored process notation in a non-ambiguous and easy to maintain manner. In combination with the view concept described above, personalized process visualizations become possible. While non-relevant process elements can be removed or aggregated with other objects, the visualization of relevant process elements can be customized to specific user or application needs.

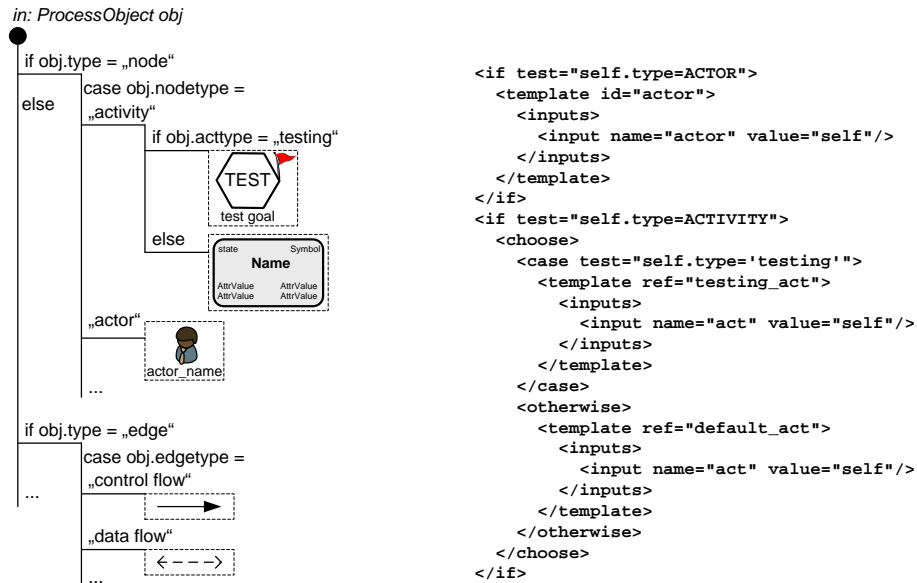


Fig. 6. Example of a Proviado template

3.3 Proviado Visualization Formats

In addition to the two visualization dimensions presented so far, Proviado enables different visualization formats for one and the same business process and further allows users to dynamically switch between these different visualizations. Examples include process diagrams, Gantt charts, trees (see [13] for a concrete approach), tables, and form-based visualizations (cf. Fig. 7).

3.4 Configuring a Process Visualization

Fig. 8 summarizes the basic steps required to automatically generate a particular process visualization. The starting point is an integrated process model, which correlates fragmented process data from different source information systems in a harmonized way. First, we restrict the visualization to that information needed by a particular user (S0). This is realized by a view component, which applies the sketched aggregation and reduction techniques to the given process model. Step S0 is followed by formatting steps S1, S2, and S3: S1 fixes the graphical symbols designed for the different process elements. For this purpose, Proviado considers information from a *visualization model*; S2 fills graphical symbols with real attribute values related to the process model or process instance that shall be displayed. Finally, in step S3 formatting parameters are customized to user preferences, e.g. by coloring the process visualization in accordance to cooperate identity guidelines.

