An Environment Supporting the Production of Live Research Objects

Massimiliano Assante[1,2], Leonardo Candela[1] and Pasquale Pagano[1]

[1] Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" – CNR, Pisa, Italy
[2] Department of Information Engineering, Università di Pisa, Italy

{assante,candela,pagano}@isti.cnr.it

Abstract. Modern science communication requires innovative environment and means for providing stakeholders with scientific outcomes. Research objects are emerging as replacements of traditional "documents" in scientific communication. These objects are multi-media and multi-part objects that aggregate all the "pieces" that contribute to a research result. Supporting these objects has gone beyond the capacity of traditional technological approaches based on locally specialized data management facilities. In this article we present an environment for producing "live research objects" by exploiting the capabilities offered by a Data Infrastructure. Such environment includes: (i) a workspace where users can organize and share with their co-workers very different items in a file-system-like environment; (ii) an editing framework where users can define the structure of a live research object and compile objects that comply with one of the defined templates; and (iii) a workflow engine where users can define the workflow governing the production of a live research object by specifying the phases and the relative responsible actors(s).

1. Introduction

Scientific research is rapidly evolving in all fields, it is multidisciplinary, networked and driven by new patterns, e.g. data-intensive sciences [1]. In this complex scenario scientific communication must go well beyond traditional scholarly communication. Specifically, it requires accessing all the elements exploited and developed during the scientific workflow to achieve a result, e.g. datasets, analysis tools, and methods [2]. This wide corpus of primarily grey elements are at the moment mostly unavailable and, even when they are available, they are not

linked to the scientific result. This makes difficult to completely understand the result and validate it.

To overcome this limitations many initiatives have been proposed in the literature as replacement of traditional "documents" in scientific communication, such as *Executable Papers* [16,17], *Enhanced Publications* [6,15], *Living Reports* [8,7]. Some of them were sponsored by major scientific publishers [12].

Lately, *Research Objects* [10] are emerging as an abstraction for communicating, sharing and reusing research results. These are multi-media and multi-part objects that aggregate all the "pieces" that contribute to a research result. Such elements, which may range from binary files to compound objects including maps, time series, and tabular data, are generally structured according to well-established templates and produced according to user-defined workflows. We extended the Research Object definition and included facilities to make some of these "pieces" contributing to a research result directly embedded in the document. The idea behind a Live Research Object is depicted in figure 1.

However, supporting Research Objects has gone beyond the capacity of traditional technological approaches based on locally specialized data management facilities. This paper discusses how an infrastructure-oriented approach aimed at promoting *sharing* and *re-use* of resources (including data, services and computational and storage resources) is an effective approach for producing Live Research Objects and poses the accent on the user-oriented facilities supporting the collaborative production of such research products.

