

Scalable Scientific Workflows Management System SWFMS

M. Abdul Rahman

Department of Computer Science, Sukkur Institute of Business Administration
Sukkur, Pakistan

Abstract—In today's electronic world conducting scientific experiments, especially in natural sciences domain, has become more and more challenging for domain scientists since "science" today has turned out to be more complex due to the two dimensional intricacy; one: assorted as well as complex computational (analytical) applications and two: increasingly large volume as well as heterogeneity of scientific data products processed by these applications. Furthermore, the involvement of increasingly large number of scientific instruments such as sensors and machines makes the scientific data management even more challenging since the data generated from such type of instruments are highly complex. To reduce the amount of complexities in conducting scientific experiments as much as possible, an integrated framework that transparently implements the conceptual separation between both the dimensions is direly needed. In order to facilitate scientific experiments 'workflow' technology has in recent years emerged in scientific disciplines like biology, bioinformatics, geology, environmental science, and eco-informatics. Much more research work has been done to develop the scientific workflow systems. However, our analysis over these existing systems shows that they lack a well-structured conceptual modeling methodology to deal with the two complex dimensions in a transparent manner. This paper presents a scientific workflow framework that properly addresses these two dimensional complexities in a proper manner.

Keywords—Scientific Workflows; Workflow Management System; Reference Architecture

I. INTRODUCTION

Over the last few years, we have witnessed a dramatic change in the way science and engineering has been conducted. In particular, computation became an established third branch of the science alongside theory and experiment. Scientific experiments can be classified into two parts, i.e. dry-lab and wet-lab. Dry-lab refers to the experiments that are conducted through computer-supported and automated computational (analysis) pipelines such as workflows in silico chemistry. Whereas wet-lab experiments attempt to focus on carrying out the experiments that involve manual tasks and human agents, for instance setting up the machines and preparing as well as measuring the samples. However, we use 'scientific experiment' as a comprehensive term which encompasses both parts of the experiments.

As a matter of fact, the management of scientific experiments is two dimensional, i.e. Application Management and Data Management. The former dimension refers to the management of the tasks (work steps) that are specific to the application domain such as collecting, preparing and

measuring the samples, setting up and using scientific machinery and equipment, and performing computations and analysis while latter dimension addresses the tasks (e.g. data provision and preparation) that are solely related to the management of the data products generated from these domain specific applications. In fact, today's complex computational / analytical applications (tools) and heterogeneity of the data raise the intricacy from both dimensions, making the scientific experiments more challenging for domain scientists. Likewise, the proliferation of data generating devices, such as plasma-mass spectrometer in the computational chemistry and sensors in the meteorological research, makes it even more difficult since the data stemming from such type of devices are mostly noisy, inconsistent, rapidly changing, highly heterogeneous and incomplete.

Obviously, handling the Application Management dimension is not a trouble-free and effortless task; however the most challenging dimension is Data Management where domain scientists are in fact uncomfortable. This is mainly due to the following key factors:

Objective and Interest: For domain scientists, the prime objective is to conduct experimental study in order to get analysis and observation results. Thus, they are comparatively more intended towards the experimental tasks (e.g. analysis, computation and observation), rather than handling the data preparation tasks. Generally, the experimental specification is developed after a theoretical study of the domain. Thus it is believed that the scientists are the main community group that is assumed to be the responsible for designing and developing the experimental tasks. Moreover, our real world experience in diverse scientific domains demonstrates that the domain scientists do not seem to be so interested in specifying data management tasks; rather they get the assistance from data experts (other technological experts) for specifying and annotating these kind of tasks. This experience clearly concludes that scientific community group is rather more intended towards the management of the first dimension (Application Management) and shows less interest towards the management of the second dimension (Data Management).

Experience and Knowledge / Familiarity: This is common understanding that designing and developing the experimental steps (first dimension) require in-depth knowledge and experience of scientific domain while designing and developing the data management steps (second dimension) urges the in-depth and extensive knowledge about data related technologies. Scientists are assumed to be the community that has knowledge, experience and expertise about

their scientific domain and hence they better know their experimental tasks and the applications used in these tasks, for instance how to setup the scientific machines, what parameters should be set and how to prepare the samples for observation, how much and what kind of calibration sample should be used and so on. On the other hand, due to their insufficient experience and expertise towards data-related technologies they are experiencing comparatively more difficulties in defining data management tasks.

