

WestminsterResearch

<http://www.westminster.ac.uk/westminsterresearch>

A model for information and action flows connecting science gateways to distributed computing infrastructures

Pierantoni, G., Frost, G., Gesing, S., Olabbarriaga, S., Jaghoori, M., Terstyanski, G. and Arshad, J.

Paper presented at the 8th International Workshop on Science Gateways (IWSG 2016), Rome, Italy, June 8-10, 2016, and published in the CEUR Workshop Proceedings, volume 1871. Available at:

<http://ceur-ws.org/Vol-1871/paper1.pdf>

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners.

Whilst further distribution of specific materials from within this archive is forbidden, you may freely distribute the URL of WestminsterResearch: (<http://westminsterresearch.wmin.ac.uk/>).

In case of abuse or copyright appearing without permission e-mail repository@westminster.ac.uk

A Model for Information and Action Flows Connecting Science Gateways to Distributed Computing Infrastructures

G. Pierantoni^{*}, D. Frost^{*}, S. Gesing[†], S. Olabarriaga[‡], M. Jaghoori[‡], G. Terstyanszky[§] and J. Arshaid[§]

^{*} Trinity Center for High Performance Computing, Trinity College Dublin, Dublin, Ireland

Email: pierantg@tcd.ie

[†] Center for Research Computing, Department of Computer Science and Engineering, University of Notre Dame, IN, USA

[‡] Academic Medical Center of the University of Amsterdam, The Netherlands

[§] Center for Parallel Computing, University of Westminster, London, United Kingdom

Abstract—To support scientists of different disciplines, different fields of Computer Science have developed tools and infrastructures with the aim of giving them access to vast computational resources in the easiest possible way. Such extremely complex structures have evolved naturally in the last decades both in depth and breath and, in addition to scientists, a plethora of heterogeneous actors (system administrators, developers, etc.) cooperate and interact with them. This complex and unstructured flow of actions and information poses difficulties in the development and usage of Science Gateways because information can be missing or hard to isolate at the right layer. In this paper, we aim to start a discussion on how to best manage these information flows to help the design and implementation of more flexible and user-friendly Science Gateways and workflow management systems in the future.

Index Terms—Workflows, eScience Portals, eInfrastructures, Science Gateways, Information Flows, Interoperability, Distributed Computing Infrastructures, Workflow Management Systems

I. INTRODUCTION

In modern days, science relies on computation to such an extent that the term *in silico* has been added to the terms *in vivo* and *in vitro*. To support scientists of different disciplines in accessing computational resources that are ever growing in size and complexity, different fields of Computational Science have developed tools and infrastructures that fall under the broad definition of *Science Gateways* (SGs) [1]–[3]. Science Gateways lie on the top of extremely complex systems and services that have evolved naturally in the last decades both in depth and breath. They span multiple layers specialized in tackling specific facets of the challenge and different communities have developed independent implementations for each layer. Furthermore, in addition to scientists, a plethora of heterogeneous actors (system administrators, developers, etc.) [4] cooperate with the scientists and interact with the infrastructure. The challenges are enormous to make all systems and persons communicate and interoperate.

The challenges posed by the need to harness distributed computing infrastructures (DCI) that vary greatly in their implementation, such as Clouds, Grids, Desktop Grids and High Performance Computing, have been at the center of

many successful efforts [3], [5]–[8]. These resulted in the construction of abstraction layers capable of interfacing with heterogeneous, distributed systems in a unified fashion.

The need to formalize and share the scientific process have also been satisfied by different scientific communities by adopting the workflows concept originally developed for the industry. Several such workflows [9]–[12] have been developed and have been adopted by different scientific communities, giving raise to the same interoperability problem as found in distributed computing. The workflows interoperability problem [13]–[15] has been addressed by building abstraction layers and intermediate languages. Nevertheless, while such efforts aim at a relative degree of freedom and interoperability across different Workflow Management Systems, they also increase the complexity of the information flows.

All these layers, of infrastructure and workflows, are connected to each other by flows of requests and replies that are unstructured and heterogeneous by nature. Requests propagate downward from the upper to the lower layers, while replies propagate upward from the lower to the upper layers and eventually reach the users who originated them. Replies carry information on the status and on the outcome of the request (often merged together), offering to the upper layer a partial view on the overall information of the lower layer.

