Institute of Architecture of Application Systems

Flexible Information Design for Business Process Visualizations

Alexander Nowak, Dimka Karastoyanova, Frank Leymann,
Andrej Rapoport, David Schumm

Institute of Architecture of Application Systems,
University of Stuttgart, Germany
{firstname.lastname}@iaas.uni-stuttgart.de

BibTeX:

```
@inproceedings{NowakKLRS12,
  author    = {Alexander Nowak, Dimka Karastoyanova, Frank Leymann,
               Andrej Rapoport, David Schumm},
  title     = {Flexible Information Design for Business Process
               Visualizations},
  booktitle = {Proceedings of the 2012 IEEE International Conference on
               Service-Oriented Computing and Applications,
               17-19 December 2012, Taipei, Taiwan},
  year      = {2012},
  pages     = {0--0},
  doi       = {TBA},
  publisher = {IEEE Computer Society}
}
```


Universität Stuttgart
Germany

Flexible Information Design for Business Process Visualizations

Alexander Nowak[1], Dimka Karastoyanova[1], Frank Leymann[1], Andrej Rapoport, David Schumm[1]
Institute of Architecture of Application Systems
University of Stuttgart,
Stuttgart, Germany
[1] {firstname.lastname}@iaas.uni-stuttgart.de

Abstract— **Profound understanding of business processes is a key success factor for Business Process Management (BPM). As more and more analytical information like runtime data from process execution or statistical data from business intelligence are available, the problem of business process complexity becomes apparent. Process-relevant information needs to be provided as fast as possible while considering easy and fast interpretation and dynamic changes in stakeholders' demands. The static and use-case specific creation or modification of process visualizations shown in current approaches and tools, however, is complex, time consuming, inflexible and thus costly. To address these shortcomings, we introduce a template-based approach that decouples the creation of visualization templates from concrete process visualizations. The binding of customization points of visualization templates to analytical process information is supported by a graphical editor that enables customization of visualizations in a fast and flexible manner. Moreover, due to the separation of concerns, our approach improves the usability of process visualizations because templates may be created by graphic experts independently from specific visualization demands. The feasibility of our concept is demonstrated by a prototypical implementation.**

Keywords- Business Process Management, Process Visualization, Process Analysis, Service-oriented Architecture

I. INTRODUCTION

Business processes are one of the most important assets of today's organizations as they determine their success or failure in global markets. The term Business Process Management (BPM) subsumes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes [1]. A crucial aspect in BPM is the monitoring and analysis of business processes [2], as it reveals how well the processes perform, where problems occur, and what needs to be changed. The need for flexible change of business processes and service compositions has already been described by Karastoyanova et al. [3]. However, due to the steady increase of complexity of business processes [4], related analytical information like runtime data from process execution, statistical data from business intelligence and analytics, or deployment configurations becomes harder to understand, manage, and change.

In addition to 'classical' approaches such as mathematical models and automatically generated reports in form of tables or text files, the necessity for visual representations and associated analysis techniques has already been identified [5]. Furthermore, Larkin and Simon [6] argue that for some problems good visual representations of information outperform textual descriptions because of the powerful and highly parallel human visual system. To address this need for visual analysis and presentation in the field of BPM, different approaches have been proposed to enrich a process with analytical, heuristic, or semantic data (e.g., augmentation patterns) and to graphically present them (e.g., presentation patterns) [7][21]. Typically, the person who creates a process visualization needs to have technical knowledge in several fields: the creation of graphical visualization, the business process structure and logic, its related data, and the individual composition language bringing the different fields together. Thus, the creation of visualizations is very process and tool-specific and can hardly be adapted easily and quickly by a non-technical user. This increases the complexity and costs when analysis demands or processes change because of communication issues between different departments (Business-IT Gap). Consequently, these kinds of approaches may be suitable for individual scenarios, however, do not provide a generic, flexible, and easily changeable solution for customizable views on information.

In this paper, we combine process enrichment and presentation in a loosely-coupled manner and provide a flexible and easy way to specify process visualizations. We introduce a concept for the separation between the concerns of the creation of visualization templates and the creation of concrete business process visualization by using points of customization within the visualization template. Based on a proposed editor, process visualizations can quickly be (ex)changed based on business demands. This, on the one hand, supports organizations and users to get detailed and focused process insight, which is crucial for successful business process change. On the other hand, graphic experts can focus on the creation of visualization templates independently from concrete process knowledge. The contribution of this work is threefold: (1) we introduce a technical concept for loosely coupled business process models, related analytical information, and visualization templates. The application of this concept allows flexible and customizable visualizations of business processes, depending on the different process stakeholder's information needs. (2) We provide an organizational setup extending the common role model in BPM to realize the concept in an organization. In particular, we introduce the role of a graphic designer and extend the previously introduced role of an information designer, respectively. (3) As proof-of-concept, we provide an architecture and a prototypical implementation. As this work focuses on an approach for configurable visualizations of business processes we do not provide an explicit evaluation of concrete visualization templates. Thus, we do not claim that the example visualization templates we developed in this work provide optimal cognitive effectiveness or comprehensiveness. However, we argue that the technical and organizational concepts we propose enable graphic experts to design such templates and let the different process stakeholders take most advantage of them.

