Electronic Communications of the EASST
Volume 37 (2011)

Workshops der wissenschaftlichen Konferenz Kommunikation in Verteilten Systemen 2011 (WowKiVS 2011)

GPMN-Edit: High-level and Goal-oriented Workflow Modeling

Kai Jander and Winfried Lamersdorf

12 pages

Guest Editors: Horst Hellbrück, Norbert Luttenberger, Volker Turau
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

GPMN-Edit: High-level and Goal-oriented Workflow Modeling

Kai Jander[1] **and Winfried Lamersdorf**

Distributed Systems and Information Systems Group
University of Hamburg, Germany
{jander@informatik.uni-hamburg.de}[1]

Abstract: Workflow modeling and tools for designing and editing workflows is an important part of business process automation. One of the challenges of workflow design has been the gap between business users and technical users. While technical users require detailed and formal models for execution, the focus for business users generally consists of more abstract and business-related aspects of the workflow such as the purpose of the workflow and business goals it derives from the business process. One approach aiming to address this gap are goal-oriented workflows which explicitly represent the business goals as part of the workflow semantic. This paper presents an approach and a tool called GPMN-Edit for modeling such goal-oriented workflows using the Goal-oriented Process Modeling Notation (GPMN). Furthermore, GPMN-Edit allows users to mask specific details of the workflow they consider irrelevant for the task at hand while still being able to continue changing the workflow.

Keywords: Workflows, Workflow Modeling, Goal-oriented Workflows, GPMN, Workflow Tools

1 Introduction

Workflow modeling is an important aspect of Business Process Management and is therefore used in almost every industry to automate business processes and increase productivity. It involves identifying the portions of a business process which can be automated and creating a workflow which automates those parts. Depending on the business process, this workflow can include the complete process in cases of highly structured and repeatable process or it can be limited to one or more parts of the business process if the process structure is weak.

After a business process has been identified, an important step is formalizing which parts of the process should be automated and which may only exist in the form of informal guideline documents or expert knowledge. As a result, additional technical information is needed in order to unambiguously specify the workflow execution semantics.

Therefore, a workflow contains a mixture of two different sets of information which include knowledge from two different groups of stakeholders involved in the specification of the workflow. The first group of stakeholders are the technical designers, who have a *technical view* on the workflow meaning they emphasize the formal specifications over the business aspects. The formal specification allows automation and execution of the workflow using a workflow engine.

The second group of stakeholders are more involved in the business aspects of a workflow. These business designers have a *business view* on the workflow. This means they are usually less interested in the execution details of the workflow but want to be aware of information concerning the business decisions involved in the workflow. These can include aspects such as business goals and Key Performance Indicators (KPI). This emphasis of business operation aspects align the workflow details with the overarching strategic goals of the business and ensure the usefulness of the workflow.

These two aspects of workflow modeling usually result in a gap between the two views. This gap can be further exacerbated by the fact that the business designers often lack the technical knowledge required for formal specification while the technical group often lacks knowledge about the business aspects and are unable to appreciate the needs of the former group.

The existence of this gap between business and technical views has led to attempts to separate them and use different languages to describe each of them. Systems which employ a combination of the *Business Process Modeling Notation (BPMN)* [BPM08] and the *Web Services Business Process Execution Language (WS-BPEL)* [WSB07] could be perceived as one of such attempts of separation; however, BPMN itself is still a quite detailed language with considerable focus on execution semantics.

A different approach towards bridging the gap between the two views is the use of goal-oriented workflow models [BPJ+10]. This approach takes concepts such as goals from the business view and uses related concepts from BDI agent technology to give them execution semantics. The advantage of these workflow models is that they integrate business concepts and execution semantics, allowing for a more intuitive approach of workflow design. In addition, they allow for flexible execution of the workflow at runtime depending on the business situation and modeled goals.

In this paper we introduce an approach for modeling goal-oriented workflows which builds on and extends the concepts described in [BPJ+10] by adding a new kind of plan and an additional edge used for goal suppression. We will introduce a software tool for modeling such workflows called *GPMN-Edit* developed in cooperation with our industry partner Daimler. This software tool uses concepts such as hiding and detail levels aimed at allowing the user to configure their current view on the workflow to the desired level of technical abstraction while still allowing the workflow to be edited on any chosen view.

