New Interactions with Workflow Systems

I. Wassink, P.E. van der Vet,
E.M.A.G. van Dijk
University of Twente
Enschede, the Netherlands
{wassinki, vet, bvdijk}@ewi.utwente.nl

G.C. van der Veer
Open University
Heerlen, the Netherlands
gerrit@acm.org

M. Roos
University of Amsterdam
Amsterdam, the Netherlands
m.roos@science.uva.nl

ABSTRACT

This paper describes the evaluation of our early design ideas of an ad-hoc of workflow system. Using the teach-back technique, we have performed a hermeneutic analysis of the mockup implementation named NIWS to get corrective and creative feedback at the functional, dialogue and representation level of the new workflow system.

Keywords

Ad-hoc workflow, teach-back, user evaluation

ACM Classification Keywords

D.3.2 **[Language Classifications],** H.5.2 [**User Interfaces**], J.3.1 **[Life and Medical Sciences]**

1. INTRODUCTION

Bioinformatics is the domain where life science meets computer science. The bioinformatician is a life scientist who uses computer tools and programming to perform biological experiments (known as in-silico experiments). An enormous amount of tools are available today as programs and web services, provided by many different organizations [2]. In a single experiment, multiple services are combined: data produced by one service is used as input for the next service. Bioinformaticians create scripts to connect the services used in an experiment. These experiments can become complex due to the huge amount of data and large number of services involved.

Workflow systems are developed to help bioinformaticians deal with the complexity of designing and running these in-silico experiments. Their chief appeal lies in the fact that they provide easy access to tools and services provided by different groups and using different protocols. A workflow system provides a graphical user interface in which task-labels represent programs and web services. The experiment itself is represented as a graph: the tasks are nodes of the graph and arrows are used to have the output of one task function as the input of another task and to indicate execution order. The user can create an experiment by dragging and dropping new tasks into the graph and connecting them.

Building a workflow is a difficult job. The bioinformatician has to choose the right services and, when services are connected, to deal with data incompatibility problems between services [2, 8]. The situation is even more complicated because in current workflow systems, the complete workflow needs to be designed in advance before it can be run. In practice, however, the complete setup of the experiment is often not known in advance [1, 4]. In such cases, the bioinformatician wants to decide on the next step of the experiment using the outcomes of steps that have been finished.

We propose a new type of workflow system, named NIWS (New Interactions in Workflow Systems). NIWS is an ad-hoc workflow system; it enables the bioinformatician to design and execute partial workflows. This system will better fit the explorative working approach of the bioinformatician. The outputs of the tasks in the partially designed workflow can be inspected to decide how the workflow will be extended and how the current output can be used as input for new tasks. The important question is, of course, will such a system satisfy the bioinformatician? To answer this question, we embarked on a systematic design approach: (1) we analyzed the domain problem; (2) we developed a view on a solution (adaptable workflows; (3) we developed a first draft design; (4) evaluated our envisioning (the current paper); and subsequently, (5) we will iterate on our design, finally build a full blown implementation, and assess its value in a real world setting. For step 4 we investigated the design's relevance and usability with bioinformaticians familiar with workflow systems, by performing a teach-back technique, a hermeneutic method to provoke the users to externalize their mental models [6].

We will first give an overview of existing studies of workflow systems. We will describe NIWS. Next, we will explain the teach-back technique. Then we will describe our empirical investigation with professional participants. After that, we will discuss our results and we will end with a reflection.

2. Workflow systems for scientific experimentation

Much research has been done on scientific workflow systems, though, only few consider the usability of these systems. There is often a big gap between the level of detail that is relevant for a life science problem and the level of detail required for the implementation of the experiment as a workflow [1]. Gordon et al. [5] performed a user study to test the usability of the Taverna workflow system. They found functionality problems due to the exploratory nature of life science life scientist need to interact with the workflow during the actual experiment. Direct interaction enables the life scientist to try parameter settings and to debug workflows [1].

