3D Representation of Business Process Models

Stefanie Betz, Daniel Eichhorn, Susan Hickl, Stefan Klink,
Agnes Koschmider, Yu Li, Andreas Oberweis, Ralf Trunko,

Institute of Applied Informatics and Formal Description Methods (AIFB)
Universität Karlsruhe (TH), Germany
{betz|eichhorn|hickl|klink|koschmider|li|oberweis|trunko}@aifb.uni-karlsruhe.de

Abstract: 3D technologies open up new possibilities for modeling business processes. They provide higher plasticity and eliminate some deficits of conventional 2D process modeling such as the limitation of the amount of information to be integrated into a process model in an understandable way. The aim of this paper is to show how the usage of an additional visual modeling dimension may support users in compactly representing and animating business process models. For this purpose, we propose an approach for 3D representation of business process models based on Petri nets. The need for the third modeling dimension is pointed out with three modeling scenarios for which we propose modeling improvements in 3D space. Early evaluations indicate the effectiveness of our approach, which goes beyond conventional modeling tools for business processes.

1 Introduction

The increasing interest in business process management by academia and industry has resulted in a multitude of modeling languages and tools supporting business process modeling. Modelers are frequently confronted with new modeling languages and sophisticated tools which may overwhelm especially those users inexperienced in process modeling. Most existing tool implementations disregard the fact that users need more support for process modeling than just a repository of graphical symbols [KHG08]. In this context, it is an essential requirement to provide a user-friendly business process visualization which facilitates a quick understanding and overview of a process model. Nevertheless, current process modeling languages are limited in their visualization capabilities because the amount of information to be integrated into a process model is usually much more than can be effectively displayed. To reduce the size of the language elements in order to integrate more information into a process model does not really solve the problem. In contrast, this approach does not disburden the life of process modelers (concerning learnability of the modeling language) or process stakeholders (concerning understandability of the process model). Thus, some approaches (e.g. [JaBu96, Sch99, MeS06, BRB07]) propose for instance different perspectives on a process model in order to improve its comprehensibility.

A deficit of current process modeling languages is that they use only two modeling dimensions (x-axis and y-axis) for visualization. This limits the amount of information to be integrated into a process model in an understandable way. By introducing the third dimension into business process modeling, we are able to represent information of a process model more compactly than current process modeling languages allow. Additionally, 3D process model visualization allows the user to change her view-point. In this paper, we extend the conventional "flat" representation of Petri nets with a concept for spatial visualization of net diagrams. In order to improve the understandability of Petri nets for novices, expressive icons may be used for places and transitions. For an easier understandable visualization of the behavior of a Petri net, model animations using graphical elements of the related application domain might be used [Ver00]. With our approach, we intend not only to improve the layout of process models (e.g. by minimizing the number of crossings of arcs [Rei93, Röl07]) but also to increase the information content of a Petri net model. Despite the prejudice that 3D displays are prevalently regarded as "gimmicks" (nice layout, no additional value) [BES00], we are confident to generate an added value with our approach – last but not least by considering the three-dimensional grasp of humans. The integration of the third modeling dimension supports a more clearly arranged representation of modeling concepts and simulation results of Petri nets. This approach gains benefits for visualizing more compactly relevant information of the process models and allows integrating new objects for user interactivity.

The remainder of the paper is structured as follows. In Section 2, we present related work in the field of 3D visualization of business process models. Section 3 surveys some deficiencies of 2D process modeling with an example. Within Section 4, we describe 3D modeling of business processes including the modeling of roles, objects and hierarchies as well as relationships between business processes. In Section 5, we depict an initial implementation of our approach. The paper concludes with a summary and a critical consideration of the results obtained so far and gives an outlook on future research.

2 Related Work

Existing work in 3D visualization of business process models can be differentiated in three categories: (1) Business process modeling, (2) 3D modeling, and (3) Layout methods.

Several methods have been proposed to model business processes with complex objects and respective events [DGB07, LHG07, LeO03], but with limited capabilities in their visualization. These approaches lack features for effectively describing all relevant information integrated in one process model. Instead, large number of dependencies between activities and objects need to be hidden because the amount of information to be integrated into a process model is usually much more than can be effectively displayed. For instance, regarding a Petri net model (or an EPC model respectively) the user can specify resources and objects, which are required for handling specific activities, on one process level, but she needs to switch to another view in order to gain insight into the dependencies between resources.

The relocation of relevant information on different views is a popular technique for breaking down the complexity of business process models [JaBu96, Sch99, MeS06, BRB07]. However, the switching between different views is a time-consuming task, because with every switching the user needs to spend time on orienting herself in the new view and comprehend the relationships between resources, roles and activities. Some tools use the third dimension for business process modeling [KGM99, BES00, KiP04, Röl07] in order to support the integration of new objects and to allow user interactivity. Our approach, which uses three dimensions for a compact representation of dependencies between objects, organization, and process models, builds on this existing research work.

Layout methods for the design of diagrams have been a research focus for several years [MaM07, HaL08]. The current tool support for diagram layout is still evolving. Especially, in 3D modeling one deficit for layout methods are efficient algorithms for handling a limited number of elements in all three dimensions. Therefore, we will use the algorithm proposed in [KaK89] in order to draw graphs, which has been evaluated as an effective technique in [Pur98].