The remainder of this paper is structured as follows: Section II describes the organizational and architectural aspects of the proposed loose coupling concept. Section III explains the novel artifacts realizing the concept, namely 'visualization templates' and 'visual configurations'. The infrastructure components introduced through the approach are also described and their mutual dependencies are discussed. An overview of our prototype is provided by walking through a case study in Section IV. Related work is discussed in Section V. The paper concludes with a summary and outlook in Section VI.

II. Organizational and Architectural Aspects

Figure 1 depicts an overview of the different tasks that have to be performed to enable the more efficient coupling of processes and their visualization, and the associated roles in the concept we propose. Initially, a process engineer models a business process to be used in a runtime environment. A graphic designer (new role) then provides visualization templates that consider cognitive effectiveness and comprehensiveness. These templates can be used to visualize the different elements of a process, like activities, control edges, data objects, KPIs etc. They are parameterized, so that they can be used in various analytical scenarios. The graphic designer does not necessarily need to have knowledge about any concrete business process model or domain; templates can thus be developed in-house or provided by third parties. Next, the information designer (a role defined in [8] and extended here) creates visual configurations for particular analytical scenarios defined by the different process stakeholders, such as the process owner, compliance officer, business analyst and others.

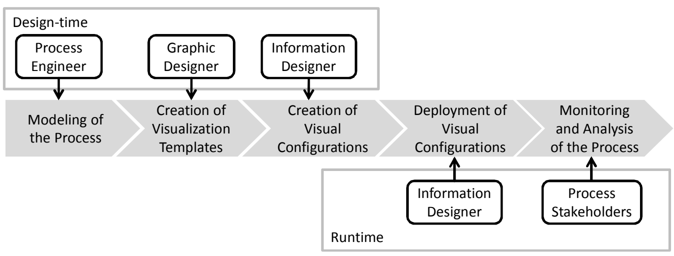


Figure 1. Overview of the different tasks and roles

The visual configuration is responsible for bringing together the visualization templates, the analytical information, and a business process model. Then, the configurations are deployed onto monitoring and analysis tools to allow the various process stakeholders to apply them to concrete analytical scenarios and corresponding processes and data.

A. Implications on the BPM Lifecycle

To position the tasks of Figure 1 in the context of BPM, we identified their relationship with the common BPM lifecycle [1], as shown in Figure 2. The K*I (Key * Indicators) comprise common economic indicators (Key Performance Indicators, KPI), Key Ecological Indicators (KEI) [9] or compliance indicators (KCI) [10]. Each of these indicators might be of particular interest to the corresponding process stakeholders. The common BPM lifecycle is usually divided into a process design time part and a process runtime phase. Design time includes the three phases of *process modeling*, *process refinement* in terms of choosing the IT resources (or their types) that perform the business process and its activities, and *process deployment* in which all artifacts related to execution are set up or configured. Runtime includes the phases *process execution*, *process monitoring*, and *process analysis* which provides improvement feedback to modeling and closes the lifecycle. The design time phase is influenced by the need to design *visualization templates* in addition to the conventional modeling of business processes.

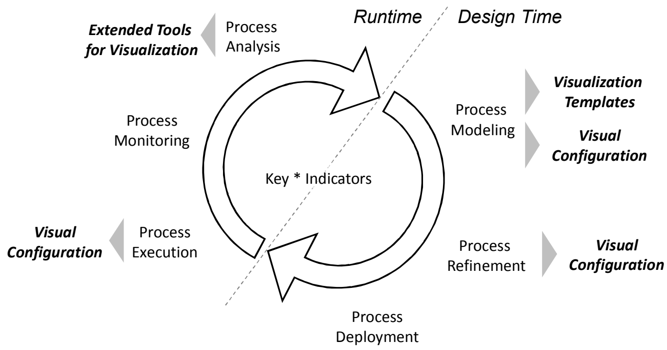


Figure 2. Extended BPM lifecycle for flexible process visualization

These templates are complex graphical shapes that contain multiple placeholders, called 'customization points'. Static information available in design time as well as analytical information produced during or after execution can be mapped to these customization points to visualize a process based on specific information needs. The mapping of information to customization points, which is specified within the *visual configuration*, also extends the scope of the design time phase. Apart from pre-defining configurations during process modeling, we also allow for defining them during process refinement, in the cases concrete services are bound to process activities in this phase, or during runtime. Thus, the configuration can then be used on demand to visualize concrete information provisioned in the process analysis phase. Analysis tools for visualization need in turn to be extended for embedding or linking externally-provided graphics.

B. Architectural Aspects

Figure 3 shows an architecture allowing organizations to benefit from flexible visual analysis support for their business processes. The architecture shows the components common in service-based process environments as well as the novel components required to realize the approach (highlighted using bold lines). Process engineers use the *process editor* to model business processes to be executed as an orchestration of automated services and human tasks. The result is a *business process* including all information required for deployment. This information is stored in a *process repository*. After deployment, the process models can be executed on a *process engine* which coordinates the invocation of services as specified in the process model. The process engineer uses the *monitoring configuration editor* to specify which analytical data needs to be sensed in order to gather all monitoring information required by the K*I definitions to calculate their values. These indicators are defined by process stakeholders to capture the different strategic objectives.