Downey [3] performed a user study to test the usability of the Kepler workflow system. One of the main features workflow users found were missing in this tool is a real-time debugger of the workflow to inspect intermediate results to make further decisions. The workflow system should guide its users to construct the workflow. Additionally, participants request for data directly being visible in the workflow diagram.

Gibson et al. [4] provided a first implementation and evaluation of an ad-hoc workflow system. The workflow designer can

ECCE 2009, September 30 – October 2, 2009, Helsinki, Finland.

design and execute partial workflows and reuse the intermediate results to further design the workflow. The results of the user study were promising; however, the system is not further developed.

3. NIWS – an adaptive workflow system

In prior work, we have discussed our workflow system named e-BioFlow [7]. This workflow system previously only supports the classical approach in which the complete workflow has to be designed in advance. NIWS is mockup implementation of the ad-hoc extension for e-BioFlow to support and stimulate explorative experiment design and execution.

Designing and running workflows in NIWS is intended to be easier than in classical workflow systems. Tasks can be executed in isolation by pressing the play button in the task box. Input ports and output ports of the tasks are present at respectively the top and the bottom of the task box. The data consumed and produced by the tasks is explicitly present in the workflow as circles. The user can inspect these data and use them as sources of inspiration how to further design the workflow. The data can be defined to be input for new tasks. NIWS does not require the user to rerun the complete workflow, but only inserted and modified tasks.

Finding suitable tasks is difficult. NIWS has a search engine to help its users find tasks based on the name, the type of operation it performs, the inputs and outputs, and the authority that hosts the application the task represents.

But NIWS is more: it supports guided analytics. Based on the type of data in the workflow, it suggests tasks that can take these data as input. This helps the user to find compatible tasks in a quick manner, but at the same time it forms a source of inspiration of possible directions in the experiment. In a similar way, NIWS can help to deliver the input required for a certain task by suggesting tasks that can produce the right data.

To connect tasks, users often have to parse and build complex data structures. NIWS helps its users doing this for XML structures. It provides so-called composer and decomposer tasks to build and to parse XML structures. In case of a composer, the user only has to provide the attribute values of the XML; in case of a decomposer, NIWS will return the attribute values.

4. Teach-back as a technique for hermeneutic analysis

People working with complex systems need a mental model of the system in order to (1) plan use; (2) actually interact with; (3) understand and assess the effect of the interaction; and (4) understand the meaning of unexpected system actions.

Mental models are knowledge structures inside people's mind, based on learning the semantics of the system and its context ("what-is" knowledge), experiencing the dialogue with the system ("how-to" knowledge), and understanding the representations of the system state, system actions and system feed-back (the "vocabulary" of the interaction).

Mental models actually develop based on a current need (to act, or to explain to a colleague, etc.), in a current context (with or without the system being at hand).

Since mental models are "mental" we cannot directly observe or register them. Hermeneutics is a philosophical method where an analyst develops understanding of the meaning an object (e.g., an artifact) has for a certain person or a certain group of people.

We apply the teach-back technique [6] for our hermeneutic analysis: We introduce prospective users of our design (professional bioinformaticians) to our early design ideas (use cases represented as realistic scenarios by introducing a realistic user persona, a typical context of use and a relevant task).

We then ask these users to *teach back* their understanding of the system to an imaginary colleague. In order to teach back, we pose, both, "what-is" questions and "how-to" questions, the latter in different degrees of similarity with the use cases shown in the scenarios. In order to record the externalized mental representations, we ask our users to write down (scribble, use key words and full text at will) their teach-back.

To interpret these representations, we first develop a scoring schema and fine tune this to a level where independent analysts reach agreement to an acceptable level. We aim at a level comparable with inter-rater reliability accepted for psychological personality measurement techniques.

5. Assessment of a design envisioning

The aim of the current study is to gain insight into the mental model bioinformaticians have about our early design envisioning, NIWS. Our study focuses on professionals (life scientists with some experience in using workflow systems), to analyse if this new system is an improvement over state of the art existing workflow systems. The study consists of three phases. First, the participants are shown a mockup of NIWS. Second, based on the scenarios four questions are asked to gain insight into the bioinformaticians' mental model of NIWS. Third, the filled protocols are scored in categories to explore the participants' mental models.

