Work Package 7: Evaluation Report Aerospace Engineering Application

Authors: Guy K. Kloss
Tomas Forkert
Frank Dannemann
Reviewer: Andreas Schreiber

Identifier: D7.2.1
Type: Deliverable
Version: 0.2
Version: December 1, 2006
Status: internal

Abstract

An implementation of the Provenance Architecture has been undertaken within the DLR application TENT [1] for aerospace engineering purposes. Potential "customers" were addressed towards the end of the Grid Provenance project's life time for evaluation. Feedback from the designated target projects and applications within the DLR is presented in this document as well as an outlook to potential other domains.

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

Contents

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

Members of the PROVENANCE Consortium:

IBM United Kingdom Limited	United Kingdom
University of Southampton	United Kingdom
University of Wales, Cardiff	United Kingdom
Deutsches Zentrum für Luft- und Raumfahrt e. V.	Germany
Universitat Politecnica de Catalunya	Spain
Magyar Tudomanyos Akademia Szamitastechnikai es Automatizalasi Kutato Intezet	Hungary

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

1 Introduction

The reference implementation from the Grid Provenance project has been used for the simulation integration environment TENT [1]. TENT is used widely in various DLR internal and external projects for simulation and automated computations in heterogeneous distributed systems. The reference Client Side Library (CSL) has been embedded within TENT for precise recording of processes' provenance. The records are stored in the reference ProvenanceService, and finally queries and analyzes are performed by the Tools Portal with various provenance specific portlets. The software required for provenance enabling the applications is available through [2]

Project members were introduced to the concepts of Provenance recording, and the resulting benefits and implications for productive use. A demonstration of a provenance enabled TENT environment (in use within the projects already) with a simple demonstration application (simple 2D computational fluid dynamics code [3] with GNUplot for visualization) was given. After conducting a simulation run the provenance of generated data could be browsed through the Tools Portal, and potential queries to answer day-to-day questions of simulational engineering were discussed, and possible additional benefits by mining the data present was projected.

1.1 Audience

Designated target projects for the implementation were SikMa [4] (simulation of complex flight maneuvers), IMENS+ [5] (thermal/mechanical stress simulation of space craft parts on re-entry into atmosphere) and the team around the DataFinder [6] (scientific/technical data management) application.

1.1.1 SikMa

The members of the SikMa projects were the ones with the highest level of provenance methodology knowledge. Within the SikMa project data and process analysis demand has been identified very early. Provenance recording as a means of solving the problem, initially has not been available in the project methodological process. So the TENT application's logging facilities were re-configured to capture the information necessary for the analysis. It was obvious that this would provide a short term solution only. Individual log files were generated on various of the distributed hosts, the information was "blurred" with application specific information used for feedback and debugging, recording was not permanent (log file roll over), merging of the files had to be preformed manually using time stamps, etc.

All this led to an early approach with the concepts of the PrIMe methodology [7] to overcome the problems. Currently the workflow of the SikMa scenario is in the initial process of being provenance enabled. The hooks for the recording of p-assertions are implemented, while concurrently the designated queries and the required information for provenance recording to answer these are being determined.

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

1.1.2 IMENS+

The IMENS+ team was informed on the subject of provenance recording and analysis based on it by a team member working on the TENT application. The simulation workflows in IMENS+ feature similar complexities as those in SikMa, so it was natural to them to also consider an implementation for the IMENS+ project. The possible demand for provenance awareness in the project is also present, but no concrete efforts have been undertaken, yet.

1.1.3 DataFinder

DataFinder as a system for scientific and technical data management deals with quite opposite use cases. But as it *manages* data undergoing various transformation, and it can also manage data that gets used or generated by the workflow system TENT, it is quite natural to contribute homogeneously meta data for provenance recording purposes homogeneously as well. Due to the mentioned reasons, and the fact that it is developed by the same team as TENT, it will be provenance enabled. Although – as the nature of the application is quite different – a mapping of the domain to the provenance architecture [8] has not been undertaken, yet.

2 The Provenance Demonstrator

Before enabling the SikMa workflow for Provenance recording it was decided to build a Provenance demonstrator. The purpose of this demonstrator is twofold:

- To provide a simplified environment for testing how to introduce Provenance recording into TENT.
- To demonstrate the benefits of Provenance recording to the SikMa community.

2.1 Functionality

The TENT Provenance demonstrator workflow (see Fig. 1) is centered around the simple computational fluid dynamics solver Nast2D [3] that is applicable to physical problems in two spatial dimensions. The workflow is constructed of the four components Action, ParVar, Nast2D, GNUplot, where only the latter three are specific for the demonstrator. Arrows connecting the components describe the directionality and type of events being exchanged on interactions. The Action component is used for initially triggering the workflow from the TENT user interface (GUI) and control is then directed to ParVar (short for "parameter variation"), and ParVar in turn is used to consecutively trigger Nast2D. ParVar generates a sequence of different values for the Reynolds number that is characteristic for each single run of Nast2D. Initially, ParVar

