Universität Stuttgart

SimTech
Cluster of Excellence

M. Sonntag K. Görlach D. Karastoyanova F. Leymann P. Malets D. Schumm

Views on Scientific Workflows

Stuttgart, July 2011

Institute of Architecture of Application Systems (IAAS)
University of Stuttgart,
Universitaetsstrasse 38
70569 Stuttgart, Germany
{sonntag, goerlach, karastoyanova, leymann, malets, schumm}@iaas.uni-stuttgart.de
http://www.iaas.uni-stuttgart.de

Abstract Workflows are becoming more and more important in e-Science due to the support they provide to scientists in computer simulations, experiments and calculations. Our experiences with workflows in this field and the literature show that scientific workflows consist of a large number of related information. This information is difficult to deal with in a single perspective and has changing importance to scientists in the different workflow lifecycle phases. In this paper we apply viewing techniques known from business process management to (service-based) scientific workflows to address these issues. We describe seven of the most relevant views and point out realization challenges. We argue that the selected views facilitate the handling of workflows to scientists and add further value to scientific workflow systems. An implementation of a subset of the views based on Web services and BPEL shows the feasibility of the approach. The presented work has the goal to increase additionally the acceptance of the workflow technology in e-Science.

Keywords Service compositions, simulation workflows, distributed simulations, BPEL, Web services.

Reference Sonntag, M., Görlach, K., Karastoyanova, D., Leymann, F., Malets, P., and Schumm, D. (2011) "Views on Scientific Workflows." In: Proceedings of the 10th International Conference on Perspectives in Business Informatics Research (BIR 2011), Riga, Latvia.

Stuttgart Research Centre for Simulation Technology (SRC SimTech)

SimTech – Cluster of Excellence
Pfaffenwaldring 7a
70569 Stuttgart
publications@simtech.uni-stuttgart.de
www.simtech.uni-stuttgart.de

Views on Scientific Workflows

Mirko Sonntag, Katharina Görlach, Dimka Karastoyanova, Frank Leymann, Polina Malets and David Schumm

Institute of Architecture of Application Systems, University of Stuttgart, Universitaetsstrasse 38, 70569 Stuttgart, Germany
{sonntag, goerlach, karastoyanova, leymann, malets, schumm}@iaas.uni-stuttgart.de

Abstract. Workflows are becoming more and more important in e-Science due to the support they provide to scientists in computer simulations, experiments and calculations. Our experiences with workflows in this field and the literature show that scientific workflows consist of a large number of related information. This information is difficult to deal with in a single perspective and has changing importance to scientists in the different workflow lifecycle phases. In this paper we apply viewing techniques known from business process management to (service-based) scientific workflows to address these issues. We describe seven of the most relevant views and point out realization challenges. We argue that the selected views facilitate the handling of workflows to scientists and add further value to scientific workflow systems. An implementation of a subset of the views based on Web services and BPEL shows the feasibility of the approach. The presented work has the goal to increase additionally the acceptance of the workflow technology in e-Science.

Keywords: Process views, BPEL, Web services, SOA, simulation workflows, scientific workflows.

1 Introduction

In the last years the application of workflows in scientific simulations and scientific computing has been experiencing an increased attention [1]. In this context workflows have been used to successfully implement, for instance, image processing tasks in physical astronomy [2], earthquake simulations in geology [3], or calculations related to biodiversity of species [4]. There are many reasons why workflows are interesting for natural or engineering scientists: (1) simulations often consist of manual steps that can be automated with the help of workflows; (2) former monolithic (legacy) scientific applications can be executed on multiple machines in a distributed manner; or (3) new simulations/calculations can be created in a graphical manner by modeling workflows, i.e. the programming effort is decreased. The main goal is to allow and facilitate scientists to concentrate on their core competencies and research topics instead of coping with IT issues.

Besides these technical improvements that workflows provide to scientists a user-friendly handling is a key concern in scientific workflow management (WFM). Workflows in e-Science possess many aspects that are interesting for scientists but hard or even impossible for a human to capture in a single perspective. Scientific workflows can consist of hundreds of activities (e.g. as described in [5]) with varying degree of importance for scientists. Scientists need to have a look at the data flow between activities to see the logical dependencies between tasks. In some cases it is required to see at a glance which tools where executed on which machines in what time. These and other information can be prepared and clarified with the help of viewing techniques for scientific workflow, which filter out unnecessary information.

While process views are a well-known technology in business WFM [6][7][8], a detailed investigation of their applicability to scientific workflows has not been carried out. Most scientific workflow management systems (SWFMS) are built from scratch and do not rely on the conventional workflow technology typical for business applications [9][10]. In the scope of the Stuttgart Research Center Simulation Technology (SRC SimTech[1]) we develop a SWFMS based on the traditional workflow technology in a SOA environment. In our work with different scientific institutes we have conducted case studies in which we have gathered requirements from scientific researchers and implemented scientific simulations with workflows and Web services (WSs) (e.g. [11]). In this paper we focus on process views that are useful to visualize the different perspectives of scientific workflows to scientists, i.e. the simulation or experiment itself and not the results of the experiment/simulation. For the description of the perspectives we build on previous work in the field of process view application scenarios [12] and viewing techniques [13]. In our former work we have developed a lifecycle definition of scientific workflows that reflects the iterative and adaptive development of scientific workflows [14] (see also Figure 1). Due to this different lifecycle it is needed that the existing viewing mechanisms and techniques are adapted to the needs of scientists and scientific applications. We selected seven views most relevant to the everyday work of scientists. These views mainly concern instance monitoring, process analysis, and abstraction. Views for process re-use have been identified important for scientific workflows but have already been proposed in former works on process view transformations [15][16][17]. To describe the views we make use of a workflow for the simulation of the ink diffusion in a glass of water as running example. We are convinced that the process views we present here add value to SWFMSs.