These multi-layered infrastructures are used by a plethora of actors with different skills [4], inclinations and priorities, which increase the complexity to a higher level. Administrators, developers, and scientists, all of them interact with one or more layers, and each of them is interested in a subset of the information flow in each layer with which she/he is likely to be best acquainted.

The complexity of these information flows poses relevant difficulties in the development and usage of Science Gateways, as information can be missing or hard to isolate at the right layer. This is true both for scientific users and administrators (e.g. error messages can be absent or difficult to understand), but also for developers as it is difficult to build systems that autonomously react to undesired events, and to dispatch the right information type to the proper users.

In this paper we start a discussion on how to best manage these information flows to help the design and implementation of more flexible and user-friendly Science Gateways and Workflow Management Systems in the future. The first step of this discussion is to propose a model to describe these information flows and the architecture that hosts them. The domain we are attempting to model is extremely vast and diverse, so we start by analyzing a sub-domain encompassing solutions with which the authors are well acquainted. Also, this paper proposes a qualitative approach without any formal description that will be attempted after the initial model has been validated. The overall goal is to increase the usability for the diverse user groups of Science Gateway systems. Our model considers standards, setting the context and suggesting methods for measuring user experience. Such standards include CISU-R (Common Industry Specification for Usability Requirements) [16] developed by the Visualization and Usability Group within NIST (National Institute of Standards and Technology) [17]. We start with an initial level with an expert evaluation [18], which is based on our own knowledge, experience and use cases. Since we are developers, providers and also users of science gateways and workflow systems, such an expert evaluation covers already a broad view.

The paper is structured as follows: Section II introduces preliminary concepts and terminology, and Section III introduces the model to describe the information flows. Section IV describes some currently used tools and technologies and Section V discusses the road ahead.

mds

May 10, 2016

II. CONCEPTS AND TERMINOLOGY

The proposed model is based on the following assumptions and concepts. Firstly, we assume that Science Gateways are composed of several *Layers*. We define a *Layer* as an entity that represents an element of a Science Gateway. A *Layer* can have different implementations and it exposes a well-defined set of functionalities to its users. Example of Layers are Presentation and Service Layers, Distributed Computing Infrastructure Layers (e.g. Grids and Clouds), Workflow Layers (e.g. TAVERNA or WS-PGRADE). Each Layer is composed of various *Elements* and is described by its *Status*

Elements of each Layer fall into two main categories: structural and transient. *Structural Elements* are static entities that deliver functions and services inside the Layer (e.g. the Job Execution Service or Information Service in a DCI). *Transient Elements* are dynamic entities created by the user inside the Layer (e.g. jobs description, files, workflows) to run a specific application. Each Layer can be accessed by *Access Components*, which are entities that enable access. Access Components can be user-oriented such as Graphical User Interfaces and Command Line Tools or programmatic interfaces such as an API.

Layers communicate through the means of *Requests* and *Replies* and are defined by the status of their Structural and Transient Elements. We define their combination as its

Information Domain; we also define *Information flows* as the exchange of information between Elements in different Layers.

The information flows that describe part of the Information Domains to the upper layers have different characteristics and related challenges. We focus here on three such challenges: *Heterogeneous information* represents the challenge of high utility information mixed with information of less relevance. *Incomplete information* represents the challenge that arises when Users cannot directly access all the required information. Finally, the problem of *Information interoperability* arises whenever different implementation of the same Layer impose the use of different languages and interaction patterns to perform the same action.

Also, from each user's perspective, information flows may be more or less useful and more or less easy to manipulate. To describe this, we introduce the concepts of Utility, Cost and Value. *Utility* defines the usefulness to the user, *Cost* describes the difficulty to obtain the information and *Value* represents the difference between the two.

To increase the overall Value of the information, we have observed that the scientific communities have devised different systems. *Heterogeneity Reduction Functions* do not modify the Utility of information but reduce the Cost associated to their fruition. *Information Extension Functions* increment the Utility of information while maintaining its Cost fixed. *Interoperability Functions* offer a unified interface to multiple implementations of the same Layer.