The following section will give an overview of related work in the area of workflow modeling tools. Afterwards, we will introduce the goal-oriented workflow models used and our approach for modeling them. Finally, we will present concepts that could be used to enhance the GPMN-Edit tool and extend the approach.

2 Related Work

Since workflow modeling is a central concept in business process management, there are a large number of workflow model languages and workflow modeling tools available. The approaches tend to differ markedly and emphasize different aspects of workflow modeling like well-defined semantics, control flow and execution flexibility. The resulting tools usually cater to their specific type of workflow model and therefore tend to differ from each other.

However, with regard to the integration of business and technical view, they can be roughly put in three categories. The first category of workflow models are more concerned with other aspects of workflow modeling and only include a few business concepts and put clear emphasis on the technical side of workflow modeling. The most extreme example of this approach is WS-BPEL, which explicitly focuses on the technical side, assuming that by the time it is used, the business aspects of workflow modeling have already been processed using a different modeling approach. Despite the fact that extensions for including aspects beyond service orchestrations such as BPEL4People [ABO$^+$07] are available, the focus remains technical.

Another example along similar lines is Yet Another Workflow Language (YAWL) [AH05]. It has a strong technical focus on well-defined semantics based on petri nets. However, deriving the language from petri nets also means that a graphical representation of the workflow is available.

The second category contains workflow models that integrate abstract business aspects alongside the technical information within the workflow model itself. This category represents an attempt to appeal to both technical and business users. The actual degree of business aspects can range from moderate with a high technical focus bordering on the first category, to extensive. An example of such an approach is BPMN. It includes aspects beyond technical execution such as pools, lanes and documents. Since BPMN is less technical than WS-BPEL, it is often used as its front end during workflow modeling. Workflows are designed in BPMN, then converted to BPEL for execution. However, even though this approach is quite common, it does tend to exhibit some problems [RM06]. BPMN attempts to integrate business aspects in the workflow model and tries to appeal to both business and technical users. However, this attempt to bridge the gap between business and technical views has led to informalities leaving some ambiguities in the semantics of BPMN including underspecified process termination conditions, exception handling for concurrent subprocess instances and unclear definition of the OR-join element [DDO08].

A second example in this category is Goal-Oriented BPMN (GO-BPMN) developed by Whitestein Technologies [RB07]. Similar to our workflow models, it uses a goal-oriented approach which employs representations of business goals as a functional feature of the workflow itself. However, our approach derives more plan and goal reasoning concepts from its technical base and includes some additional elements such as intermediary plans inserted between goals.

The third category includes comprehensive workflow modeling approaches which include a large variety of business aspects. However, unlike approaches from the second category, these approaches use additional models to represent the various business aspects. A prominent comprehensive approach of this type is the *Architecture of Integrated Information Systems (ARIS)* (see [SS06] and [SN00]), which includes a workflow modeling language called *Event-driven Process Chains (EPC)* with similar features regarding the integration of business views as the second category, but also offers tools for modeling additional aspects about the workflow.

The concepts used in the third category allow the workflow designer to include additional aspects beyond mere execution semantics in the workflow. However, the result is a large number of sometimes unrelated models which may be difficult to keep in a consistent state. For example, some of the modeled aspects are non-functional and exist solely for documentation purposes. While this is unquestionably useful, it involves the risk of the workflow being changed and slowly diverging from its documentation over time unless rigid discipline is maintained.

The advantage of the second category are the additional aspects included alongside the execution semantics within the workflow model. In cases such as GO-BPMN, the business aspects form an integral part of the workflow. This mitigates the consistency problem of approaches in the third category by showing the workflow engineer potential inconsistencies. It would there-