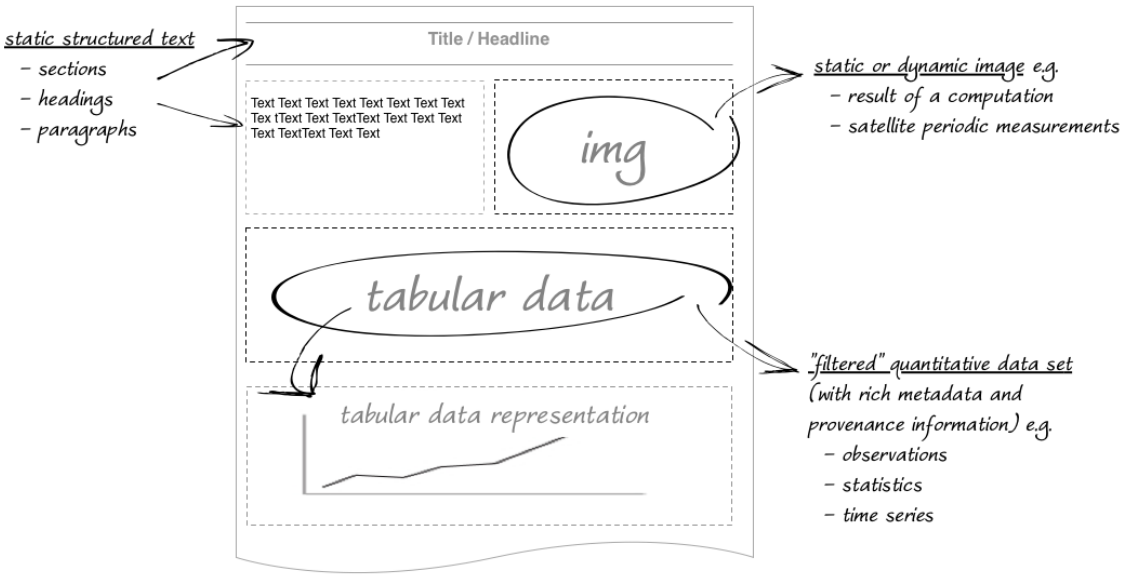

Fig. 1. The Idea behind a Live Research Object

The production environment includes: (*i*) a *workspace* where users can organize and share very different items (from binary files to compound objects) in a file-system-like environment; (*ii*) an *editing framework* where users can define the structure of a live research object (a template indicating sections, layout, active elements) and compile objects that comply with one of the defined templates by entering content or taking it from the workspace via *drag and drop*; and (*iii*) a *workflow engine* where users can define the workflow governing the production of a live research object by specifying the phases and the relative responsible actors(s).

Accordingly, the article is structured as follows. Section 2 describes the requirements and the main design decisions driving the development of the proposed approach. Section 3 presents the

environment developed to offer Live Research Objects production. Finally, Section 4 concludes the article and summarizes its results.

2. Motivations and Design Philosophy

We feel that the classic notion of documents viewed as static entities has to be abandoned. This notion should be moving towards one where documents are constantly evolving, by enriching them with components that yield seamless data access to contents, user cooperation, collaboration, interaction, and personalization.

However, this notion cannot evolve if not supported by progresses in the way documents are created, produced and made available too. In fact, thanks to the new technologies, scientists, researchers or experts in a field, can produce novel research outputs based on new resources (in terms of hardware, services and content) that were not available in the past.

Despite that, this production can still require a lot of work due to *(i)* the complexity of interfacing with different sources and tools, and *(ii)* the people involved in the task that may need coordination, concurrent and rule-based access to the same research object or part of it. In particular, for the research object instance case the resulting work may also not meet the requirements of its consumer, *i.e.*, the reader, since it could present a picture of the subject at the time of its production and not at the time in which the information produced is accessed and used.

The latter point illustrates the potential for new scientific advances. For example, the Food and Agriculture Organization of the United Nations (FAO) exploits its rich information sources,

ranging from raw data sets to graphs and map archives, to periodically prepare reports on the status of the agriculture and fishery, per country. This activity often:

- make necessary to have access to a wealth of textual, graphical and tabular information located in several (external) data sources;
- requires several people working together in the collection, collation, drafting, and reviewing;
- demands for "freshness" of the information reported.

To overcome these problems we decided to experiment the construction of a workflow[1]-driven Live Research Objects production environment. This environment *(i)* exploits the facilities provided by an underlying *Hybrid Data Infrastructure* [18] for accessing and exploiting the entire spectrum of resources needed to achieve a research result including data, services and computing resources "as-a-Service", *(ii)* uses a component-oriented flexible document model for the representation of the research objects, *(iii)* benefits from a workflow driven mechanism to ensure concurrent and rule-based access to its users, and *(iv)* is accessible through a thin client (namely a web browser).

The solution for the described approach can be logically divided in two main modules: one module realizes an environment for producing (namely defining and editing) innovative research objects, one module realizes an environment driving users (namely assigning actions and notifying the right actor(s) when needed) while producing such research objects.

[1] Commonly agreed, workflow is the series of steps procedure taken to complete a given task or job.