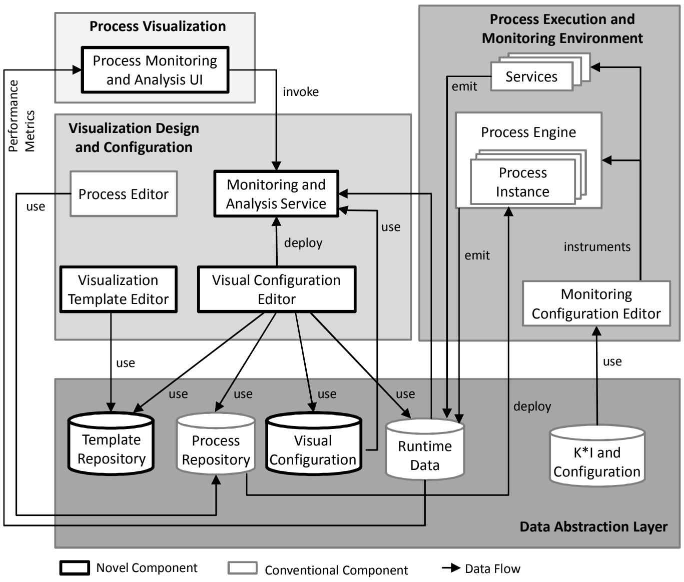


Figure 3. Architecture of loosely coupled business process visualization

As depicted in the architecture in Figure 3, several additional components are required to enable the proposed approach. The graphic designer uses the *visualization template editor* to create graphics considering the design requirements like cognitive effectiveness and comprehensiveness. The designer also inserts appropriate customization points that can be filled with concrete process runtime data and, thus, makes these graphics customizable. The visualization templates resulting from this work are stored in a *template repository* that provides versioning, search, and thus eases reuse of templates. The information designer uses the *visual configuration editor* to link all artifacts used for visualization into a visual configuration. That is, mapping the activities and other elements of a process to specific visualization templates and mapping the process data to the customization points of a template. The visual configurations containing a process model, visualization templates, analytical data format and a mapping definition are stored in the *visual configuration* repository. Before a configuration can be used it needs to be deployed onto the *monitoring and analysis service*. The *monitoring and analysis service* is described in detail within the prototype walkthrough in Section 4 as this service depends on the concrete implementation of the visualization templates environment. This service can be invoked by the *process monitoring and analysis UI* to generate process visualizations, i.e. the user interface enables the selection of particular processes and their corresponding analytical data to trigger the creation of visualizations and to display them to the users. These visualizations can be generated based on historical data or even by (incomplete) data generated during process execution.

III. Elements of Loosely Coupled Business Process Visualization

Our approach contains two key extensions to existing BPM architectures: (1) the template editor and the template repository dealing with *visualization templates*, and (2) the visual configuration editor and the visual configuration repository that deal with the creation and management of *visual configurations*. These extensions are necessary in order to accommodate the novel artifacts we introduce in this work. They are described in the following, starting with a conceptual model showing their constituents and their relations to common BPM artifacts.

A. Conceptual Model

The conceptual model depicted in Figure 4 describes how the new elements relate to each other. A *process visualization* is generated from a *visual configuration* by using the *monitoring and analysis service*. The *monitoring and analysis service* in turn uses the *analytical data* of the process whose structure is captured in the *analytical data format*. The *visual configuration* contains: (1) an *analytical data format*, (2) a *process model* that describes the business process of interest, and (3) *visualization templates* that contain one or more *customization points* that enable the dynamic insertion of *analytical data*. Moreover, the *visual configuration* realizes a list of mappings that connect process activities (e.g. service

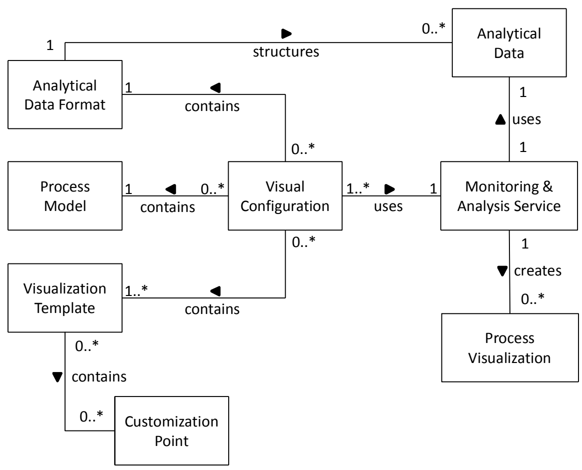


Figure 4. Conceptual model of the elements of loosely coupled visualizations

invocation activities) with a visualization template (e.g. default template) and the customization points of that template (e.g. [size]) with an element of the *analytical* data format (e.g. energyConsumption). The *analytical data format* is defined by the information designer based on the individual visualization requirements. There is no need for a specific format, however, the format needs to be captured in a XML schema definition. In some cases an additional transformation of the analytical data might be necessary. This ensures that an individual data format can be used for creating the visual configuration.