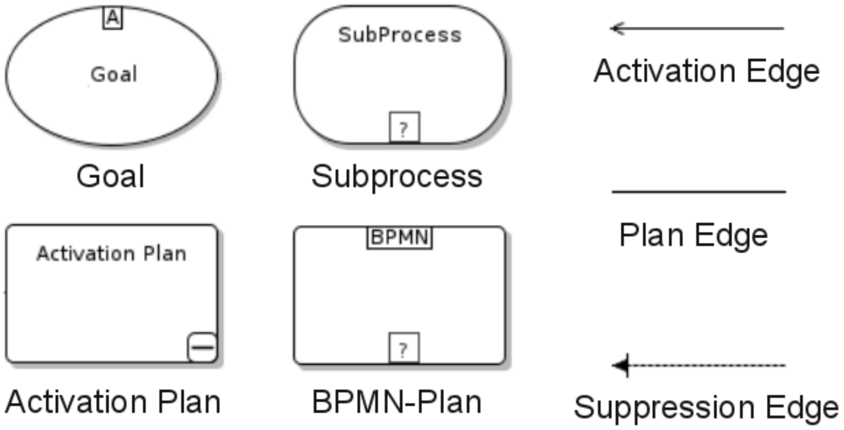

Figure 1: The graphical elements used in GPMN

fore be desirable to have an integrated approach similar to those in the second category which includes as many aspects of the third category as possible.

An approach for simplifying the view on workflow models is the Proviado Framework (see [BRB07]). The framework has goals similar to our approach which is the presentation of a different view on the workflow for different groups of users. However, our approach differs in that its application is focused on the use in goal-oriented workflows and provides a linear set of detail levels which increasingly focus on the business goals of the process.

This also applies to similar approaches for adapting views to different groups of users for workflow notations like BPMN. While the workflows can be simplified by omitting or replacing workflow elements such as branches and activities, the workflow itself is centered around activities and, unlike goals, do not directly convey information about the purpose of activities. Our approach uses the basic idea of simplifying workflows and aims at integrating aspects of the third category of workflow concepts into a concept following the second category.

3 Approach for modeling GPMN-Workflows

This section will describe the approach used by GPMN-Edit for modeling goal-oriented workflows. The first part includes an overview of the graphical modeling language GPMN, describing goal-oriented workflows which are then used as workflow models in the workflow engine as well as GPMN-Edit itself. The next part introduces GPMN-Edit and describes how the tool increases the abstraction of the basic GPMN model by employing concealment of technical detail.

3.1 GPMN Workflows

GPMN-Edit is used to create and modify workflows defined using the *Goal-oriented Process Modeling Notation (GPMN)* [BPJ$^+$10]). GPMN consists of a number of graphical elements as shown in Figure 1. The technical basis for executing GPMN processes are a type of software agent technology called BDI agents, the concrete implementation used for the workflows being Jadex (see [PB09]).

The term BDI stands for Beliefs, Desires and Intentions. Beliefs is knowledge acquired by the agent. If a BDI agent is used to implement a GPMN workflow, the beliefs of the agent are referred to as workflow *context*. The context contains all information concerning the business process that are managed by the workflow instance. This is usually process-specific information which includes, for example, customer addresses, lists of purchased items and change request information. The information contained in the context directly influences the behavior of the workflow.

The desires of a BDI agents are abstract concepts of what the agent wants to achieve. Desires are represented both in agents and workflows as goals. Once a goal becomes active, the agent strives to fulfill this goal. Since goals can contradict each other, conditions and configurations of goals set by the user can influence which goals are pursued and which are suspended. In terms of a workflow, the decision is usually based on information contained in the workflow context. This goal reasoning distinguishes goals from subprocesses in workflow notations like BPMN.

Finally, intentions, which are expressed as plans for both BDI agents and GPMN workflows, represent a set of concrete actions which the agent or workflow can chose to fulfill active goals. The decision as to which plan is the most appropriate is influenced both by association with a goal and, again, rules and conditions as defined by the user based on information contained in the context. This allows the workflow engineer to implement several options for achieving a goal which may be appropriate in some situations but excluded by condition in others.

In GPMN goals are represented by the goal element. Motivated by the reason as the underlying BDI infrastructure, GPMN currently includes four goal types (see also [BPML05]): *Achieve goals* aim to achieve a specific workflow state, *perform goals* initiate the performance of a task, *query goals* acquire information and *maintain goals* continuously monitor the workflow state and intervene if it enters an undesired state.

Since the workflow engineer also has to include the means to accomplish goals, plans need to be available. Currently, GPMN supports two kinds of plans. The first kind of plan is the activation plan, which allows goals to have subgoals. The activation plan can be connected to subgoals using activation edges. During execution, the plan causes connected subgoals to become active in a configurable manner such as parallel sequential activations.