III. A MODEL FOR INFORMATION FLOWS

The domain we attempt to model spans multiple layers and many different implementations for each layer, therefore it is therefore arduous to draw a conclusive and exhaustive schema. Nevertheless, we observed some recurring architectural patterns that suggest to adopt an abstraction encompassing four main layers:

- *Scientific Domain* layer for interaction with the scientific user using domain concepts.
- *Generic Portal* layer for interaction with generic users and to offer tools and APIs to build the applications of the above layer.
- *Workflow Management* layer, where the processing orchestration is described and executed.
- *Distributed Infrastructures* used for computing, storage and data, which are normally represented by one or more DCIs.

To model how the different layers and actors interact through Information Flows, we try to simplify such a complex system and then adapt step by step the model to the complexity of real systems.

The layers are examined under the consideration of CISU-R, which defines three levels of compliance for usability. Level 1: Context of use must consider individually:

- The stakeholders.
- The intended user groups.
- The main goals for each user group.

- The intended computing or technical environment.
- The intended physical and social environments.
- Scenarios of use specifying tasks in context.
- Any prerequisite documentation/training materials.

Level 2: Measures must include:

- Performance measures, i.e. achieving user goals.
- Satisfaction measures via known questionnaires.

Level 3: The test method specifies how it is planned to evaluate that the requirements are met.

The model focuses at this stage on setting the context regarding level 1: from stakeholders (science gateway and workflow management providers), intended user groups (e.g., domain scientists, administrators) and main goals for each group. The computing and technical environment as well as scenarios are analyzed via case studies. Keeping the model as generic as possible, we aim to incorporate and apply it for diverse physical and social environments.

A. The Model of one Layer

Here we model a single layer of the full stack and its interactions with the users. It is important to stress that this does not attempt to model an entire stack as a Single Layer but rather to model a generic layer of the full stack.

Figure 1 presents a simple model of a generic layer of the stack, which could be used to model a portal, a workflow submission system or a DCI middleware.

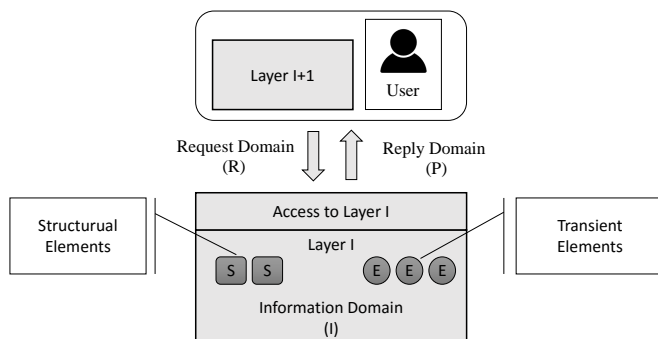


Fig. 1. Model of a Single Layer

The proposed model comprises the following entities:

- *Layer I* describes a generic Layer in the structure such as a portal, a Workflow Management System, or DCI). Layer I will be described by its *Information Domain* that includes its status. The description of the status has to take into account the dual, interconnected nature of the Layer: that of its own structure and that of the actions it is performing. An example of this is the possibility of a job to fail because of the inconsistent status of the DCI or because of a failure of the job itself.
- *Access to Layer I* is an Access Component that models APIs for programmatic access as well as command line and graphical user interfaces for direct human interaction. Access Components can restrict access to the Layer depending on Authorization policies.

- *Structural Elements* model the internal components of the layer.
- *Transient Elements* model the objects defined by the user that are currently handled by the layer (e.g jobs or Workflows being managed).

External to the layer, there are either human users or other programmatic entities that connect to it. We model the interactions between these entities as represented in Figure 1 by employing the concepts of *Layers*, *Requests*, *Replies*, *Structural Elements*, and *Transient Elements*.

Layer I+1 issues Requests (possibly involving Transient Elements) to the Layer I and obtains Replies in return. It is important to highlight that an action may modify the status of both Structural and Transient Elements of the layer, but the ending status of these entities does not strictly follow the ones that preceded the action (as other events may have occurred while the action was executed). Requests are detailed by parameters that may include Transient Elements or references to them. As an example, the submission of a job to a DCI can be modeled as a Request of a submission action of a Transient Element describing a job that will take as parameters the job description itself, additional parameters and details of the identity of the entity submitting the job. Replies include different, heterogeneous elements such as an exit/error code, job results and logging information.

We also model the situation when users may not be able allowed to issue all Requests to a layer and that they may not be able to directly access the entire set of the Information Domain of the Layer. This can be the case of Authorization policies. We define the subset of the Information Domain accessible by each user as being *Directly Accessible*.