B. Visualization Templates

Recall, that we propose the use of visualization templates to separate the knowledge of comprehensive visual communication from concrete process design and execution knowledge. A *graphic designer* models such templates in order to visualize different aspects of processes and analytical information of these processes, respectively. Thus, the graphic designer has two main tasks:

(T1) The designer decides how to visually represent activities and other elements of a business processes.

(T2) The designer enriches the visualization template with extension points, so-called *customization points* that are used to dynamically change the appearance of a visualization based on analytical data. The intention of using customization points is to allow a template designer to create one or more variable points that allow not only to set specific values from analytical data but also to use specific icons or colors depending on these data.

During design time, the customization points can be mapped to analytical data by using a corresponding schema definition representing the structure of the analytical data. To cope with these tasks a textual language for describing the graphical representations is needed. One suitable language is Scalable Vector Graphics (SVG) [11]. SVG files are platform- and device-independent and use a hierarchical XML structure to define elements and their corresponding attributes. Using the SVG format has two main advantages which directly address the tasks (T1) and (T2), respectively: (i) it allows the use of user-defined graphics and free scaling and (ii) it provides the possibility to add extensions for customization points to the XML source. Three different steps need to be performed to insert a customization point: (1) Locate the attribute or node value that should become variable. (2) Add a new customization attribute or node value next to the identified attribute. (3) Define all properties of the customization point within the new attribute.

The location of a customization point may be derived from existing scenarios, experience, or the design visual shape of a template. As an example, an attribute of interest could look like the following: *name="Used 0 kWh in total"*. In this example, the amount of kWh used by a certain activity is subject of a customization point as it should reflect a value from runtime data. Consequently, a new attribute needs to be added that provides the means for such a customization point. In terms of XML syntax, the new customization attribute could be represented as shown in Figure 5. First, the attribute refers to a special *namespace*. This enables compatibility to standard SVG and allows the usage of standard SVG tools for visualization template editing. The attribute value in the example is then comprised by a *constant prefix*, a *constant suffix*, and a *variable part*. To map the variable part to concrete analytical information it is separated into an *ID*, a *name*, and a *type* that is valid for this value. That is, for example, a string or an integer value. In case there is no analytical data mapped to a customization point the original value from the template will stay in place, i.e. the customization point will be ignored.

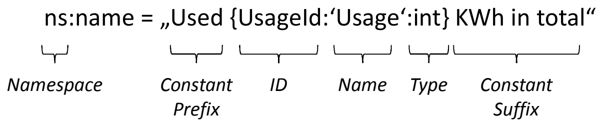


Figure 5. Structure of customization points in SVG

An example visualization template for a process activity could look like the one proposed in Figure 6. The template on the left side contains five different customization points: a *type icon* (enumeration, icons may be embedded in the template) that represents the type of the activity, the *title of the activity* (string) that represents the name of the activity, a *state icon* (enumeration) that indicates the current execution state of the activity, a *progress bar value* (string), and the *progress bar filling* (integer) that represents the overall progress of the activity. Before rendering, the values of the original attributes will be replaced with the ones calculated from the customization point and associated data. An example of a mapping to concrete information is shown on the right side of Figure 6. With the example data (HUMAN_TASK / "Approve Loan" / RUNNING / 79). It is also possible to map one and the same analytical data to multiple customization points (here: 79).

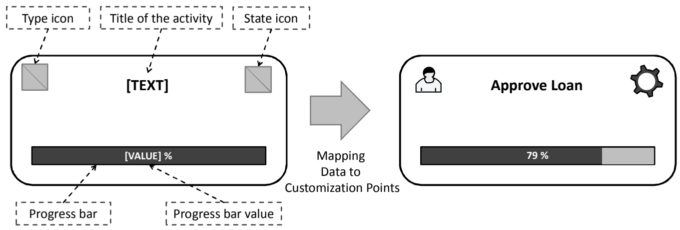


Figure 6. A visualization template including customization points (left), and a visualization using the template with instance data (right)

C. Visual Configuration

The visual configuration is responsible for coupling the visualization templates, the analytical data structure, and the process model. As stated above, the configuration maps process activities to visualization templates and customization points of the templates to the analytical data format. Note, that the concrete analytical information is not associated at the time of creating the configuration, only its structure. The insertion of concrete analytical information is performed afterwards by the *monitoring and analysis service*. This service will be used each time a process visualization should be generated. The general structure of a visual configuration is shown in Figure 7. A visual configuration contains mappings, the data format of analytical data to be displayed, the process model, and metainformation. The mapping of process activities to a template is represented by a *MappingsType* that references an arbitrary number of *MappingTypes*. A single *MappingType* is composed of an *ActivityType* and a *TemplateType*. The *ActivityType* references a set of activities by using either an activity type (e.g. task, wait) or a specific activity attribute (e.g. name). This is represented by the attributes *attrName*, *attrValue*, and the *nodeName*. A *TemplateType* references a template by its *name* and contains an arbitrary number of *ParameterTypes* that describe how the template must be filled with concrete analytical data. The *ParameterType* has three attributes: a parameter identifier *paramid*, a customization point identifier *vpid*, and a *staticValue* that can be used to assign a value to the customization point that remains constant for all visualizations. Figure 7 shows the usage of visualization templates for process activities. The approach, however, is also applicable to other process elements like control or data edges and data objects.