Activation plans and edges allow a workflow engineer to construct *goal hierarchies*, in which top level goals are accomplished through their subgoals, which again can be divided into even more subgoals. This goal decomposition continues until the goal is sufficiently concrete to accomplish it using a simple sequence of activities.

An example of a goal hierarchy can be seen in Figure 2. This example is part of a larger workflow used at Daimler which handles change management for a process. This specific section shows the generation of a change request description. The top level goal starting the hierarchy states that the goal of this workflow fragment is the description of the change request, which is then delegated to more specific subgoals using activation plans and edges.

Once a goal in the hierarchy becomes concrete enough for a simple implementation of a concrete sequence of activities, the second kind of plan called *BPMN-plan* is used. BPMN-plans define a series of activities in BPMN notation which are aimed at achieving the goal it is connected to using a plan edge.

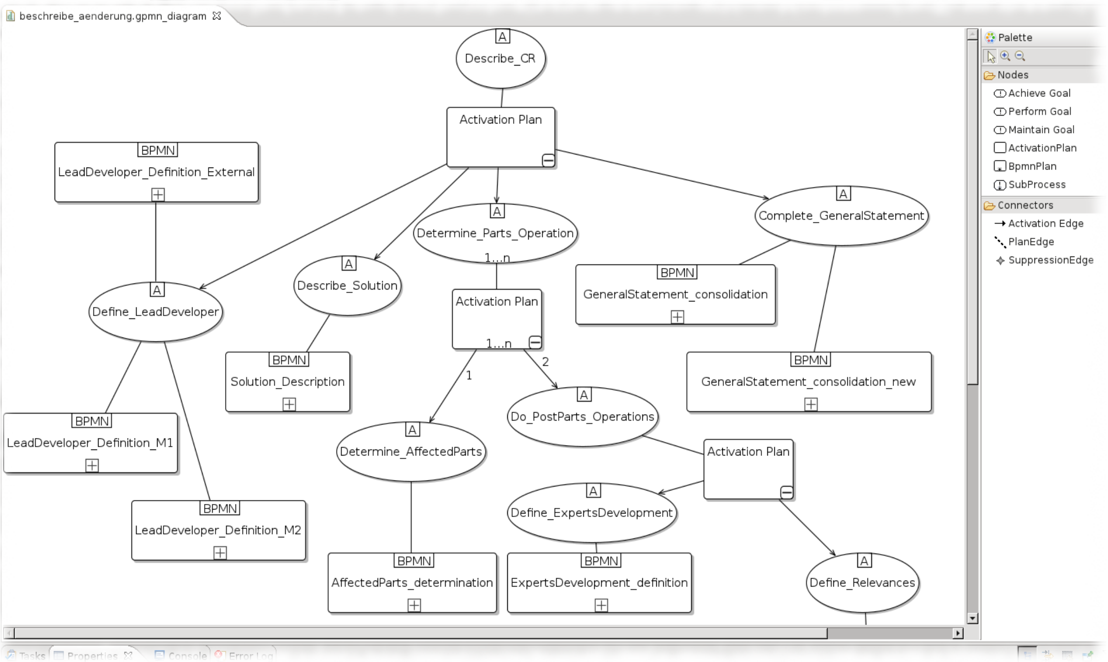

Figure 2: Part of a change management workflow used by Daimler, this section shows the generation of a change request description

In addition to the elements motivated by the underlying BDI agents, GPMN also includes subprocesses. Subprocesses are started by an activation plan which has an outgoing activation edge connected to the subprocess. The subprocess is treated just like a goal from the perspective of the activation plan, meaning the process is started either sequentially or in parallel with other connected subprocesses and plans and the activation plan is considered finished once all of them have terminated.

Subprocesses allow the workflow engineer to modularize the workflow, reducing the size of the individual goal hierarchies. Depending on the configuration details, subprocesses can also make use of concurrencies and allow a form of parallel execution of the workflow.

The final element of GPMN is the suppression edge. This edge can connect goals with other goals and indicates a suppression relation between goals. This means that once a goal with outgoing suppression edges becomes active, it temporarily suspends all goals connected with the suppression edges. On goal termination, the temporary suspension is released and the other goals can continue to execute plans. This allows goals which have an overriding importance to suspend conflicting or distracting goals until they have finished.

