A Novel Framework for Monitoring and Analyzing Quality of Data in Simulation Workflows

Michael Reiter[1], Uwe Breitenbücher[1], Schahram Dustdar[2], Dimka Karastoyanova[1], Frank Leymann[1], Hong-Linh Truong[2]

[1]Institute of Architecture of Application Systems (IAAS), University of Stuttgart
{reiter, breitenbuecher, karastoyanova, leymann}@iaas.uni-stuttgart.de

[2]Distributed Systems Group, Vienna University of Technology
{dustdar,truong}@infosys.tuwien.ac.at

***Abstract*—In recent years scientific workflows have been used for conducting data-intensive and long running simulations. Such simulation workflows have processed and produced different types of data whose quality has a strong influence on the final outcome of simulations. Therefore being able to monitor and analyze quality of this data during workflow execution is of paramount importance, as detection of quality problems will enable us to control the execution of simulations efficiently. Unfortunately, existing scientific workflow execution systems do not support the monitoring and analysis of quality of data for multi-scale or multi-domain simulations. In this paper, we examine how quality of data can be comprehensively measured within workflows and how the measured quality can be used to control and adapt running workflows. We present a quality of data measurement process and describe a quality of data monitoring and analysis framework that integrates this measurement process into a workflow management system.**

Key words: workflow management system; e-science; simulation workflow; quality of data; monitoring and analysis

I. I. INTRODUCTION

Workflows are compositions of tasks by means of data or causal dependencies that are carried out on a computer by using a workflow management system (WfMS) [1]. In recent years, the workflow technology has been widely applied to the scientific domain, and scientific workflows for e-science are well covered in [2].

Simulations, as a subset of scientific applications, are compositions of complex calculations and data management tasks. Therefore, workflows have been widely used in simulations to simplify the execution and to reduce errors made by humans. For example, workflows have been used in finite element method (FEM)-based simulations [3] [4]. Several scientific WfMSs have already been created to carry out scientific and simulation workflows, such as Askalon [5], Kepler [6], Pegasus [7], Taverna [8], Triana [9], and Trident [10]. However, most WfMSs do not measure, monitor and analyze quality of data (input, intermediate result, and output) associated with workflows in a comprehensive way to support the control of the execution of complex scientific workflows based on monitored quality of data (QoD) at runtime. There is a lack of generic methods to measure QoD within the invoked applications or workflow activities. The reason for this is that in the scientific area the usage of generic data quality concepts is at the beginning [11]. That is surprising, since unacceptable QoD in long-running scientific applications leads to huge waste of computing resources and human effort without reaching any simulation goal. In order to obtain exact and reliable simulation results, the quality of all processed data must be guaranteed to an acceptable degree.

In the business domain different concepts for data quality are established, such as methods for measuring quality metrics [12]. Stimulated by these methods, we investigate and adopt them for simulation workflows which typically have complex data dependencies [4]. In this paper, we first present a novel QoD measurement process. We have identified fundamental concepts to implement this process with the aim to control and to adapt the workflow execution. We characterize in general how simulation workflows can be driven by QoD. Second, we present a novel QoD monitoring and analysis framework that realizes our QoD measurement process and integrates them into WfMSs. We present different mechanisms which can be used to evaluate QoD metrics for simulation workflows and describe the usefulness of our framework through a real-world FEM-based simulation. To our best knowledge, this is the first effort on supporting the monitoring and analysis of QoD metrics for scientific workflows in a comprehensive way, as existing scientific WfMSs focus only on time-based performance monitoring and analysis.

The rest of this paper is organized as follows: Section II describes the background and related work. Section III presents our QoD measurement process. In section IV we present our QoD monitoring and analysis framework. Experiments demonstrating our concepts and framework are shown in section V. We conclude the paper and outline our future work in section VI.

II. BACKGROUND AND RELATED WORK

Several scientific WfMSs have been developed, such as Askalon [5], Kepler [6], Pegasus [7], Taverna [8], Triana [9], and Trident [10]. Most of them include a special component for or integrate with a tool for monitoring and analysis of workflows. Typically, such a component or tool can provide detailed information about execution status and performance of workflows at different levels, such as activities and workflow regions. However, only Kepler and Taverna support the monitoring and analysis of QoD metrics, activities, or workflow regions, but in a quite limited way. Na'im et al. [13] described an approach based on Kepler to define data quality threshold values as well as to monitor and visualize QoD findings during runtime. However, there is no detail technique of how to measure and analyze QoD metrics. Missier et al. [14] presented on top of Taverna the Qurator workbench that can use given QoD findings to control scientific workflows. Qurator assumes that quality evidence is presented so that quality assertions can be made but Qurator does not measure QoD metrics.

Many methods exist in scientific domains to measure QoD in a specific context. Most of them depend on a specific application and cannot be used in a general-purpose simulation WfMS. Gray et al. [15] calls for the control of QoD in simulations and for the development of QoD-aware applications. They refer in particular to FEM-based simulations – without making any concrete implementation suggestion. In this paper, we include different methods to evaluate QoD and we make evaluated QoD metrics available for other related applications.