Finally, we model the different profiles of actors connecting to the layer through three main profiles: *Result-Oriented*, *Layer-Oriented*, and *Development-Oriented* actors. Henceforth we will refer to all actors accessing Layer I as *Users* encompassing in this generic term both human users and programmatic components. In any case, even software components will have to be executed with a certain identity either by delegation, robot certificates or other means. *Result-oriented* users model actors whose main interest is in the results provided by the layer. They want to be shielded as much as possible from the technical details of the layer. Ideally, a Result-Oriented User would like to treat the entire layer as a black box that would either return the results correctly or, in case of failure, deliver within expected time the result along with a contact point for addressing the issue. Since jobs or tasks can be active over long periods of time, it is important to provide and visualize information for monitoring active jobs. *Layer-oriented* users model actors that have an opposite view. They are interested in the internals and status of the layer, which should be seen as a transparent box allowing complete access and manipulation of the inner workings. They are mainly concerned with the maintenance of the Structural Elements of the layer. Such users include Workflow Management and DCI Providers, who need detailed information optimization and error resolution. *Development-Oriented* users

model actors whose main focus is the development of either the Structural Elements that compose layers or Transient Elements such as Workflows on behalf of Result-Oriented users.

We also have to model the fact that layers have multiple implementations as presented in 2. In this case *Layer I+1* triggers actions and receives results from two separate implementations of the same model. As there is no explicit interoperability provision between the two implementations, *Layer I+1* will have to support two separate access modalities (syntax that defines Requests, Transient Elements, Reply formats, etc...) by explicitly dealing with two separate Requests, Replies and Information Domains.

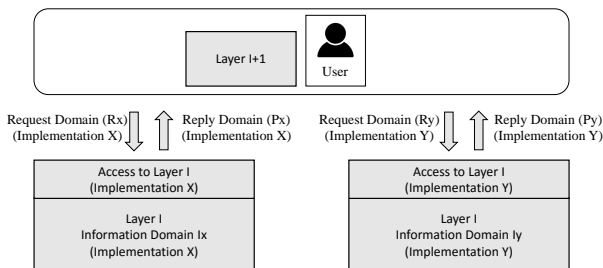


Fig. 2. Model of a Single Layer with Multiple Implementations

B. Information Value

The concepts of *Value*, *Utility* and *Cost* are fundamental in this model, which can be expressed differently for each of the user Profiles. Utility describes how useful the information contained within a Reply is to any particular user profile. The Cost describes how difficult it is to obtain that information, thus covering both the action of extracting information from the Reply and issuing the related Request (e.g. extracting the relevant information about the failure of a Workflows executed on multiple DCIs may be very hard to perform). Finally Value describes the difference between the Utility and the Cost.

Utility, Cost and Value need to be quantified for usage in the proposed model, which is still topic of ongoing debate. A possibility would be to use a real numbers in the range from 0 to 1, where 0 represents low scores. For example, a Request returning useless information (Utility=0) that is very difficult to understand for a particular user profile (Cost=1) would have a Value of -1. Another Request returning useful information (Utility=1) that is very easy to understand (Cost=0) would have a Value of +1.

C. Value Increasing Functions

There are three main characteristics that reduce the Value to different user profiles. Firstly the information that has the highest Utility to different User profiles is often mixed with information that has less Utility. We refer to this problem as that of *Heterogeneous Information*. Secondly, the required information may not be directly reachable by a user, a problem that we refer to as *Incomplete Information*. Finally, different implementations of the same layers impose the use of different

languages and interaction patterns to perform the same action (e.g. the execution of a workflows), which we coin *Information Interoperability*.

Science Gateway developers have devised different solutions to these problems which we attempt to model as either a Structural Element-Value Increasing Structural Element (VISE), or as a smart Transient Element - Value Increasing Transient Element (VITE). VISE's are usually result of the effort of Layer-Oriented Users that modify Structural Elements of one Layer to increase its usability by one or more Users. VISE's are usually the result of efforts by Development-Oriented or Result-Oriented Users that modify job descriptions or workflows to increase the usability of one Layer, for example jobs or Workflows that internally manage information flows and/or automatically perform value-adding actions.