Due to the two dimensional complexity of scientific experiments, scientists are facing two types of key management challenges, i.e. application specific management and data related management. From the two key factors mentioned above, two important conclusions can be drawn about the relationship of domain scientists with these management challenges. One, their main focus is on handling the first dimension (Application Management) and thus they are comparatively less intended towards the management of the second dimension (Data Management). Two, they are experiencing comparatively more problems in managing the second dimension (Data Management). As a result, handling both the intricate dimensions by scientists, in an unstructured way, results in focus divergence.

Therefore, Independent and separate specification of both application and data management dimensions would help them reduce the workflow complexities. Thus, a mechanism that implements conceptual separation between both management dimensions is direly needed.

II. RELATED WORKS

A scientific Workflow Management System (SWfMS) is the framework that offers the means to completely define, manage, monitor, and execute the scientific experiments in terms of scientific workflows. The design of a generic architecture at an appropriate level of abstraction that properly and transparently addresses the essential requirements for SWfMSs is critical and challenging.

The formal concept of workflow has existed in the business world for a long time. The Workflow Management Coalition (WfMC) [1] has proposed a reference architecture for business workflows. Since then, the reference architecture and its variants [2] have been widely adopted in development of business workflow management systems [3, 4, 5, 6]. However, in [7], authors have convincingly argued that these reference architectures are not suitable for scientific workflow management systems since scientific workflows have different goals.

During the past years, several scientific workflow management frameworks have been emerged [8, 9], which offer fairly much experiences for future research and development. In this section we will report some popular systems and also provide a comparative discussion.

Kepler [10, 11] is one of the popular open source scientific workflow systems with contributors from a range of application-oriented research projects such as Ecology [12], Biology [13], and Geology [14]. Kepler is built on Ptolemy II, a PSE (Problem Solving Environment) from electrical

engineering and thus inherits actor-oriented feature from it. Kepler is dataflow-centric and uses proprietary modeling language so called MoML [15] for workflow specification.

Taverna [16, 17] is also an open source scientific workflow management system like Kepler; it is a part of myGrid project which aims to employ Grid technology to develop high level middleware for supporting personalized in silico experiments [18] in biology. Taverna is implemented as a service-oriented architecture based on Web Service standard, thus the data channel between two services works on SOAP based XML messages.

Triana [19, 20] is an open source workflow based graphical problem solving environment PSE, aiming at defining, analyzing, managing, executing and monitoring the workflows that handles a range of distributed elements such as grid jobs, web services and P2P communication. Although Triana was developed for data-analysis scientists in GEO 600 project, it can be used in many different ways and a rich library of units currently exists covering a broad range of applications.

Pegasus [21] is a framework that can manage the execution of complex scientific workflows on distributed resources. Pegasus is the part of GriPhyN [22] project which aims at supporting large-scale data management in physics experiments such as astronomy, high energy physics, and navigation wave physics. Pegasus enables scientists to design workflows on application level without the need to worry about the actual execution environment. Thus abstract workflows designed by the domain scientists are independent of any resources they will be executed on. Pegasus basically provides the functionality to map the scientific workflows onto distributed resources at a Grid middleware.

ASKALON [23, 24] is a framework developed for Grid application development and computing environment. Its ultimate goal is to simplifying the development and optimization of scientific workflows that can harness the power of Grid computing. In ASKALON, workflows are defined using its property XML-based language known as Abstract Grid Workflow Language AGWL. Like Pegasus, the language enables users to define workflows on abstract level without involving into the middleware complexities and dynamic nature of the Grid. Workflows are composed by using atomic units of works called Activities interconnected through control-flow and data-flow dependencies. The language provides a rich set of constructs to express sequence, parallelism, choices, and iteration workflow structures.

SODIUM [25] (Service Oriented Development In a Unified FraMework) is a platform which provides a set of languages, tools, and corresponding middleware, for modeling and executing scientific workflows composed of heterogeneous services. The system is implemented as Service-oriented architecture and the main objective is to provide seamless access to different types of services such as Web services, Grid Services and P2P services. The overall functionality is achieved by three phases. First, user need to model requirements for services which will satisfy specific workflow task.