5.1 Setup

The mockup of NIWS is an animated slideshow presentation containing a narrative of a bioinformatician performing experiments, showing text and sketchy mockups of the system. A voice-over reads the text in the slideshow to make the presentation vivid and realistic. The presentation contains two scenarios that show various features of the envisioned system and suggest new possibilities when using this system. The scenarios are based on real-life situations in bioinformatics, but worked out using our system ideas. The presentation of the scenarios takes about ten minutes.

The four questions consists of a "What is" question, probing a semantic mental model, and three "How to" questions, probing procedural mental models. In the first question, the participant is asked to explain to an imaginary colleague Tom, who is familiar with workflow systems but does not know NIWS, what NIWS is. In the three "How to" questions, the participant is asked to explain to Tom how to perform a particular task using NIWS. These tasks are not explicitly covered by the scenarios, but using NIWS could be inferred from them in relation to the individual participant's mental model.

The questions are distributed on paper. To respond, participants can write, scribble, make drawings, etc. The participants get five minutes to answer each question. They are, however, not allowed to discuss or to ask questions, since we are interested in what they believe the system can do. We do explicitly mention that the questions are not to test the participants' knowledge: there are neither right nor wrong answers. Participation is anonymous and voluntary. All participants are rewarded for participation with a 1 GB USB key.

5.2 Participants

In total, there are 50 respondents, originating from different countries, though most of them are Dutch. The participants have different backgrounds (biology, bioinformatics, chemistry, computer science) and their expertise in using workflow in bioinformatics experiments differs from beginner to experienced user. These respondents are recruited during six sessions: during visits at life science research groups, courses in the Taverna workflow systems and a meeting of the BioAssist Group. The size of the groups ranges from 1 to 20 persons.

A strict protocol is handled in these sessions in order to keep the experiment reproducible. In each session, an experimenter is present to start the scenarios, to distribute and collect the protocols and to manage the time. No information about NIWS is given to the participants other than the scenarios. The participants received the reward when they handed in the form.

5.3 Scoring the protocols

The result of teach-back may consist of both creative and corrective feedback. Creative feedback will encompass new features the participants expect to exist based on the scenario. In corrective feedback, the participants mention features they do not like, expect not to work, or want to be improved.

The feedback is analyzed regarding three levels of the system: (1) feedback related to the functionality: what the participants believe the system can do and what its limitations will be; (2) feedback related to the dialogue; (3) feedback related to the representation of the workflow experiment and the system interface.

A scoring scheme is set up to analyze the forms in an unambiguous and reproducible way. This scheme consists of rules and examples how to categorize the feedback. To set up this scheme, two analysts (authors) separately analyzed five forms. They discussed their findings with a third author and built a scoring scheme. The two analysts separately scored another three protocols and compared their scorings to test the agreement on the scoring scheme, which confirmed interpretation and scoring reliability. Consequently a single analyst was sufficient to score the remaining 42 protocols.

6. Results

The results are grouped along the scoring levels. In 6.1 we show the results (illustrated by examples from the protocols) for functionality:

- Corrective feedback: functionality (“what-is” knowledge) as indicated in the scenarios that we found back in the protocols and that is consistent with the scenarios, as well as functionality understood by the participants that is inconsistent with our scenarios, and indications of functionality aspects not appreciated by the participants.
- Creative feedback: We will show examples of functionality found in the protocols not mentioned in the scenarios, that makes sense as extension of the design.

Based on this we intend to repair, expand and improve the functionality of NIWS in the next phase of this project.

In 6.2 we will report on corrective feedback and creative feedback regarding the dialogue (“how-to” knowledge) of NIWS, and in 6.3 we will do the same for feedback regarding the representations. On this last aspect we need to keep in mind that the scenarios as presented by us are describing our NIWS design ideas at a global level, focusing on the functionality, and hinting the dialogue, but being vague on the actual representation of the system interface and on the users’ actions.