Batini and Scannapieco [12] consider that data quality consists of six main dimensions: accuracy, completeness, currency, timeliness, volatility, and consistency. Those dimensions can be used in business as well as in the scientific applications. In practice, the term data quality is used with several meanings as well as weights in both domains and must be regarded specifically respective to the application domain. Nevertheless, data quality can be measured and expressed in general as a value or a set of values [12], for example by using metrics defining a concrete algorithm for the determining of the data quality. In this paper, we are focusing on analyzing and evaluating QoD metrics within simulation workflows that has not been considered before, by modifying and extending concepts in [12] to separate the measurement from other components.

As has already been pointed out, the term data quality is used in many different domains and each domain has its own interpretation of its meaning and appliance. For example, in many scenarios the completeness of data is associated directly with the quality of the data regarding completeness (e.g. [12]). In other domains the completeness has maybe only a characteristic meaning and must be explored in a special context to analyze the effective data quality, i.e. a single value describing the completeness of data can lead to different data quality values – each value interpreted in and dependent on a special context. Because of that we can interpret a high value of completeness as a bad quality, when our algorithm prefers a low level of completeness to work best to get a result very fast. Therefore, we should not mix the analysis and evaluation of data characteristics with that of data goodness, but separate them in different phases of the same process. This example shows a big problem every domain has to deal with: The ambiguity of the term – what is quality of data? The lack of clarity leads to misunderstandings, especially when different domains with different views have to work together. To solve these problems, we present a well-defined process to measure QoD in a consistent manner. First, we identify building blocks for data quality driven simulation workflows in a more abstract way. After that, we are focusing on implementation based on the so-called conventional workflow architecture.

III. QUALITY OF DATA MEASUREMENT PROCESS FOR SIMULATION WORKFLOWS

A Quality of Data Measurement Process

In our view, QoD measurement is not a monolithically process that one can determine in a single step whether data is good or bad. We must distinguish data characteristics, e.g., the completeness of data, from data goodness, e.g., whether a value of the completeness is good enough for a solver to continue to process the data. To this end, our QoD Measurement Process (QoDMP) has three important principles that are described in the following:

Decoupling data characteristics and the data goodness: Data characteristics have no judgmental value. Without considering them in a specific context (e.g., in a particular simulation purpose), they are neither good nor bad and describe only characteristic properties. In our QoDMP (see Figure 1), data characteristics are represented by metrics measured by our measurement and analysis process. Data goodness is the result of the interpretation of measured characteristics in a special simulation context. This allows the evaluation of certain characteristic properties depending on the context to different QoD values. For example, an input data can have a high QoD for the processing by Algorithm 1, but a low QoD for the processing by Algorithm 2, although there was only one measured characteristic (e.g. accuracy) for the input data. In this case, the evaluation contexts are associated with Algorithm 1 or Algorithm 2. In principle, there are strong dependencies between measured characteristics of data, evaluation context and the final QoD.

As a result of such dependencies, we divide QoDMP into two phases. The first phase analyzes the data for certain characteristics, resulting in metrics (without taking any context into consideration). This phase is called "Analysis-Phase" and returns a set of metrics that describes the characteristics of the data being analyzed. In contrast to Missier et al. [14] we do not require a partial order on this set. The second phase evaluates the results of the Analysis-Phase by considering the analysis results (characteristics) in a special context. This is done by interpretations in the "Evaluation-Phase" and results in the goodness of data. In this phase, we require, as well as Missier et al. [14], a partial order on this set. Both phases

together determine the final QoD. Hence, the data goodness value represents the QoD with respect to a context and specific metrics. Figure 1 shows these phases.

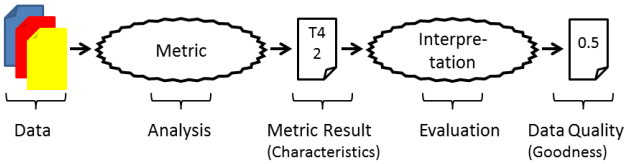

Figure 1: Two-step process to determine QoD.

Metric analysis and interpretation processes: Metric analysis and interpretation processes are in principle implemented via algorithms. A metric measurement process will return metrics as input for possible interpretation processes. For example, a metric measuring the completeness of a data set can have a value between 0.0 (incomplete) and 1.0 (complete). A simple interpretation gets this value as input and evaluates whether the data can be used as input for a certain algorithm or not by simply checking if the threshold of 0.5 is exceeded. Therefore, the interpretation context is implemented implicitly and directly into the interpretation algorithm. There could be different interpretation contexts. For example, it is possible to rate high completeness specifically for different types of use or for different algorithms by different interpretations of the independent characteristics. Of course there are many use cases where an explicit interpretation of measured data characteristics to QoD is not needed. For example, often it is sufficient to take the value of completeness (assume a value between 0.0 and 1.0) directly as QoD without interpreting this value explicitly in the Evaluation-Phase (In this case, we could consider there is an implicitly interpretation using the identity function as the interpretation algorithm).