The paper contributes (1) an advancement of the current state-of-the-art of scientific WFM by exploiting viewing techniques that have been identified recently in the field of traditional process management; (2) extensions and refinements to existing process views that meet different requirements of scientific computing; and (3) a proof of the concept by the implementation of a subset of the proposed views in a SWFMS based on BPEL [18] and WSs.

The rest of the paper is structured as follows. Section 2 discusses related work in the field of process views and views in existing SWFMSs. Section 3 presents a collection of views on scientific workflows and provides details about the extensions that were necessary. Section 4 shows a prototypical implementation of a subset of the views of Section 3. Section 5 closes the paper with conclusions and an outlook.

---

[1] `http://www.simtech.uni-stuttgart.de`

2 Related Work

In business process management (BPM), viewing techniques are currently gaining momentum. The increasing complexity of business processes requires the use of advanced abstraction and visualization techniques. Viewing approaches for the omission and aggregation of tasks [19] as well as those related to analysis, monitoring and graphical display [12][13] are relevant to our work. Process monitoring using viewing techniques has been thoroughly investigated (e.g. in [20]). These monitoring views can also be applied to ease work in complex scientific workflows as we discuss in Section 3.3. These different views and the concepts behind them need extensions in order to fit the needs of scientists and scientific applications.

In [21] Petre argues that for different groups of people, different graphical notations, icons, shapes and so on needs to be provided to account for different understanding. While some icons used in business process automation frameworks might be universally applicable, in scientific workflows other shapes might be useful to ease understanding. We take a first step into this direction by proposing the use of custom icons for scientific computation services as discussed in Section 3.5.

Cohen-Boulaki et al. [22] used viewing techniques for different levels of granularity and abstraction to solve the provenance challenge. They demonstrated that such techniques are well applicable to support reasoning about all the intermediate and final data produced in the course of execution of scientific workflows.

Existing SWFMSs also make use of views. In e-BioFlow[2] it is possible to switch between control and data flow perspectives. The current status of the workflows is displayed in a table. Kepler[3] and Triana[4] allow the modeling of complex activities that hide more complex workflow logic from the users. Both make use of a monitoring view to display the runtime status of workflows. Taverna[5] also provides a view for the status of running workflows. Pegasus[6] contains a monitoring component for the analysis of past workflow runs.

3 Views on Scientific Workflows

Scientists and their applications impose new requirements on workflow systems, e.g. data-centricity, tool integration, hiding of technical details, clear arrangement of workflows when applied in the field of scientific research [9][10]. Most of these requirements can be met by existing process views from conventional workflow technology [13]. However, in our former work we have observed that the lifecycle phases modeling, execution, monitoring and adaptation of conventional workflows are alternating and continuously repeating in scientific workflows, i.e. scientific workflows are developed in a trial-and-error approach (see Figure 1) [14]. The reason

[2] `http://sourceforge.net/projects/e-bio-flow/`
[3] `https://kepler-project.org/`
[4] `http://www.trianacode.org/`
[5] `http://www.taverna.org.uk/`
[6] `http://pegasus.isi.edu/`

is that the direction of an experiment may not be predictable and hence an adaptation of running experiments is required. This and the fact that scientists play all roles participating in the workflow lifecycle motivate the need for an integrated tool that supports scientists in all lifecycle phases. A modeling tool for scientific workflows hence has to be able to also steer workflow execution, to monitor workflows and to adapt running workflows. Some of the process views therefore need extensions to be applicable for scientific workflows. The views are not automatically derived visualizations of aspects of a scientific workflow in another tool. They are part of the modeling tool and thus can be used to model scientific workflows. Another reason for extensions is that information about the computing infrastructure is of interest to scientists, e.g. properties of the employed servers.

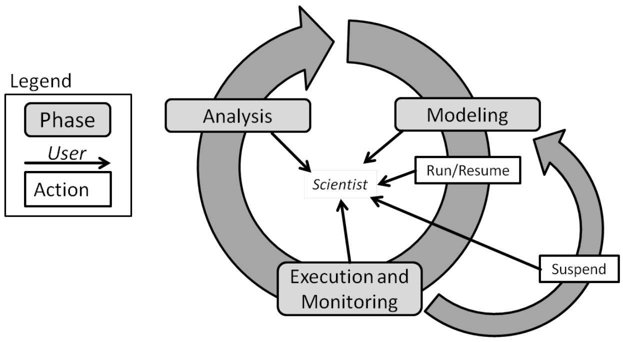

Fig. 1. Definition of the lifecycle of scientific workflows [14]

We have chosen a scientific workflow for the simulation of ink diffusion in a glass of water over a period of time as running example. Core of the simulation is Dune[7], a C++ toolbox to solve partial differential equations (PDEs) with the help of grid-based methods (e.g. finite elements method (FEM), finite volumes). Note that we (and the Dune community) understand the term “grid” (also called mesh) here as a graph of nodes and edges used for complex calculations and not as a computer infrastructure. We implemented this simulation with BPEL to orchestrate DUNE services; the DUNE services are WSs providing Dune modules for remote use in a network. For the purpose of legibility we created a BPMN [23] representation of this BPEL workflow with the most important steps only (see Figure 2).