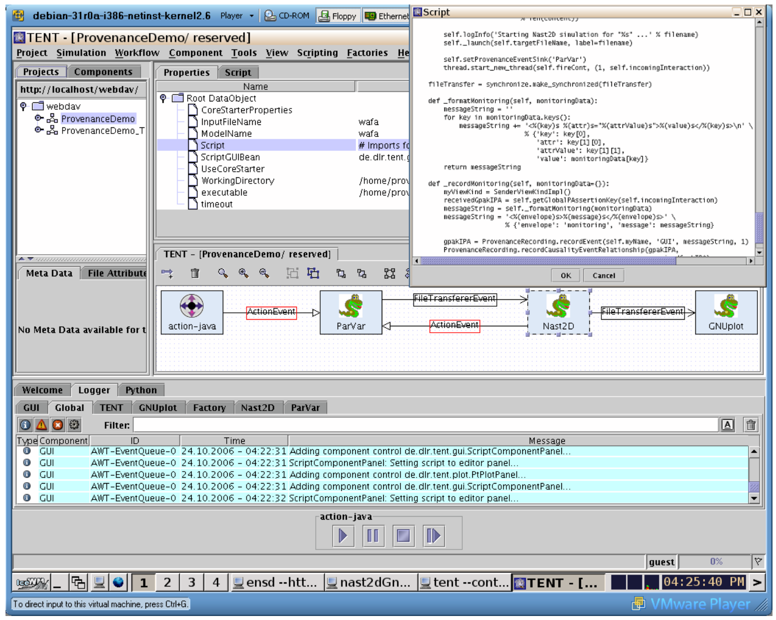

Figure 1: TENT provenance demonstrator workflow with scriptcomponents

generates a specific input file (using the generated Reynolds number) and sends it to Nast2D by means of a file transferrer event.

After the solver has determined the proper flow field for the given Reynolds number, the Nast2D component triggers the GNUplot component by sending a file containing the resulting 2-dimensional flow field using the file transferrer event. GNUplot, the recipient of the event, subsequently visualizes the results.

While GNUplot is used for rendering the final 2-dimensional flow field, the Nast2D component additionally offers the possibility to gather and display numerical data describing the computational progress while the solver is still active. This monitoring data is collected and sent to a TENT-internal monitoring tool (not represented by any of the components visible in the TENT workflow panel, see Fig. 2).

Finally, Nast2D notifies the ParVar component on completion of the single simulation by sending an appropriate event. In turn, ParVar generates a new Reynolds number and sends an updated input file to Nast2D continuing with the computation loop. Altogether, three different values for the Reynolds number are generated, thus once started, this simulation loop in the workflow executes three times.

The demonstrator workflow has been built based on a flexible kind of TENT component, the "scripting component." A number of these components allows the main functionality of an individual component to be implemented by means of Python scripts that are being run using an embedded Jython interpreter.

The advantage of this approach is the good transparency of the internal processing in the components as opposed to using statically compiled Java components. Additionally, the provenance interface only needed to be incorporated in a limited number

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

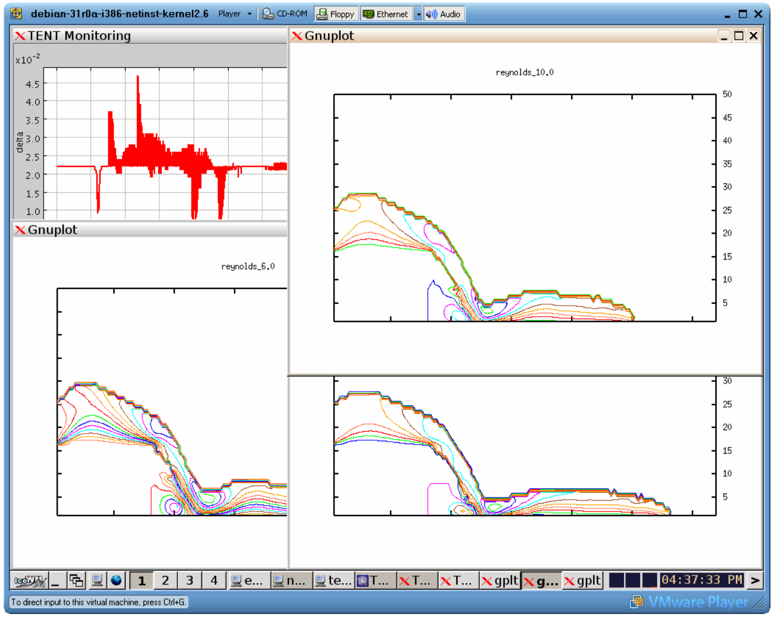

Figure 2: TENT provenance demonstrator result visualization

of code locations. Care has been exercised to keep the calls to the provenance API on the script level as concise, yet as clear as possible.

2.2 Making TENT Provenance-Aware

The inherent structure of the TENT integration environment makes it a suitable candidate for becoming Provenance enabled since the basic components that TENT workflows are being composed of are easily identified as provenance actors.

Apart from the obvious actors in TENT there are, however, "hidden" ones that have to be identified carefully. In general, they are related to the initial sources and final sinks of information related to the setup and outcome of the simulations to be performed within TENT.