Subjective and objective determination: Metric analyses and interpretations can also be performed manually by scientists. The following example describes the need to support manual analysis and interpretation: A simulation workflow iterates a calculation a million times. During this calculation several characteristics of the involved data are measured (Analysis-Phase). This information is aggregated and transmitted to the mobile phone of the responsible scientist, who can evaluate the data goodness (Evaluation-Phase), and pass the result (maybe a "Good" or a "Bad") via user interface to the WfMS which decides, based on the result, whether to continue the execution or not. Hence, the concept allows objective metric analyses and interpretations executed by a machine as well as subjective metric analyses and interpretations performed by a human being. As soon as one phase is executed by a human being, the whole QoD determination is influenced by subjective values. We call this subjectively determined QoD. Otherwise, if no human being is involved and the complete calculation is done by a machine, we call this objectively determined QoD.

So in general, we distinguish between the evaluation done by software and the evaluation done by human beings. The objective evaluation is executed automatically by software-based algorithms. The subjective evaluation is performed manually by a human activity and is based on expertise, instructions, regulations, and rules. If QoD is determined objectively any further QoD evaluation with the same metric analyses and interpretations, the same data set, and the same context will lead to the same QoD, as the evaluation is clearly reproducible. In the subjective case, QoD results can differ for each evaluation, even with the same conditions as above, because human beings not always have the same knowledge, or think, feel and act in the same way. Solving this non-reproducible problem is not our objective. Instead we focus on the integration of software-based and human-based determination into one single integrated QoD evaluation system.

B Fundamental concepts for QoD driven simulation workflows

To develop QoD driven simulation workflows, we have to identify the most important building blocks that a suitable scientific workflow environment must support.

Data handling: A complex simulation workflow uses different kinds of data represented in different formats and stored in different sources. Their QoD should be determined, as long as they have an influence on the simulation result, performance, and consumed resources of workflows. For example, the final result of a FEM-based simulation depends on the quality of material parameters (input data) and of matrix coming from solving differential equations (intermediate results). Because of the dependencies, the QoD of these types of data must be examined together to guarantee simulation results with an optimum quality.

QoD dimensions: Different dimensions of QoD must be supported in simulation workflows. However, concrete QoD dimensions are dependent on specific simulations or domains. QoD dimensions can be domain-independent or domain-specific. Thus generic evaluation techniques as well as custom techniques should be supported.

QoD measurement: As mentioned before, our QoDMP is divided into an Analysis-Phase and an Evaluation-Phase. In the Analysis-Phase specific characteristics will be determined. At the Evaluation-Phase those characteristics will be interpreted to determine the goodness of data, i.e. the final QoD. Given a QoD metric, in many cases, more than one measurement methods exist for the metric. Therefore, one must select the suitable method. Furthermore, we need to associate determined QoD metrics with additional meta information for provenance and analysis reasons, such as, for which workflow activity the measurement is taken, the types of data to be determined, the measurement technique (e.g. to get information about the precision of measurements), and the scale on which results are reported (e.g. 0.0 is bad and 1.0 is good). The QoD measurement can be done automatically by software or manually by human. The reason is that for many scientific data, software tools cannot determine QoD but human expert can. Thus, both objective and subjective QoD metrics will exist.

Aggregation of metrics and interpretations: To analyze the QoD we distinguish between basic metrics or