The ink diffusion in water simulation consists of four main phases. Firstly, a new simulation instance in the Dune framework is created (steps 1-6). The input parameter file is unpacked; the parameters are inserted into the Dune/simulation source code which then gets compiled. Secondly, the raw grid is created that describes the glass of water as graph of nodes and edges (step 7). This grid is the discretization of the glass of water. Then, the grid is refined by multiplying the nodes and edges with the goal to gain much more detailed results (steps 8-9). Thirdly, the simulation is conducted on the refined grid (step 10). Multiple iterations are needed to simulate how the ink distributes in the water. Each loop step represents a simulation time step. Finally, the simulation is stopped and the simulation instance is closed (steps 11-13).

[7] Distributed and Unified Numerics Environment, `http://www.dune-project.org`

In the rest of the section we present selected views. All views are shown in the same way by giving a motivation, listing prerequisites, describing the approach itself including a figure and considering challenges.

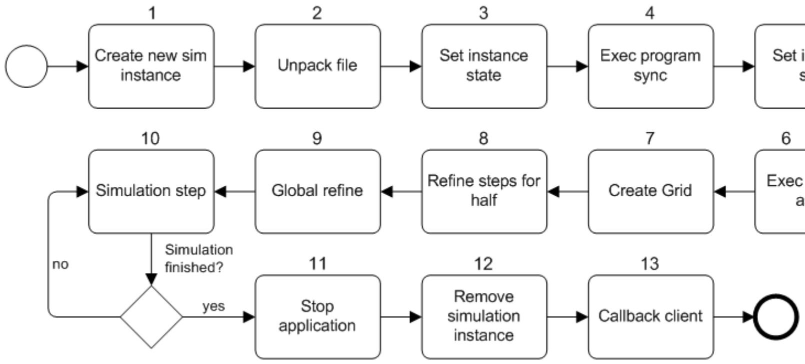

Fig. 2. BPMN diagram of the scientific workflow for simulation of ink diffusion in water. All activities denote service invocations. All other activities are omitted for the reason of legibility.

3.1 Aggregation of Complex Workflow Logic

Motivation. Scientific workflows often implement complex logic that can be divided in different functional parts. For example steps 7-9 in Figure 2 create and refine an FEM grid needed for the subsequent simulation steps. Such a self-contained logic/workflow fragment implements well-known behavior. It can comprise a huge number of activities and may have multiple entries and exits.

Figure 3a shows another example, a fragment for the robust allocation of a scientific service from a resource manager for a computing intensive task (e.g. a PDE solver service). If service allocation fails, the service request is retried a predefined number of times. The fragment implements useful behavior, but the workflow's legibility suffers from its complexity (especially for non-computer scientists). A view is needed to aggregate selected complex workflow behavior so that it looks like a single entity and the scientists can focus on the relevant experiment's logic.

Prerequisites. The subset of aggregated activities should be restricted to a connected set of activities. This prevents from ambiguity and cycles in the workflow graph when aggregating the activities.

Approach. Aggregation views in BPM are usually automatically derived by certain algorithms. In scientific WFM the scientist manually selects a number of activities in a workflow model that ought to be aggregated. This provides maximal flexibility for a customization that fits the needs of a scientist. A transformation step then automatically translates the workflow model into a (graphical) representation where the selected activities are replaced by a single activity that now stands for the complete behavior of the aggregate. Note that BPMN has a built-in mechanism to represent aggregates: collapsed activities. The aggregation view on the fragment "robust allocation of a service" is shown in Figure 3b. In contrast to outsourcing workflow logic with the help of sub-workflows the aggregation of logic effects only the modeling and not to the runtime of workflows. In particular it is possible to gain insight into the aggregated workflow logic in a straight-forward way because the logic is still part of the workflow model. The aggregation view represents logic as a black box where the implementation details can be accessed on demand. As opposed to this, the logic of sub-workflows is not visible in the parent/invoking workflow.

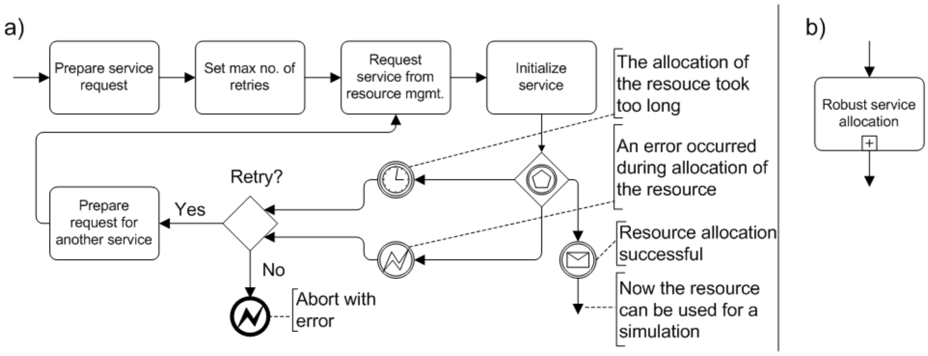

Fig. 3. Process fragment for robust service allocation (a) and the aggregation view on the same fragment (b).