The first module is delegated to solve the issues related to the complexity of interfacing with different sources and tools, providing users with a familiar and easy-to-use environment to produce research objects. The following design decisions have been taken for the implementation of the proposed approach:

- The management of different data sources should be made transparent to the end-users (by the hybrid data infrastructure) and promote the seamless access and sharing of a rich array of information objects;
- The research objects production environment should enable a WYSIWYG[2] editor (similar to Google docs) to give users an immediate feeling on how the actual research object will look like;
- The production environment should enable the production of research objects sharing a common structure, when needed. Thus it supports a production based on two phases: *(i) template definition*, to define a basic research object structure (including sections and layout) to be adopted by research objects expected to be compliant with such a structure, and *(ii) reporting*, to produce the actual research object in compliancy with one of the defined templates;
- Supported Research Objects should enable the definition of living elements, i.e. Research Object elements that are potentially willing to evolve whenever an user accessed the object;

[2] WYSIWYG is an acronym for "what you see is what you get". A WYSIWYG editor is one that allows a user to see what the end result will look like while the document is being created.

- The produced research object should be available in different exporting formats, including OpenXML, PDF, and HTML.

The second module of the solution instead is in charge of coupling the research object with a workflow driven mechanism and ensuring through a series of steps, rule-based access to the same instance. For the implementation of this module the following design decisions have been taken:

- support for visual representations of the workflow, through workflow diagrams outlining the whole process;
- support for workflow states;
- support for manual and automatic (policy-based) routing;
- support for routing to individuals and to groups (set of users having the same role).

Accordingly, the next section presents the environment developed to offer workflow driven Live Research Objects production facilities.

3. The gCube Live Research Objects Environment

gCube[3] is software system enabling the building and operation of an Hybrid Data Infrastructure. In a nutshell, it offers a number of mediators for interfacing with (data) providers and a number of services for data management over a rich array of data types. The gCube Live Research Objects Environment is one of these data management services and it implements the

[3] www.gcube-system.org

environment envisaged in the previous sections to realize Live Research Objects. It is made of three main components: (i) the *Virtual Workspace*, that is a virtual file system promoting the sharing of a rich array of information objects, (ii) the *Research Object Editing Environment*, that is a graphical editor and renderer of a Live Research Object, and (iii) the *Research Object Workflow Manager*, that allows to create and associate workflows to a given Research Object and to manage and control its status.

3.1. Virtual Workspace

The Virtual Workspace (WS) is the core element of the cooperation environment. It is conceived to resemble a classical folder-based file system any user is familiar with. As a consequence, the operations it supports on the items are the expected ones, namely items creation, deletion and their organization in folders and subfolders. Thus every single user is free to organize its items.

However, the real added value of this file-system-like environment is represented by the types of items it can manage in a seamless way. They range from binary files to information objects representing tabular data, species distribution maps, and time series. Every item in the workspace is equipped with a rich metadata including bibliographic information like title and creator as well as lineage data.

Another distinguishing feature is represented by the sharing that is fundamental since users need to work collaboratively on the same research object and rely on common research materials. Sharing can be performed per single item as well as per folder and it is invite-based. Any item or

folder and, in turn, its content can be shared with other users. The users involved in this sharing are alerted by a notification mechanism in charge of delivering the related invite.

The end-user interface A web application has been developed to offer users the possibility of viewing and managing their research object instances and as well as any other information object. As shown in Figure 2, this web application offers a remote file system view similar to any Operating System over the content available within the data infrastructure along with files management facilities such as copy, delete and upload.