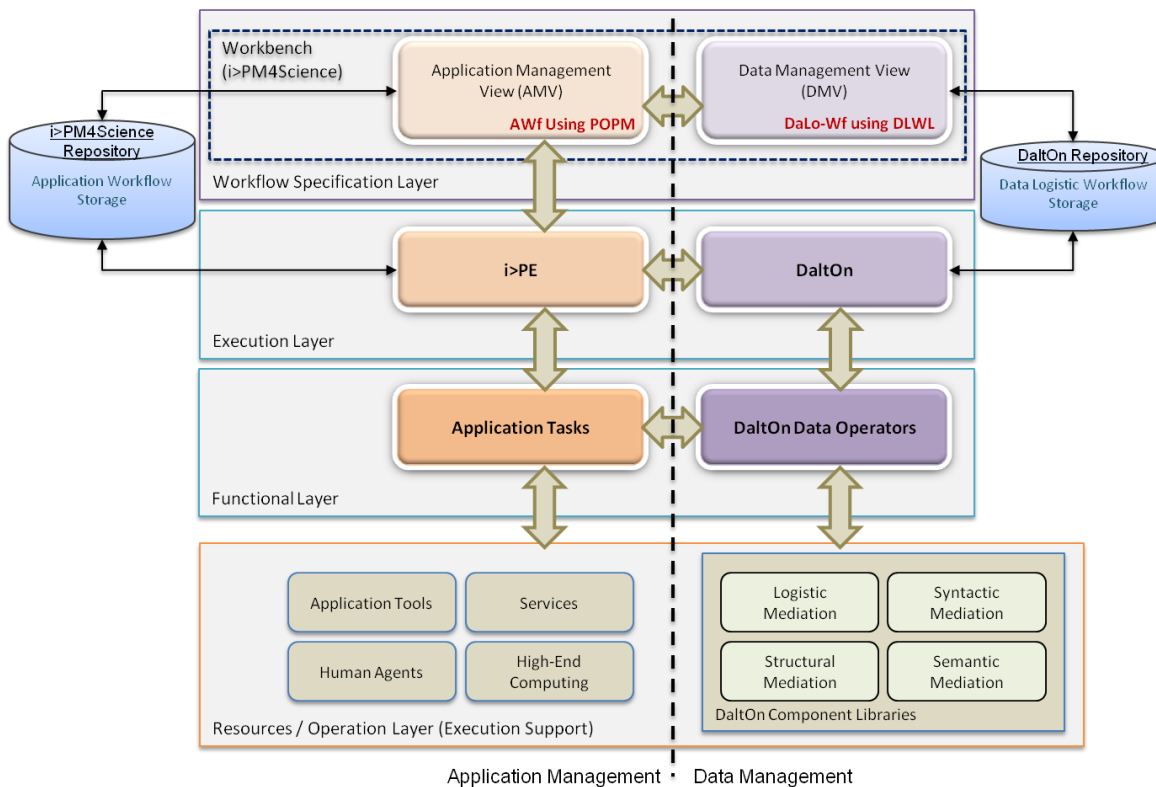


Fig. 1. Architecture of scienceFLOW (a Scientific Workflow Framework)

e-Bioflow [Wass08] is workflow design system which, considering the fact that workflow designers from different domains prefer different perspectives, enables users to model a workflow from three different perspectives: the control flow perspective, the data flow perspective, and the resource perspective. e-Bioflow is inspired by the context of scientific collaborative environment such as e-BioLab and relies on an existing system to have the workflow enacted; workflow models developed by e-Bioflow are enacted by the open-source workflow system Yawl [Van04].

These systems are well developed and are powerful in offering very rich libraries of pre-developed computational components, in executing workflows through the distributed and high computing environments such as Grids and P2P, in managing provenance information, in managing data on grid based technologies, and also in providing many novel and innovative features. However, an architectural reference that can provide a high level management of sub-systems and their interactions in a scientific workflow framework is still an open issue. The current systems have either not an explicit architectural design or the architecture is proprietary and restricted greatly by the legacy system that the frameworks are built upon [7]. For example, Kepler is built on the Ptolemy II, and hence, each new requirement that is needed to be incorporated by the framework is based on the extensions to the architecture of the underlying system, i.e. Ptolemy II. Pegasus, on the other hand, is built upon Condor and Dagman by adding another workflow mapper on the top of these two systems.