Challenges. The main difficulty is the replacement of selected activities if the aggregate has multiple entries or exits. Another challenge is an intelligent visualization to enable expanding and collapsing of the aggregates.

3.2 Phases in Simulation Workflows

Motivation. In our work with scientists we have implemented several simulation use cases with WSs and workflows (e.g. [11]). Usually the simulations follow a simple pattern of phases: (1) the pre-processing comprises data and simulation preparation such as parameter specification, data import, FEM grid generation, or directory setup; (2) the actual simulation is often a loop of resource-demanding calculations, e.g. solving a PDE; (3) during the post-processing phase result data is visualized and the simulation environment is cleaned up (e.g. de-allocation of resources, deletion of intermediate files). Although a simulation consists in principle of these phases, a simulation workflow can be huge and complex. The phases consist of more than a simple sequence of service calls. Fault handling, robust service allocation (see Figure 2a) and usage, as well as transaction logic can convolute the workflow logic so that scientists can lose the overview. A view on the workflow that reflects the simulation phases is deemed useful.

Prerequisites. The phases in a simulation workflow should be detectable and in sequential order. Otherwise a simulation could conduct a step back, e.g. from simulation to pre-processing, contradicting the common understanding of the phases.

Approach. The view on the simulation phases can be created automatically or manually. Automatic generation is based on a mapping of activities on phases. A mapping could, e.g., be geared to the usage of specific services (e.g. an activity for the invocation of a visualization service belongs to the post-processing phase) or activity names (e.g. an activity "solve PDE" belongs to the simulation phase). For the manual creation of the view the scientist selects a number of activities and assigns them to a phase. Not all simulations have to be mapped on the three afore-mentioned phases. It may be required to introduce finer grained phases, e.g. post-processing for intermediate results. Therefore it should be possible for scientists to customize the simulation phases that the workflow modeling tool provides (e.g. reordering, renaming). After the mapping of the workflow logic on phases is complete (Figure 3a), a transformation step translates the workflow model with the help of an aggregation view into a workflow that only consists of the resulting phases as aggregates (Figure 3b). Then, the phases are illustrated as arrows that prescribe their sequential ordering (Figure 3c). The implementation details of the simulation are hidden to the scientist but can be requested on demand (similar to the aggregation of complex workflow logic in Section 3.1).

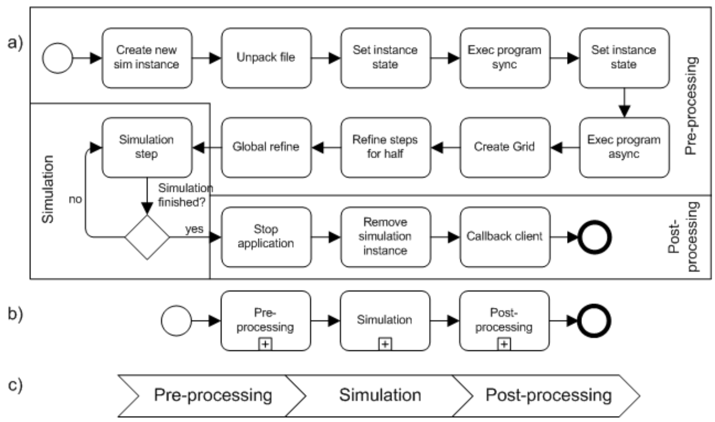

Fig. 4. Aggregation view on a simulation workflow to show only the high-level simulation phases. The simulation workflow is mapped on simulation phases (a). An aggregation view on the activities of the phases reduces the complexity of the model (b). The aggregates are then visualized as arrows (c).

Challenges. The main challenge is to find an appropriate visualization of the simulation phases if they are interleaved to a certain degree.

3.3 Status of a Scientific Workflow

Motivation. For the trial-and-error scientific workflow development it is needed to enrich workflow models in the modeling tool with information about the execution status of workflow instances. Scientists can then monitor the progress of their simulations while still modeling them—a requirement completely new to business workflows where modeling and monitoring are accomplished by different tools.

Prerequisites. In order to visualize the runtime status of workflow instances in a modeling tool the tool needs access to instance data. This can be achieved e.g. by querying the audit trail or by listening to execution events that are published by the employed workflow engine. Note that the used technique strongly depends on the API the workflow engine offers. Another prerequisite is that the workflow model of the instance a scientist wants to monitor is available in the modeling tool. Otherwise the instance state cannot be monitored.

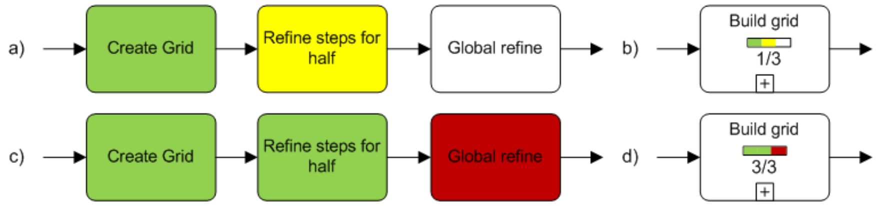

Fig. 5. Workflow model with annotated runtime information. Completed activities are green (a, c), running activities yellow (a), and failed ones red (c). Aggregated activities must have a visualization that reflects the state of their contained activities, e.g. by a multi-colored status bar (b, d).