3 Running Example

We will demonstrate benefits of 3D business process modeling by presenting a simple example. Figure 1 shows a conventional 2D representation of a decision process for flood crisis management. Precisely, the depicted process is part of a complex process of controlling rehabilitation measures [Pla02, NeK05].

This process is modeled with Petri nets where circles represent conditions and rectangles represent actions. A transition box inscribed with two vertical bars indicates that the respective transition is refined by another Petri net. Repeated refinement of transitions leads to a hierarchical representation of a process.

To solve the decision problems concerning an efficient resources and action allocation, the involvement of several organizational units (e.g. executive staff, administration member) is required. The first part of the process handles the priority of the disaster notifications, which are analyzed in the second part of the process. Before selecting an action scenario concerning the selected notifications, the availability of resources needs to be checked (e.g. manpower, technical and electrical equipment) where some specific activities are performed by roles (e.g. executive staff as *role 1* or PR and media staff as *role 2* in Figure 1). For comprehensibility reasons we annotated only few roles. Finally, a decision will be made and the responsible persons and involved parties will be informed.

Although this standard process model is on a high level of abstraction with a clearly arranged number of roles and activities, the understandability is not satisfying. Additionally, to gain insight into the relationship of roles and process hierarchies users need to open another process or organization model, which requires spending time in order to understand the relationships. In this context, the next section presents our approach of a compact representation of relevant information between a business process model and an organization model.

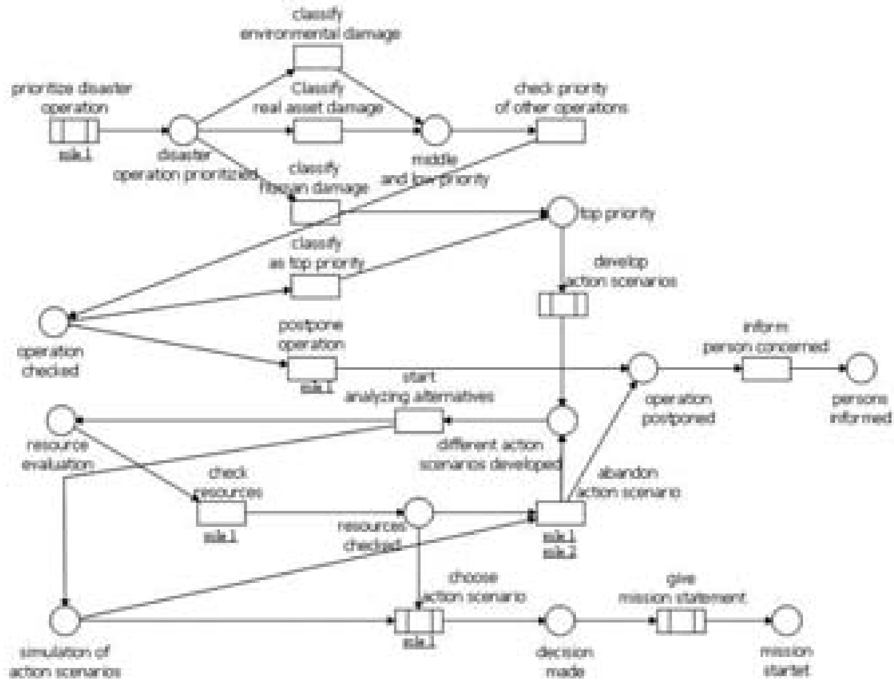


Figure 1: Decision process for flood crisis management

4 Modeling Business Processes with 3 Dimensions

In this section we first introduce Petri net constructs, which are used in the following in the three scenarios for 3D business process modeling with respect to modeling roles, resources, semantic relationships and hierarchies of business processes. Naturally, other processes or data resulting from process metrics (e.g. time, costs) can also be modeled as 3D objects. But in the initial stage of exploring 3D business process modeling, we restrict ourselves in this paper on the three scenarios mentioned before that will be explained in Section 4.2 to 4.4.

4.1 Petri nets

Petri nets are a widely accepted graphical language for the specification, simulation and verification of information systems' behavior. Formally, a Petri net is a directed bipartite graph with two sets of nodes (places and transitions) and a set of arcs which can be described by the triple $N = (P, T, F)$, where P is the set of places, T is the set of transitions (which is disjoint from P) and $F \subseteq (P \times T) \cup (T \times P)$ is a flow relation. Elements of P are graphically represented as circles (e.g. *disaster operation prioritized* in Figure 1), elements of T as rectangles (e.g. *prioritize disaster operation*) and elements of F as directed arcs between places and transitions. A place p is an input place of a transition t, if there exists an arc from *p* to *t*. A place *p* is an output place of a transition *t*, if there exists an arc from *t* to *p*. The set of all input places of a transition *t* is denoted by •*t* and is called pre-set. The set of all output places is denoted by *t*• and is called post-set. Numerous Petri net variants have been proposed, which can be subsumed in elementary or high-level Petri nets. In elementary Petri nets, places contain tokens (black dots), which represent anonymous objects, whereas the flow of the tokens simulates the process flow. In high-level Petri nets tokens represent identifiable objects.