2.2.1 Queries

The type of p-assertions to be recorded, and the locations where they have to be generated, depends on the type of query that is intended to be used for examination. As it can be seen from the example above, a typical workflow for numerical simulations of physical phenomena involves a limited number of actors. They are activated and executed in a loop until certain termination criteria are fulfilled.

With respect to a workflow as mentioned above, some of the most basic provenance related questions to be asked are *"Given a particular workflow, and a specific set of result data, ..."*

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

- What have been the initial conditions for this simulation?
- What were the termination criteria?
- How often has the workflow been executed?
- What specific algorithm/version has been used in computations?

Provided that TENT has been designed to be a distributed system, specific questions relating to that might be asked, as well:

- What were the hosts participating in the simulation run?

With respect to the demonstrator workflow the most suitable provenance related questions to be formulated are:

- For a given set of results, what was the input data?
- For a given single result, what were the parameters used?

As will be seen below, some of the questions might be answered implicitly (e. g. "How often has the workflow been executed?"), but the queries relating input and output data require explicit information about the current workflow state to be stored. Before going into more detail, we need to discuss how the terms "set of results" and "input data" are to be defined in the TENT context.

Generally, when running simulations from the aerospace context within TENT, the input data is a combination of one or more input files and single pieces of information as numerical values and/or state variables, descriptors, and workflow specific meta information. The possible outcome of the simulation run also comprises of files and single pieces of information, without changing the meta information.

For the demonstrator workflow this view has been limited to the input file describing the physical boundary conditions and the particular value of the Reynolds number that is characteristic for a specific run. This number is also part of the input file, but since it is the characterizing quantity it is treated separately as well. A set of output data is defined to consist of the output file produced by the solver Nast2D and the monitoring data. Since the provenance service is not designed to be a file store, the file specific information is limited to descriptive information: file name, location in the file system, and a hash value for verification of consistency. This information has been made part of the "pay load" of the exchanged TENT events and as such has been prepared for provenance storage and retrieval.

2.2.2 Interactions

Most of the interactions are already visualized in the TENT workflow panel by means of wires representing the lines along which information is propagated in the form of so called TENT events. There are a number of different events that might be sent, each one having a different semantics and possible “pay load”. An event sender might be connected to more than one event recipient, that is, TENT events are being broadcasted to all connected listeners and not generally sent to individual ones. Thus generally, to the event sender the recipient is anonymous. Messages that are sent to “hidden” actors clearly are not visible in the TENT workflow. As mentioned earlier, the TENT online monitoring facility (part of the GUI) is one of the “hidden” actors. Thus, for provenance purposes the information is treated like having been sent by a fictitious “monitoring event” to the plot panel in the the actor “GUI”.

2.2.3 Recording of P-Assertions

As already mentioned, the most natural starting point for making TENT provenance aware are the TENT components that are easily identified with provenance actors. The basic component interaction is sending and receiving TENT events. Thus, provenance event information is being recorded upon sending and receiving a TENT event.

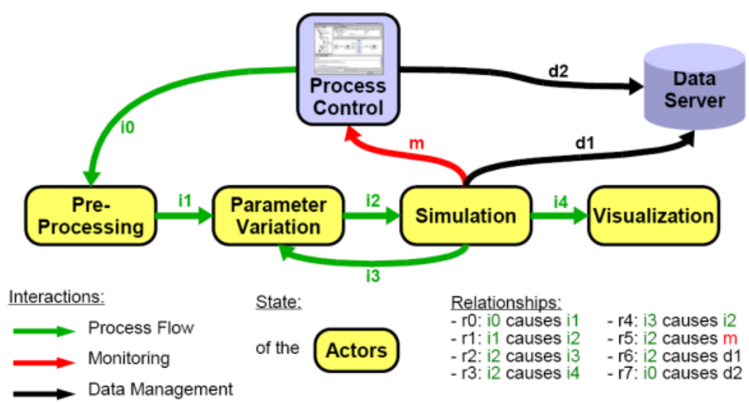

Figure 3: TENT demonstrator with all actors, interactions and relationships

For linking the record of the sending side of an interaction to the receiving side’s record, a unique interaction ID is generated by the event sender and transferred along with the event. This interaction ID, together with the source and sink of the interaction, forms the interaction key for recording p-assertions. Due to the broadcast nature of TENT events, which requires anonymous endpoints to be inserted into an interaction key, additonal actor state p-assertions had to be recorded in order to capture the actual components names.

In order to get a continuous chain of interactions for a computation case (a simulation) the recorded interaction p-assertions need to be linked. This is accomplished by recording additional relationship p-assertions (see Fig. 3), by every actor *B* that generates events which have been caused by a certain incoming event on this actor. For simplicity, only "caused by" relationships have been used.

2.3 Results

The results of analyzing a run of the TENT demonstrator by means of the Provenance Tools (as developed by the University of Cardiff) are shown in Fig. 4 and Fig. 5.