GPMN workflows are stored in an XML-based file format. This file format only supports the seven described elements as described in the metamodel seen in Figure 3. Information about graphical layout are stored in a separate file. The GPMN workflow model is closely related to the underlying model of BDI agents. This allows the workflow description to be easily translated into BDI agents.

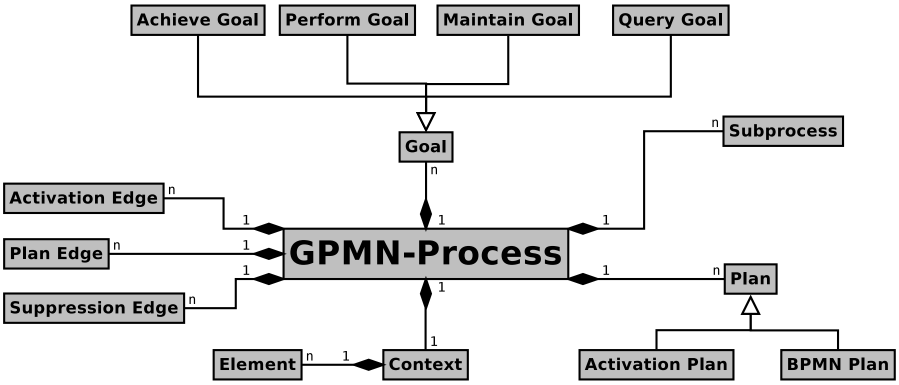

Figure 3: The GPMN file format meta model describes the file format used to store GPMN workflows

In the following part we will describe how the editor uses this format to introduce more abstract concepts while still maintaining the simple format to describe GPMN workflows.

3.2 GPMN-Edit Concepts

While the basic GPMN elements could be used to design goal-oriented workflows, they contain a considerable amount of technical information that is not always needed by the workflow engineer.

One example of an element that is not always required and is only useful on occasion for modeling specific details of the workflow is the activation plan. While including activation plans in the workflow concepts allows the configuration of such plans for parallel and sequential behavior, and allowing the mixture of BPMN-plans and subsidiary goal hierarchies as execution options for goals enhances the expressiveness of the language, it is an element primarily concerned with execution semantics.

As a result, user feedback received from our industry partner Daimler suggested that some users may wish for an abstracted view of the workflow. Therefore we concluded that it should be possible to avoid explicitly displaying the activation plans if the user is not interested in them; nevertheless, the relationship of goals activating subgoals should still be visible.

This step necessitated the introduction a new element to represent the activation relationship between goals. Since this new element would have no impact the underlying workflow model, it should only be part of the visual representation of the user and not of the workflow model itself.

This new edge replaces the activation plan, a plan edge and an activation edge and represents an activation relationship between goals. Since the relationship it represents is similar to that of an activation edge, it is called a *virtual activation edge*. However, since the edge actually replaces three components which potentially connect more than two other elements, the replacement of the activation plan and connecting edges with the virtual activation edge is non-trivial.

In addition to hiding the activation plans, GPMN-Edit allows the workflow engineer to conceal BPMN-plans, allowing the reduction of the graph to the business goals of the workflow.

Unlike concealing activation plans, this feature is considerably simpler since BPMN-plans do not include any outgoing connection.

While both features can be used on individual elements at any time, GPMN-Edit provides three default visibilities which employ the two concealment features in order to represent three levels of abstraction. The first level exposes all technical details of the model including all activation plans and BPMN-plans reflecting how it will be implemented as a BDI agent in the workflow engine.

The second level is an intermediate level of abstraction which hides the activation plans but exposes the BPMN-plans. This level hides the most intricate technical details of the workflow while still displaying the execution option for each of the business goals in the workflow. The abstraction level is similar to models used in other goal-oriented workflow systems like GO-BPMN.

Finally, the third level has the highest level of abstraction and focuses solely on the business goals. This level is intended to give business users an overview of the objectives of the workflow and their relationships while hiding the technical details.

All three abstraction levels are controlled using a detail slider. Setting the detail slider to the desired level automatically conceals or reveals the elements appropriate for the level. However, the user can still individually conceal or reveal particular elements if more or less detail are required locally. In addition, GPMN-Edit allows manipulation of all visible elements, including virtual or purely visual ones, automatically adjusting the underlying model using defaults based on common use cases.