III. SCIENTIFIC WORKFLOW FRAMEWORK – SCIENCEFLOW

In order to define workflow specifications of both application and data management dimensions in an independent way, the paper presents an integrated framework, called scienceFLOW [Figure 1] that implements both dimensions in a transparent manner. The framework is composed of four layers having various operational sub-systems at each layer.

A. Layers and Sub-systems

Figure 1 depicts the architectural view of our framework that integrates a number of operational sub-systems. In the following, we will provide a detailed discussion of each operational sub-system on each layer.

Workflow Specification Layer: On this layer, in order to define a scientific workflow we have built two separately graphical sub-systems ‘**Application Management View AMV**’ and ‘**Data Management View DMV**’ for specifying two categories of workflows ‘**Application Workflow AWF**’ and ‘**DataLogistic Workflow DaLo-Wf**’ respectively. The ‘**Workbench**’, known as *i>PM4Science*, of the framework integrates both sub-systems and thus provides two different graphical tabs (Views) [Figure 3] in order to represent them disjointedly. In this way, the workbench promotes two community groups to work together but on their corresponding tabs, i.e. scientists on ‘AMV’ and data experts on ‘DMV’. In the first tab ‘AMV’ [Figure 3(a)], ‘AWF’ is defined using *POPM* notion [29] that allows scientists to express scientific operations on very abstract level without involvement into application and data technicalities.

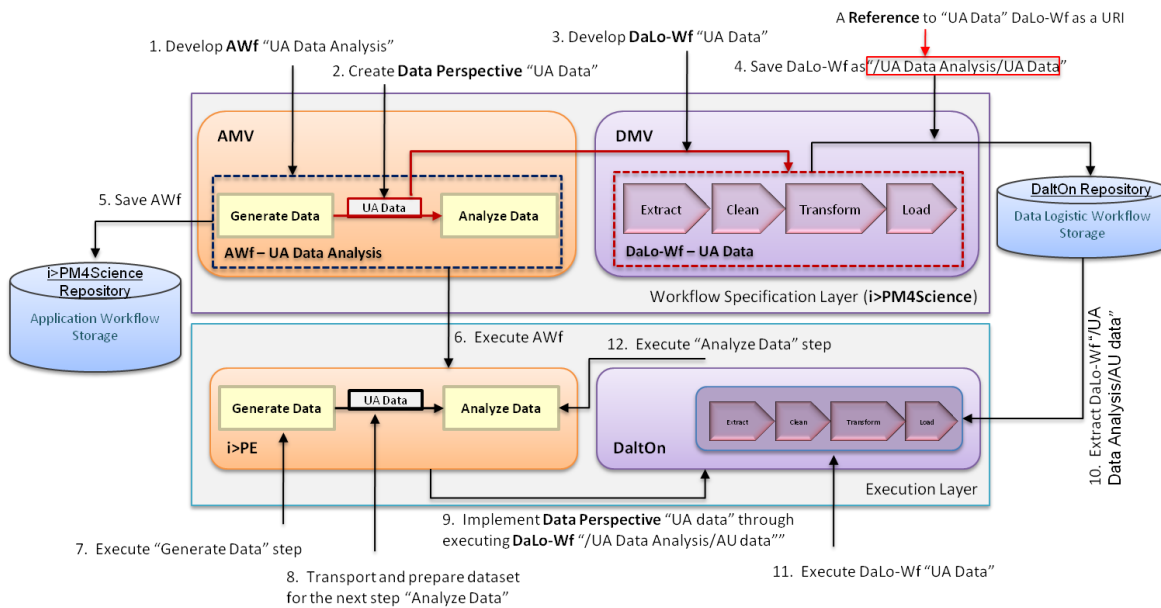


Fig. 2. A Simple Example – showing the course of actions and the flow of Information in the framework

In the second tab ‘DMV’ [Figure 3(b)], for each of ‘Data Perspective’ specified in ‘AWf’ a ‘DaLo-Wf’ is defined using a specialized (data centric) language.