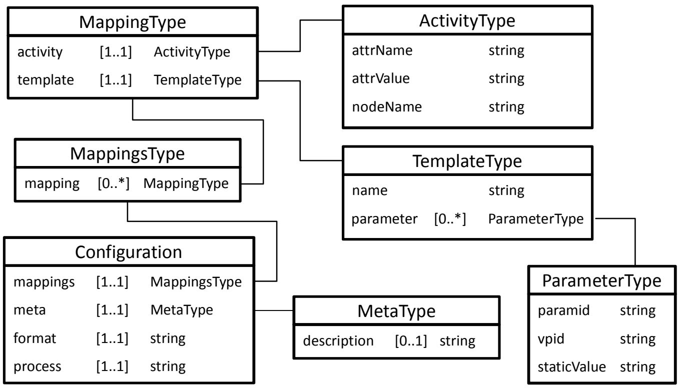


Figure 7. Structure of a visual configuration

IV. CASE STUDY AND PROTOTYPE

As a proof of concept we implemented a prototype [12] that realizes the new components presented in Section 3. The prototype consists of a visual configuration editor, a service that uses the configuration to generate the visualization, and a shared library providing functionality used by both of them. We implemented also a user interface that allows invoking the visualization generation service. As integration technology we used Web services. The corresponding repositories are realized through file system storage. As visualization template editor we used Inkscape [13], which is an open source SVG editor. The insertion of customization points was also made using Inkscape but could have also been made using a common XML editor. The current support of visualization templates is limited to process activities. For the implementation of the prototype we assume a service-based BPM landscape which means that a business process manifests itself as an orchestration of automatic or human-provided services operating in different environments [14]. With this approach we enable in particular the customizable visualization of monitoring and analysis data of BPEL processes [15] in a user friendly manner.

A. Description of the Business Process Example

In this example, the key stakeholder is the executive of a building architecture office, whose goal is to assess the environmental impact of his business processes. A showcase business process is depicted in Figure 8, and is modeled using the Business Process Model and Notation (BPMN) [16]. The process describes the tasks necessary to plan and implement a bridge construction: The process is triggered whenever the architecture office accepts a construction order from a customer. Based on the customer's requirements a Computer-Aided Design (CAD) system is used to develop a computer model of the bridge. After the first version of the CAD model is created, several simulations are performed to further improve the model. For instance, the simulations verify the stability of the bridge under extreme conditions. As these kinds of simulations are very resource intensive, the simulation environment is hosted on high-performance compute nodes. In case one of the simulations reveals problems in the bridge model it has to be corrected and/or optimized and the simulations have to be performed again with the updated model. Then, the process finishes. The executive wants to improve the environmental impact of this business process as a more eco-friendly image provides a good selling point to customers. However, the executive does not know which parts of the business process cause a significant environmental impact that could be decreased. Thus, he decides to put measurement facilities in place that gather the energy consumption data of the different IT systems used to execute the activities of the business process.

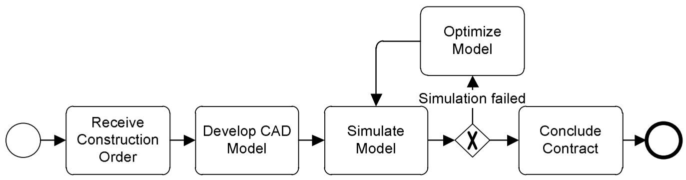
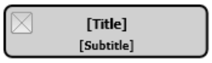
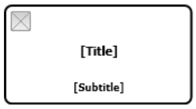
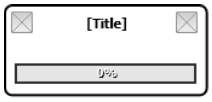


Figure 8. Planning process for bridge construction

B. Development and Selection of Visualization Templates

To analyze the energy consumption of the business process appropriate visualizations need to be integrated into the company's process management and analytics dashboard. The first step of providing a customized process visualization requires either the graphic designer to develop a new template that allows capturing the needs of the organization, or the information designer has to choose a template from a catalog of available templates. In the latter case, the information designer may also buy a template at a template marketplace. The intention of using independent visualization templates naturally implies the internal use of a template repository that allows easy reusability of different templates. In our example, the architecture office is already in possession of different templates. An overview of some of the templates is shown in Table 1. Assuming that it is required to focus only on a single ecological characteristic, namely energy consumption, the information designer decides to use the *Colored Box* template shown in Table 1. This template keeps the structure of the process but only changes the visual appearance of the activities. In case of other visualization requirements other templates may be chosen, for example, presenting activities of a process regardless their original structure. The chosen template is available as SVG and contains four different customization points to be filled with values during the creation of the visualization. In the upper-left corner an icon for the activity type can be inserted. In the center of the box the name of the activity can be inserted (*[Title]*) as well as a corresponding subtitle (*[Subtitle]*).