In the following we will describe how users may benefit from the usage of the third dimension for modeling business processes. Our approach is applicable for both elementary and high-level Petri nets.

4.2 Modeling Roles and Objects

Usually, activities and roles are stored in different models (process model and organization model) and users manually insert a link between roles and activities. Roles may be assigned to activities in the process model but an organization model is still required to describe the relationships between organization units. Thus, the user needs to switch to another view in order to gain a deeper insight into roles or resources. Activities represent tasks, roles describe capabilities of a resource, which is an entity performing the assigned tasks in the business process.

With respect to the running example, the deficit of switching between different models is overcome by using a third dimension for representing relationships between process and organization model. Thus, users can easily catch the position of a specific role, e.g. always at the beginning/in the middle/at the end of a process or if the role appears in several parts of the process. Additionally, our modeling technique supports the generation of specific views on interesting details of relationships between activities and roles respectively resources. This is not directly possible in the business process model shown in Figure 1.

For this, we define the assignment of a role to an activity by a mapping *mapToRoles:* $A \rightarrow \mathcal{P}(R)$ where $R = \{r_1 , ..., r_m\}$ is the set of all roles, *m* is the number of roles, *A* is the set of activities and $\mathcal{P}(R)$ is the power set of *R*. This mapping can be visualized as shown in Figure 2, where the task *abandon action scenario* is performed by two roles ("execution staff" and "PR and media staff"), which are modeled in a different dimension than the process elements.

Additionally, the usage of the third dimension supports the aggregation or the generalization of roles, respectively. In Figure 2 the role *Execution staff* and the role *PR and media staff* are aggregated to the role *Civil servants*. The aggregation and generalization of roles are differentiated by the type of arc. In a generalization relationship the arcs are directed. For aggregation we use *simple* lines. The benefit of this representation is that users do not need to switch to another view in order to orient themselves.

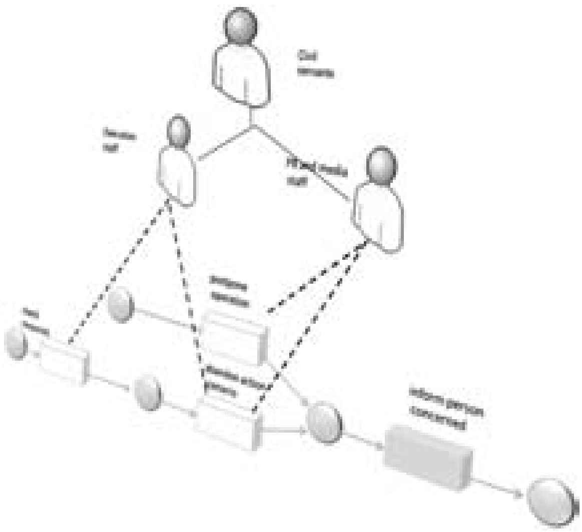


Figure 2: Using the third dimension for representing the relationship between roles and activities

On business process instance level, activity instances require to be associated with resources which can be used to perform the activities. Consequently, we define a mapping from roles to capable and available resources *mapRoleToResource* $: R \rightarrow \mathcal{P}(RES)$, where $RES = \{res_1, .., res_n\}$ is the set of all resources, n is the number of different resources and $\mathcal{P}(RES)$ is the powerset of *RES*. The visualization of the combination of the mappings *mapToRoles* and *mapRoleToResource* is shown in Figure 3, where, for instance, Anne and Gina have the capability of the role *Execution staff*.