6.1 Functionality

Most respondents react positively on the system presented. Many of them mention NIWS is like other workflow tools, but then more intuitive, simpler or easier to use. As one said, “a big plus is that you can add additional processing anywhere in the chain, without having to re-run everything as it caches intermediate results”. Another respondent mentioned “You don’t have to rerun the workflow every time. Therefore you will save a lot of time”. It is also easier to use for beginners: “NIWS is this new workflow system that has this cool feature of giving you hints when you don’t know what to do. Ideal for beginners like me ;-).” However, one respondent said the questions were easier to solve without using a workflow system.

Many respondents picked up the idea of designing workflows step by step. Intermediate results can be used to further design the workflow. “The nice thing is that one can execute every process in isolation and that one can inspect the outputs of the workflows at any moment.” NIWS enables one to execute the partial workflow, to test and debug the workflow. One respondent describes this as “kneading” de workflow.

Eight respondents propose a two step approach to design and run workflows in case large data sets are analyzed. First, design a workflow using a small example data set. Second, when the design is finished, run the workflow for the entire data set. Another respondent suggests to create a workflow for one data item, and to embed this one into a larger workflow that runs it for each data item of the complete set in parallel.

To find services, 27 respondents recommend using the search facility of NIWS, though some of them found the use of this facility to be unclear. One respondent expects the search function to be smart: meta-data can be used to further refine the search. For example, the database name can be used to find blast services that have access to that database. Others recommend using external resources, such as Google or colleagues, to find services. NIWS is expected to provide access to many different types of web services, such as BioMOBY, REST, XML-RPC and SOAP/WSDL.

The feature of NIWS to suggest services that can take data available in the workflow as input is picked up by 11 respondents. Three of them explicitly mentioned that they expect the suggested services to be compatible with the data in its current format; so no data conversion should be needed.

NIWS’s functionality to automatically compose and decompose XML data is found useful by many respondents. Sixteen respondents even expect these facilities to solve all data format problems and data conversion to be a built-in feature of NIWS. Others, however, were skeptic about the automatic data conversion facilities: “If this went well, e.g. if you would never experience data compatibility issues, is questionable, because the output of one service needs to know what kind of format is expected as input of the other service”. Some respondents expect support for scripting facilities, including query languages, to perform the data conversion. These scripting facilities could be used to perform data transformations, but also to affect the control flow of the workflow. Others recommend searching for external data conversion services that will hopefully transform the data into the right format.

6.2 Dialogue

The scenarios show the drag and drop facilities of NIWS. Similar, two respondents expect copy and paste functionality to be available for easily reusing parts of workflows. A few respondents expect the option to embed workflows previously designed or designed by others in larger workflows.

Many respondents have picked up to use the play button in the task box to run a task in isolation. From the scenarios, it is not clear whether tasks upstream in the workflow will be executed automatically. One respondent supposes this to be the case. NIWS will ask its user to enter missing data. In case only a fixed set of options is valid as input, NIWS will lists them to let users choose from them. One respondent mentioned it would be nice if users can also choose from data already in the workflow.

Many respondents perceived that composer tasks and decomposer tasks can be added by right-clicking on respectively the input and output ports of a task to feed correct input or parse output. One respondent described this as some magic: "The blast service needs some magic before we can use it, so we must tell [NIWS] to do its magic. The result is two boxes which we can give our [user] name and sequences". He refers to the composition tasks to deliver the correct XML input. Some other respondents expect right-clicking on ports is used to set default input values. Two respondents expect NIWS to add composer tasks and decomposer tasks automatically. Two other respondents declared that building and parsing hierarchical XML structures can be established by a chain of composition or decomposition tasks.

NIWS is expected to warn about the existence of data compatibility problems. "The system will give an error if the outputs don't match the inputs. When that is the case (and yes, this will happen) write a small converter script to process the output." To create these scripts, a good editor with auto-completion and syntax highlighting is desirable.

One respondent recommends limiting the amount of mouse interaction required: “It looks like a lot of clicking is required, for every composer and decomposer and to execute a task, you have to click.”

