Dynamic Measurement Scenarios in the Virtual Laboratory system

Marcin Lawenda, Norbert Meyer, Marcin Okoń, Tomasz Rajtar,
Dominik Stokłosa, Damian Kaliszan, Maciej Stroiński
Poznań Supercomputing and Networking Center
Supercomputing Department
Noskowskiego 10, 61-704 Poznań, Poland
Marcin.Lawenda@man.poznan.pl

Abstract

In the paper some dynamic measurement scenarios conception is presented. The Dynamic Measurement Scenarios (DMS) are used for advanced workflow control in the Virtual Laboratory system. It facilitates defining and monitoring the measurement process from the preparation stage through experimental and computational processes to results analysis (based on the achieved visualization data). DMS often consists of many different tasks connected by links with specified conditions. To execute DMS we need a special application (during the definition process) on the one hand and on the other hand a sophisticated (and complex) module is needed for manage and control specified workflow. The described issues are illustrated by Nuclear Magnetic Resonance (NMR) laboratory examples.

1. Introduction

Virtual Laboratory (VL) is a heterogeneous, distributed environment, which allows scientists all over the world to work on a common group of projects [1]. This environment should allow conducting experiments with the usage of physical devices, doing simulation using the computational application, and communication between users working on the same topic. Similarly to typical laboratory tools and research techniques depending on the specific field of science, virtual laboratories can benefit from some collaboration techniques as tele-immersion, but they are not mandatory [2].

Research work of the authors is focused on virtual laboratory aspects that have significant influence on creating a general system which will allow taking control on many different devices. Another issue (which we want to discuss in this paper) is managing the jobs workflow to assure appropriate comfort to the user and spare his/her time.

The Virtual Laboratory system is developed in Poznań Supercomputing and Networking Center in collaboration with the Institute of Bioorganic Chemistry and Radioastronomy Department of Nicolaus Copernicus University.

2. Motivations

Experiments executed in the science laboratories are complex and consist of many stages [3]. Thus, we can define a graph which describes the execution path specified by a user. Nodes in this graph correspond to experimental or computational tasks. Edges (links) correspond to the path the measurement execution is following. In nodes we have a defined type of application and its parameters. In links the passing conditions which substitute decisions made by the user are defined. These conditions are connected with applications and let us determine if the application is finished with the desired results (if not, the condition is not met). This execution graph with specified nodes and links (with conditions) is called the Dynamic Measurement Scenario (DMS). The term ”dynamic” is used, because the real execution path is determined by the results achieved on each measurement stage. All of these parameters can be written down using a special language: the Dynamic Measurement Scenario Language (DMSL).

Let us resume, the most important advantages of MDS:

- connection of different types of jobs (experimental i computational),
- speed up of tasks sequence execution,
- simplifying scenario monitoring,
- possibility of multiple use of a given scenario,
- legible way of the workflow control,
- possibility of defining many measurement execution ways.

The concept of the Dynamic Measurement Scenarios allows defining the process of an experiment in any way, from pre-processing, through executing the experiment, to the post-processing and visualization tasks. Defining the measurement scenario allows to spare a lot of time during computation. The user does not have to wait for the end of a given process stage to submit another one. It is made automatically.

To increase the possibility of the jobs scenario the dynamic measurement scenario (DMS) was defined. In the DMS model, besides the definition of the tasks execution sequence, we can also define some extra connections (e.g. loops, parallel connections), conditions on the connections and different lengths of the execution paths. Thanks to the possibility of the conditions defining on the connections paths, the real path is determined during execution and can depend on computational results.

To properly define the Dynamic Measurement Scenario the system has to have knowledge about available applications and connections which are enabled to create. In case of the laboratory type, where the processing software is well known, it seems to be easy. The issue will be more complex when we want to define the DMS model for a wider range of virtual laboratories.

To solve this problem we have defined a special language (DMSL) which determines the following conditions: names of the connected applications, a condition (or conditions) connected with a given path, an additional condition which has to be met to pass a given path (e.g. when special conversion between application is needed) and, finally, a list of input or output files for applications.

3. Designing

The DMS schema must be designed by the VLab administrator before it is available for users. Creating a new Dynamic Measurement Scenario needs a special attitude. We assume that some stages will be performed by the computer science engineer and scientist from a scientific discipline DMS concerns, and some by application.