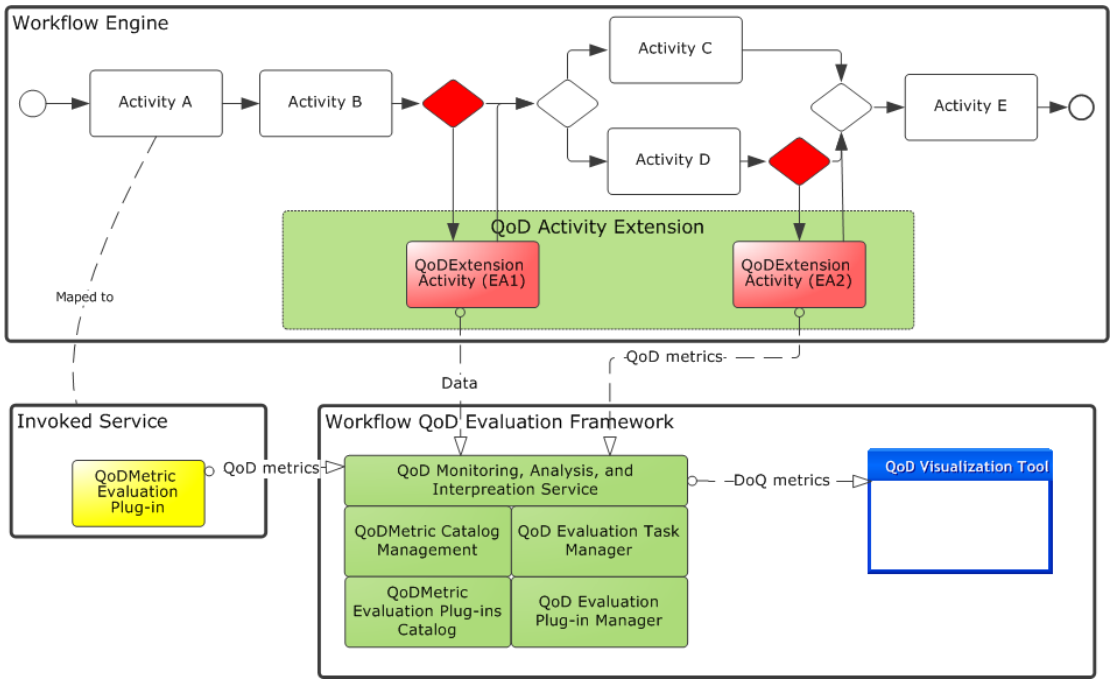

Figure 2: Architecture of QoD Monitoring and Analysis Framework

interpretations and aggregated metrics or interpretations. Basic metrics or interpretations use one measurement or evaluation method that cannot be spitted meaningful. An aggregated metric consolidates several basic or aggregated metrics. The same also applies for interpretation. Hence, an aggregated metric or interpretation can be established via aggregation operators/algorithms. We support basic metrics and interpretations as well as the aggregation of metrics and interpretations.

QoD-based control / monitoring / adaptation of workflow execution: During the execution of a simulation workflow QoD metrics must be monitored. Based on monitored QoD metrics simulation workflows can be adjusted by using different data or different algorithms. Therefore, it is necessary to have such controlling and monitoring functionalities within a WfMS. Because simulation workflows are typically long running workflows that deal with huge amount of data, using adjusting functionalities at runtime a QoD driven simulation workflow can reduce the waste of computing resources and human effort. The QoD based workflow adaptation can be done automatically by software or manually through human activities.

C *Methodologies by using a conventional workflow architecture*

To implement our QoDMP we focus on special properties of the conventional workflow architecture [1]. Nevertheless, as our approach is generic it can be adapted to other workflow architecture architectures, e.g. data flow-oriented workflow architectures. In short, conventional WfMSs rely on control flow-oriented workflow languages, such as BPEL, and separates between modeling and runtime environment [1]. A process model that describes the structure of a real world process is created within the modeling environment. The process model defines all possible paths through the workflow and includes the rules that define which paths should be taken. Additionally, it defines all actions that need to be performed the workflow. Examples of conventional scientific WfMS are Trident and [3].

The general idea behind QoD driven simulation workflows based on conventional workflow architecture is to manipulate the set of values that determines the actual path through the workflow. To adjust the simulation workflow for QoD findings, we manipulate control flows by creating new activities for evaluating QoD which need to access data and data references. Furthermore, depending on specific requirements, we need to manipulate the internal structure of invoked services (e.g. a matrix solver service) in order to measure QoD within these services.

IV. QoD Monitoring and Analysis Framework for Workflows

In this section we present different possibilities to realize and integrate our QoDMP into workflow environments. We developed three different mechanisms to measure QoD metrics: (i) by integrating QoD evaluation plug-ins into invoked services which perform the function described by workflow activities, (ii) by providing an independent QoD evaluation service that can be used by any consumer to evaluate QoD metrics, and (iii) by providing an extension activity executed within workflow engines to measure quality of data metrics.

Figure 2 presents how QoD evaluation-relevant components are integrated with WfMSs. The overall system consists of several components working together to integrate our proposed QoDMP into a WfMS. These components can be divided into two groups: *Integration*

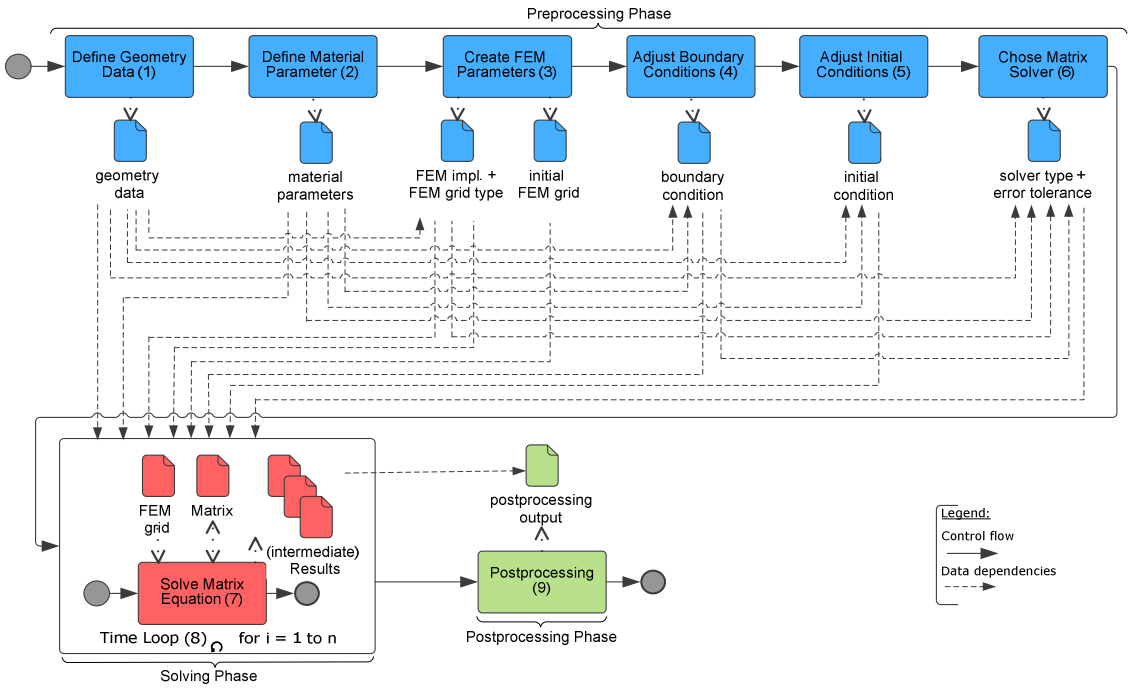

Figure 3: Finite element method (FEM) based simulation workflow with data dependencies.