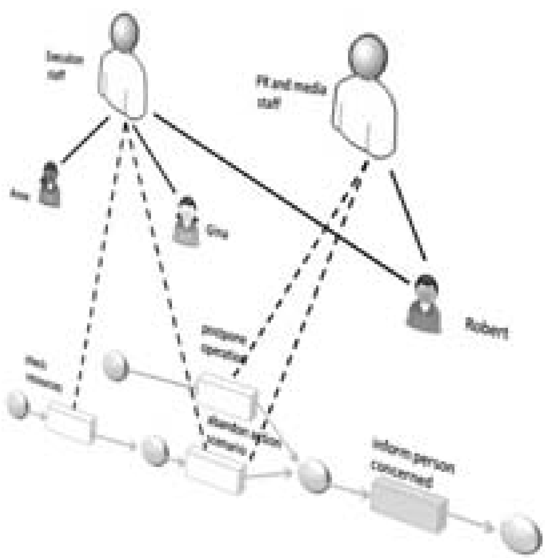
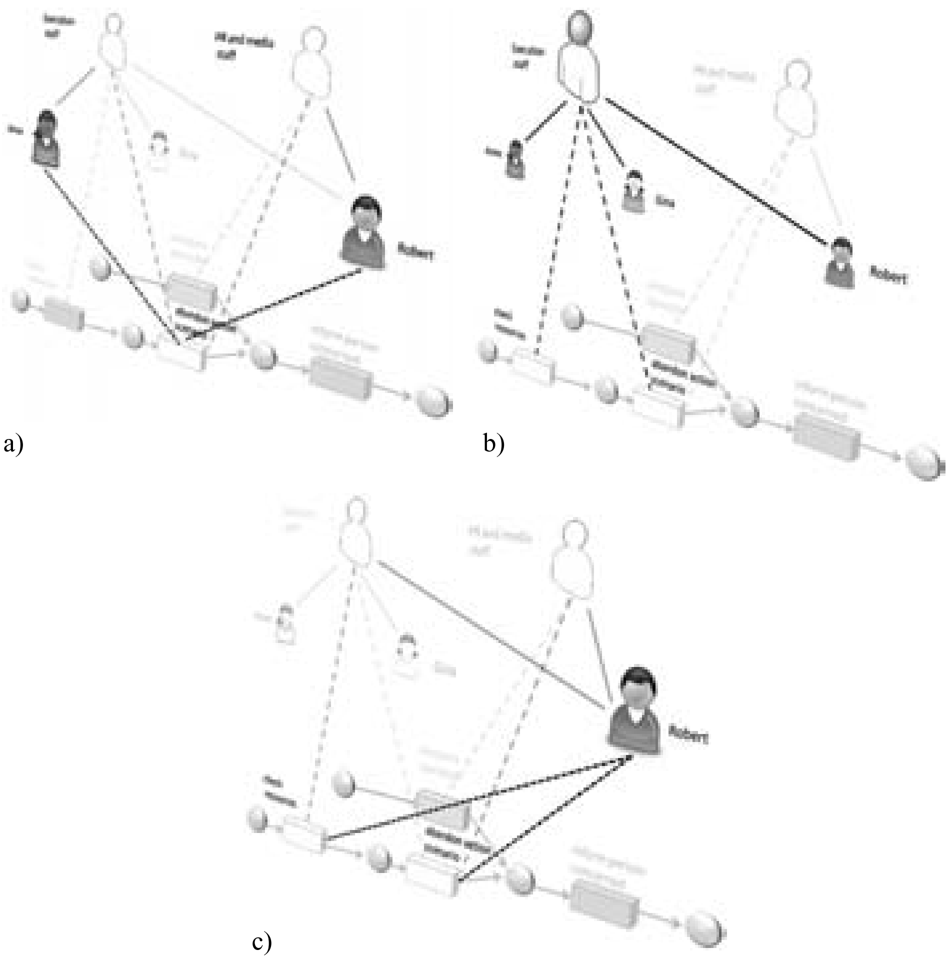


Figure 3: Representing the relationships between roles, resources and activities

The two mappings support the generation of specific views of interesting activities, roles or resources. Figure 3 shows a restricted view on a) roles performing specific activities, b) involved resources of specific roles and c) activities which require specific roles/resources. The benefit of this restriction is that users are able to see how frequently specific roles are involved in the process (i.e. by counting the number of relationships between specific roles and activities). With respect to the given example, this representation overcomes the limitation of switching between process and organizational models in order to achieve this information.



a) b)

c)

Figure 4: Using the third dimension for a view on a) roles, b) resources, and c) activities

4.3 Modeling Semantic Relationships between Business Processes

Besides 3D visualization of one single business process as described above, we will focus in this section on relationships between several business processes. Relationships between two or more business processes increase the amount of information to be integrated into a process model. In this modeling scenario, business processes are represented in parallel planes and the third dimension is used to model relationships in particular semantic relationships, which overcome the limitation of a controlled vocabulary for process elements.

For this, we first calculate similarities between process elements according to [EKO07]. Four different similarity measures are used, which operate on the syntactical, the linguistic, the structural and the abstraction level of process element names. The syntactical measure unveils typos in process element names and the linguistic similarity measure detects synonyms by using the WordNet taxonomy[1]. Homonyms are revealed by the structural measure, which considers a so-called context of process elements. The abstraction level similarity measure detects hyperonyms/hyponyms, which take into account the depth of terms in lexical reference systems such as WordNet. The total similarity value is calculated by a weighted sum of all four measures, which is only computed for the same type of elements (e.g. places vs. places or transitions vs. transitions). If a certain threshold of the total similarity value is reached[2], these two elements are defined as *similar* and a connection will be shown in the modeling environment.

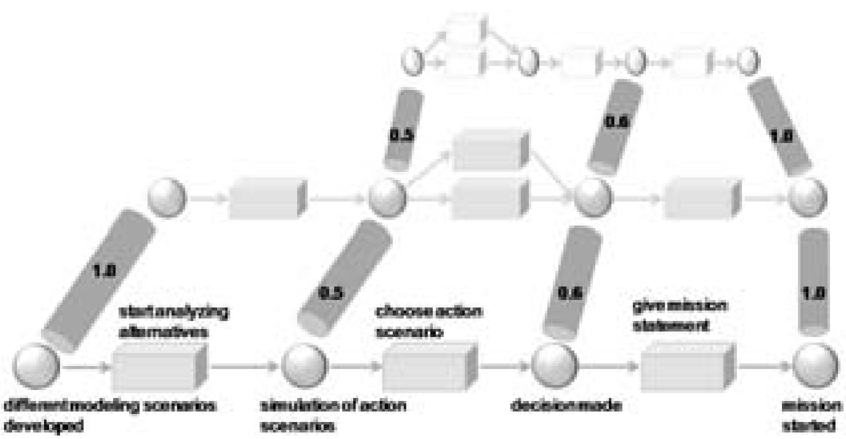


Figure 5 : Representation of similarities between process elements

The value of similarity is visualized analogously to social network analysis tools [WaF94, KRW06, SWS08], e.g., with thickness (more or less bold) or color of the connection cylinders (e.g. scale from bright over gray to black or from green over yellow to red). Users can visualize semantic similarities between places, transitions or both. Figure 5 shows only similarities between places visualized by more or less thick cylinders. The actual similarity value is shown inside the connection bar.