The usefulness of detail levels can be seen when applied to a moderately complex workflow such as the Daimler change management workflow fragment shown in Figure 2. The workflow fragment contains a goal hierarchy which includes three activation plans and eight BPMN-plans. One of the activation plans is configured to activate its subgoals sequentially while the rest activate their subgoals in parallel.

The goals “Define_LeadDeveloper” and “Complete_GeneralStatement” are connected to multiple BPMN-plans using plan edges. This means once the goal becomes active, one plan is chosen depending on the process context. In the full-detail level shown, the workflow contains a considerable amount of technical information alongside more abstract information such as goals. A user who is not interested in BPMN-plans or the detailed configuration of activation plans may wish to chose to select the most abstract level, focusing on the business goals.

As an example, we will use the earlier workflow fragment to demonstrate the highest level of abstraction with the least technical detail (see Figure 4). The goal is to reduce the apparent complexity of the workflow and reduce the number of visible elements which could distract from the business goals of the workflow.

Despite this considerable reduction of visible elements, the workflow can still be modified. For example, virtual activation edges can be reconnected to different goals. This means the editor has to adjust the underlying activation plan and edges to reflect this change, which can be done transparently for the user.

For example, if two goals share an activation plan which connects to common subgoals, it would be represented by virtual activation edges connecting each of the two goals to each of the subgoals. If the workflow engineer connects one of the virtual activation edges to a different subgoal, merely reconnecting the activation edge of the underlying activation plan is insufficient

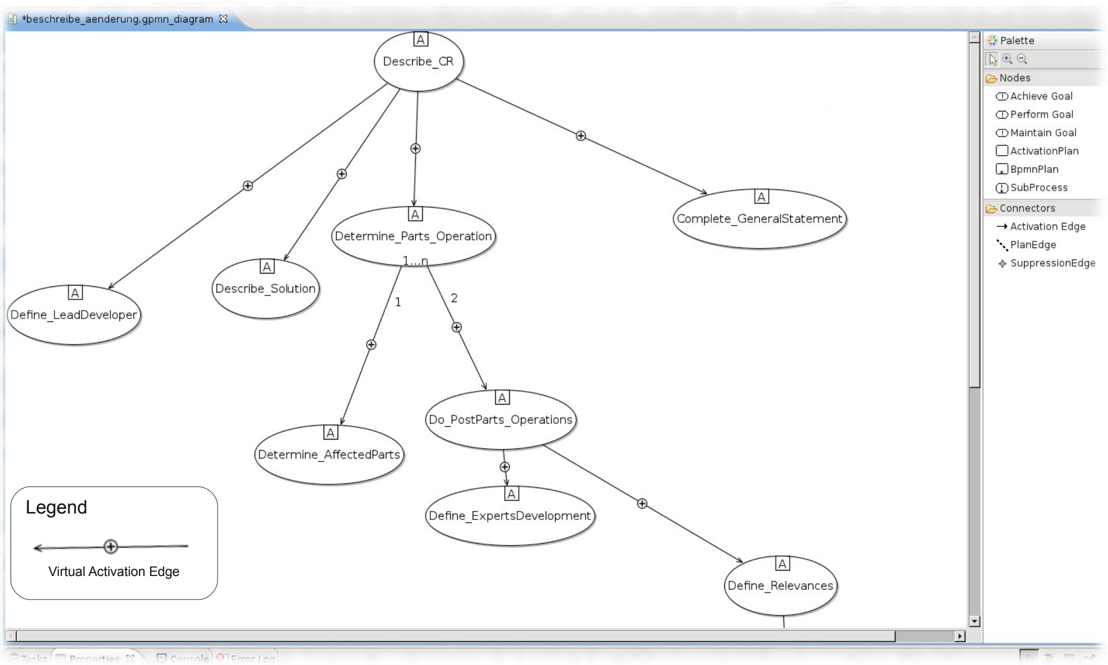

Figure 4: The workflow plans have been hidden and activation plans have been replaced by virtual activation edges, focusing the view of the workflow on the business goals

since the modification only applies to a single top-level goal. The editor has to generate a copy of the shared activation plan and reconnect the plan and activation edges accordingly to represent the new relationship between the goals.