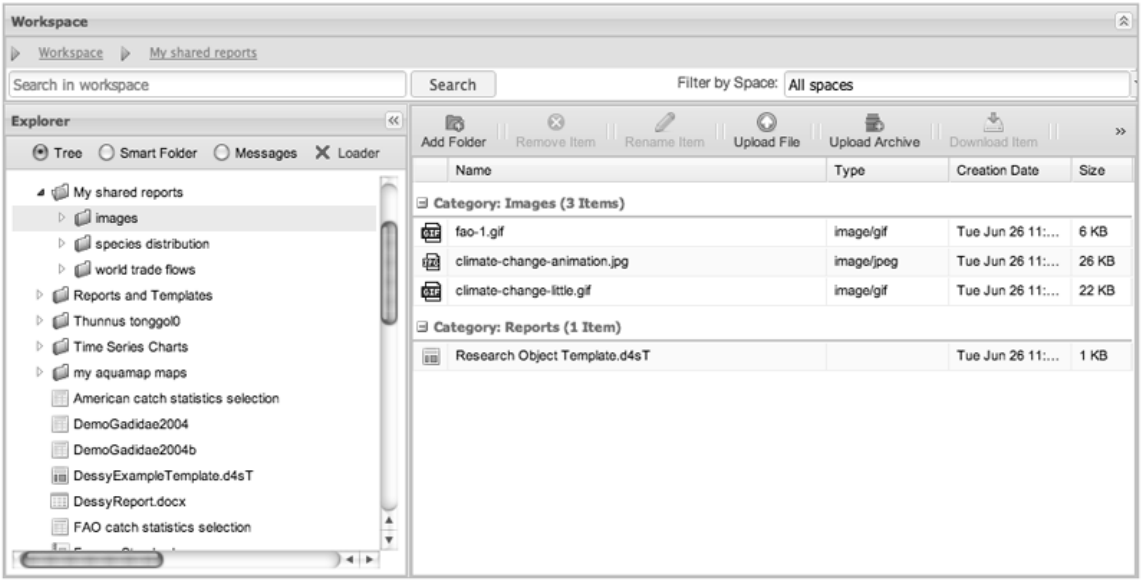

Fig. 2. The Virtual workspace UI

3.2. Research Object Editing Environment

Document templates motivate their uptake by easing the strain of repetitive manual work. It is quite common that a document's structure and style is maintained through time and used for multiple document instances. This is the reason why the Research Object Editing Environment, that is a web application developed by using the Google Webtool Kit framework [13], is logically divided in two phases: the Template Definition and the Research Object Editor. The first one is needed to define research object templates that, during this stage, are dynamically and statically completed. The second instead is capable of loading these templates to produce actual research objects by filling out their dynamic parts. In the following we explain our design approach from a functional description point of view together with actual examples for both phases.

Template definition During this phase, document templates can be created through user interfaces by exploiting an extension of the Document Editor web application, called Template Creator.

The Template Creator adopts a component oriented approach for templates composition and supports several component types such as structured and styled text areas (title, headings, body), images, tables, table of contents (ToC), bibliography, page breaks. Template components are divided in two classes: *static* and *dynamic*. A static component of a template is, as the name suggests, a part that is not meant to change across diverse research objects sharing the same template, such as filled-in texts or images. A dynamic component instead is meant to be completed in the second phase and, in turn, can belong to the following two categories:

- *dynamic text*: empty text areas, including headers and empty tables, fall into this category. An example would be having the same template for a set of research objects aiming at representing a study on a species. One would change species' specific data while the structure and the presentation would be uniform for all of them.
- *dynamic spot*: empty rectangular spots designed to host an image, a table, a diagram or a chart, to be instantiated using a given data source or data set. This is a key category as during the second phase these spots become configurable, providing users with the possibility of entering configuration parameters. Examples vary depending on the "hosting type" and will be explicated in the following.

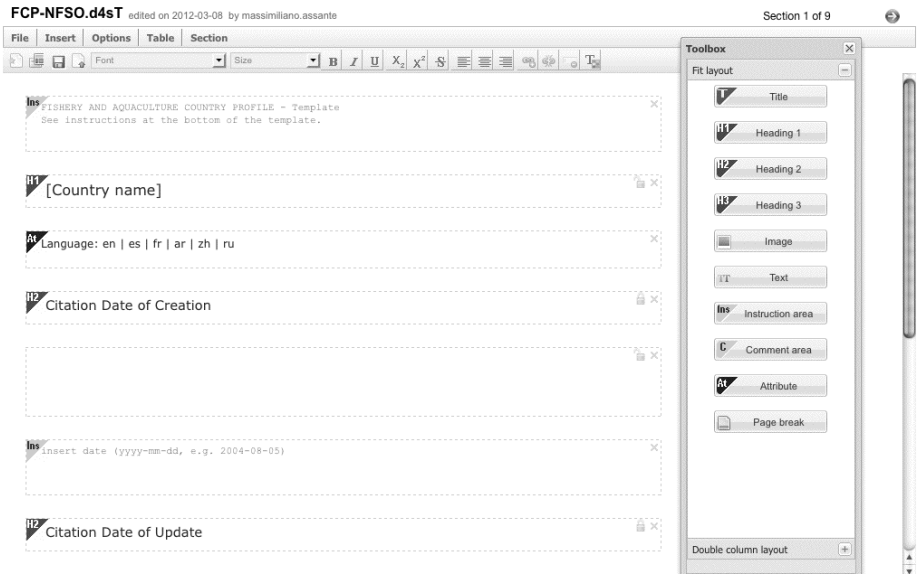

Fig. 3. A template section being created in the Template Creator

The template components belonging to a template can be grouped into sections. Figure 3 illustrates a template section being created using the Template Creator. As one can see, it is a