Figure 2 depicts a simple example for developing and executing a scientific workflow through the framework, demonstrating the course of actions and the flow of information throughout the sub-systems in the framework. The course of actions consists of 12 stages and the ‘Workflow Specification Layer’ covers 1 to 6 stages. As an example, a scientist develops a scientific experiment for analyzing weather data (UA dataset) stemming from a sensor device (Ultrasonic Anemometer). At first stage, the scientist develops an ‘AWf’ (named “UA Data Analysis”) using ‘AMV’, including only two experiment related steps ‘Generate Data’ and ‘Analyze Data’. In the second stage, (s)he creates a ‘Data Perspective’ (named “UA Data”). At the third stage, a ‘DaLo-Wf’ (named “UA Data”) is developed - against the created ‘Data Perspective’ - using ‘DMV’, including data related steps ‘Extract’, ‘Clean’, ‘Transform’ and ‘Load’. After successful specification of the ‘DaLo-Wf’, at the fourth stage it is stored into ‘DaltOn Repository’, at the location “/UA Data Analysis/UA Data”. In this way ‘Data Perspectives’ defined into ‘AWf’ do not contain the concrete specification of respective ‘DaLo-Wfs’; rather holds a **Reference** of the location of such workflows. By default the name of ‘DaLo-Wf’ in the ‘DaltOn Repository’ is the same as that of ‘Data Perspective’ in the ‘AWf’ with the contextual path defined in the ‘AWf’, nevertheless users can change it. Moreover, the ‘Data Perspective’ can refer already created ‘DaLo-Wf’ by just annotating it with the reference of the location of a particular workflow. In this way the reusability of ‘DaLo-Wfs’ in multiple application workflows can easily be achieved. Basically, the reference of the location is a URI and thus can be qualified with the address of the system where ‘DaltOn Repository’ resides, for instance “http://132.180.195.110/UA Data Analysis/UA Data”; by default the repository is expected to be at the local system. At the fifth stage the complete specification ‘AWf’ is stored into

the ‘i>PM4Science Repository’ for future use. At the sixth stage the complete ‘AWf’ specification is passed to the execution environment (i<PE) for executing it.

The screenshots of our implemented workbench are shown in Figure 3 where (a) represents ‘AMV’ and (b) depicts ‘DMV’. Since all the ‘DaLo-Wfs’ are stored into ‘DaltOn Repository’, ‘DMV’ also provides the ability to search - from the repository - already defined ‘DaLo-Wfs’. ‘DMV’ fundamentally consists of three elements, i.e. design panel (right) for defining ‘DaLo-Wf’ using DLWL notion, Data Perspective explorer panel (left) for browsing through ‘DaLo-Wfs’, and search panel (upper left) for searching previously defined ‘DaLo-Wfs’.

Execution Layer: At this layer two dedicated sub-systems are employed, i.e. ‘i>PE’ [30] and ‘DaltOn’ [31] for executing both categories of workflows separately and independently. The sub-system ‘i>PE’ (integrated Process Executor) is a process-centric workflow engine founded on POPM paradigm, that makes available the environment for executing ‘AWf’ specified via POPM notion. During the course of ‘AWf’ execution, whenever the engine comes across ‘Data Perspective’ between two work steps it invokes the contiguous sub-system, i.e. ‘DaltOn’, in order to implement data perspective under the focus. The sub-system ‘DaltOn’ is a data processing system which is specifically designed to provide the data management mediation to the scientific workflows by implementing data management part (Data Perspective).

The Figure 2 (lower layer) demonstrates the course of actions and the information flow through the execution sub-systems employed at this layer. The course of actions at this layer encircles 6 stages (from 7 to 12). At the seventh stage, the execution engine ‘i>PE’ starts executing the ‘AWf’ through the invocation of the first work step ‘Generate UA Data’ that aims at extracting the dataset from sensor device. At the eighth stage, the engine flags the next step ‘Analyze Data’ as an “executable” and identifies the data transportation and preparation task for the step. Then at the ninth stage, in order to

implement the 'Data Perspective' (i.e. "UA Data") the engine invokes and requests the 'DaltOn' to execute the respective 'DaLo-Wf', by passing the location reference of the workflow (i.e. "/UA Data Analysis/UA Data"). After getting the execution request from 'i>PE', the 'DaltOn' extracts the corresponding 'DaLo-Wf' (graph of data processing tasks) from 'DaltOn Repository' at the tenth stage and implements

the 'Data Perspective' (i.e. "UA Data") by executing the respective 'DaLo-Wf' at the eleventh stage. In this way 'DaltOn Repository' plays an interface role between 'DMV' and 'DaltOn' sub-systems. Finally at the twelfth stage, 'i>PE' executes the last step (i.e. "Analyze Data") in the example 'AWf'.