Approach. The current states of the instantiated workflow model elements are attached to the workflow model as well as other information that is relevant for the visualization of a workflow instance (e.g. number of completed iteration in loops). The illustration of the workflow model is changed according to the augmented runtime information. Instance states are mapped to colors; these colors are used to display activities (see Figure 5a and 5b). In loops it is useful to visualize the number of passed loop iterations. Special attention has to be paid to aggregations. The status of an aggregation depends on the states of all contained activities. It is thus required to calculate the aggregation state and/or to filter information not needed to illustrate the instance state (e.g. state changes of activities in an aggregation). The state of an aggregation could be displayed with the help of a status bar (see Figure 5b and 5d).

Challenges. The main difficulty is how to bridge the separation between workflow models and instances. The question is how to visualize different instances in a modeling tool that is inherently unaware of workflow instances. The tool needs an extension to allow users to select the workflow instance to monitor. Selection of instances should be based on given or generated meta-data or the starting time instead of instance IDs. Nevertheless, the instance IDs of the currently monitored instances are important to correlate runtime information to the monitored instances.

3.4 Data Flow Visualization

Motivation. Current languages in conventional workflow technology are control flow-oriented, e.g. BPEL, BPMN. In contrast, existing SWFMSs usually follow a data flow-oriented modeling paradigm [1][24]. The reason is that scientists think in a data-oriented way. This fact creates the need for an explicit visualization of data aspects in control flow-oriented workflow languages when being used for scientific applications.

Prerequisites. The size of data items can be calculated not until runtime.

Approach. The workflow languages used in the conventional workflow technology usually provide means to implicitly specify the flow of data by means of variables. Based on this implicit data flow a transformation step can calculate the explicit data dependencies between two activities or between an activity and a data source or sink (e.g. variables, databases, files) (Figure 6). Two activities A and B have a data dependency if activity A produces data that is used as input for activity B or vice versa. After the transformation step the data links have to be visualized. We recommend views that exclusively illustrate control dependencies, exclusively data dependencies, or both combined. That way the scientist can select the view that best displays the detail he is interested in (e.g. the control flow view is best for displaying loops, the data flow view can help optimize a workflow by task parallelization). Another aspect of a data flow view is to visualize the amount of data that is transferred within the workflow and between used services. This information is useful to reduce expensive data transfer operations. The amount of data can be represented by the data link color or width (Figure 6). The data size has to be collected at runtime. The generated data links of the workflow model have to be augmented with this information and the illustration of the data links can be adapted accordingly.

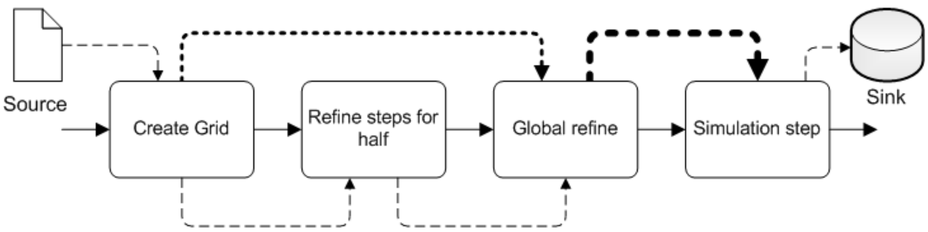

Fig. 6. View for explicit data flow between activities and sources/sinks by dotted arrows. The arrow width is an indicator for the data size.

Challenges. Determining the size of data is not straight-forward because many data objects exist only in the memory of the workflow system. If data is passed by reference it has to be specified if the reference itself or the referenced data counts for the data size. Techniques have to be developed to determine the size of data that is external to the workflow engine (e.g. files).

3.5 Custom Icon for Service Invocation

Motivation. One of the most important requirements on sWfMSs is usability since the majority of users of sWfMSs are no computer scientists. A scientist needs the support of easy-to-use tools and self-explaining visualizations in these tools. Scientific workflows can encompass a lot of service invocations and other activities. Existing workflow tools (e.g. the Eclipse BPEL Designer) or graphical workflow languages (e.g. BPMN) foresee the visualization of activities on per activity-type-basis, i.e. all activities of a certain type have the same icon. Activities are usually customized by their names. This makes it difficult and time-consuming to orient oneself in a forest of nodes, edges, labels and recurring icons. The idea is to allow customization of activity icons in the workflow modeling tool to facilitate orientation in a workflow graph by different visual symbols.

Prerequisites. The customization has to be done with mnemonic and self-explanatory icons to be of real value to scientists.

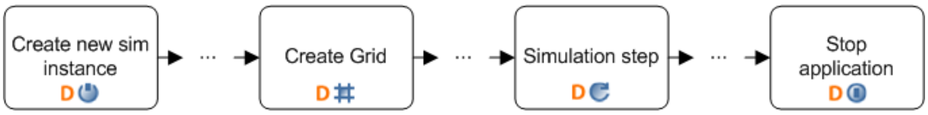

Fig. 7. Customized icons for service invocations of the Dune-library.

Approach. Customization of activity icons can be done manually or automatically. In the manual case the scientist simply selects an activity and specifies a new icon. The modeling tool should provide icons for standard cases (e.g. different sensors, databases, visualization applications such as Gnuplot). Additionally, it should be possible to import and select custom icons. In the automatic case a mapping of activities on icons is needed. This can be realized by an external file that maps URLs of icons on activities by certain criteria (e.g. the invoked service operation, the location of the invoked operation, the service binding, the activity's name). Another option is a WSDL extension where the URL of an icon is attached to a WSDL operation. In a workflow modeling tool the icon can then be loaded and used for the activity that calls this operation. Figure 7 shows the service calls of Dune-library in our "ink diffusion" example with customized icons.