TABLE I. OVERVIEW OF VISUALIZATION TEMPLATES

Template	Customization Points	Preview
Small Box	Title Activity Type Icon	
Colored Box	Title Subtitle Background Color Activity Type Icon	
Sizeable Box	Title Subtitle Width Activity Type Icon	
Progress Bar	Title Progress Bar Progress Bar Value Activity Type Icon Activity State Icon	
Drop	Title Subtitle Width Height	

Moreover, to ease the visual identification of the energy consumption of each activity the background color can be adapted depending on the measured values. The information designer defines four different colors: white is used when no energy consumption value is available for the activity; green is used if the energy consumption is below 80% of the given target value; yellow is used if the energy consumption is between 80% and 100% of a given target value; red is used if the energy consumption is above 100% of a given target value. The target values are set for each activity of the process separately. In our example, the target values are based on vendor information about state-of-the-art computer systems used for the corresponding tasks.

```
<text id="text6658" y="0" x="50%"
  ns:NodeValue="{NameId:'Title text':string}"
     style="font-weight:bold;text-align:center;text-
         anchor:middle;
         fill:#000000;font-family:Tahoma;font-
         size:8pt;">[Title]</text>
```

Figure 9. Visualization template - customization point for NodeValues

In the following we present how customization points are appropriately included in the (textual representation of the) visualization template in order to enable enriched visualizations. Our prototype uses both variable *NodeValues* and variable *Attributes*. For space reasons, we only present one code example for each type. All other customization points are defined using the same syntax. The customization point in Figure 9 shows how the *NodeValue* 'Title' is made variable. The customization point is described by its ID (*NameId)*, a description of the customization point that can be shown to a user to ease configuration (*Title Text*), and the expected type (*string*) of the inserted data. If there is no analytical data mapped to that customization point, the default value *[Title]* will be used.

The second example shown in Figure 10 describes how the value of an attribute can be made variable. The different background colors are defined within the template by using different Cascading Style Sheet (CSS) style classes. The default class is *colorlevel0* indicating that the background color of the template is white. The customization point consists of an ID (*BackColorId*), a description (*Background Color*), and an enumeration that describes the allowed values for this customization point. That is, 0,1,2,3 which represent the colors white, green, yellow, and red, respectively, that can be used as background color.

```
<rect id="rect6648" width="98%" height="93%"
      style="stroke:#000000;stroke-width:2;"
      class="colorlevel0"
         ns:class="colorlevel{BackColorId:'Background
                 Color (0 to 3)':enum:0,1,2,3}" />
```

Figure 10. Visualization template - customization point for attributes

C. *Creating a Visual Configuration*

After choosing a suitable template that covers the visualization requirements of the process stakeholder the information designer can build a configuration file that groups all information necessary for using the template in the monitoring environment. First, to set up an appropriate configuration, he has to identify the data format of the analytical data that should be used (or define the format in case of composite data). This step is necessary to establish the mapping between customization points and the corresponding analytical data. We used two different templates that distinguish between invoke activities (service invocations, represented by the *template-color.svg* template) and all other activities (represented by the *template-default.svg* template) that may be present within the corresponding BPEL process. The mapping used in this example is shown in Figure 11.

```
<mappings>
 <mapping>
  <activity nodeName="invoke"></activity>
  <template name="template-color.svg">
  <parameter vpid="BackColorId"
             paramid="Efficiency"/>
  <parameter staticValue="ns:type" vpid="IconId"/>
  <parameter staticValue="ns:name" vpid="TitleId"/>
  <parameter vpid="SubtitleId"
             paramid="UsedEquipment"/>
  </template>
 </mapping>
 <mapping>
  <activity nodeName=""></activity>
  <template name="template-default.svg">
  <parameter staticValue="ns:type" vpid="IconId"/>
  <parameter staticValue="ns:name" vpid="NameId"/>
   </template>
 </mapping>
</mappings>
```

Figure 11. Configuration of a visualization template

Invoke activities are mapped to the *template-color.svg* template. This template consists of four different customization points: The background color (*BackColorId*) and the *subtitle* of the activity are mapped to the analytical data *Efficiency* and *UsedEquipment*, respectively. Furthermore, the prototype also uses the keywords *type* and *name* that ease the usability by directly mapping the customization point to information from the process model. Thus, these values are described as *staticValue*. All other activities used in the process model are using the *template-default.svg* template. The parameters are only mapped to the *name* and the *type* of the underlying process model. In [17] we provide a complete demo of the configuration editor (named *Vipro*) and show how a visual configuration is created.