human-friendly interface assuming the form of an HTML page. This application provides users with toolbars and toolboxes from where they have the possibility of adding or removing template components, adding or removing sections, formatting text and saving their work into the Virtual Workspace.

Research Object Editor: The Document templates, created through the Template Creator extension, can be loaded by accessing the Virtual Workspace from within the Reporting Editor through user interfaces.

The Research Object Editor in this phase offers the possibility to complete or instantiate the dynamic components that might be present in a template. Depending on their type, this action can be performed in two ways: *(i)* by typing in, for components belonging to the *dynamic text category* (formatting text as one would in a word processor by using a formatting bar) or *(ii)* by providing a configuration for components belonging to the dynamic spot category. This configuration can vary depending on the *dynamic spot* type. For instance, users would specify which column and which row intervals to show for *tables* or, what data to be set on x and y axis for *charts* and so on.

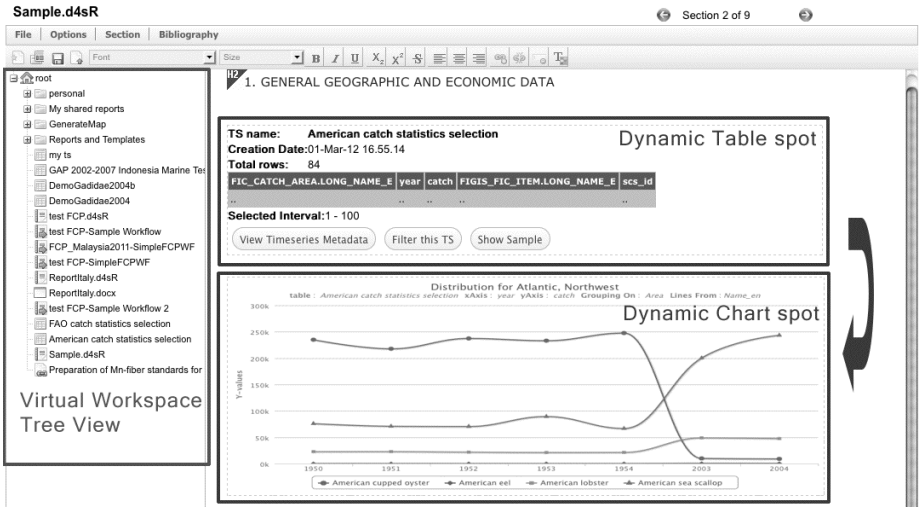

Fig. 4. A partial view of a template being completed in the Research Object Editor

Figure 4 illustrates the Research Object Editor user interface, a tree view of the Virtual Workspace is presented next to the *working area*. This tree view is needed to connect dynamic spots to the actual data they need to display. This connection is made explicit through *Drag and Drop* operations (by dragging the desired item over the desired spot).

Figure 4 also shows an example of template section being completed next to the tree view. The section is composed by three components, a static heading component on the top and two dynamic spots below, the first of type Table and the second of type Chart. In the example, both

spots have been connected to the same *quantitative dataset*[4], that is represented as an item in the user's Virtual Workspace, and describes a catch statistics *time series*[5] (reporting statistics on catches of marine species on a give geographic area, per year). The example shows the type Table being configured while the type Chart, that has been already, is actually displaying the related chart.

It is important to stress the fact that the *dynamic spot* component during this phase becomes *living, interactive* and capable of redisplaying itself depending on the parameters specified by the user.

Research Object instances can be successively exported into different formats by using functions provided by this editor, such as OpenXML (docx), PDF, HTML and saved into the user's personal Virtual Workspace.