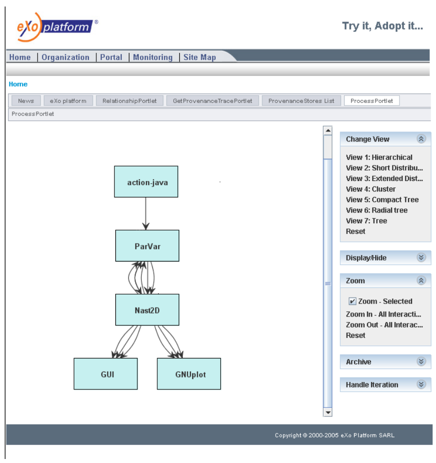

Figure 4: TENT provenance demonstrator process view

The process portlet provides a view of the various actors and the mutual interactions. It completely matches the view from the TENT workflow panel when taking into accout that "GUI" represents a hidden actor not being visible in Fig. 1. The arrows connecting the various actors correctly reflect the number and direction of the interactions involved.

Another display of the interactions is provided by the relationship portlet (Fig. 5). As opposed to the process view, in which actors are represented as boxes, a box in the relationship view represents an interaction. Thus each of the connecting lines from Fig. 4 is now represented as a box with arrows indicating the (causal) relationships between them. The relationship portlet provides something that, using a mathematical terminology, might be interpreted as the "first derivative" of the process view. The causalities are displayed bottom-up, i. e. the initial interaction between the Action and ParVar components is represented by the lower most box. Labelling of the boxes indicates source and sink of the interaction in question and provides information about the type of the interaction. As can be seen quite easily, the diagram extends over three levels with repeating patterns of causalities, thus, reflecting the fact that this workflow is executed consecutively three times.

In order to have a look at the "pay load" of the interaction in question, a right click on the box lists the contents. The information displayed depends on the type of

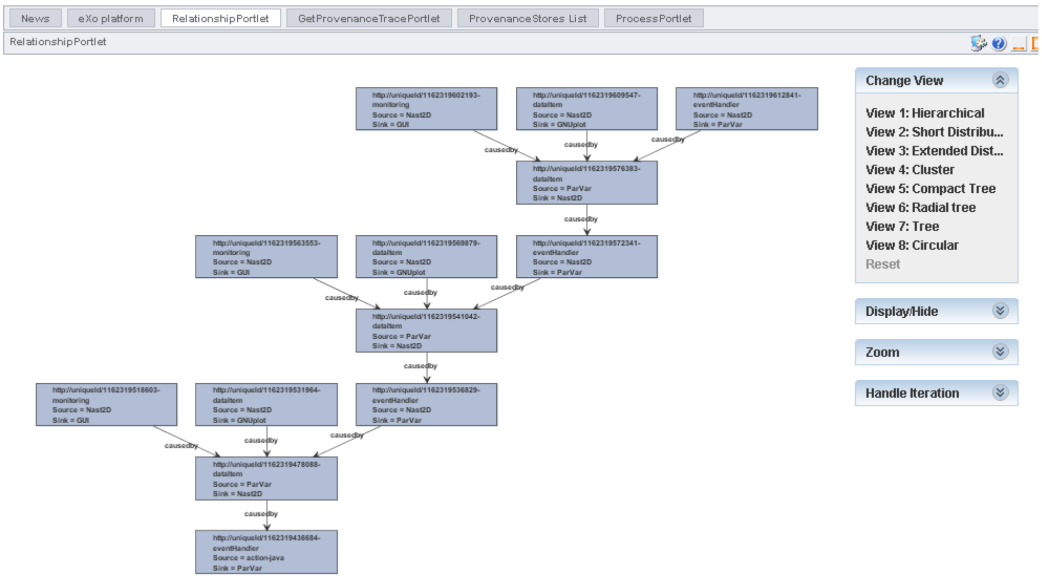

Figure 5: TENT provenance demonstrator relationship view

interaction involved and is formatted with an appropriate data accessor. Fig. 6 lists the parameters describing the contents of a file transferrer event used for sending a new version of the Nast2D input file “wafa.in” from the parameter variation component ParVar to Nast2D.

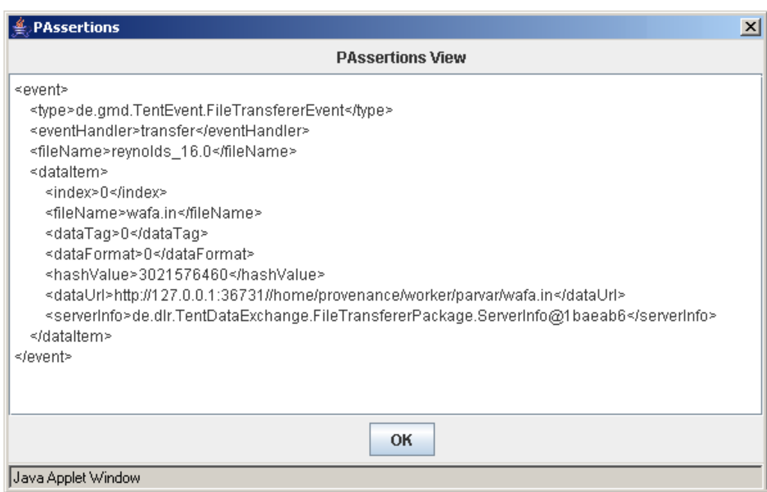

Figure 6: Relationship portlet data accessor view

Analyzing the information provided by the relationship portlet provides answers to the queries that have been identified in section 2.2.1. This enables one to track which particular output file or set of monitoring data corresponds to a given value for the input parameter *Reynolds number*, that is characteristic for each individual run of the

The PROVENANCE project receives research funding from the European Commission’s Sixth Framework Programme

workflow.