Since the workflow detail is reduced at higher levels of abstraction, it also diminishes the expressiveness of the notation. However, the default action performed by the editor reflect the most common use cases and thus only requires the additional expressiveness in special cases. In any event, the workflow modeler always has the choice of overriding the abstraction level locally, by unhiding elements of the workflow to obtain access to required details.

3.3 GPMN-Edit Implementation

Since one of the goals for the implementation of the modeling tool was inclusion in an integrated development environment (IDE), GPMN-Edit has been implemented as a plug-in for Eclipse [Ecl]. Since some specialized workflows may require integration with custom systems, such as industrial robots, the integration of the workflow editor in the same development environment as the technical integration has the advantage of simplifying test procedures for such workflows during development. In addition, integration allows the use of other facilities of the IDE such as repository and project management.

Development effort for the tool was reduced by basing it on the Eclipse Modeling Framework (EMF) and the related Graphical Modeling Framework (GMF). These frameworks enable model–driven development by allowing developers to specify a metamodel and automatically generate the appropriate code for internal model consistency, graphical interface and model storage. The model storage already separates the business model from representational information such as layout.

The automatically generated code is extended with additional features such as element hiding, virtual activation edges and levels of detail. Furthermore, some of the interface components and element representations have been modified to enhance ease of use.

Additional modification have been added to correctly implement user manipulations. For example, reconnecting virtual activation edges requires modification of the business model.

In the next section we will describe potential enhancement to the current modeling approach. This includes additional levels of detail and other concepts to provide additional information and further ways of abstraction for GPMN workflows.

4 Future Work

In this section we will present further concepts which could help to broaden the range of abstraction levels available to the workflow engineer. Since the editor is still being tested and developed, a final evaluation of its performance in a business environment has not been conducted, however, initial responses by our industrial partner have so far been positive. Once development has reached a sufficient state, a full evaluation will be conducted, centering less on quantitative performance gains over competing concept but more on qualitative gains such as the possibility of modeling low-structure knowledge and ad-hoc processes.

While GPMN-Edit is already capable of displaying significant technical detail as needed, some of the detail such as the BPMN workflow fragments and their effects are still hidden. While BPMN-plans do not have outgoing edges, a task is available for use in BPMN-plans which can reference goals in the GPMN model and allows the activation of those goals(Goal Activation Task). This type of goal activations may depend on runtime data in case of gateways and is currently not graphically represented in the model. Since activations of this type do not always trigger during runtime they can be considered *potential activations*.

We therefore propose an additional level of detail which includes more technical information than the level exposing activation plans. In addition to displaying BPMN-plans and activation plans, this level would also display the potential activations between BPMN-plans and goals. Unlike the activation edges which express definite activations at runtime and plan edges which represent deliberation options with regard to actions, the relationship between BPMN-plans and goals is a form of goal activation, yet dependent on runtime information and decisions similar to plan edges. As a result purely visual edge type called *potential activation edge* would be needed to express this relationship. These edges would only be displayed in the new high detail level.

Following the idea of allowing modification on all levels of detail, these edges could also be manipulated by changing the targeted goal. The editor would rewrite the BPMN workflow fragment by replacing the targeted goal.

Another concept we are currently investigating are *goal patterns*, which represent process patterns consisting of a specific arrangement of goals and plans used in the workflow model. This can include arrangements such as combinations of activation plans using parallel and sequential execution and reactivations of parts of the goal hierarchy. Goal patterns could ultimately be expressed by a single complex element in GPMN-Edit on a higher level of abstraction, yet represented as multiple elements in both the underlying workflow model and on high levels of detail in GPMN-Edit.

Finally, GPMN-Edit could be enhanced to allow even higher levels of abstraction than the ones currently offered. In addition to concealing plans and plan details and potentially goal patterns, specific goals could be hidden to further focus the view on core business goals. This would mean concealing high-detail goals first, which tend to appear lower in the goal hierarchy.