The designing of the DMS consists of the following stages:

- application analyzing,
- connection diagram preparing,
- describing additional dependencies in the connection diagram,
- applications and links description generating,
- measurement scenario description generating.

All stages are tightly connected between themselves and should be executed in the presented sequence. The first three phases should be performed, as we mentioned before, by the VLab administrator and engaged scientist (see 3.1, 3.2, 3.3). The last two can be done by a special application which take into consideration user's rights and information introduced by the user (see 3.4, 3.5). Next, we will analyse each stage of DMS designing.

3.1. Application analyzing

Each application available in the dynamic scenario must be first analyzed from the functional point of view. Input and output parameters have to be taken into consideration. Also, input and output format files must be described. An exemplary option to describe in these files follows: file type (binary, text), filename format (if exists): filename mask, filename extension. If the file format type is text, the analyst can analyze information within. A specified pattern connected with parameters important for the final user (calculated by application) can be used to specify link conditions.

3.2. Connection diagram preparation

After the application analysis the VLab administrator can make the connection diagram. In this phase applications are assigned to execution stages. Exemplary stages for the laboratory of NMR spectroscopy follow: acquisition, processing, visualization, structure determination and structure visualization.

Each application from each stage is connected to another one taking into consideration data achieved in the previous analysing step (see 3.1).

At this stage the necessity of cooperation between a computer science engineer and a scientist connected with a given science discipline is strictly required.

In Fig. 1 we present an exemplary diagram for the Virtual Laboratory of NMR Spectroscopy.

3.3. Describing additional dependencies in the connection diagram

At this stage we present conditions on the connections between applications which were created at the previous stage (see 3.1). We should focus our attention on: connection conditions, conversion issues and files types related to links.

Conditions defined on the connections can influence the measurement execution path. After the end of each application execution they are verified and in this way the following execution path is determined. For more information please go to 4.

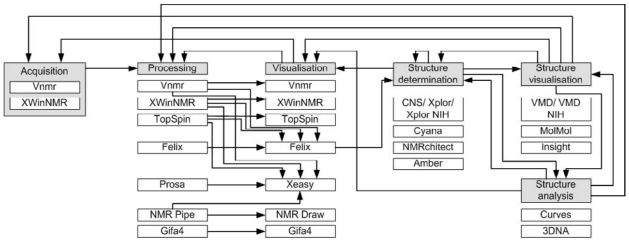

Figure 1. An exemplary connection diagram for the Virtual Laboratory of NMR Spectroscopy

Conversion is performed when two connected applications have a different input-output file format. It should be described for each connection separately. Depending on the connected applications conversion can be realized in a different way. More about the conversion issues can be found in 5.

It is necessary to determine which type of file can be used as an input file to the target application. Therefore, a set of output files of a given application must be related to each link. Note that a different set of files can be related to each link. This means that some output files of a source application can be used for one target application but not for another one and vise versa.

3.4. Applications and links description generation

In the following two subsections the generation of application specification, links description and the final measurement scenario file are characterized.

The next steps can be performed using the Scenario Submission Application (SSA) based on the information from the previous stages.

DMS is encoded in the Dynamic Measurement Scenario Language (DMSL). DMSL base on the XML and XSD standard.

The general DMS consists of a description of all possible applications with all parameters available for users.

XSD schema

XML Schemas express shared vocabularies and allow machines to carry out rules made by people. They provide means for defining the structure, content and semantics of XML documents. An XML schema is an XML based alternative to DTD (Document Type Definition) and describes the structure, content and semantics of an XML document (a set of simple data types which can be associated with XML element types and attributes).

The purpose of an XML Schema is to define the legal building blocks of an XML document, just like a DTD. An XML Schema:

- defines elements that can appear in a document,
- defines attributes that can appear in a document,
- defines which elements are child elements,
- defines the order of child elements,
- defines the number of child elements,
- defines whether an element is empty or can include text,
- defines data types for elements and attributes,
- defines default and fixed values for elements and attributes.

The XML Schema language is also referred to as the XML Schema Definition (XSD).

The next paragraphs describe the XSD schemas which are used by the VLab System. The XSD schemas have been created for the Dynamic Measurement Scenarios definition.

Components Description Schema

The Components Description Schema (CDS) defines the Java swing components (JComponent) which are used for displaying the resource properties (See RDS description).