D. Applying a Visual Configuration

After creating the visual configuration for binding the customization points to concrete analytical information, the information designer deploys the visual configuration to the *monitoring and analysis service* that handles the actual merge of visualization template and concrete data. This service calculates a layout for the process graph to be visualized considering the actual sizes of the instantiated visualization templates. To enrich the visualization template with concrete data, the data needs to be provided in the correct format, i.e. the same format that has been used during the creation of the configuration. A sample snippet of the analytical data of the activity *DevelopCADModel* is shown in Figure 12. The activity is identified by its name and provides the parameters *Efficiency* and *UsedEquipment* that have been mapped to the customization points of the visualization template.

```
<activities>
  <activity attrName="name"
            attrValue="DevelopCADModel">
    <parameter name="Efficiency">1</parameter>
    <parameter name="UsedEquipment">Power Mac G5
    </parameter>
  </activity>
  ...
</activities>
```

Figure 12. Example analytical data for visualization

The result returned from the *monitoring and analysis service* is a process visualization in SVG format that shows the average energy consumption of all activity instances that have been measured. The process visualization which is enriched by analytical data is shown in Figure 13, rendered in the Firefox browser. As defined in the configuration, the invoke activities are represented by the colored template, all other activities by the default template. The icons as well as the names of the activities are directly derived from the underlying BPEL process. The subtitle and the colors of the invoke activities are representing the analytical data from runtime monitoring. Basically, the analytical data may represent data from a single or from multiple instances of a process. Therefore, it is also possible to visualize, e.g., loops of activities within a single process instance by correspondingly pre-processing the data with respect to the number of invokes. The accumulated energy consumption, for example, and the number of service invokes may then be mapped to specific customization points, respectively. The mapping of analytical data as well as visualization templates defining the appearance of an activity can easily be changed retrospectively within the visual configuration editor, based on changing visualization demands or changes in the process model, for example. Based on the process visualization of our example the executive can now identify that the activities *DevelopCADModel* and *CheckResult* exceed their corresponding target value for energy consumption.

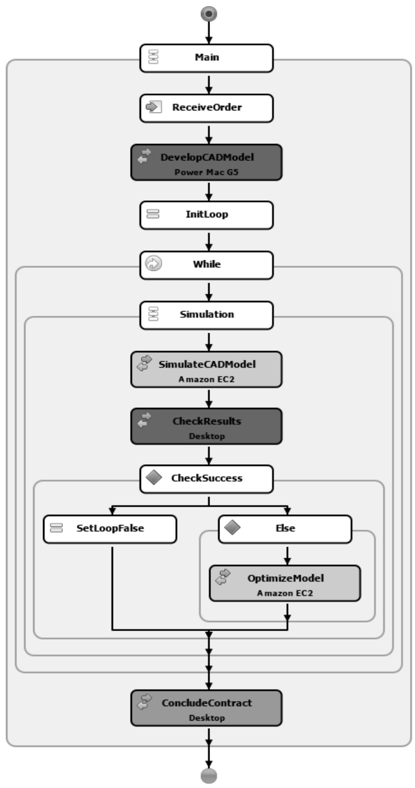


Figure 13. Business process visualization of the bridge planning process, data about energy consumption, and color-coded visualization templates for activities

V. RELATED WORK

The visualization of complex data is an inherent research challenge. For example, Santos and Brodlie [18] extended the Haber-McNabb dataflow model for scientific visualizations to visualize n-dimensional numeric data, or Müller et al. [19] proposed an approach to visualize software systems. These approaches, however, do not address BPM in particular or consider a separation of concerns in creating and using such visualizations. The basic concept of information design for BPM was sketched by Schumm [8], however, without providing a detailed insight into the organizational, architectural and technical aspects and challenges. A corresponding prototype has been proposed presented for visualizing a business process [18], however, it is limited to display instance-specific runtime information and to highlighting using one single visualization template. The approach presented in this work outperforms these initial concepts by an advanced approach for loose coupling of process elements, graphics and almost arbitrary analytical data.

Aside from these works, further approaches were proposed towards visual configurations. The main shortcomings of these approaches, compared to the presented concept, are the insufficient provision of all relevant analytical information and the approaches considering a rich and expressive graphical representation require a lot of manual and time-consuming work. These approaches also lack in decoupling the different knowledge areas of the different stakeholders. The major work in that area is discussed in the following: The works proposed by Bobrik et al. [21][22] describe an approach for configuring and personalizing the visualization of a business process. The notation definition contains template definitions that consist of three different parts: input parameters that map the process elements, representations of the graphical symbol, and parameters that map process runtime data. The application of a template definition to a process is done by a scripting language. Note, this requires the template designer to declare generic data requests within the template and thus to have thorough knowledge of process, runtime data, and template definition. This results in a tight coupling of the different components.

VI. Conclusion and Outlook

In this work we present an approach for enhanced visualizations by enriching business process visualizations with process analytics information. The main concept of our approach is to decouple the business process model, the graphical visualization, and the analytical information of interest. This allows separating tasks and skills between the different stakeholders that are involved in the creation and use of a process visualization. The separation provides great advantage as experts of graphic design, process design, and business analysis can focus on their specific area of knowledge without the need to know too many details from other areas. Using the proposed concepts and visualization editor, different requirements of new visualizations can be created in a highly flexible manner. The editor encapsulates all the work necessary to glue the single components together in a visualization configuration: selection of processes, visualization templates, and runtime data as well as mapping the runtime data to customization points of visualization templates. After that, the configuration is processed by the introduced monitoring and analysis services resulting in the customized visualization of the business process model.