components and *core components* of the framework. In order to analyze and interpret QoD, we utilize different types of tasks. In the following, we describe them in detail.

A. QoD measurement mechanisms

The integration components are included into workflow engines or invoked services in order to capture measurements used for QoD evaluation or even QoD metrics. The first measurement mechanism is the *QoDMetricEvaluation Plug-in* component (Note: In this section evaluation stands for evaluate a metric and not for evaluation in the sense of interpretation such as in the previous section). This type of component (shown for Activity A in Figure 2) can be used by any individual invoked service of a workflow to measure QoD metrics within this service. The *Plug-in* component has to be instrumented into corresponding services. On the one hand, it helps determining some QoD metrics for data which cannot be seen by the workflow. Thus, it can be useful for long-running invoked services. On the other hand, it helps reducing overhead due to QoD evaluation as this type of plug-in can access data to be evaluated locally. An example of how this Plug-in can be used is the following: An activity implementation A is responsible for solving a matrix equation. At some following activities, the workflow needs a accuracy quality of this matrix to make a decision. The activity implementation A can measure the quality directly and send the result to the *QoD Monitoring, Analysis and Interpretation Service*. When the workflow needs the quality, it retrieves the quality from the *QoD Monitoring, Analysis and Interpretation Service.*

The second measurement mechanism uses the *QoD Monitoring, Analysis and Interpretation Service* to determine the QoD. This service offers interfaces for any consumer (e.g., *QoDMetricEvaluation Plug-in* components) to send data or reference to data to the service. Based on that and requested QoD metrics, the service will return the evaluated QoD metrics. This mechanism is built based on previous work on evaluating data in Data-as-a-Service [16].

The third measurement mechanism is performed by the *QoDExtensionActivity* component (shown in Figure 2 at QoD Activity Extension). This type of components is automatically instantiated and executed by the workflow engine in order to determine QoD metrics within the workflow. The scientist, for example, can specify which QoD metrics should be determined for which data flows. Based on that, the workflow engine will create a *QoDExtensionActivity*. A *QoDExtensionActivity* can perform the measurement of QoD metrics itself (EA1) or utilize the *QoD Monitoring, Analysis, and Interpretation Service* (EA2) to support its measurement. In any case, measured QoD metrics will be sent/stored to/in the *QoD Monitoring, Analysis and Interpretation Service.*

B. Core components for evaluating QoD

There are five core components interacting together to support evaluation of QoD metrics (shown in Figure 2 at Workflow QoD Evaluation Framework): The *QoD Monitoring, Analysis, and Interpretation Service* offers features for storing, analyzing, and interpreting QoD. For example, this service can be called by any application to measure or monitor QoD of data that the application passes to the service (or the reference to the data is passed to the service). The process of QoD measurement is organized by a task concept. Basically, tasks can be used

to execute algorithms for determining and interpreting QoD. After receiving requests for analyzing QoD, the service passes them to the *QoD Evaluation Task Manager* which is responsible for processing the tasks. For that, the manager looks up the appropriate metric by calling the *QoDMetric Catalog Management* component which organizes all available metrics and handles the flow of input data. After the respective metric is found the manager prepare the task, processes the calculation of the metric and returns the measured values to the requestor. The system is extensible which is achieved by two components: *QoDMetric Evaluation Plug-ins Catalog* and *QoD Evaluation Plug-in Manager*. Both components work together and provide methods to register and search existing implementations of metric analysis and interpretation algorithms. Available QoD values can be sent to a *QoD Visualization Tool* which visualizes the values to the end-user (e.g. a scientist). Similarly, other tools can also subscribe and receive QoD metrics from the *QoD Monitoring, Analysis and Interpretation Service*.

C. Tasks for calculating and interpreting QoD

To analyze and evaluate data the *QoD Evaluation Task Manager* uses different types of task. First, *MetricCalculationTasks* are used to calculate QoD metrics. The resulting metrics can (but not have to) be evaluated by *InterpretationCalculationTasks*. We decide to implement this task concept because it allows specifying a *MetricCalculationTask* whose results have to be evaluated by a specified *InterpretationCalculationTask* in one single message so both tasks can be transferred (and calculated) within one service call.

Our service supports objective QoD analysis and interpretation by automatically computable algorithms as described above and subjective QoD metrics by human tasks. To support human tasks, the core system provides a communication interface which allows sending HumanTasks [17] to an external component and routes the results back to the corresponding task. For every objective task there is a corresponding subjective human task.