Also, a set of component attributes is available. Some of the attributes are required by the component, some are optional. They are all used to define the component behaviour i.e. whether the component should be visible, editable, etc.

Using the schema described above the following Java components have been defined:

- check box (JCheckBox),
- text field (JTextField),
- list (JList),
- combo box (JComboBox),
- text area (JTextArea),
- date field,
- radio button (JRadioButton),
- spinner field (JSpinner).

Links description schema (LDS)

The LDS schema describes the available connections between resources. It also specifies the conditions which have to be taken into consideration while connecting resources.

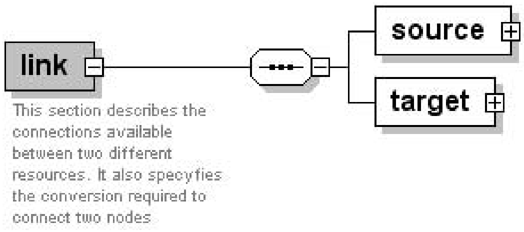
link
This section describes the connections available between two different resources. It also specyfies the conversion required to connect two nodes
source
target

Figure 2. The link element

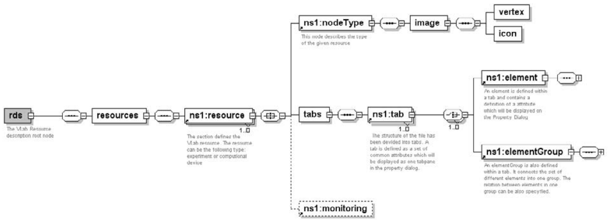
rds
The VLab Resource description root node
resources
ns1:resource
1..∞
The section defines the VLab resource. The resource can be the following type: experiment or computional device
ns1:nodeType
This node describes the type of the given resource
image
vertex
icon
tabs
ns1:tab
1..∞
The structure of the file has been devided into tabs. A tab is defined as a set of common attributes which will be displayed as one tabpane in the property dialog.
1..∞
ns1:element
An element is defined within a tab and contains a definition of a attribute which will be displayed on the Property Dialog
ns1:elementGroup
An elementGroup is also defined within a tab. It connects the set of different elements into one group. The relation between elements in one group can be also specyfied.
ns1:monitoring

Figure 3. Resource description schema

The Link element

The link element, as it is shown in the figure below contains the connection definition between two nodes (Fig. 2).

The connection attributes are presented in the table below.

No	Name	Description
1	id	the link identifier
2	resourceId	component name
3	externalConversion	necessity of external conversion

Resource Description Schema

The Resource Description Schema (RDS) has been created for the VLab resource description. The general resource description structure is presented in the figure below (Fig. 3).

The schema defines a list of the resources which are all attached to the resources node. Every resource element contains the following sections:

- nodeType - defines the type of the given resource, i.e. NMR experiment of the spectrometer; it also defines the type properties which are used in the Scenario Submission application like type icon, resource icon, etc.,

- tabs - the resource properties are grouped in tabbed panes which are displayed in the JTabbedPanes; each tab node contains a set of elements; each element represents a resource property; the elements can also be put together in a group; the relationship between elements in the group can also be defined; an example of the spectrometer properties is shown in the figure below,

- monitoring - this optional section contains information about the given resource in the VLab system.

3.5. Measurement scenario description generation

The user is welcome to create the measurement diagram using the Scenario Submission Application (SSA).

The applications and links definition specified in the previous stage are imported to the SSA. A List of imported applications take into account the user's rights. Thus, it consists of only these applications descriptions (and possible connections) which are available for the user.

After finishing the measurement scenario defined by the user, a new DMS description is generated. It is defined on the basis of the diagram and information added by user. Only the chosen application descriptions with specified parameters are placed there.

4. Connection aspects

The user can define conditions on the connections which can influence the measurement workflow. These conditions concern the computational or experimental results. A set of available conditions to be specified by the user (while a new measurement is defined) is determined on the "describing additional dependencies" stage of the DMS creation process (see 3.3). To enable a given connection all conditions defined there have to be met.

The user is also welcome to define logical conditions between links. There are two types of conditions: *OR* (default) and *AND*. Logical conditions can be defined in the beginning and end of connections. Their meaning depends on their localization.