3 Discussion of Provenance Enabled Applications

In the following the implementation process of TENT for SikMa will be discussed.

As already stated at the beginning of section 2, the demonstrator has been used as a test environment for the integration of provenance recording into TENT. Nevertheless, the SikMa workflow deviates from the demonstrator workflow in several important aspects, as outlined below.

SikMa workflows fall into two categories:

- stand-alone workflows
- coupling workflows

3.1 Simula–Tau Coupling

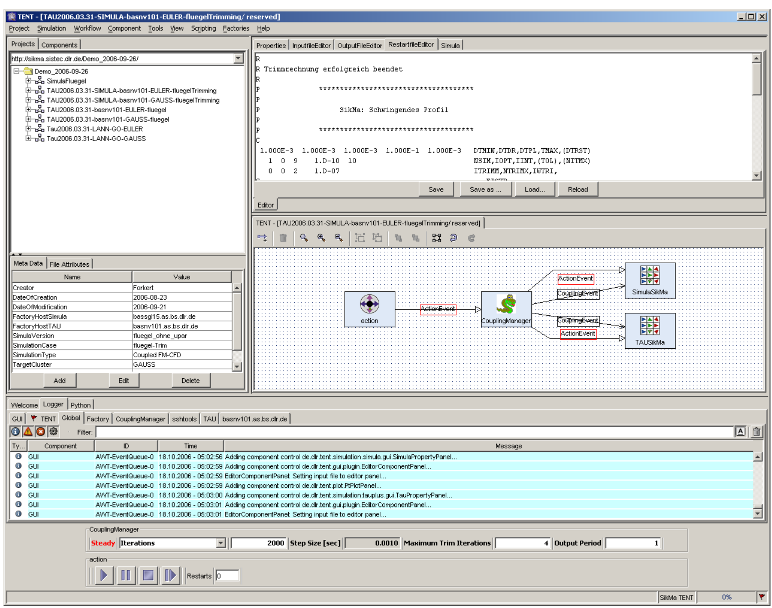

Figure 7: TENT flight mechanics coupling workflow

The layout of a typical SikMa coupling workflow will be seen in Fig. 7. The presented workflow is the one for coupled computations between the two numerical simulation codes *Simula* (flight mechanics) and *Tau* (computational fluid dynamics). The same workflow layout is being used for the aero-elastic SikMa coupling workflow that involves *Tau* and the structural mechanics code *UCJN* (not shown here).

The central component is the *Coupling Manager*, which is used for executing the correct coupling sequence and exchanging coupling information between the codes.

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

In the workflow panel unidirectional connections appear between the *Coupling Manager* and the simulation components. The respective wires denote the already known *ActionEvent* and another kind of TENT event which has not been used in the demonstrator. According to its usage this event is named *CouplingEvent* and is designed for passing coupling information from the *Coupling Manager* to the simulation components. Despite its apparent unidirectionality, the information flow is, in fact, bidirectional! This is why all components in this workflow (except for the *Action* component) in terms of provenance recording are both sending *and* receiving actors, although this is not apparent from the graphical workflow representation. Except from this peculiarity, the *CouplingEvent* is the most suitable of the possible TENT events for being provenance enabled since it is addressable. In spite of the already mentioned general broadcasting nature of the TENT events, it requires a distinguished recipient to be given, and this in turn is checked by the actual recipients.

Apart from the obvious actors, there will be several more "hidden" actors than in the demonstrator workflow, among the most important ones being the integrated applications themselves. This comes about since in the SikMa workflows applications are controlled explicitly at run time, whereas in the demonstrator workflow the communication between the Nast2D application and the associated TENT component is limited to starting the application and waiting for its completion (black box). The information exchange directed towards the application, therefore, has to be subject to provenance recording, itself.