Each task, whether is software-based or human-base, can be specified and executed in synchronous or asynchronous manner. If a task is executed in the synchronous manner, the workflow execution will wait until the task finishes and returns QoD metrics or interpretations. For asynchronous tasks, tasks can return QoD metrics and interpretations via callbacks implemented using Web services.

V. EXPERIMENTS

We have implemented a prototype of our framework and chose the BPEL-based simulation WfMS presented by Görlach et al. [3] that makes use of conventional workflow architecture specification.

To perform experiments we conducted different configurations of a FEM-based simulation workflow presented in [4]. Figure 3 shows the workflow and data dependencies. For all simulations we use the physics specific framework PANDAS[1] which supports simulations based on the theory of porous media (TPM) [18]. A TPM model with mass exchange calculates the volume fraction and the density function of a solid body can be used for different kinds of simulations, e.g. to calculate the stability of dyke during high water or determine structural changes in human tissue. Since the purpose of our experiments is to examine QoD metrics, not time-based performance, we decided to run our workflows on a Linux-based dual-core machine (2.6 GHz) with 3 GB main memory.

As the result of various input data for activities 1 to 6 in Figure 3 two FEM-based simulation having different physics and characteristics are executed: a ”fluid-saturated elastic column in an impermeable rigid tub” (EC) and a ”rigid slab on a fluid-saturated elastic half space” (RS). We conducted both simulations with a set of material parameters (activity 2 in Figure 3) that have different quality. We use two basic metrics to measure QoD as well as to discourse data and QoD dependencies: Material Parameter Accuracy and Vector Condition.

Material Parameter Accuracy: The correctness of the material parameter to describe the phenomenological behavior of the material depends on the accurate description of all relevant parameter. To simplify our experiments we focus only on the parameter “Lame constant”. We define a well proven Lame constant parameter as reference (good). After that, we multiply this parameter with the factor 0.9, 0.95, 1.0, 1.05 and so on until 1.3. Hence, the characteristics (the first step in our QoD measurement process) of the Material Parameter Accuracy (MPA:char) are 0.9, 0.95, 1.0, 1.05 and so on. In a next step, we make interpretations of those characteristics. A simple interpretation is 1.0 divided by MPA:char (1.0/MPA:char) if MPA:char is greater or equal 1.0 and MPA:char divided by 1.0 (MPA:char/1.0) otherwise. That implies that the accuracy, the goodness, as well as QoD of the changed Lame constant parameter is good if the values are similar to the original Lame constant parameter, and bad if the values vary significantly. For example, if MPA:char is 1.2 the goodness (MPA:good) is 1/1.2 = 0.83. Based on domain scientist’s experience, we interpreted the goodness of the final QoD as:

- good if MPA:good is between 1.0 and 0.95,
- limited if MPA:good is between 0.94 and 0.85,
- poor if MPA:good is between 0.84 and 0.75,
- bad if MPA:good is lower than 0.75.

Vector Condition: To solve a matrix equation such as **A**x = b with numerical methods, the condition of a given matrix **A** and a given vector b determines the solving performance and the quality of the solution vector x. If the difference regarding the least absolute value and the least maximum value of the main diagonal of A and of vector b and x is ”too big” numerical errors can be estimated.

[1] http://www.mechbau.uni-stuttgart.de/pandas/index.php

For the sake of brevity, we represent only the Vector Condition (VC) of b and x. For the Vector Condition of b (VC_b), we define b_l as the least absolute value with b_l unequal 0 and b_m as the maximum absolute value of b. Hence, the characteristic of vector b is the pair of values b_l and b_m. For the interpretation, first we calculate b_l divided by b_m. If b_l/b_m greater than 1.0^{-8} no numerical error must be estimated by using the PANDAS framework. This and the following numbers are based on experience. In our experiments, we defined the goodness and the QoD as:

- 1.0 (good) if b_l/b_m is greater than 1.0^{-8},
- 0.75 (limited) if b_l/b_m is between 1.0^{-8} and 1.0^{-16},
- 0.5 (poor) if b_l/b_m is between 1.0^{-16} and 1.0^{-20},
- 0.0 (bad) if b_l/b_m is less than 1.0^{-20}.

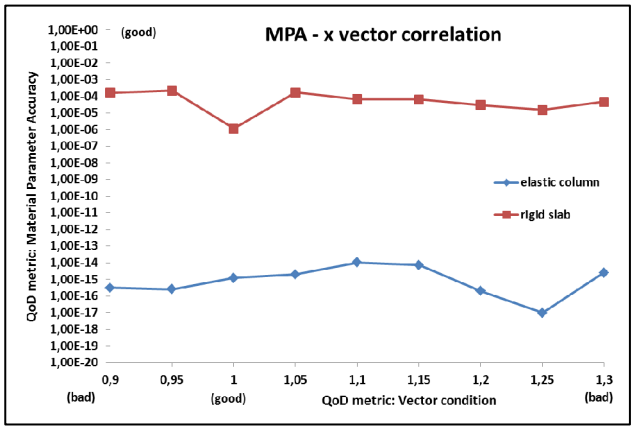

Figure 4: Results of elastic column (EC) and rigid slab (RS) relating to the material parameter accuracy (MPA) and the vector condition (VC) of vector x

The same applies for the characteristic and interpretation for the Vector Condition of x (VC_x). Table 1, Figure 4 and Figure 5 summarize the results of our experiments.