With respect to the given example, when calculating the similarities between another business process and the process from the example the insertion of semantic similarities

[1] http://wordnet.princeton.edu/
[2] Our experiments have shown that a threshold of 0.4 is a satisfying value.

would graphically complicate the understandability of the process models. Therefore, we regard a third dimension in this modeling scenario as effective for easily creating and viewing relationships between process models.

The next section describes how users can efficiently navigate through a hierarchically structured process model.

4.4 Modeling Hierarchies

For Petri nets, several formal net transformations such as refinement and coarsening were proposed to realize a stepwise hierarchical design of processes [Pet73]. In a transition refinement, a transition is refined to a sub-process. It is useful to consider only transition refinements with distinguished input and output transitions [Des05]. Thus, the pre-set of the input transition and the post-set of the output transition are equal to pre- and post-set of the transition to be refined. Coarsening is inverse to refinement where more specific process elements are subsequently linked together to coarse-grained process models.

The current tool implementations support the navigation through the different modeling levels of a business process. Continuing our running example, the navigation is a time consuming task because users need to click inside each sub-concept manually and only one single hierarchy level can be shown at one point of time. We use the third dimension for an efficient navigation through a sequence of refinements. Additionally, through the usage of the third dimension we can restrict the view of hierarchies on relevant refinements and highlight interrelated border elements of the sequence of refinements.

To realize the navigation through a sequence of refinements, we first introduce the notion of a sequence of refinements. A refinement (respectively a coarsening) is a surjective net morphism introduced by [Pet73][3]. Let $B_1,\ldots,B_n$ be business process models with disjoint sets of nodes and $f_1,\ldots,f_{n-1}$ mappings such that (B_{i+1}, B_i, f_i), $i=1,\ldots, n-1$ are quotients. A net morphism (B_2,B_1,f) is called a quotient (B_2 is called a refinement of B_1), iff the mapping $f: X_2 \rightarrow X_1$ denoted by $B_2 \rightarrow B_1$ is surjective on both the nodes and the arcs. Then $SR = (B_1, f_1,\ldots,f_{n-1},B_n)$ is called a sequence of refinements.

Next, we can highlight the same pre- and post-sets of all sequences of refinements. Thus, users can easily catch the relationship between processes. To highlight the same pre- and post-sets we consider a set of business processes and a sequence of refinements. If a business process has a refined transition then we assign the same color to the pre-set of the input transition to be refined and to the post-set of the output transition to be refined.

Figure 6 shows a sequence of refinements and the highlighted pre- and post-set of transitions to be refined. When double clicking on a refined transition, the user can open the next hierarchy level. Users can restrict the navigation through the sequence of refine-

---

[3] Roughly speaking, a net morphism is a mapping from elements of a source net to elements of a target net which respects the bipartition and the flow relation of the source net [DeM91].

ments by moving processes in the background and thus processes appear "*invisible*". In Figure 6, the user opened four refinements and moved one refinement in the background.