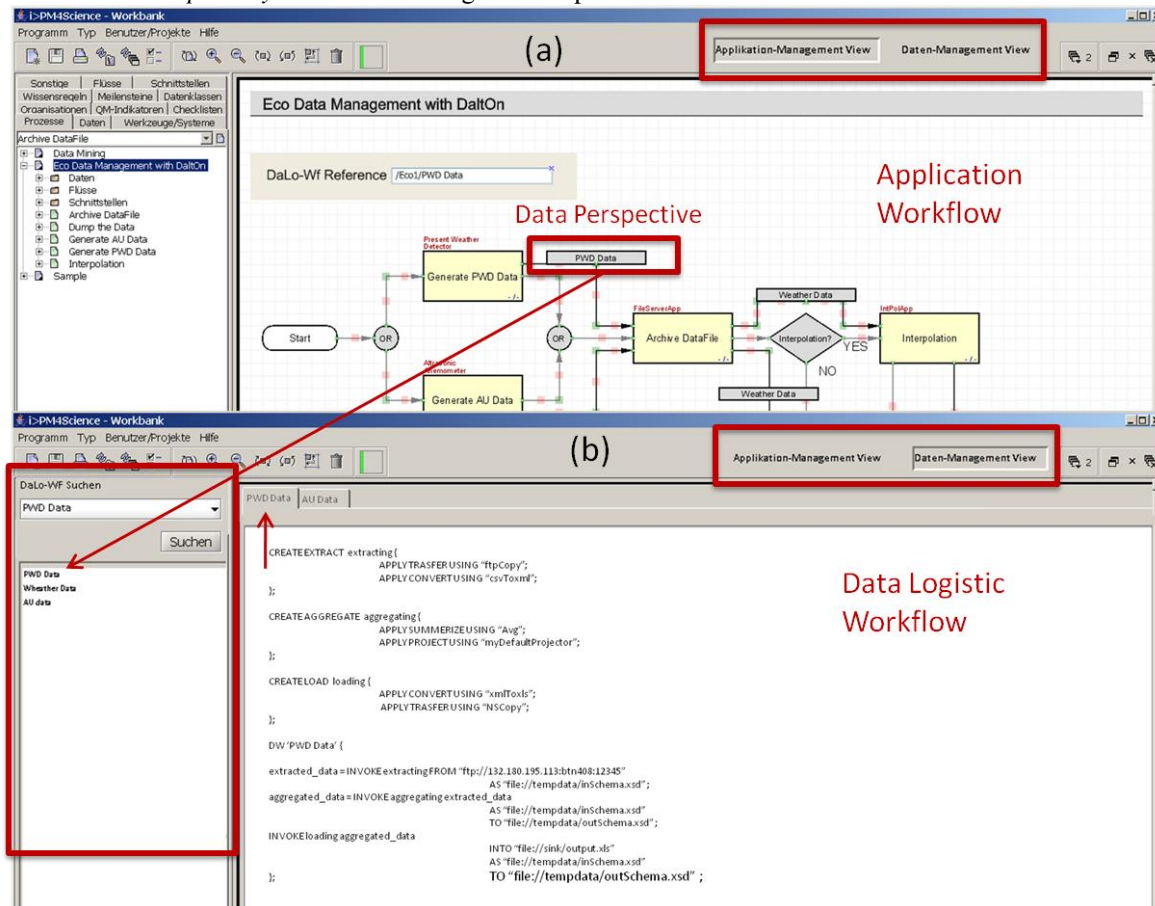


Fig. 3. i>PM4Science Workbench – (a) Application Management View - AMV, (b) Data Management View - DMV

Functional Layer: On this layer, in order to provide a generic execution support to both sub-systems on 'Execution' layer we built up two sub-systems, i.e. 'Application Tasks' and 'DaltOn Data Operators'. Basically, both the sub-systems constitute the libraries for different kind of logical operations (tasks) with the purpose of supplying execution entities to the execution engines.

'Application Tasks' sub-system aims at providing the library of commonly used experiment related tasks that play the role of building blocks for 'AWf' development. The sub-system also supports some features that are required for managing and maintaining a component library, for instance adding, removing and editing the experimental tasks. The tasks provided by such sub-system are completely abstract entities, thus the physical implementation of each of these tasks is realized by the underlying concrete functions or applications. Therefore, a single task can be utilized with multiple implementations in a generic way.