3.3. Research Object Workflow Manager

The idea behind the Research Object Workflow Manager (ROW) is to work collaboratively to the creation of Research Objects. During this phase the Research Object is being created and is still incomplete. It is initiated by an actor, generally identified by a role, and then passed to other

[4] A quantitative data set represents a systematic compilation of measurements intended to be machine readable. The measurements may be the intentional result of scientific research or information produced by governments or others for any purpose, so long as it is systematically organized and described [4].

[5] A time series is a sequence of data points, measured typically at successive time instants spaced at uniform time intervals [1].

people involved in the realization. In the following, to distinguish this phase we will use the term Report instead of Research Object.

For example, one author is asked to start drafting the Report, successively several iterations can be possible between authors and editors, before the Report is sent for review and authorization. To support such scenarios, ROW is equipped with a *workflow roles editor* that, as the name suggests, allows to distinct the users involved in the creation of the Report into categories *e.g.*, *author, editor, publisher*.

The ROW is also equipped with a *workflow templates editor*, a graphical editor and renderer of a workflow diagram. A *workflow template* (WT) is naturally represented as a digraph (directed graph) where vertices, *i.e.*, workflow steps, are connected by directed edges, or arcs. WTs support the possibility to apply *labels*, *i.e.*, *roles*, on edges to indicate the required role to perform the transition from one step to another and are always defined by an entry and an ending point, i.e., Start/End steps are mandatory in any WT. An example of a workflow template, loaded in this editor is depicted in Figure 5.

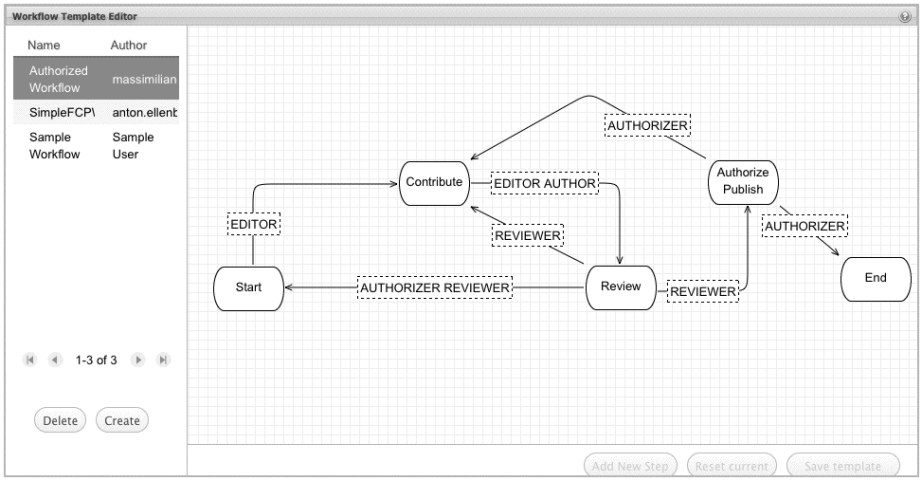

Fig. 5. A workflow template diagram loaded in the Workflow Templates Editor.

The presence of workflow templates is justified by the fact that reuse is highly desirable. Often one would like to reuse the same workflow diagram for multiple Reports, alternating the users involved but the process.

Roles and Templates are indeed complementary components for the ROW: they are required for the creation phase of a workflow document, where a Report's draft is linked to a Workflow Template, and for the functioning and monitoring of the reporting activity, *i.e.*, the step-by-step procedure needed to complete the job.

In particular, the creation phase of a new workflow document is performed by an entitled user and requires the three stages illustrated in Figure 6. In the first stage the user associates a Report draft to a Workflow Template: selects the Report to work with, created with the Research Object Editor by accessing the Virtual Workspace and couples it with an already available template

(created previously through the workflow template editor). The second stage requires the user to specify, for each step, the associated roles and their relative permissions. It is possible to associate any Role to a given step, however Roles (labels) connected to a step's outgoing edges are mandatory and already present for permissions to be added. It is worth noting that the permissions attached to a role can vary from one step to another, *e.g.*, during the step "review" an author has read-only permission while during the step "contribute" an author has both read and update permissions. During the third stage instead the steps are not interested while roles, defined in the workflow report during the second stage are. In fact this is the part where these roles are linked to the actual users partaking in the workflow report production.