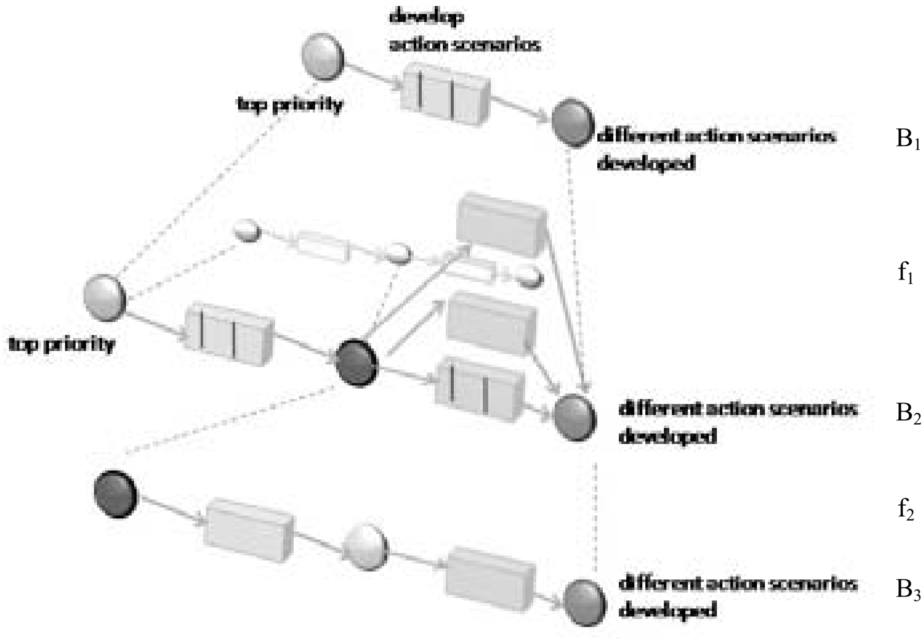


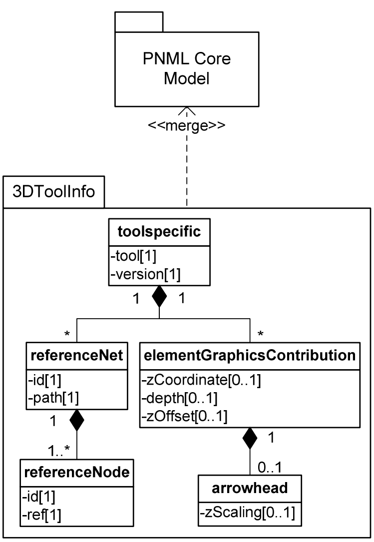
Figure 6: A sequence of refinements and highlighted pre- and post-sets of refined transitions

The next section presents some implementation aspects of our 3D process modeling environment.

5 Implementation

The Petri net Markup Language (PNML) [JKW00] is a standardized XML-based format for persistently storing Petri net models and exchanging them among different tools. The current version of PNML defines a large variety of concepts for describing Petri nets, e.g., Core Model, Multisets, Partitions, High-level Core Structure, Symmetric Nets, etc. Nevertheless, it still provides no means to specify properties for 3D representation of Petri nets, e.g., graphical position and size of Petri net elements in 3D space. Therefore, as the first step of implementing 3D visualization of Petri nets, we extend the PNML grammar with 3D-specific concepts. The extension is limited within the element type "toolspecific.element" that is abstractly defined in all versions of PNML Core Model to store tool specific information. In this way, the concepts for 3D representation of Petri nets are relatively independent from other standardized PNML definitions and can be flexibly reused if the PNML used by a tool needs to be upgraded. By adhering to this limitation, 3D Petri net models can also be interpreted and used as 2D models (despite semantic loss) by other PNML-based tools if they provide no support for 3D visualization of Petri nets. To conform to standard PNML grammar, the concepts for 3D representation of Petri nets are defined in RELAX NG [ClM01], a simple schema language for XML, which has in some aspects advantages over XML Schema, e.g., support for unordered and non-deterministic content, ability to be algorithmically converted into both XML Schema and DTD.

Figure 7 shows on the left side a UML class diagram that defines the package "3DToolInfo" as an extension of the PNML Core Model. On the right side, a code fragment of the corresponding RELAX NG schema is displayed. For simplification purposes, we define in this package only 3D-specific concepts, which can be combined and interleaved with other concepts defined elsewhere in the same-named type "toolspecific.element" by using the RELAX NG-specific attribute "combine" with the value "interleave". The element "toolspecific" defined in "toolspecific.element" has the two child elements "referenceNet" and "elementGraphicsContribution". The element "referenceNet" is used to store reference relationships between Petri net models and contains information to identify and locate the reference net, e.g., ID and path of the reference net, ID and reference ID of reference nodes in the net. In 3D space, these relationships can be depicted by drawing for example dotted lines between nodes and corresponding reference nodes. The element "elementGraphicsContribution" contributes to Petri net element graphics with dedicated properties defined on the third coordinate axis. For instance, "zCoordinate" specifies the position of Petri net elements in the third dimension; "depth" defines together with "width" and "height" the 3D size of the elements; "zOffset" is used to position labels whose location is determined by the offset to the position of their owners; "zScaling" specifies the scale of arrow heads of arcs or directed lines in the third dimension.




```
<?xml version="1.0" encoding="UTF-8"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0"
  xmlns:a="http://relaxng.org/ns/compatibility/annotations/1.0"
  datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
  <define name="toolspecific.element" combine="interleave">
    <element name="toolspecific">
      <attribute name="tool"><text/></attribute>
      <attribute name="version"><text/></attribute>
      <zeroOrMore>
        <ref name="referenceNet"/>
        <ref name="elementGraphicsContribution"/>
      </zeroOrMore>
    </element>
  </define>
  <define name="referenceNet">
    <element name="referenceNet">
      <attribute name="id"><data type="ID"/></attribute>
      <attribute name="path"><data type="anyURI"/></attribute>
      <oneOrMore><ref name="referenceNode"/></oneOrMore>
    </element>
  </define>
  <define name="referenceNode">
    <element name="referenceNet">
      <attribute name="id"><data type="ID"/></attribute>
      <attribute name="ref"><data type="IDREF"/></attribute>
    </element>
  </define>
  … …
</grammar>
```

Figure 7: The package 3DToolInfo and corresponding RELAX NG schema

Organization models, roles and resources are important process objects in business process modeling. With our technique, they are persistently stored in separated XML files typified by RELAX NG schemas and bound to PNML documents via links. To better visualize these process objects in 3D space, 3D graphics and images are used whose path, z-offset, depth and other properties like transparency are also specified in the respective RELAX NG schemas.

Based on the extended PNML grammar and related RELAX NG definitions for process objects, a prototype for the purposed 3D visualization of Petri net has been implemented. Figure 8 shows a screenshot of a 3D business process modeled with our prototype.