For short, the results show that in both simulations the QoD of the material parameter accuracy has no relevant impact on the QoD of the vector condition in relation to the vector b. In contrast, in the elastic column (EC) simulation the QoD of the material parameter accuracy has an impact to the QoD of the vector condition in relation to the vector x. Based on this finding, numerical problems can be detected at runtime in detail. Especially in complex simulation this knowledge can be used to control or adapt a simulation workflow. This testifies to the importance of QoD measurement by executing simulations with complex data dependencies. Note that in these experiments we examined only objective QoD metrics and interpretations and thresholds are defined for specific interpretations. Using such metrics to steer the workflow execution should be dependent on specific interpretations and is a complex problem which is out of scope of this paper.

We also conducted larger-scale experiments based on a multi-scale and multi-domain simulation workflow that calculates structure changes within a human bone [19] to evaluate the overhead of QoD measurements. We run the simulation applications on a Linux-based quad-core machine (2.3 GHz) with 6 GB main memory and the workflow middleware on a Windows-based dual-core machine (2.3 GHz) with 6 GB main memory.

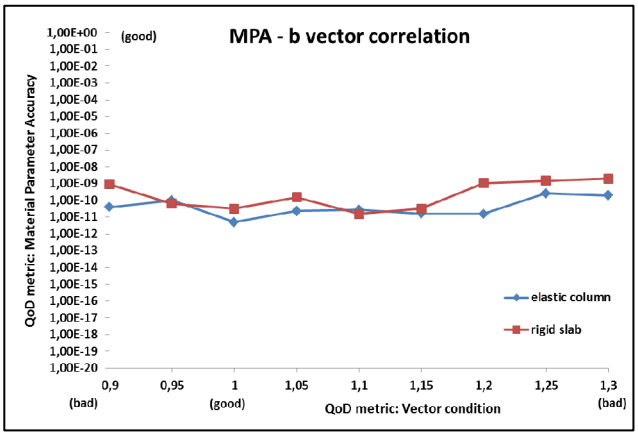

Figure 5: Results of elastic column (EC) and rigid slab (RS) relating to the material parameter accuracy (MPA) and the vector condition (VC) of vector b

Experiment	MPA: char	MPA: good	QoD: MPA	VC_x: char	VC_x: good	QoD: VC_x
EC 0.9	0.9	0.9	Limited	3.1^{-16}	0.5	poor
EC 0.95	0.95	0.95	good	2.6^{-16}	0.5	poor
EC 1.0	1.0	1.0	good	2.1^{-15}	0.75	limited
EC 1.05	1.05	0.95	good	2.0^{-15}	0.75	limited
EC 1.1	1.1	0.91	limited	1.1^{-14}	0.75	limited
EC 1.15	1.15	0.87	limited	7.2^{-15}	0.75	limited
EC 1.2	1.2	0.83	poor	2.0^{-16}	0.5	poor
EC 1.25	1.25	0.8	poor	9.9^{-18}	0.5	poor
EC 1.3	1.3	0.77	poor	2.6^{-15}	0.75	limited
RS 0.9	0.9	0.9	limited	1.6^{-4}	1.0	good
RS 0.95	0.95	0.95	good	2.2^{-4}	1.0	good
RS 1.0	1.0	1.0	good	1.2^{-6}	1.0	good
RS 1.05	1.05	0.95	good	1.7^{-4}	1.0	good
RS 1.1	1.1	0.91	limited	7.0^{-5}	1.0	good
RS 1.15	1.15	0.87	limited	6.5^{-5}	1.0	good
RS 1.2	1.2	0.83	poor	3.1^{-5}	1.0	good
RS 1.25	1.25	0.8	poor	1.5^{-5}	1.0	good
RS 1.3	1.3	0.77	poor	4.8^{-5}	1.0	good

Table 1: Results of elastic column (EC) and rigid slab (RS) relating to the material parameter accuracy (MPA) and the vector condition (VC) of vector x

The overhead of QoD measurements strongly depends on the implementation of the QoD metrics. Negligible measurable overhead was identified if we implement the QoD metric Material Parameter Accuracy on the Windows-machine on which the workflow middleware is running and if we invoked the QoD task in an asynchronous manner. In contrast, the execution time was doubled if we run the QoD metric Vector Condition on the same system than the simulation applications, invoked the QoD task in a synchronous way, and stored all QoD results for provenance reasons on a relational database (PostgreSQL DB). Furthermore, we have implemented the evaluation of the Vector Condition as a subjective human task and invoked this activity in a synchronous manner. Hence, the speed of execution depends mostly on the time a scientist will need to make a decision.