3.2 Tau Stand-Alone

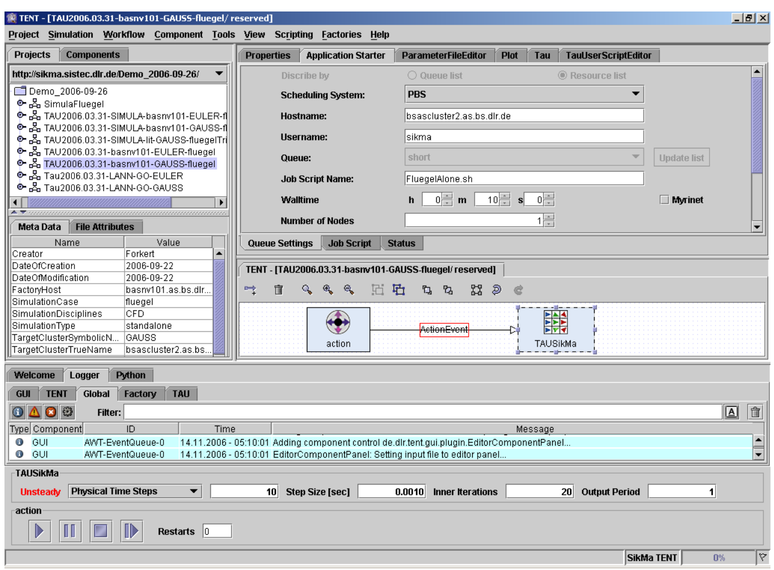

Figure 8: TENT Tau stand-alone workflow

The importance of recording the interactions with the application as a hidden actor becomes fully evident when looking at the workflow in Fig. 8. Only one *ActionEvent* is being sent in the beginning. All further activity will take place *inside* the component and, therefore, the provenance of the outcome of this process is only recoverable when the hidden interaction between the Tau *component* and *application* is recorded properly. This is due to the fact that the Tau component is connected to the application's main loop, and control information is exchanged continuously.

Another hidden actor at run time is the so called "data object" of a component. The data object is a container holding all important control and state variables of a TENT component and is partially modifiable by the user before application start-up. It is also used for automatically triggering activities during application run time when certain conditions are fulfilled. All the relevant state variables and their respective changes in the data object are being made available for recording through actor state and interaction p-assertions.

4 Results

4.1 Demonstrator

Building the provenance demonstrator as a first step has proven to be a very good choice. On the one hand it helped to decrease the initial efforts for team members in the development. On the other hand it aided immensely in demonstrating the provenance enabled TENT application to the interested audience, since the provenance enabled TENT system has not been deployed for specific end user scenarios in SikMa and IMENS+, yet. It put forward a real working example in easily understandable terms in a use case touching most engineers' demands for a simulation environment.

4.2 SikMa Project

The provenance demonstrator has left a positive impression on the members of the SikMa project's team in that it has led to a good grasp of understanding towards the issues involved in provenance related data recording and analysis.

The previously unsatisfactory need to resort to the application's logging messages in order to recover provenance related information for individual runs of the SikMa workflows had already been identified as a major issue in the project.

Introducing the provenance awareness to the TENT infrastructure offers solutions to pending problems. This is, however, only the technical aspect of finding the solution, since the scientists and engineers involved will also have to adapt their work style to become "provenance compatible". On the algorithmic side this requires a strict management of versions for the developed numeric solvers and the exposure of this version information to the provenance system for recording. With respect to the data that are generated in simulation runs it is no longer advisable to simply modify input files or overwrite output files in place on the data servers, as this will result in

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

corruption of the information consistency of referenced information *not* stored within the provenance service (e. g. large result sets, retrieved input models).

In view of these facts a short questionaire has been developed, presented and discussed with the SikMa team members.

- Do you think provenance recording adds a benefit to your every day work?
- Will it be difficult for you to adapt to a "provenance" style of working with your simulations?
- Are the demonstrated queries a good starting point for your work?
- Is the provided documentation helpful in understanding the concepts and implications of a "provenance"enabled style of work?

There has been a very positive feedback obtained with respect to the presented questionaire. A clear majority of SikMa team members expressed their positive attitude towards the benefits of provenance recording and the implications for their work.

Consequently, extending TENT with the additional provenance recording capabilities that are specific for the type of workflows used in the SikMa project has started in a straight forward and efficient way.

4.3 IMENS+ Project

The project domain within IMENS+ is similar to the one of the SikMa project. But the demand for post process analysis is not as high, so the project's team members have decided to wait for first experiences from "real world usage" in the SikMa project.

Judging from the current impression – trying out the demonstrator and discussions with the other projects – it is, however, probable that a deployment of a provenance enabled system to their problem domain will be undertaken in the near future.

5 Conclusion

The provenance architecture has been designed to fit the various needs of problems in quite different domains. It seems also very well suited for solving the problems encountered in the field of aerospace engineering. Of course recording p-assertions to a remote server adds an overhead, but so did the remote logging facility. Delays due to this were easily circumvented by placing recording code into a separate thread freeing the system to continue almost uninterruptedly to spend it's computational resources on solving the numerical problems.

Managing access to the recorded p-assertions for process analysis and querying relieved the GUI and TENT environment code from being bloated with too much distracting functionality. The exclusive use of the web browser based portal eased familiarization as to the end user the TENT GUI remained unchanged.

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

The only issue users were having is initially changing the every day working style. Due to provenance recording and analyses performed upon that information the working style also has got to be "provenance enabled". But this is considered to be just a matter of training and time.

6 User Requirements Compliance Table

The following table shows to which degree the requirements of the users and developers of the TENT system were fulfilled through the provenance architecture and it's implementation within TENT. The listed user requirements are the outcome of the Provenance User Requirements Document [9].