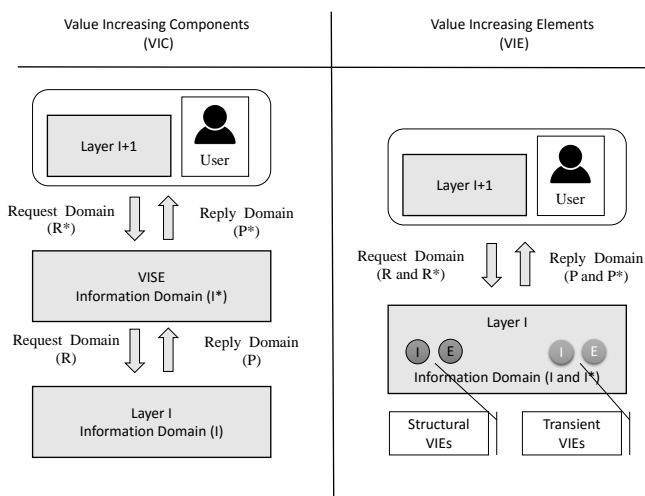


Fig. 3. Value Increasing Components and Elements

We envisage four main types of operations to increase the value of Information Flows, the so called .

1) *Heterogeneity Reduction Function*: They are filtering functions that isolate sub-set of a Reply in order to make it more accessible to different user profiles. An example would be a function that filters job results, error codes and logging information. Result-Oriented Users will see only the job result, if any, while Layer-Oriented Users will see only error codes and logging information, if directly available. These functions do not increase *Utility* but reduce the *Cost* thus increasing *Value*.

2) *Information Extension Function*: The second type extends the information domain directly available to the user. An example would be a an automatic operation that automatically retrieves information on the status of the Layer (e.g. retrieval and parsing of log files) on the failure of a job. These functions do not modify the status of the Transient Elements in the layer, but the Information Domain made available to the user has a greater *Utility* and the same *Cost* thus resulting in increased

Value¹.

3) *Compound Actions*: The third type of functions, coined *Compound Actions*, can perform a variety of actions that can modify the status of the Transient Elements in the Layer. An example would be the execution of pilot jobs prior to the submission of the real job to foresee problems in the layer and/or the automatic execution of diagnostic jobs and routines on the failure of jobs.

4) *Interoperability Actions and Functions*: The final type of functions, which we coin *Interoperability Functions* offer a unified interface to multiple implementations of the same layer. Examples are the submission to multiple DCIs infrastructures by a single set of commands or the possibility to execute a workflow written for one Workflow Management System on a different system. These functions offer the combined Utility of several implementation while requesting the same Cost of one implementation, thus radically increasing the overall Value

D. Interactions between VISEs, VITEs and Layers

We argue that there are five different ways in which Layers and VISEs can interact. We also define a set of *Improved Requests*, *Improved Replies* and *Improved Information Domains* provided by the VISE.

- The first solution is to use a VISE as an additional, separate Layer as presented in the left part of Figure 4. In this case the upper Layer can access both Improved and Original sets of Requests and Replies. This arrangement supports VISEs that offer abstractions of different implementations of the same Layer allowing for interoperability. Examples are a VISE that optimizes the Information Flow of different DCIs or a VISE that optimizes the information flows from different Workflow Management Systems
- The central part of Figure 4 shows a VISE embedded within the access component of Layer I
- The right part of Figure 4 shows an opposite solution where a VISE is embedded in Layer I+1 that uses it in a totally transparent way
- Figure 5 shows a *Multi-Layer VISE* that connects to different layers on the left and a *Multi-Layer Multi-Implementation VISE* on the right. A multi-layer VISE has the advantage of offering improved Requests and Replies that combine the Information Domains of multiple layers. A Multi-Layer, Multi-Implementation VISE extends the functionalities of a Multi-Layer VISE across different implementations of the same layer.

IV. EXAMPLES

The examples presented here represent the experience from a large of Science Gateway-oriented solutions of knowledge to the authors.