Beginning conditions

The Beginning conditions are connected with the results achieved at the computational stage. Each link is marked by logical ***1*** if all conditions are met or logical ***0*** otherwise. The default logical operation defined between links is *OR*, which means that each execution path can be done separately. In case of the *AND* operation simultaneously execution of two or more paths is possible on condition that achieving results in the source application are met (value ***1*** on each link).

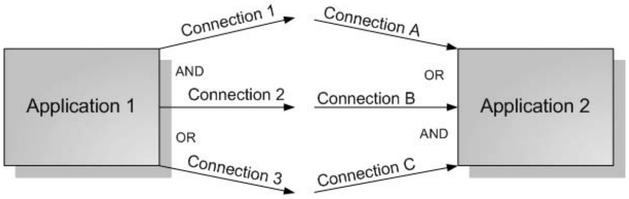

Figure 4. Defining in-connections conditions

Ending conditions

The ending conditions are defined to assure that all necessary data is available before the start of the next processing task.

Similarly as before, for the *OR* condition the link is independent from the other ones. This means that if conditions on a given link are met, the measurement process can go on without waiting for other processes.

The *AND* condition is used when results computation from a few sources is needed. In this way we can force waiting for finishing a number of processes and only then run the next task.

These discussed situation are presented in Fig. 4.

5. Conversion

In this section we want to focus on the conversion issues. As we mentioned before, conversion is made when input and output file application formats are different. We assume that conversion may be done in four different ways:

- setting up the switch in the source application to export data in an appropriate format,
- using a program on the source server to convert the output file (mini postprocessing),
- using a program on the source server to convert the output file (mini preprocessing),
- setting up the switch in the target application to import data in an appropriate format.

The conversion way has to be matched with a given connection. The preferred way is using the application switches because it usually does not cause additional CPU power consuming. Please note that not every conversion way will be possible in each situation.

6. Conclusions

The Dynamic Measurement Scenario is a flexible and very convenient way to define workflow in many types of virtual laboratories. It allows to spare a lot of user's time and speed up of the project realization. Thanks to the Submission Scenario Application the user can easily define, submit and monitor the progress of the DMS realization. In a case of an interactive task realization, it is possible to reserve a time slot on a laboratory device or computational server and notify the user about that. Therefore, the user avoids waiting for free resources. Of course, the scheduling and reservation processes are made by other modules, but on the SE (which executes the DMS) request.

Despite the necessary time to prepare the DMS (application analysing, connection defining, etc.) the Dynamic Measurement Scenario idea seems to be worth being introduced in many systems where the defining complex workflow is needed to preform the science job.

To prove our assumptions, and as an exemplary implementation, it is planned to make the NMR Spectrometer (Bruker Avance 600) and radiotelescope (diameter 32m) available to the Grid users [4].

References

[1] Afsarmanesh, H., Benabdelkader, A., Kaletas, E.C., Garita, C., and Hertzberger, L.O.: Towards a Multilayer Architecture for Scientific Virtual Laboratories. In: Bubak, M., Afsarmanesh, H., Williams, R., Hertberger, B., (Eds.), Proc. Int. Conf. High Performance Computing and Networking, Amsterdam, May 8-10, 2000, Lecture Notes in Computer Science 1823, 162-176, Springer, 2000

[2] Afsarmanesh, H., Belleman, R.G., Belloum, A.S.Z., Benabdelkader, A., van den Brand, J.F.J., Eijkel, G.B., Frenkel, A., Garita, C., Groep, D.L., Heeren, R.M.A., Hendrikse, Z.W., Hertzberger, L.O., Kaandorp, J.A., Kaletas, E.C., Korkhov, V., de Laat, C.T.A.M., Sloot, P.M.A., Vasunin, D., Visser, A., Yakali, H.H.: VLAM-G: A Grid-based virtual laboratory. Scientific Programming, (Special issue on Grid Computing) vol. 10, nr 2 pp. 173-181. (R.H. Perrott and B.K. Szymanski, editors), IOS Press, 2002. ISSN 1058-9244.

[3] Lawenda, M., Meyer, N., Rajtar, T., Okoń, M., Stokłosa, D., Stroiński, M., Popenda, Ł., Gdaniec, Z., Adamiak, R.W.: General Conception of the Virtual Laboratory. International Conference on Computational Science 2004, LNCS 3038, pp. 1013-1016, Cracow, Poland, June 6-9, 2004

[4] Virtual Laboratory project `http://vlab.psnc.pl/`