However, since GPMN workflows can potentially contain multiple separate goal hierarchies, this poses a potential problem how one would treat multiple hierarchies when a specific level of detail has been chosen. In addition, assigning a specific detail level to goals may pose a challenge and using the goal hierarchy by itself may not be enough. This may require the user to define the importance of individual goals with respect to the level of detail. This question and the question of the usefulness of multiple goal hierarchies would need to be investigated in practice using feedback during modeling of workflows with our industry partner Daimler.

5 Conclusion

In this paper we presented an approach and tool for modeling goal-oriented workflows and proposed a path for integrating business and technical perspectives of workflow modeling. The tool allows users to customize their view on the workflow by concealing details which are currently relevant for the user. In addition, we have attempted to maintain the ability of users to modify all visible aspects of the workflow, including elements which represent an abstraction of more complex underlying workflow elements.

Furthermore, we have presented approaches for extending this concept to extend the range of possible perspectives of the workflow to both include even more detail or even higher levels of abstraction as desired. This would include the use of goal patterns, goal concealment and potential activations.

Finally we have elaborated on potential problems in applying those concepts to GPMN-Edit, which will require further investigation with our industry partner Daimler to evaluate their usefulness in practice.

Acknowledgements: We would like to thank the DFG for supporting the technology transfer project Go4Flex.

Bibliography

[ABO+07] Adobe, BEA, Oracle, A. Endpoints, IBM, SAP. WS-BPEL Extension for People (BPEL4People), Version 1.0. edition, 2007. http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/BPEL4People_v1.pdf

[AH05] W. M. P. van der Aalst, A. H. M. ter Hofstede. YAWL: yet another workflow language. *Information Systems* 30(4):245–275, June 2005.

[BPJ+10] L. Braubach, A. Pokahr, K. Jander, W. Lamersdorf, B. Burmeister. Go4Flex: Goal-oriented Process Modelling. In al. (ed.), *Proc. 4th International Symposium on Intelligent Distributed Computing (IDC-2010)*. P. 7787. Springer-Verlag, 9 2010.

[BPM08] Object Management Group (OMG). Business Process Modeling Notation (BPMN) Specification. Version 1.1 edition, Feb. 2008.
http://www.bpmn.org/Documents/BPMN_1-1_Specification.pdf

[BPML05] L. Braubach, A. Pokahr, D. Moldt, W. Lamersdorf. Goal Representation for BDI Agent Systems. In *Proc. of (ProMAS 2004)*. Pp. 44–65. Springer, 2005.

[BRB07] R. Bobrik, M. Reichert, T. Bauer. View-Based Process Visualization. In Alonso et al. (eds.), *BPM*. Lecture Notes in Computer Science 4714, pp. 88–95. Springer, 2007.

[DDO08] R. M. Dijkman, M. Dumas, C. Ouyang. Semantics and analysis of business process models in BPMN. *Information & Software Technology* 50(12):1281–1294, 2008.

[Ecl] Eclipse Foundation. Eclipse Integrated Development Environment.
http://www.eclipse.org/downloads/

[PB09] A. Pokahr, L. Braubach. From a Research to an Industrial-Strength Agent Platform: Jadex V2. In Hans Robert Hansen (ed.), *Business Services: Konzepte, Technologien, Anwendungen - 9. Internationale Tagung Wirtschaftsinformatik (WI 2009)*. Pp. 769–778. sterreichische Computer Gesellschaft, 2 2009.

[RB07] G. Rimassa, B. Burmeister. Achieving Business Process Agility in Engineering Change Management with Agent Technology. In *WOA*. Pp. 1–7. 2007.

[RM06] J. Recker, J. Mendling. On the Translation between BPMN and BPEL. In Krogstie et al. (eds.), *Proceedings of the Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD'06)*. Pp. 521–532. Namur University Press, Namur, Belgium, EU, 2006.

[SN00] A.-W. Scheer, M. Nüttgens. ARIS Architecture and Reference Models for Business Process Management. In *Business Process Management*. Springer, 2000.

[SS06] A.-W. Scheer, K. Schneider. ARIS Architecture of Integrated Information Systems. In Bernus et al. (eds.), *Handbook on Architectures of Information Systems*. International Handbooks Information System, pp. 605–623. Springer Berlin Heidelberg, 2006.

[WSB07] Organization for the Advancement of Structured Information Standards (OASIS). Web Services Business Process Execution Language (WSPBEL) Specification. Version 2.0 edition, 2007.