Interoperability has been tackled at both the Workflow and DCI layers within a set of related projects (SCI-BUS [19],

¹The status of the Layer and its Transient Elements can be modified in subtle ways that are not covered at the moment by our model

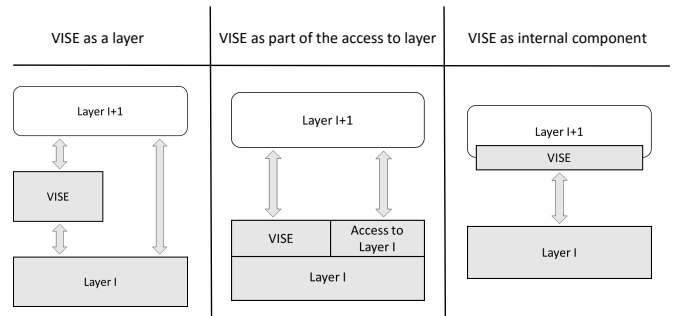


Fig. 4. Different ways to connect Layers with Value Increasing Structural Elements

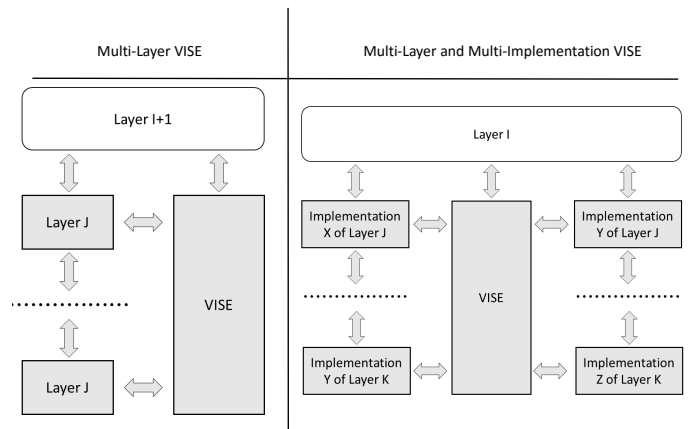


Fig. 5. Multi-Layer and Multi-Implementation VISEs

SHIWA [20] and ER-FLOW [21]). These projects brought together domain experts and technology providers to work on a platform centered on the gUSE/WS-PGrade technology [22] to lower the complexity in the use of multi-layered infrastructures and many of these solutions can be seen as Value Increasing Structural Elements (VISE's) and Value Increasing Transient Elements (VITEs). From a broad perspective, the entire gUSE/WS-PGrade/SHIWA suite of components is a framework that can be used directly as a general Science Gateway to serve multiple scientific communities (Layer-Oriented and Result-Oriented users) or can be used by Developer-Oriented Users to create Customized Gateways that act as Value Increasing Components built in the topmost layer for Result-Oriented Users of specific communities.

Interoperability at computation infrastructure level is supported by the gUSE technology, which offers an abstraction layer to multiple heterogeneous providers (Grid, HPC resources, Desktop grids) called DCI-Bridge. It can be combined with another VISE offering an abstraction layer to different Cloud Providers called CloudBroker [23]. A VISE offering data transfer compound action across heterogeneous storage systems called DataAvenue [24] can also be connected to gUSE to support domain experts who need to manage large data sets across multiple sites.

Interoperability between different implementation of Work-

flow systems has been achieved by the SHIWA Interoperability platform [14], a VISE that acts as an abstraction layer to different Workflow Management Systems. Issues related at the reduction of expressiveness and richness of the set of actions posed by the abstraction layer have been investigated and experimented with by four different communities in the ER-FLOW project [21]. Another example is presented by [7] to tackle the problem of Interoperability of Workflows across heterogeneous infrastructures: Workflows are decorated with extra nodes to orchestrate the creation and destruction of Virtual Infrastructures that are enacted by VISEs operating in Clouds.

Interestingly enough, the solutions fall under two main categories. The technology-oriented partners in SCI-BUS composed by Layer-Oriented and Development-Oriented experts created VISE's that had structural relevance in the overall architecture of the system. The domain expert partners composed mainly by Result-Oriented and Development-Oriented experts developed multi-layered design patterns for their own Workflows that acted as Value Increasing Transient Elements (VITEs). Three out of four communities in ER-FLOW developed Value-Increasing Transient Elements by creating multi-layered patterns of Workflows [25]–[27] that exposed a higher Value to the user than their individual parts and performed compound actions when executed. One community took a more general approach by developing a VISE, called Processing Manager [8], that offered compound actions and increasing the Value of the information to the user by implementing an abstraction layer to different levels of middleware.