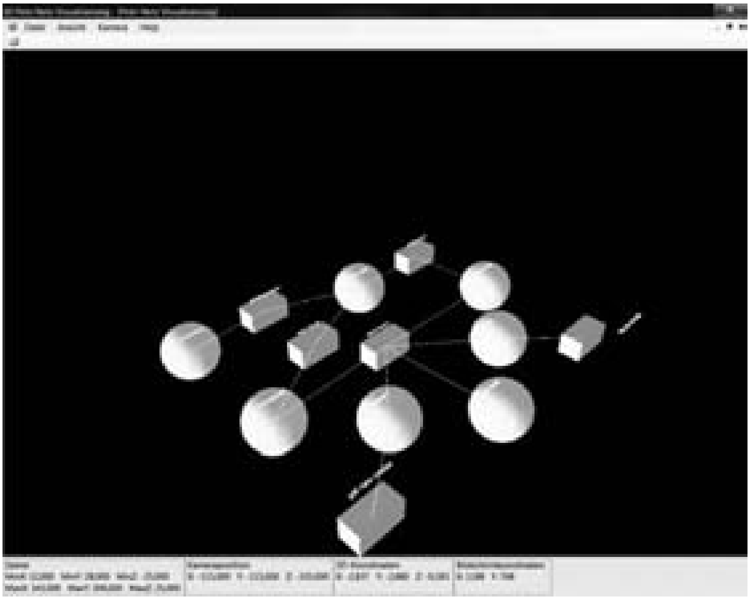


Figure 8: Screenshot of the 3D modeling environment

6 Conclusion and Outlook

In this paper we presented a Petri net-based approach for the integration of a third dimension into the representation of business process models. We presented related work in the areas of business process modeling, 3D modeling, and graphical layout methods. Based on this, we discussed some deficits of current 2D process modeling and showed how to improve layout and to increase information content of Petri net models by introducing a third modeling dimension. The need for a third modeling dimension was pointed out in three different modeling scenarios: modeling relationships between roles, resources and activities, modeling semantic relationships between business processes, and modeling hierarchies. Within these scenarios we showed that the 3D representation of processes facilitates the access to process-specific information. Whilst in conventional 2D representation process-specific information interfere with each other, in 3D the information becomes straightforward. By turning the process model in its 3D environment, one can examine different views and gather easily process-specific information. For the implementation of our approach we extended the PNML grammar by defining concepts for visualization of 3D Petri nets in RELAX NG.

Future work comprises the evaluation of our approach and the integration of the implemented prototype into the Petri net-based process modeling framework INCOME2010 [KLO08]. In addition, 3D visualization and animation of other process objects (e.g. process metrics such as time, cost, etc.) will be explored in our future research. In this contribution, we are focusing on a 3D representation concerning the control flow of processes. A further step in the future will be a 3D representation concerning the data flow of processes (e.g. XML documents in XML nets). Furthermore, for simulation purposes we intend to introduce a dynamic third dimension for process elements (e.g. places, symbolized as balls in 3D, change their volume and/or color according to the number of tokens they contain).

References

[BES00] Ballegooij, A.; Elliens, A.; Schönhage, B.: 3D Gadgets for Business Process Visualization – A Case Study. In: Proceedings of the fifth symposium on Virtual reality modelling language, pp. 131-138, 2000.

[BRB07] Bobrik, R.; Reichert, M.; Bauer, T.: View-based process visualization. In: 5th International Conference on Business Process Management, Springer, pp. 88-95, 2007.

[ClM01] Clark, J.; Murata, M.: RELAX NG Specification, OASIS Committee Specification, 2001.

[DGB07] Decker, G.; Grosskopf, A.; and Barros, A.: A Graphical Notation for Modeling Complex Events in Business Processes. In: Proceedings of the 11th IEEE international Enterprise Distributed Object Computing Conference, IEEE Computer Society, Washington, DC, 2007.

[DeM91] Desel, J.; Merceron, A.: Vicinity respecting net morphisms. In: Proceedings on Advances in Petri Nets, G. Rozenberg, Ed. Springer-Verlag New York, New York, NY, 2001, 165-185.

[Des05] Desel, J.: Process Modeling Using Petri Nets, In: M. Dumas, W.M.P. van der Aalst, A.H.M. ter Hofstede, Process-Aware Information Systems, pp. 147-177, 2005.

[EKO07] Ehrig, M.; Koschmider, A.; Oberweis, A.: Measuring Similarity between Semantic Business Process Models. In: Proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling, volume 67, pp. 71-80. Australian Computer Science Communications, Ballarat, Australia, 2007.

[HaL08] Hadar, I.; Leron, U.: How intuitive is object-oriented design? Communications of the ACM, vol. 51(5), pp. 41-46, 2008.

[JaB96] Jablonski, S.; Bussler, C.: Workflow Management: Modeling Concepts, Architecture and Implementation, Itp New Media, October 1996.

[JaG07] Jablonski, S.; Goetz, M.: Perspective Oriented Business Process Visualization. In: Proceedings of the Business Process Management Workshops, Lecture Notes in Computer Science 4928, pp. 144-155, Springer, BPM 2007.

[Jen94] Jensen. K.: An Introduction to the Theoretical Aspects of Coloured Petri Nets. In: Reflections and Perspectives, Lecture Notes in Computer Science, pp. 230-272. Springer, 1994.