'DaltOn Data Operators' aims at constituting the library of the most common data operations that play the role of basic building blocks for 'DaLo-Wf' development. Like above sub-system, it also supports some basic features that are needed for managing and maintaining a component library. Data operators offered by this sub-system are logical entities, thus the physical implementation of each of these operators is realized by the underlying concrete functions. Therefore, a single operator can be utilized with multiple implementations in a generic way. For instance, the operator 'Convert' reflects an abstract operation that can be utilized in a number of implementations such as 'csv2xml', 'ua2xml', and 'xml2xls' - by just providing a specific low level function.

Resource / Operational Layer: Fundamentally, this layer aims at supplying physical resources to the upper layer in order to implement the logical and abstract experimental tasks as well as data operators. In order to support experimental tasks we maintain the libraries of application / software tools, services such as Web or Grid services, human agents. In order

to support the data operators we implemented comprehensive and classified component libraries of diverse functions.

IV. CONCLUSION

The paper presented a generic and scalable scientific workflow framework '*scienceFLOW*', whose originality is a clear isolation of two management concerns, i.e. application and data management. The design solution of the framework was motivated by two factors, i.e. the identified requirements and the method for e-Science. The basic 'requirement' (i.e. application specific and data related issues should be handled entirely in a separate manner) is nicely fostered in the framework since the whole framework is divided into two segments, i.e. the application management and the data management. In order to manage the issues occurring in both segments dedicated sub-systems are employed not only at the design level but also at the execution level.