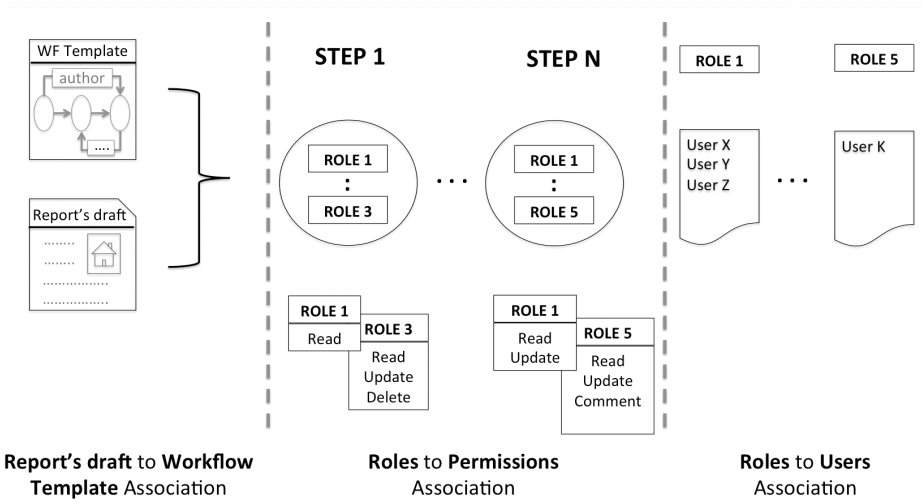

Fig. 6. The three stages needed for the creation of new documents workflows.

Regarding ROW *functioning*, manual routing and automatic routing are sup- ported as by requirement of Section 2. The former is achieved by providing the workflow reports owner

(WFO), *i.e.*, the user who created workflow reports through the three stages procedure previously described, the possibility to decide when and where performing step transitions, for the latter instead, the concept of forward action has been introduced. A *forward action* is a sort of green flag users make us of to indicate their work is completed for the current step (in which they are required to do a job). For instance, suppose there are three authors assigned to a step "contribute", each author completes his job and then performs a forward action towards the next step, *e.g.*, "review", ROW's logic recognizes it and consequently executes the transition (automatically) towards the expected step, *i.e.*, "review" for the sake of this example. It is worth noticing that the policy applied to automatic transitions is the one of executing it if, and only if, the totality of the actors involved in a job have forwarded. However, this can be changed to majority (instead of totality), by setting a parameter during the workflow report's creation phase.

Regarding ROW *monitoring*, the WFO can continuously monitor the users activities involved in a workflow. Clearly, a WFO can see and monitor only the workflow reports he owns. For each workflow report it is possible to *(i)* check the current status of the workflow, that is characterized by the current step and the forward actions performed until that time, *(ii)* check the workflow history, a chronological reversed list of user actions on the workflow.

The concurrent access, that is required since multiple users could work on the same report instance at the same time, is guaranteed by a locking mechanism provided by the ROW. When a user is editing a report instance a lock is acquired over it and no other user can access the report until the editing one finishes his work and commits the changes. The information a user has over any workflow report he needs to work with are the following: *name, current status, his role, date of creation, last action performed, grants (read, update etc.), whether the report is locked or not.*

4. Conclusion

Scientific research is nowadays multidisciplinary, networked and driven by new patterns, *e.g.*, data intensive sciences. This calls for innovative approaches and tools that are capable to cope with disciplines and tasks that can not be tackled by researches operating in isolation. Moreover, research products expected to be produced are richer than traditional research products, namely scholarly publications.

In this paper, an innovative environment for the production of live research objects was presented. In particular, a number of facilities supporting the whole lifecycle leading to the production of Live Research Objects was described. These facilities include: *(i)* a shared workspace resembling a classical file system and enabling users to store and organize any research artifact leading to a research product; *(ii)* an editor conceived to support the definition of research object structures, to promote the realization of research objects compliant with a given structure by implementing a number of user friendly features, *e.g.*, drag and drop of object constituents from the user workspace; and *(iii)* a workflow engine supporting the definition and operation of workflows driving the realization of a live research object in a collaborative and distributed approach.