Challenges. The WSDL extension for operation icons entails additional work to service providers. They have to create an icon and publish it together with the service. The workflow modeling tool has to be able to deal with the icon's format. The icon's size should have an upper boundary because it has to be downloaded during workflow modeling. Problems arise when the computer that is used to model the workflows does not have an internet connection or the service is temporarily down. In this case the standard icon could be taken until the predestined icon is available.

3.6 Performance Analysis

Motivation. In the analysis and optimization of scientific workflows, time plays a crucial role (e.g. the time particular resources are occupied, the time needed to transfer large amounts of data, the runtime of (parts of) the workflow execution). The resource consumption is another determinant of performance. The idea is to build a view that makes the performance perspective of a workflow instance visible to a scientist in the workflow design environment. The impact of concrete machines to the workflow throughput is not considered in business process views where the servers are transparent to the workflow.

Prerequisites. Firstly, the “effective” (executed) workflow model has to be constructed that might differ from the originally designed workflow model due to adaptation steps (e.g. automatic insertion of data transfer activities). For this construction we can benefit from process mining tools and techniques (e.g. the ProM framework [25]). Secondly, the effective workflow model has to be imported into the modeling tool and augmented with runtime information about the time and resource consumption aspects of the workflow execution (e.g. the duration of activities/service invocations, the duration of data transfers, consumed processing units (e.g., measured in FLOPS)). This information needs to the annotated to the activities contained in the effective model.

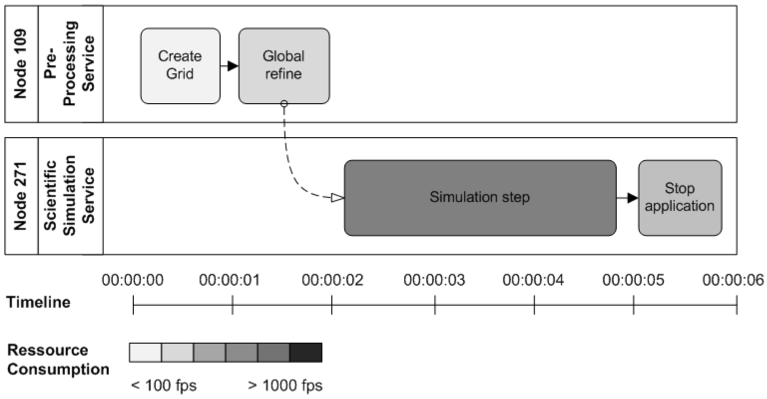

Fig. 8. Performance analysis is supported through stretching long-running activities and arranging them along a timeline. The degree of resource consumption is visualized by using different levels of grey for activity shapes.

Approach. As a first step, we unfold loops to clearly arrange the number of iterations, to identify problematic branches and differences in the execution of iterations. As a result, we obtain a complex workflow model containing possibly hundreds or thousands of activities. As a consequence, we need to apply abstraction techniques to simplify this model. This encompasses removing dead and unreachable paths which do not have an impact on the performance. Further, we can omit all activities which are in terms of duration or resource consumption below a certain threshold that needs to be specified on a per case basis. The resulting model is then layouted and displayed. We propose to arrange the activities along a timeline, and stretch their shapes in order to express the time dimension. To distinguish between the different computing nodes and the used scientific services it is meaningful to apply pools and swim lanes. The consumption of computing resources can be visualized by coloring the activities (see Figure 8).

Challenges. For the measurement of resource consumption in used servers these servers have to be instrumented accordingly. Another challenge is the mapping of the performance data to activities of the effective model. Finally, creating a well-readable layout is not trivial for long-running, complex scientific workflows that orchestrate services on lots of nodes.

3.7 Access to Runtime Information of Used Services

Motivation. Besides the workflow instance status information during runtime (see Section 3.3) the status of used services and resources as well as an insight into used and produced data is also of interest. This makes the simulation infrastructure visible and gives scientists the full control over their experiments and simulations. In contrast to this, in BPM scenarios the used machines are invisible.

Prerequisites. The used services need to provide operations that allow querying information about the machines they are installed on as well as information about used and produced data.

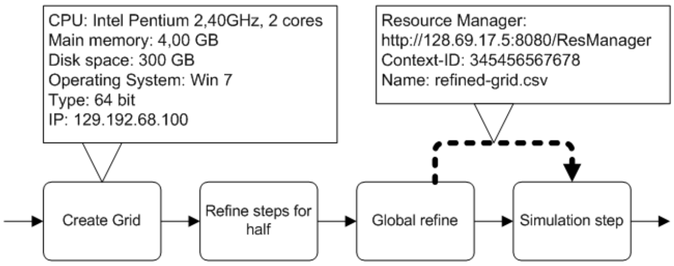

Fig. 9. View on service and data information.

Approach. This view collects and visualizes simulation instance data beyond the state of workflow activities. This includes information about the machines scientific services are running on (e.g. CPU, storage), the content of variables (or data passed via data links) in the workflow and data external to the workflow (e.g. files, databases) (see Figure 9). It is important to mention that this view does not augment the workflow model with the collected information. In fact the data should be accessed on demand by the scientist because it can encompass huge amounts of data (e.g. result files of hundreds of mega bytes). The data is distributed in the scientific workflow infrastructure and only loaded to the workflow modeling tool on explicit request. After that it is visualized in a user-friendly way, e.g. by a pop-up.