[JKW00] Jüngel, M.; Kindler, E.; Weber, M.: Towards a Generic Interchange Format for Petri Nets. In: Meeting on XML/SGML based Interchange Formats for Petri Nets, pp.1-5, Aarhus, Denmark, 21st ICATPN, 2000.
[KaK89] Kamada, T.; Kawai, S.: An algorithm for drawing general undirected graphs. Information Processing Letters 31, 1989.
[KHG08] Koschmider, A.; Habryn, F.; Gottschalk, F.: Real Support for Perspective-compliant Business Process Design. In: 4th International Workshop On Business Process Design. Milan, Italy, 2008.
[KiP04] Kindler, E.; Páles, C.: 3D-Visualization of Petri Net Models: Concept and Realization. In: Proceedings of International Conference of Applications and Theory of Petri Nets, pp. 464-473, Springer, 2004.
[KRW06] Klink, S.; Reuther, P.; Weber, A.; Walter, B.; Ley, M.: Analysing Social Networks Within Bibliographical Data, Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA), Krakow, Poland, September 2006.
[KLO08] Klink, S.; Li, Y.; Oberweis, A.: INCOME2010 - a Toolset for Developing Process-Oriented Information Systems Based on Petri Nets. In: Proceedings of International Workshop on Petri Nets Tools and APplications (PNTAP 2008, associated to SIMUTools 2008). Marseille, France, 2008.
[KGM99] Krallmann, H.; Gu, F.; Mitritz, A.: ProVision3D - Eine Virtual Reality Workbench zur Modellierung, Kontrolle und Steuerung von Geschäftsprozessen im virtuellen Raum. Wirtschaftsinformatik 41(1): pp. 48-57, 1999 (in German).
[LeO03] Lenz, K.; Oberweis, A.: Inter-Organizational Business Process Management with XML Nets. In: Petri Net Technology for Communication-Based Systems, Advances in Petri Nets volume 2472 of LNCS, pp. 243-263. Springer- Verlag, 2003.
[LHG07] Li, L.; Hosking, J.; Grundy, J.: Visual Modelling of Complex Business Processes with Trees, Overlays and Distortion-based Displays. In: Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing VLHCC, IEEE Computer Society, Washington, DC, 137-144, 2007.
[MaM07] Maier, S.; Minas, M.: A Pattern-Based Layout algorithm for Diagram Editors. In: Proceedings of the Workshop on the Layout of (Software) Engineering Diagrams, 2007.
[MeS06] Mendling, J.; Simon, C.: Business Process Design by View Integration, In: Business Process Management Workshops, pp. 55-64, Springer, 2006.
[Mil08] Miles,R.: Microsoft XNA Game Studio 2.0. In: Microsoft Press Corp., 2008
[NeK05] Nestmann, F.; Kron, A.: Frühwarnung und Management von Katastrophen am Beispiel Hochwasser. In: Naturgefahren und Kommunikation, Beilage der Zeitschrift GAIA – Ökologische Perspektiven für Wissenschaft und Gesellschaft Heft 14/3, 2005.
[Pet73] Petri, C.A.: Concepts of Net Theory. Mathematical Foundations of Computer Science. In: Proceedings of Symposium and Summer School, High Tatras, Mathematical Institute of the Slovak Academy of Sciences, pp.137-146, 1973.
[Pla02] Plate E.J.: Flood risk and flood management. In: Journal of Hydrology 267 2–11, Elsevier, 2002
[Pur98] Purchase, H. C.: Performance of Layout Algorithms. In: Comprehension, not Computation, Journal of Visual Languages & ComputingVolume 9, Issue 6, pp. 647-657, 1998.
[Rei93] Reiss, S.: A Framework for Abstract 3D Visualization. In: Proceedings of the IEEE Symposium on Visual Languages, pp 108-115, IEEE Computer Society Press, 1993.
[ReR98] Reisig, W.; Rozenberg, G. (Eds.): Lectures on Petri Nets I: Basic Models. Volume 1491, Lecture Notes in Computer Science, Springer, 1998.
[RiB06] Rinderle, S.; Bobrik, R.; Reichert, M.; Bauer, T.: Business Process Visualization - Use Cases, Challenges, Solutions. In: Proceedings of the Eighth International Conference on Enterprise Information Systems (ICEIS'06), pp. 204-211, 2006.
[Röl07] Rölke, H.: 3-D Petri nets. In: Petri Net Newsletter, No. 72, pp 3-9, April 2007.
[Sch99] Scheer, A.-W.: ARIS – Business Process Modeling, Springer Verlag, Berlin, 1999.

[SWS08] Said, Y. H., Wegman, E. J., Sharabati, W. K., and Rigsby, J. T. 2008. Social networks of author-coauthor relationships. Comput. Stat. Data Anal. 52, 4 (Jan. 2008), 2177-2184. DOI= http://dx.doi.org/10.1016/j.csda.2007.07.021

[Ver00] Verbeek, E.: ExSpect 6.4x product information. In: Mortensen, K. (Ed.): Petri Nets 2000: Tool Demonstrations, pp 39-41, 2000.

[WaF94] Wassermann, S; Faust, K.: Social Network Analysis: Methods and Applications, Cambridge University Press, New York, 1994.