Req. ID ***Flag***	**Description**	**Outcome**	**Remark**
AR-2-1 *essential*	The provenance architecture should be able to store all kinds of information that is needed to trace back the preceding process of data transformation within a workflow.	fulfilled	All relevant information could be stored in the provenance service.
AR-2-2 *essential*	Recorded provenance information should make it possible to automatically restart workflows or parts of a workflow by the TENT system.	partially fulfilled	The respective information is not yet provided to the provenance system. The provenance system provides the necessary functionality, but the workflow system is still lacking full capability of warm starting computations.

AR-2-3 *essential*	The provenance architecture should be able to provide a trusted historical record of user access to produced data during a workflow (including intermediate data, result data and associated meta-data as well), which can be used as evidence that the given data set has been accessed only by authorized users (as specified by the initiator of the workflow).	partially fulfilled	Computations are conducted in a controlled and protected environment (including the provenance service). The partners have agreed that for this reason user access is not to be tracked through all services individually. Too many points of access would need to be targeted towards this specific environment that are not well re-usable. Therefore security measures will have to be provided by the setup on deployment.
AR-2-4 *essential*	The provenance architecture should make it able to identify unauthorized accesses to produced data during a workflow (including intermediate data, result data and associated meta-data). Access rights are specified by the initiator of the workflow.	partially fulfilled	The respective information is not yet provided to the provenance system. (See also AR-2-3.)
TR-1-1-B-1 *essential*	For the output of the TAU module the version information of the involved TAU code should be recorded by the provenance system.	partially fulfilled	The respective information is not yet provided to the provenance system. The provenance system is capable, but the precise version information is not yet queryable from the simulation code.

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

TR-1-1-B-2 *essential*	For a given output of the TAU module the processed input files should be recorded by the provenance system.	partially fulfilled	The respective information is not yet provided to the provenance system. The provenance system is capable, but this functionality is needed for restart capability. (See also AR-2-2.)
TR-1-1-B-3 *essential*	For a given output of the aero-elastic module the processed input files should be recorded by the provenance system.	partially fulfilled	The respective information is not yet provided to the provenance system. The provenance system is capable, but this functionality is needed for restart capability. (See also AR-2-2, TR-1-1-B-2.)
TR-1-1-B-4 *essential*	Rejection of job submission by the TENT framework to cluster batch systems should be recorded by the provenance system, so this event can be recognized by TENT and it can restart the workflow or the appropriate modules.	not applicable	Such an event requires user interaction and cannot be handled automatically. Only the reason as given from the scheduling system can be recorded, but a correction for the re-submission would require user intervention.
TR-1-1-B-5 *essential*	The provenance architecture shall provide a way to map TENT access rights to ensure that no misuse of provenance data will take place.	partially fulfilled	The respective information is not yet provided to the provenance system. (See also AR-2-3.)
TR-3-5-A *essential*	The system should support the storage of recorded provenance data for a complete simulation session. Run time for a simulation session is between 1 minute and 1 month, its typical value is a few days.	fulfilled	The provenance data resides in an XML database for further processing.

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

TR-3-6 *essential*	The system should be able to archive recorded provenance data.	fulfilled	The Provenance services includes a backup functionality.
TR-3-7 *essential*	The system should be able to export recorded provenance data for external usage.	fulfilled	Using the tools provided with the provenance service p-assertions can be dumped to the file system.
TR-4-2 *essential*	The architecture should support the dynamic processing of provenance data, i. e. recorded provenance data should be instantly queryable even if a recording session (recording of interrelated provenance records belonging to e. g. the same workflow) is still in progress.	fulfilled	The provenance service can be queried at any time on the basis of all (so far) received p-assertions.
TR-4-3 *essential*	The provenance architecture should support the storage of results of analysis and reasoning operations performed on the provenance data by tools that are not part of the generic architecture (3rd party tools on the application layer).	fulfilled	The libraries and tools support access to the information in a unified way, so both the queries and results can be stored for further use in any way.
TR-5-3 *essential*	The provenance architecture should be deployable as an integrated part of a system, as a service within the same administrative domain as the client system and as a 3rd (external) party operated service, too.	fulfilled	For example, the TENT system is deployed using a 3rd party deployment tool (InstallAnywhere) that is easily extendable for including the Provenance Client Side library.
TR-6-1 *essential*	The architecture should support a rich set of generic APIs that allow analysis and reasoning tools to be built upon.	fulfilled	The reference implementation's Client Side Library (CSL) provides all the interfaces necessary to build upon.

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

TR-6-4-A *essential*	The provenance architecture should provide a programmatic interface for the administration of the system.	not applicable	Tools are provided for the management of the provenance service's setup and management. No provenance-specific management functionality has been identified in a provenance store, hence the project has not defined a management interface or API.
TR-6-4-B *essential*	The administrative interface of the provenance architecture should be able to be accessed and controlled through it's API. It has to be integrable into TENT or at least be accessible through the TENT system. Therefore some kind of user authentication may additionally be needed.	not applicable	Access to management functionality is only possible through the provenance service's setup tools using command line wrappers as no provenance-specific administrative functionality has been identified in a provenance store. (See also TR-6-4-A.)