Acknowledgments. The work reported has been partially supported by the D4Science-II project (FP7 of the European Commission, INFRA-2008-1.2.2, Contract No. 239019) and the iMarine project (FP7 of the European Commission, FP7-INFRASTRUCTURES-2011-2, Contract No. 283644).

References

1. T. Hey, S. Tansley, and K. Tolle. The Fourth Paradigm: Data-intensive Scientific Discovery. Microsoft Research, 2009.
2. Boulton, G.; Campbell, P.; Collins, B.; Elias, P.; Hall, D. W.; Laurie, G.; O'Neill, O.; Rawlins, M.; Thornton, D. J.; Vallance, P. & Walport, M. Science as an Open Enterprise. *The Royal Society,* 2012
3. http://en.wikipedia.org/wiki/timeseries. TimeSeries Definition, 2012.
4. M. Altman and G. King. A Proposed Standard for the Scholarly Citation of Quantitative Data. 13(3/4), Apr. 2007.
5. T. Blanke, L. Candela, M. Hedges, M. Priddy, and F. Simeoni. Deploying general purpose virtual research environments for humanities research. Philosophical Transactions of the Royal Society A, 368:3813–3828, 2010.
6. OpenAIRE EU Project Website - What is an Enhanced Publication? http://www.openaire.eu/en/component/content/article/76-highlights/344-a-short-introduction-to-enhanced-publications 2012
7. L. Candela, F. Akal, H. Avancini, D. Castelli, L. Fusco, V. Guidetti, C. Langguth, A. Manzi, P. Pagano, H. Schuldt, M. Simi, M. Springmann, and L. Voicu. DILIGENT: integrating Digital Library and Grid Technologies for a new Earth Observation Research Infrastructure. International Journal on Digital Libraries, 7(1- 2):59–80, October 2007.
8. L. Candela, D. Castelli, P. Pagano, and M. Simi. From Heterogeneous Information Spaces to Virtual Documents. In E. A. Fox, E. J. Neuhold, P. Premsmit, and V. Wuwongse, editors, Digital Libraries: Implementing Strategies and Sharing Experiences, 8th International Conference on Asian Digital Libraries, ICADL 2005, Lecture Notes in Computer Science, pages 11–22, Bangkok, Thailand, December 2005. Springer
9. G. Crane, A. Babeu, and D. Bamman. eScience and the humanities. International Journal on Digital Libraries, 7(1-2):117–122, October 2007.
10. Belhajjame K., Goble C., & De Roure D. Research object management: opportunities and challenges. Data Intensive Collaboration in Science and Engineering (DISCOSE) workshop, collocated with ACM CSCW 2012.
11. G. Crane, A. Babeu, and D. Bamman. eScience and the humanities. International Journal on Digital Libraries, 7(1-2):117–122, October 2007.
12. The Executable Paper Grand Challenge, Elsevier http://www.executablepapers.com/
13. Google Inc. Google Webtool Kit. http://developers.google.com/web-toolkit/
14. J. Lave and Wenger. Situated Learning: Legitimate Peripheral Participation. Cam, 1991.
15. S. Woutersen-Windhouwer, R. Brandsma, P. Verhaar, A. Hogenaar, M. Hoogerwerf, P. Doorenbosch, E. Durr, J. Ludwig, B. Schmidt, B. Sierman, "Enhanced Publications", edited by M. Vernooy-Gerritsen, SURF Foundation, Amsterdam University Press, 2009
16. Van Gorp, P. & Mazanek, S. SHARE: a web portal for creating and sharing executable research papers. *Procedia Computer Science*, 2011, *4*, 589-597

17. Nowakowski, P.; Ciepiela, E.; Harezlak, D.; Kocot, J.; Kasztelnik, M.; Bartynski, T.; Meizner, J.; Dyk, G. & Malawski, M. The Collage Authoring Environment. *Procedia Computer Science,* 2011, *4*, 608-617
18. Candela, L.; Castelli, D. & Pagano, P. Managing Big Data through Hybrid Data Infrastructures. *ERCIM News,* 2012, 37-38