One of communities used a powerful feature offered by gUSE that allowed the easy development of interfaces [28] specific for each workflow. This solution as VISE in the upper layer, by selecting the set of information relevant to Result-Oriented actors, thus combining the rapidity offered by template-driven development with the effectiveness of information heterogeneity-reduction actions. This solution acted on three different aspects: First, it reduced the Cost of issuing requests for Result-Oriented Users by visually isolating the relevant actions (submission and retrieval of Workflows), Second, it reduced the cost of building the Workflows for the Developer-Oriented users by providing re-usable Sub-Workflows that could be combined in higher-level Meta-Workflows. Finally, it created Meta-Workflows that had an increased Utility but the same Cost to execute thus increasing the final Value.

V. CONCLUSIONS

This research is still its infancy and further examination will be necessary to evaluate its worth. Firstly, the model has now been used to describe systems that are closely related with each other and developed by interconnected communities, so it may fail when used to describe solutions based on different philosophies such as the HubZero platform. Should the proposed model succeed in describing additional platforms, it could become the basis for further refinement and a more formal approach.

ACKNOWLEDGMENT

This research would not have been possible without the help and support of all the partners of the ER-FLOW, SCI-BUS and HELIO projects. The authors are particularly grateful to Prof. Kacsuk, Prof Herres-Pawlis, Dr. Sciacca, Dr. Becciani, Dr. Castelli and Dr. Taffoni.

REFERENCES

- [1] N. Wilkins-Diehr, "Special issue: Science gateways - Common community interfaces to grid resources," *Concurrency Computation Practice and Experience*, vol. 19, no. 6, pp. 743–749, apr 2007. [Online]. Available: <http://doi.wiley.com/10.1002/cpe.1098>
- [2] P. Kacsuk, *Science gateways for distributed computing infrastructures: Development framework and exploitation by scientific user communities*, P. Kacsuk, Ed. Cham: Springer International Publishing, 2014. [Online]. Available: <http://link.springer.com/10.1007/978-3-319-11268-8>
- [3] S. Gelsing, T. Kiss, and G. Pierantoni, "Editorial: Science Gateways Applying Clouds, Grids and HPC - 5819z007.pdf," 2014. [Online]. Available: <http://www.computer.org/csdl/proceedings/iwsg/2014/5819/00/5819z007.pdf>
- [4] S. Olabarriga, G. Pierantoni, G. Taffoni, E. Sciacca, M. Jaghoori, V. Korkhov, G. Castelli, C. Vuerli, U. Becciani, E. Carley, and B. Bentley, "Scientific Workflow Management - For Whom?" in *2014 IEEE 10th International Conference on e-Science*, vol. 1. IEEE, oct 2014, pp. 298–305. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6972277>
- [5] K. Karozkai, A. Kertesz, and P. Kacsuk, "Brokering Solution for Science Gateways Using Multiple Distributed Computing Infrastructures," in *2015 7th International Workshop on Science Gateways*. IEEE, jun 2015, pp. 28–33. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7217925>
- [6] Z. Farkas, P. Kacsuk, and A. Hajnal, "Connecting Workflow-Oriented Science Gateways to Multi-cloud Systems," in *2015 7th International Workshop on Science Gateways*. IEEE, jun 2015, pp. 40–46. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7217927>
- [7] P. Kacsuk, G. Kecskemeti, A. Kertesz, Z. Nemeth, A. Visegradi, and M. Gergely, "Infrastructure Aware Scientific Workflows and Their Support by a Science Gateway," in *2015 7th International Workshop on Science Gateways*. IEEE, jun 2015, pp. 22–27. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7217924>
- [8] M. M. Jaghoori, S. Shahand, and S. D. Olabarriga, "Processing Manager for Science Gateways," in *2015 7th International Workshop on Science Gateways*. IEEE, jun 2015, pp. 1–7. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7217921>
- [9] J. Liu, E. Pacitti, P. Valduriez, and M. Mattoso, "A Survey of Data-Intensive Scientific Workflow Management," *Journal of Grid Computing*, mar 2015. [Online]. Available: <http://link.springer.com/10.1007/s10723-015-9329-8>
- [10] N. Cerezo, "Conceptual workflows," Ph.D. dissertation, Université Nice Sophia Antipolis, 2013. [Online]. Available: <https://tel.archives-ouvertes.fr/tel-00942559/document>
- [11] E. Deelman, D. Gannon, M. Shields, and I. Taylor, "Workflows and e-Science: An overview of workflow system features and capabilities," *Future Generation Computer Systems*, vol. 25, no. 5, pp. 528–540, may 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167739X08000861>
- [12] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers, "Examining the Challenges of Scientific Workflows," dec 2007. [Online]. Available: <http://eprints.soton.ac.uk/271187/1/computer07.pdf>
- [13] E. Elmroth, F. Hernández, and J. Tordsson, "Three fundamental dimensions of scientific workflow interoperability: Model of computation, language, and execution environment," *Future Generation Computer Systems*, vol. 26, no. 2, pp. 245–256, feb 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167739X09001174>
- [14] G. Terstyanszky, T. Kukla, T. Kiss, P. Kacsuk, A. Balasko, and Z. Farkas, "Enabling scientific workflow sharing through coarse-grained interoperability," *Future Generation Computer Systems*, vol. 37, pp. 46–59, 2014.