REFERENCES

- [1] Workflow Management Coalition: "The Workflow Management Coalition (Homepage)". <http://www.wfmc.org>, [Oct 2016]
- [2] Grefen, P. and de Vries, R.: A Reference Architecture for Workflow Management Systems," *Data Knowledge Eng.*, vol. 27, no. 1, pp. 31-57, 1998
- [3] YAWL: Yet Another Workflow Language: "YAWL Foundation (Homepage)". <http://www.yawlfoundation.org/> [Oct 2016]
- [4] van der Aalst, W., Aldred, L., Dumas, M. and ter Hofstede, A.: Design and Implementation of the YAWL System, *Proc. Center for Advancement of Informal Science Education Conf. (CAiSE '04)*, pp. 142-159, 2004
- [5] Wil M. P. van der Aalst: *Business Process Management: A Comprehensive Survey*, Hindawi Publishing Corporation, ISRN Software Engineering, Volume 2013
- [6] Ming Gao, Lei Yang, Chunhua Zhang, Wen Guan and Ailing Li: Towards Unified Business Process Modeling and Verification for Role-based Resource-oriented Service Composition, *International Journal of Hybrid Information Technology* Vol.9, No.3 (2016), pp. 145-158, 2016
- [7] Lin, C.; Lu, S.; Chebotko, A.; Pai, D.; Lai, Z.; Fotouhi, F. and Hua, J.: A Reference Architecture for Scientific Workflow Management Systems and the VIEW SOA Solution. *IEEE Transactions on Services Computing*, Vol. 2(No. 1): p. 79-92, 2009
- [8] AL Lemos, F Daniel, B Benatallah: *Web Service Composition: A Survey of Techniques and Tools*, *ACM Computing Surveys*, Vol. 48, No. 3, Article 33, December 2015
- [9] Scientific Workflow Survey (home page), <http://www.extreme.indiana.edu/swf-survey/>, [Oct 2016]
- [10] Daniel Crawl, Alok Singh, and Ilkay Altintas: Kepler WebView: A Lightweight, Portable Framework for Constructing Real-time Web Interfaces of Scientific Workflows, *Procedia Computer Science*, Volume 80, Pages 673–679, 2016
- [11] Ludäscher, B.; Altintas, I.; Berkley, C.; Higgins, D.; Jaeger-Frank, E.; Jones, M.; Lee, E.; Tao, J.; Zhao, Y.: *Scientific Workflow Management and the Kepler System*. In *Concurrency and Computation: Practice & Experience*, Vol. 18 No. 10, Citeseer, 2006
- [12] SEEK: Science Environment for Ecological Knowledge (home page), www.seek.ecoinformatics.org, [Oct 2016]
- [13] Clotho: Design environment for synthetic biological systems (home page), <http://www.clothocad.org/>, [Oct 2016]
- [14] GEON: The NSF Information Technology Research (ITR) program, <http://www.geongrid.org>, [Oct 2016]
- [15] Lee, E. A. and Neuendorffer, S.: *MoML — A Modeling Markup Language in XML — Version 0.4*, Technical Memorandum UCB/ERL M00/12, University of California, Berkeley, CA 94720, March 14, 2000
- [16] Katherine Wolstencroft1, Robert Haines, Donal Fellows: *The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud*, *Nucleic Acids Research Advance Access*, Oxford University Press, May 2013
- [17] Taverna: An open source scientific workflow system (home page), <http://www.taverna.org.uk>, [Oct 2016]
- [18] Oinn, T.; Addis, M.; Ferris, J.; Marvin, D.; Greenwood, M.; Goble, C.; Wipat, A.; Li, P.; Carver, T.: Delivering web service coordination capability to users, *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, p. 438-439, ACM New York, NY, USA, 2004
- [19] Taylor, I. J.; Deelman, E.; Gannon, D. B.; Shields, M.: *Workflows for e-Science: Scientific Workflows for Grids*, Springer-Verlag London Limited, 2007
- [20] Triana - Open Source Problem Solving Software (home page), www.trianacode.org [Oct 2016]
- [21] Ewa Deelman, Karan Vahia, Gideon Juve: *Pegasus, a Workflow Management System for Science Automation, Future Generation of Computer Systems*, Elsevier, Volume 46, Pages 17–35, May 2015
- [22] Grid Physics Network (home page): <http://www.usatlas.bnl.gov/computing/grid/griphyn>, [Oct 2016]
- [23] Fahringer, T: *ASKALON Grid Environment, User Guide Version 1.2, Distributed and Parallel Systems Group, Institute of Computer Science, University of Innsbruck*, September 2015
- [24] Fahringer, T.; Prodan, R.; Duan, R.; Hofer, J.; Nadeem, F.; Nerieri, F.; Podlipnig, S.; Qin, J.; Siddiqui, M.; Truong, H. L.: *Askalon: A development and grid computing environment for scientific workflows, Workflows for eScience, Scientific Workflows for Grids*, Springer, 2007
- [25] Tsalgatidou, A.; Athanasopoulos, G.; Pantazoglou, M.; Pautasso, C.; Heinis, T.; Gronmo, R.; Hoff, H.; Berre, A. J.; Glittum, M.; Topouzidou, S.: *Developing scientific workflows from heterogeneous services*, *ACM Sigmod Record*, Vol 35, No. 2, ACM, 2006
- [26] Wassink, I.; Rauwerda, H.; van der Vet, P. E.; Breit, T.; Nijholt, A.; Elloumi, M.; Küng, J.; Linial, M.; Murphy, R. F.; Schneider, K.: *e-BioFlow: Different perspectives on scientific workflows*, 2nd International Conference on Bioinformatics Research and Development (BIRD), Springer, 2008
- [27] The First International Workshop on Data and Process Provenance WDPP (homepage <http://www.itee.uq.edu.au/~dasfaa/workshop/wdpp/WDP09.htm>), 2009
- [28] Greenwood, M.; Goble, C.; Stevens, R.; Zhao, J.; Addis, M.; Marvin, D.; Moreau, L.; Oinn, T.: *Provenance of e-Science experiments - experience from bioinformatics*, In Cox, S., ed.: *Proceedings of UK e-Science All Hands Meeting 2003*, Nottingham, UK, 2003
- [29] Jablonski, S., Bussler, C.: *Workflow management. Modeling concepts, architecture and implementation*, Thomson, London, UK, 1996
- [30] Jablonski, S.; Faerber, M.; Götz, M.; Volz, B.; Dornstauder, S.; Müller, S.: *Configurable Execution Environments for Medical Processes*, 4th Int'l Conf. on BPM, BPM Demo Session 2006, Vienna, 2006
- [31] Jablonski, S.; Curé, O.; Rehman, M. A.; Volz, B.: *Architecture of the DaltOn Data Integration System for Scientific Applications*, *Proceedings of the 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid*, IEEE Computer Society, 2008.