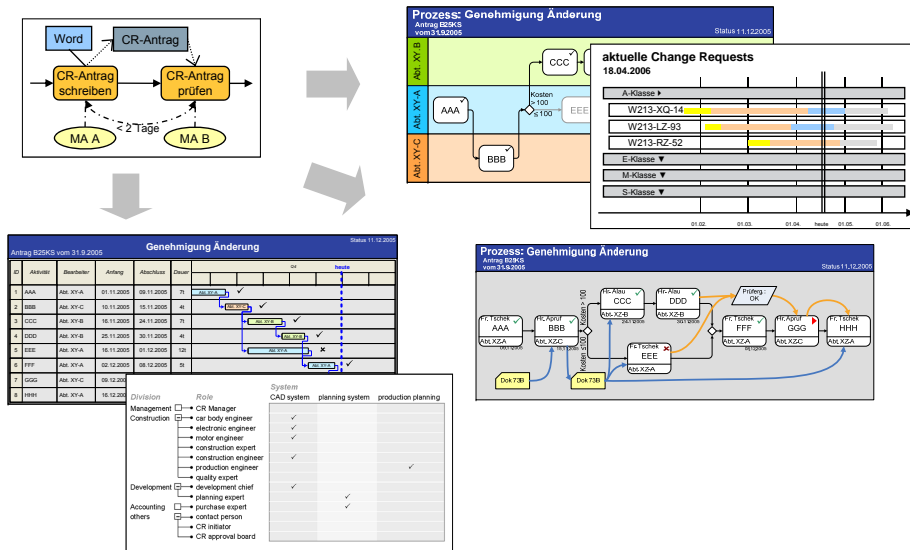


Fig. 7. Examples of different visualization formats

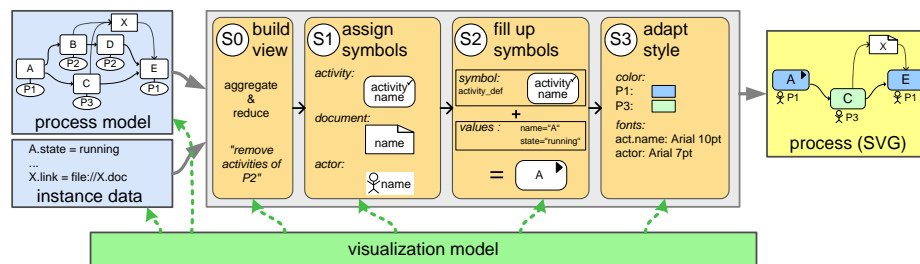


Fig. 8. Generating a process visualization in Proviado

4 Application Example

Consider again the process model from Fig. 3. Assume that an instance of this process shall be visualized for an actor from the engineering domain. For this purpose, non-relevant process elements have to be discarded. Automated steps for transforming and exchanging data (e.g. Steps 4 and 5), for example, shall not be displayed. The same applies to selected interactive steps (e.g. Steps 2 and 3). Finally, control edges capturing forward and backward jumps shall be removed. Altogether this process view can be realized by applying the following view-building operations (listed in brackets for each operation):

*Aggregation:*¹ {1, 2}, {11, 12, 13, 14, 15}
Reduction: {3}, {4, 5}, {10}, {17, 18}, {20}, {21}
DeleteEdge: {22, 23, 24}, {25, 26}

The resulting process view would still contain a large number of satellite nodes (representing actors, systems, etc.) which usually shall not be displayed. Proviado visualization models allow omitting such nodes and assigning their data values to other visualization objects, e.g. activity boxes (cf. Fig. 9). Furthermore, with the *Proviado Template Mechanism* any desired appearance of the process view to be displayed can be realized. For example, the visualization from Fig. 9 contains information like change reason, change description, and involved product parts. Furthermore, a header has been added. Other data like a detailed change request (CR) description can be accessed via a tool tip. Finally, activities being of particular importance for engineers are highlighted.

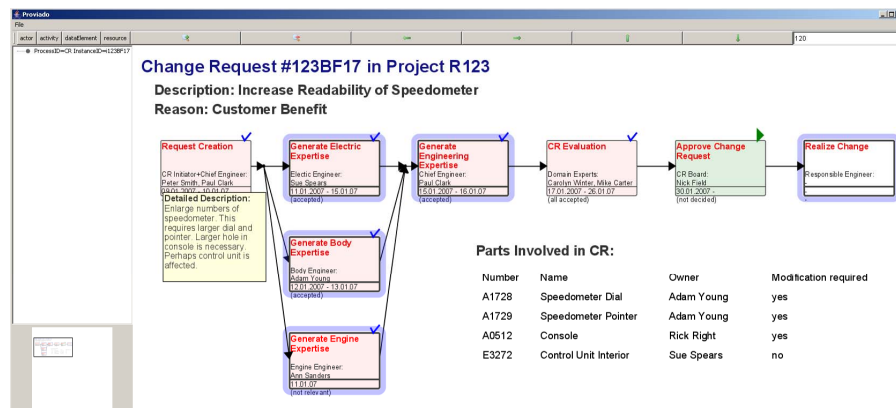


Fig. 9. Visualization of a CR process instance for engineers

Note that the created process drawing (cf. Fig. 9) constitutes one possible abstracted visualization of the process model from Fig. 3. Depending on specific user requirements, for example, Proviado allows providing different visualizations of the same process view, e.g. using a standardized notation like BPMN. Basic to this exchangeability of visual representations is the *Proviado Template Mechanism*. Generally, different information and layouts can be presented. Furthermore, new process views (with same or different appearance) can be easily realized. For example, for managers each of the five phases of the CR process could be aggregated to one single activity and only information about deadlines, delays, resources, and the final decision be visualized (cf. Fig. 10)

¹ Each operation is listed in brackets. The aggregations result in activities "Request Creation" and "CR Evaluation"

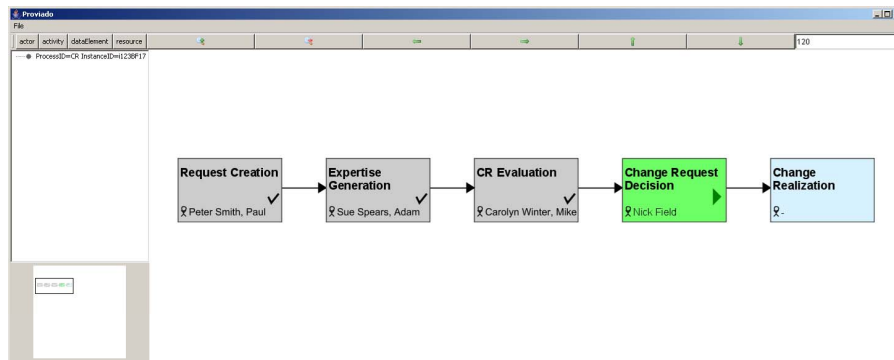


Fig. 10. Visualization of a CR process instance for managers

5 What are the Next Steps?

In the meantime, several other approaches for creating process model abstractions based on process views have been proposed (e.g., [14, 15, 16]). Like Proviado, these proposals focus on creating and visualizing abstracted process views, but do not consider another fundamental aspect of PAISs: change and evolution [3, 17]. More precisely, they do not allow changing a large process model through editing or updating any of its view-based abstractions. As a consequence, process changes still must be directly applied to the original process model, which constitutes a complex as well as error-prone task for domain experts, particularly when confronted with large process models. To overcome this limitation, in addition to view-based process abstractions, users should be allowed to change large process models through updating respective process views. However, this must not be accomplished in an uncontrolled manner to avoid inconsistencies or errors.