Challenges. Many different data sources (e.g. resource manager, audit trails, servers) and representations (e.g. files, databases) have to be integrated, processed and visualized. Since the data is spread over the infrastructure it may be difficult to find data related to a particular simulation run. Sophisticated correlation mechanisms have to be used (e.g. a global context ID for each simulation run that is attached to all messages and hence known to all participating services).

4 Implementation

We are developing a SWFMS based on the WS technology and BPEL. The idea is to introduce the numerous advantages of conventional workflows to e-Science [24], e.g. robust workflow execution, fault handling and transactions on the workflow level, asynchronous messaging. But the technology needs thorough extensions to become interesting for scientists, e.g. support for trial-and-error workflow development (see lifecycle in Figure 1), integration of stream data and sensors, or data as first class citizen. Our current prototype of the SWFMS implements some of these extensions and a subset of the views presented in this paper. The latter is subject of this section. The prototype is based on the Apache Orchestration Director Engine (ODE) as BPEL engine, the Eclipse BPEL Designer as modeling tool and Apache Active MQ as message queuing system. A demonstration of the prototype is available in [26].

We extended the visualization of `invoke` activities to implement the view *custom icon for service invocation*. If an invoke activity gets configured with a partner link and an operation of a WSDL, then the modeling tool looks up whether an icon is registered with this WSDL operation. If so, the icon is fetched and used for this invoke activity (Figure 10a, activity “Global Refine”). Otherwise the standard icon of the modeling tool is taken (Figure 10a, e.g. activity “Create Grid”). The custom icons are attached as URI to WSDL operations by the new attribute `icon` (see Listing 1).

```
<wsdl:definitions
    xmlns:icon="http://iaas.uni-stuttgart.de/wsdlIconExtension">
  <operation name="thisOp"
      icon:icon="http://exampleService/thisOp/icon.gif">...
  </operation>
</wsdl:definitions>
```

Listing 1. WSDL extension for custom operation icons.

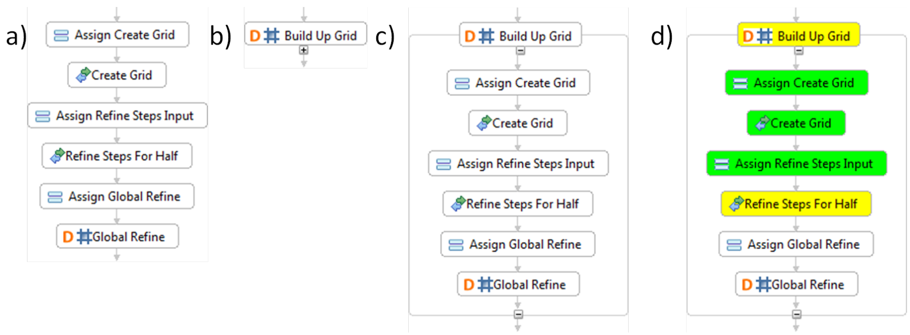

Fig. 10. Eclipse BPEL Designer extension for custom icon for service invocation (a), aggregation collapsed (b) and expanded (c), and monitoring (d).

The *aggregation* view is in parts already supported by the Eclipse BPEL Designer: structured activities can be collapsed so that they appear as a single basic activity. We extended this mechanism so that a set of consecutive activities can be marked by a user and aggregated explicitly via the context menu. The aggregated activities are put into a collapsed sequence (Figure 10b). The sequence activity inherits the icon of the last invoke activity of the aggregate. The user can expand the activity to gain insight into the aggregate’s logic (Figure 10c). Vice versa it is also possible to disaggregate the activities via a context menu command.

Finally, the *status of a scientific workflow* view is realized as follows. The BPEL Designer is extended so that scientists can start the opened workflow from within the tool. The workflow model is then deployed on an engine and a new instance is started. The extended engine publishes execution events over a topic. The modeling tool subscribes to this topic, receives these events, correlates them to the opened model and colors activities according to the received activity instance state (Figure 10d).

5 Conclusions

Scientific workflows possess properties that make it difficult to handle them during modeling, monitoring and analysis (e.g. size of the workflows, data and data flow as first class citizen, used resources). In this paper, we have shown how process viewing techniques can be used to ease dealing with workflows to scientists and to reveal potentials for the optimization of workflow execution. Process views are well elaborated on in BPM. In the context of scientific workflows, however, an investigation of their applicability was missing. We have filled this gap by presenting the concept of seven views relevant to scientists and scientific workflows. We are convinced that the implementation of these views adds value to any SWFMSs. As a proof of concept we have implemented a subset of the proposed views. Some of described concepts could also be generalized and provided back to BPM (e.g. custom icons, performance analysis).

In BPM, efforts were made to find icons for recurring actions in business processes [27]. With the implementation of the view *custom icon for service invocations* and the associated WSDL extension we have developed the technical basis for a similar approach in scientific WFM. Further research will show whether frequently recurring actions can be identified and standard icons can be found. In future we want to implement the missing views and find more views, e.g. for stream data or security.