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

TR-7-1 *essential*	There should exist different levels of system documentation, including the following: • a detailed API documentation for programmers who intend to integrate the provenance architecture into their systems, • a detailed description of the administrative interface of the system for system administrators, • a detailed description of other human-computer interfaces presented by the system e. g. for analysis and reasoning. Different audiences should be taken into account here including end-users as well, who want to use the provided tools as a stand-alone applications.	partially fulfilled	A methodology for provenance enabling applications and domains has been created, and documentations of various levels of details for the provenance service, the client side library, and tools have been provided by the project partners.
CR-1-1-B *essential*	Within TENT the execution overhead due to provenance recording has the upper constraint of not affecting the interaction with the system in a significant manner. In terms of the applications used in TENT workflows: due to typical execution times of e. g. the flow solver TAU, overhead has to be kept at minimum level.	fulfilled	Has been verified by means of the TENT Provenance demonstrator.

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

CR-1-2-B *essential*	There are the same constraints for storage overhead as for execution overhead (see CR-1-1-B), but less restricted.	fulfilled	Has been verified by means of the TENT Provenance demonstrator.
CR-4-3 *essential*	Access rights to the provenance system must be consistent with access rights to the rest of the TENT system. The provenance system should provide a way to map access rights information of TENT into its security subsystem. Access rights are stored in TENT in an LDAP server.	fulfilled	Access rights are granted as specified. Though, within WP7 they are not used. Computations are conducted in a controlled and protected environment (including the provenance service). The partners have agreed that a sensible mapping of access rights would be not ideal and complicated. Additionally it would not be very well re-usable in other environments. Therefore security measures will have to be provided by the setup on deployment.

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

CR-4-4-A *essential*	The provenance architecture should be configurable in a way that assigns the following access rights to the given user groups: • User: Access to provenance data directly involved in the data manipulation process of the simulation (s)he has started and configured. • System designer: Access to secondary provenance data, which is the collection of all user provenance data and derivations from them for analyzing and reasoning purposes. • System developer: Access to all kinds of provenance data. This especially includes data coming directly from the TENT core components. This data has to be visible only for this user group.	fulfilled	As used in WP7 access privileges are not granted on a per-action basis, but rather on the deployment of the whole computational environment. (See also CR-4-3.)

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

CR-5-3 *essential, critical*	The provenance system should be capable of handling huge amounts of provenance data coming in very frequently from the application itself. It should not create any bottle necks disturbing the system.	fulfilled	Recording too many atomic interactions in a short time (data capture loop) will create a bottle neck on recording. The latency, however, was not too big to be able to easily work around this problem by aggregating interactions and submitting them (A) collectively and (B) in a separate thread to not have a negative impact on the system's overall performance.
CR-5-4 *essential, critical*	The provenance system should provide more and more detailed information about the different data and control flows taken place during workflow execution.	fulfilled	The provenance enabled TENT application records process specific information that is chained and semantically tied to the problem domain. Therefore, additional information on the process is available both in higher quantity and quality (structurally linked).

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme

CR-5-5 *essential, critical*	On top of the API of the provenance system TENT must be able to access all its functions and provide them to the users through appropriate interfaces.	fulfilled	All provenance system functions essential for use with the TENT system have been interfaced and are available within the various components. Some functionality did not need to be implemented for direct use within TENT as external portal and management systems are provided.

References

[1] TENT Website. [Online]. Available: http://www.dlr.de/sc/tent

[2] Grid Provenance Project Software Website. [Online]. Available: http://twiki.gridprovenance.org/bin/view/Provenance/ProjectSoftware

[3] NaSt2D/NaSt3D Website. [Online]. Available: http://www5.informatik.tu-muenchen.de/software/NaSt.html

[4] SikMa Project Website. [Online]. Available: http://www.dlr.de/as/sikma

[5] IMENS+ Project Website. [Online]. Available: http://www.dlr.de/sc/desktopdefault.aspx/tabid-1205/1631_read-3193/

[6] DataFinder Website. [Online]. Available: http://www.dlr.de/datafinder

[7] S. Munroe, S. Miles, V. Tan, P. Groth, S. Jiang, L. Moreau, J. Ibbotson, and J. Vázquez-Salceda, "D3.3.1: PrIMe: A Methodology for Developing Provenance-Aware Applications," University of Southampton, Tech. Rep., Nov. 2006. [Online]. Available: http://eprints.ecs.soton.ac.uk/13046/

[8] P. Groth, S. Jiang, S. Miles, S. Munroe, V. Tan, S. Tsasakou, and L. Moreau, "D3.1.1: An Architecture for Provenance Systems," University of Southampton, Tech. Rep., Oct. 2006. [Online]. Available: http://eprints.ecs.soton.ac.uk/13196/

[9] A. Andics (ed.), "D2.1.1: User Requirements Document," MTA SZTAKI, Tech. Rep., Feb. 2005. [Online]. Available: http://www.gridprovenance.org/deliverables/GRID_PROVENANCE-UserRequirements-D211-Month6.pd

The PROVENANCE project receives research funding from the European Commission's Sixth Framework Programme