- [15] K. Plankensteiner, R. Prodan, M. Janetschek, T. Fahringer, J. Montagnat, D. Rogers, I. Harvey, I. Taylor, Á. Balaskó, and P. Kacsuk, “Fine-Grain Interoperability of Scientific Workflows in Distributed Computing Infrastructures,” *Journal of Grid Computing*, vol. 11, no. 3, pp. 429–455, sep 2013. [Online]. Available: <http://link.springer.com/10.1007/s10723-013-9261-8>
- [16] National Institute of Standards and Technology, “Common Industry Specification for Usability Requirements (CISUR),” <http://zing.ncsl.nist.gov/iusr/documents/whatistheCISUR.html>, 2006, [Online; accessed 10-May-2016].
- [17] —, “Visualization and Usability Group,” <http://www.nist.gov/itl/iad/vug/index.cfm>, 2016, [Online; accessed 10-May-2016].
- [18] J. L. Gabbard, D. Hix, J. E. Swan II, M. A. Livingston, T. H. Höllerer, S. J. Julier, D. Brown, and Y. Baillot, “Usability Engineering for Complex Interactive Systems Development,” in *Proceedings of Human Systems Integration Symposium 2003, Engineering for Usability*, jun 2003, p. 1.
- [19] “SCI-BUS Project.” [Online]. Available: <https://www.sci-bus.eu/>
- [20] “SHIWA Project.” [Online]. Available: <http://www.shiwa-workflow.eu/project>
- [21] “ER-FLOW Project.” [Online]. Available: <http://www.erflow.eu/>
- [22] P. Kacsuk, Z. Farkas, M. Kozlovsky, G. Hermann, A. Balasko, K. Karoczkai, and I. Marton, “WS-PGRADE/gUSE Generic DCI Gateway Framework for a Large Variety of User Communities,” *Journal of Grid Computing*, vol. 10, no. 4, pp. 601–630, 2012.
- [23] “CloudBroker Project.” [Online]. Available: <http://cloudbroker.com/platform/>
- [24] “DataAvenue Project.” [Online]. Available: <https://data-avenue.eu/>
- [25] S. Herres-Pawlis, A. Hoffmann, T. Rosener, J. Kruger, R. Grunzke, and S. Gesing, “Multi-layer Meta-met workflows for the Evaluation of Solvent and Dispersion Effects in Transition Metal Systems Using the MoSGrid Science Gateways,” in *2015 7th International Workshop on Science Gateways*. IEEE, jun 2015, pp. 47–52. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7217928>
- [26] G. Pierantoni and E. Carley, “Met workflows and Workflow Interoperability for Heliophysics,” in *2014 6th International Workshop on Science Gateways*. IEEE, jun 2014, pp. 79–84. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6882073>
- [27] G. Pierantoni, D. Perez-suarez, and P. T. Gallagher, “A WORKFLOW-ORIENTED APPROACH TO PROPAGATION MODELS,” vol. 15, no. 3, pp. 271–291, 2014.
- [28] G. Pierantoni and C. Eoin, “HELIOGate: A Portal for Heliophysics,” in *Science Gateways for Distributed Computing Infrastructures*, P. Kacsuk, Ed. Springer International Publishing, 2014, ch. HELIOGate., pp. 195–207. [Online]. Available: <http://link.springer.com/10.1007/978-3-319-11268-8>