Acknowledgements. The authors 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. Taylor, I., Deelman, E., Gannon, D., Shields, M.: Workflows for e-Science: Scientific Workflows for Grids. Springer (2007)
2. Berriman, G.B., Deelman, E., Good, J. et al.: Generating Complex Astronomy Workflows. In: Taylor, I. et al.: Workflows for e-Science. Springer (2007)
3. Maechling, P., Deelman, E., Zhao, L. et al.: SCEC CyberShake Workflows—Automating Probabilistic Seismic Hazard Analysis Calculations. In: Taylor, I. et al.: Workflows for e-Science. Springer (2007)
4. Jones, A.: Workflow and Biodiversity e-Science. In: Taylor, I. et al.: Workflows for e-Science. Springer (2007)
5. Deelman, E., Mehta, G., Singh, G. et al.: Pegasus: Mapping Large-Scale Workflows to Distributed Resources. In: Taylor, I. et al.: Workflows for e-Science. Springer (2007)
6. Eshuis, R., Grefen, P.: Constructing Customized Process Views. In: Data & Knowledge Engineering, 64(2), pp.419-438. Elsevier (2008)

7. Zhao, X., Liu, C., Sadiq, W. et al.: Implementing Process Views in the Web Service Environment. In: World Wide Web Journal, 14(1), pp. 27-52. Springer Publisher (2011)
8. Polyvyanyy, A., Smirnov, S., Weske, M.. Business Process Model Abstraction. In: International Handbook on Business Process Management. Springer (2009).
9. Sonntag, M., Karastoyanova, D., Leymann, F.: The Missing Features Of Workflow Systems For Scientific Computations. In: Proc. of the 3rd Grid Workflow Workshop (GWW), GI-Edition LNI, Volume P-160 (2010)
10. Sonntag, M., Karastoyanova, D., Deelman, E.: Bridging The Gap Between Business And Scientific Workflows. In: Proc. of the IEEE 6th Int. Conf. on e-Science. IEEE Press (2010)
11. Sonntag, M., Hotta, S., Karastoyanova, D., Molnar, D., Schmauder, S.: Workflow-Based Distributed Environment For Legacy Simulation Applications. In: Proc. of the 6th Int. Conf. on Software and Data Technologies (ICSOFT 2011), Poster Presentation, 2011
12. Schumm, D., Anstett, T., Leymann, F., Schleicher, D.: Applicability of Process Viewing Patterns in Business Process Management. In: Proc. of the Int. Workshop on Models and Model-driven Methods for Service Engineering (3M4SE 2010). IEEE Press (2010)
13. Schumm, D., Leymann, F., Streule, A.: Process Viewing Patterns. Proc. of the 14th IEEE International EDOC Conference (EDOC 2010), IEEE Computer Society Press. (2010)
14. Sonntag, M., Karastoyanova, D.: Next Generation Interactive Scientific Experimenting Based On The Workflow Technology. In: Proc. of the 21st IASTED Int. Conf. on Modelling and Simulation. Acta Press (2010)
15. Goderis, A., Sattler, U., Lord, P., Goble, C.: Seven Bottlenecks to Workflow Reuse and Repurposing. In: Proc. of the 4th Int. Semantic Web Conference. Springer (2005)
16. Avrilionis, D., Cunin, P.Y., Fernström, C.: OPSIS: A View Mechanism for Software Processes Which Supports Their Evolution And Reuse. In: Proc. of the 18th Int. Conf. on Software Engineering (ICSE). IEEE Computer Society (1996)
17. Schumm, D., Leymann, F., Streule, A.: Process Views to Support Compliance Management in Business Processes. In: Proc. of the 11th Int. Conf. on Electronic Commerce and Web Technologies (EC-Web). Springer (2010)
18. Organization for the Advancement of Structured Information Standards (OASIS): Business Process Execution Language 2.0 (BPEL). (2007)
19. Bobrik, R., Reichert, M., Bauer, T.: View-Based Process Visualization. In: Proc. of the 5th Int. Conf. on Business Process Management (BPM'07). Springer-Verlag (2007)
20. Reichert, M., Bassil, S., Bobrik, R., Bauer, T.: The Proviado Access Control Model for Business Process Monitoring Components. In: Enterprise Modelling and Information Systems—An International Journal, 5(3), pp. 64-88, ISSN 1860-6059 (2010)
21. Petre, M.: Why Looking Isn't Always Seeing: Readership Skills and Graphical Programming. In: Communications of the ACM, 38(6). (1995)
22. Cohen-Boulakia, S., Biton, O., Cohen, S., Davidson, S.B.: Addressing the Provenance Challenge using ZOOM. In: Concurrency and Computation: Practice and Experience, volume 20 (2008)
23. Object Management Group (OMG): Business Process Model and Notation (BPMN). OMG Available Specification, Version 2.0, Beta 1. (2009)
24. Goerlach, K., Sonntag, M., Karastoyanova, D. et al.: Conventional Workflow Technology for Scientific Simulation. In: Yang, X., Wang, L., Jie, W. (eds.) Guide to e-Science. Springer-Verlag (2011)
25. Process Mining Group. ProM (2011) `http://prom.win.tue.nl/tools/prom/`
26. Sonntag, M.: Views on scientific workflows. Prototype demo (2011) `http://www.iaas.uni-stuttgart.de/institut/mitarbeiter/sonntag/indexE.php#viewsVideo`
27. Mendling, J., Recker, J., Reijers, H.A.: On The Usage Of Labels And Icons In Business Process Modeling. In: Int. Journal of Information System Modeling and Design (IJISMD), 1(2). IGI-Global (2009)