To improve the approach we are currently working on two aspects: (1) a concept for automatic generation of legends corresponding to the different visualization. Similarly to cartography, legends are required in order to provide the reader with all information needed to make the visualization clear and unambiguous. (2) We want to apply different views [7] to the visualizations to provide stakeholders with, for example, additional information to even better fit their demands. We also want extend the approach for the use with Key Ecological Indicators of business processes.

Acknowledgment

This work was partially funded by the BMWi project Migrate! (01ME11055). D. Schumm and D. Karastoyanova would like to thank the German Research Foundation (DFG) for financial support of the project within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart.

References

[1] F. Leymann, D. Roller, Production Workflows – Concepts and Techniques. PTR Prentice Hall, 2000.

[2] G.T. Lakshmanan, P.T. Keyser, A. Slominski and F. Curbera, A Business Centric Monitoring Approach for Heterogeneous Service Composites. Proc. of SCC 2011, IEEE, 2011, pp.671-678.

[3] D. Karastoyanova, F. Leymann, J. Nitzsche, B. Wetzstein, D. Wutke, Parameterized BPEL Processes: Concepts and Implementation. Business Process Management, vol. 4102/2006, 2006, pp. 471-476.

[4] J. Vanhatalo, H. Völzer and F. Leymann, Faster and more focused Control-flow Analysis for Business Process Models though SESE Decomposition. Proc. of the 5th Intl. Conf. on Service-Oriented Computing (ICSOC), Springer, 2007, pp. 43-55.

[5] K. Vergidis, A. Tiwari and B. Majeed, Business Process Analysis and Optimization: Beyond Reengineering. Transactions On Systems, Man, and Cybernetics: Applications and Reviews, vol. 38(1), 2008, pp. 69-82.

[6] J.H. Larkin and H.A. Simon, Why a Diagram is (Sometimes) Worth Ten Thousand Words. Cognitive Science, vol. 11(1), 1987, pp. 65-100.

[7] D. Schumm, F. Leymann and A. Streule, Process Viewing Patterns. Proc. of the 14th IEEE Intl. EDOC Conference, IEEE, 2010, pp. 89-98.

[8] D. Schumm, Information Design for Business Process Management. 5th Summer School on SOC, Poster Presentation, 2011.

[9] A. Nowak, F. Leymann, D. Schumm and B. Wetzstein, An Architecture and Methodology for a Four-Phased Approach to Green BPR. Proc. of ICT-GLOW 2011, Springer, 2011, pp. 150-164.

[10] P. Silveira, C. Rodríguez, F. Casati, F. Daniel, V. D'Andrea, C. Worledge and Z. Taheri, On the Design of Compliance Governance Dashboards for Effective Compliance and Audit Management. Proc. of NFPSLAM-SOC2009, Springer, 2009, pp. 208-217.

[11] World Wide Web Consortium, Scalable Vector Graphics (SVG). http://www.w3.org/TR/SVG/, 2010.

[12] A. Rapoports, Configurable Visual Representations for Business Process Monitoring. University of Stuttgart, Faculty of Computer Science, Electrical Engineering, and Information Technology, Diploma Thesis No. 3181, 2011.

[13] Inkscape, Open Source SVG Editor. http://inkscape.org/, 2012.

[14] C. Fehling, F. Leymann, D. Schumm, R. Konrad, R. Mietzner and M. Pauly, Flexible Process-Based Applications in Hybrid Clouds. Proc. of CLOUD 2011, IEEE, 2011, pp.81-88.

[15] OASIS, Web Services Business Process Execution Language Version 2.0. OASIS, 2007.

[16] OMG, Business Process Model and Notation (BPMN). OMG, 2012.

[17] A. Rapoport, Vipro Demo. http://www.iaas.uni-stuttgart.de/forschung/views/v-demo.avi, 2011.

[18] S. dos Santos, K. Brodlie, Gaining understanding of multivariate and multidimensional data through visualization. Computers & Graphics, 28(3), 2004, pp. 311-325.

[19] R. Müller, P. Kovacs, J. Schilbach, U. Eisenecker, Generative Software Visualizaion: Automatic Generation of User-Specific Visualisations. Proc. of IWDE 2011, ACM, 2011, pp. 45-49.

[20] D. Schumm, G. Latuske and F. Leymann, A Prototype for View-based Monitoring of BPEL Processes. Technical Report 2011/04, University of Stuttgart, 2011.

[21] R. Bobrik, Konfigurierbare Visualisierung komplexer Prozessmodelle (German). PhD Thesis, University of Ulm, Germany, 2008.

[22] R. Bobrik, T. Bauer and M. Reichert, Proviado – Personalized and Configurable Visualizations of Business Processes. Proc. of EC-WEB2006, Springer, 2006, pp. 61-71.