In the proView project we address these and other challenges by providing powerful view-creation operations similar to the ones suggested in the context of Proviado; i.e., the operations allow abstracting process models through the reduction and aggregation of process elements as well as through changes of the process model notation. In addition, view-update operations allow adapting process views and propagating the respective changes to the underlying process model as well as to other related process views [18, 19]. A series of user experiments is planned to evaluate whether process model abstractions and tailored process visualizations contribute to ease the change and evolution of complex process models.

References

1. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring Large Process Model Repositories. *Computers in Industry* **62**(5) (2011) 467–486

2. Bobrik, R., Reichert, M., Bauer, T.: Requirements for the visualization of system-spanning business processes. In: Proc. DEXA'05 Workshops, Copenhagen (2005) 948–954
3. Reichert, M., Weber, B.: Enabling Flexibility in Process-aware Information Systems - Challenges, Methods, Technologies. Springer (2012)
4. Mutschler, B., Reichert, M., Bumiller, J.: Unleashing the effectiveness of process-oriented information systems: Problem analysis, critical success factors and implications. IEEE Transactions on Systems, Man, and Cybernetics **38**(3) (2008) 280–291
5. Michelberger, B., Mutschler, B., Reichert, M.: Process-oriented information logistics: Aligning enterprise information with business processes. In: 16th IEEE International EDOC Conference (EDOC 2012), IEEE Computer Society Press (2012)
6. Groenewegen, J., Hoppenbrouwers, S., Proper, E.: Playing archimate models. In: Enterprise, Business-Process and Information Systems Modeling. 11th Int. Workshop BPMDS'10 and 15th Int. Conference EMMSAD'10. Volume 50 of Lecture Notes in Business Information Processing. (2010) 182–194
7. Kühn, H., Bayer, F., Junginger, S., Karagiannis, D.: Enterprise model integration. In: Proc. 4th Int. Conf. E-Commerce and Web Technologies (EC-Web'03). (2003)
8. Davis, R.: ARIS Design Platform: Advanced Process Modelling and Administration. Springer (2008)
9. Reichert, M., Bassil, S., Bobrik, R., Bauer, T.: The Proviado access control model for business process monitoring components. Enterprise Modelling and Information Systems Architectures - An International Journal **5**(3) (2010) 64–88
10. Bobrik, R., Bauer, T., Reichert, M.: Proviado – personalized and configurable visualizations of business processes. In: Proc. EC-WEB'06. LNCS 4082 (2006) 61–71
11. Bobrik, R., Reichert, M., Bauer, T.: View-based process visualization. In: Proc. BPM'07. LNCS 4714 (2007) 88–95
12. Reichert, M., Kolb, J., Bobrik, R., Bauer, T.: Enabling Personalized Visualization of Large Business Processes through Parameterizable Views. In: Proc. ACM SAC'12, Riva del Garda (Trento), Italy (2012)
13. Kolb, J., Reichert, M.: Using Concurrent Task Trees for Stakeholder-centered Modeling and Visualization of Business Processes. In: Proc. S-BPM ONE 2012, CCIS 284. (2012) 237–251
14. Tran, H.: View-Based and Model-Driven Approach for Process-Driven, Service-Oriented Architectures. TU Wien, PhD thesis (2009)
15. Chiu, D.K.W., Cheung, S.C., Till, S., Karlapalem, K., Li, Q., Kafeza, E.: Workflow view driven cross-organizational interoperability in a web service environment. Inf. Techn. and Mgmt. **5**(3-4) (2004) 221–250
16. Smirnov, S., Reijers, H.A., Weske, M., Nugteren, T.: Business process model abstraction: a definition, catalog, and survey. Distributed and Parallel Databases **30**(1) (2012) 63–99
17. Reichert, M., Rinderle-Ma, S., Dadam, P.: Flexibility in process-aware information systems. T. Petri Nets and Other Models of Concurrency **2** (2009) 115–135
18. Kolb, J., Kammerer, K., Reichert, M.: Updatable Process Views for Adapting Large Process Models: The proView Demonstrator. In: Proc. of the Business Process Management 2012 Demonstration Track, Tallinn, Estonia (2012)
19. Kolb, J., Kammerer, K., Reichert, M.: Updatable Process Views for User-centered Adaption of Large Process Models. In: Proc. Intl. Conf. on Service Oriented Computing (ICSOC'12), Shanghai, China (2012)