VI. Conclusion and Future Work

In this paper, we present a novel QoD measurement process for simulation workflows. We have described how this process and its components can be integrated into a conventional workflow management system for simulation workflows. As the next step, we will continue to enhance our prototype and conduct experiments in larger scales, in particular, integration specific plug-ins for QoD metric evaluation and expert activities in evaluating QoD metrics. We will investigate the integration of our QoD framework into non-conventional WfMSs such as Kepler or Pegasus. Furthermore, we will focus on adapting and optimizing workflows based on QoD metrics.

Acknowledgement

The work presented in this paper has partly been funded by the DFG Cluster of Excellence Simulation Technology (http://www.simtech.uni-stuttgart.de) (EXC310) and by the Vienna Science and Technology Fund (WWTF), project ICT08-032.

References

[1] Leymann, F.; Roller, D.: Production Workflow: Concepts and Techniques. Prentice Hall, Englewood Cliffs, NJ (1999).

[2] Taylor, I.; Deelman, E.; Gannon, E.: Workflows for e-Science - Scientific Workflows for Grids. Springer, London, UK (2007).

[3] Görlach, K.; Sonntag, M.; Karastoyanova, D.; Leymann, F.; Reiter, M.: Conventional Workflow Technology for Scientific Simulation. In: Yang, Xiaoyu (Eds.); Wang, Lizhe (Eds.); Jie, Wei (Eds.): Guide to e-Science, Springer (2011).

[4] Reimann, P.; Reiter, M.; Schwarz, H.; Karastoyanova, D.; Leymann, F.: SIMPL – A Framework for Accessing External Data in Simulation Workflows, In Proc. 14. GI-Fachtagung Datenbanksysteme für Business, Technologie und Web (2011).

[5] Fahringer, T.; Jugravu, A; Pllana, S.; Prodan, R.; Seragiotto, C.; Truong, H.: ASKALON: a tool set for cluster and Grid computing: Research Articles, Concurrency and Computation: Practice & Experience, v.17 n.2-4, p.143-169 (2005).

[6] Altintas, I.; Berkley, C.; Jaeger, E.; Jones, M.; Ludascher, B.; Mock, S.: Kepler: An Extensible System for Design and Execution of Scientific Workflows. SSDBM (2004).

[7] Deelman, E.; Blythe, J.; Gil, Y.; Kesselman, C.; Mehta, G.; Patil, S.; Su, M.-H.; Vahi, K.; Livny, M.: Pegasus: Mapping Scientific Workflows onto the Grid. Lecture Notes in Computer Science, Volume 3165/2004, Second European AcrossGrids Conference, Springer, pp. 11-20 (2004).

[8] Oinn, T.; Greenwood, M.; Addis, M.; Nedim Alpdemir, M.; Ferris, J.; Glover, K.; Goble, C.; Goderis, A.; Hull, D.; Marvin, D.; Li, P.; Lord, P.; Pocock, M.R.; Senger, M.; Stevens, R.; Wipat, A.; Wroe, C.: Taverna: Lessons in Creating a Workflow Environment for the Life Sciences. Concurrency and Computation: Practice and Experience, 18(10):1067-110 (2006).

[9] Churches, D.; Gombas, G.; Harrison, A.; Maassen, J.; Robinson, C.; Shields, M.; Taylor, I.; Wang, I.: Programming Scientific and Distributed Workflow with Triana Services. Concurrency and Computation: Practice and Experience. Special Issue on Scientific Workflows (2005).

[10] Barga, R. et. al.: The Trident Scientific Workflow Workbench. In: Proc. of the 4th International Conference on e-Science, Indianapolis, Indiana (2008).

[11] Hey, T., Tansley, S., Tolle, K.: The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft, Redmond, WA (2009).

[12] Batini, C.; Scannapieco, M.: Data Quality: Concepts, Methodologies and Techniques, ser. Data-Centric Systems and Applications. Springer (2006).

[13] Na'im, A.; Crawl, D.; Indrawan, M.; Altintas, I.; Sun, S.: Monitoring data quality in Kepler. In: Proc. of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC '10). ACM, New York, NY, USA, 560-564.

[14] Missier, P.; Embury, S.; Greenwood, M.; Preece, A.; Jin, B.: Managing information quality in e-science: the qurator workbench. In: Proc. of the 2007 ACM SIGMOD international conference on Management of data (SIGMOD '07). ACM, 1150-1152 (2007).

[15] Gray, J.; Liu, D. T.; Nieto-Santisteban, M.; Szalay, A.; DeWitt, D. J.; Heber, G.: Scientific data management in the coming decade, SIGMOD Rec., vol. 34, no. 4, pp. 34–41 (2005).

[16] Truong, H.; Dustdar, S.;, "On Evaluating and Publishing Data Concerns for Data as a Service," Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific , vol., no., pp.363-370, 6-10, doi: 10.1109/APSCC.2010.54 (2010).

[17] OASIS: WS-BPEL Extension for People (BPEL4People).

[18] Ehlers, W.; Foundation of multiphasic and porous material, In Foundation of Multiphasic and Porous Material, W. Ehlers and J. Bluhm, Eds. Springer (2002).

[19] Krause, R.; Markert, B.; Ehlers, W: A Porous Media Modell for the Description of Adaptive Bone Remodelling. Proceedings in Applied Mathematics and Mechanics (PAMM) (2010).