6.3 Representation

In total, 28 respondents use drawings to explain Tom how to use NIWS. In many drawings, the tasks boxes have inputs and outputs explicitly visible at top and bottom. In the scenarios, data are presented as circles, which makes the workflow graphs look like colored Petri nets. Data are explicitly present in drawings of 19 respondents. These data are connected by arrows coming from output ports and arrows going to input ports. Some respondents drew data using boxes instead of circles. So, the difference between these two symbols seems to be unclear. One respondent used stacked circles to represent collections of data in case a task returns multiple items.

7. Discussion

A scenario-based mockup implementation is an easy and a fast way to evaluate design ideas (e.g., of NIWS) in an early stage of the design. Applying the teach-back technique we found that NIWS is a significant improvement over traditional workflow interfaces. Many respondents put high value on the ability to inspect intermediate results to further design the workflow. Helping with data conversion and finding and suggesting services are other features these respondents put high value on.

Besides positive feedback, respondents gave feedback about desired **functionality** of a workflow system, even of aspects not shown in the scenarios. Furthermore, the respondents gave directions to improve the **interaction** with the system presented and other workflow systems.

The results of this study will be used to develop an interactive implementation of NIWS in our workflow tool e-BioFlow.

8. Acknowledgement

Our thanks to L. Packwood for her help to design the mockup of NIWS. We like to thank J.A.M. Leunissen, P.B.T. Neerincx, M. van der Bruggen, H. Rauwerda, T.M. Breit and M. van Driel for their help to find participants for the study.

This work was part of the BioRange programme of the Netherlands Bioinformatics Centre (NBIC), which is supported by a BSIK grant through the Netherlands Genomics Initiative (NGI).

9. References

[1] M. Addis, J. Ferris, M. Greenwood, P. Li, D. Marvin, T. Oinn, and A. Wipat. Experiences with e-Science workflow specification and enactment in bioinformatics. In S.J. Cox, editor, *e-Science All Hands Meeting 2003*, pages 459–466, Nottingham, United Kingdom, 2004.

[2] V. Curcin, M. Ghanem, and Y. Guo. Web services in the life sciences. *Drug Discovery Today*, 10(12):865–871, 2005.

[3] L.L. Downey. Group usability testing: Evolution in usability techniques. *Journal of Usability Studies*, 2(3):133–144, 2007.

[4] A. Gibson, M. Gamble, K. Wolstencroft, T. Oinn, and C. Goble. The data playground: An intuitive workflow specification environment. In S.J. Cox, editor, *E-SCIENCE '07: Proceedings of the Third IEEE International Conference on e-Science and Grid Computing*, pages 59–68, Washington, DC, USA, 2007. IEEE Computer Society Press.

[5] P. M. K. Gordon and C.W. Sensen. A pilot study into the usability of a scientific workflow construction tool. Technical Report 2007-874-26, Sun Center of Excellence for Visual Genomics, 2007.

[6] M.C. Puerta Melguizo, C. Chisalita, and G.C. Van der Veer. Assessing users mental models in designing complex systems. In A. El Kamel, K. Mellouli, and P. Borne, editors, *IEEE International Conference on Systems, Man and Cybernetics*, pages 1–6, Yasmine Hammamet, Tunesia, 2002.

[7] I. Wassink, H. Rauwerda, P.E. van der Vet, T.M. Breit, and A. Nijholt. e-BioFlow: Different perspectives on scientific workflows. In M. Elloumi, Josef Küng, Michal Linial, Robert F. Murphy, Kristan Schneider, and Christian Toma, editors, *2nd Internationall Conference on Bioinformatics Research and Development (BIRD)*, pages 243—257, Vienna, Austria, 2008.

[8] I. Wassink, P.E. van der Vet, K. Wolstencroft, P.B.T. Neerincx, M. Roos, H. Rauwerda, and T.M. Breit. Analysing scientific workflows: why workflows not only connect web services. In LJ. Zhang, editor, *SERVICES 2009 (Part I)*, pages 314–321, Los Angeles, USA, 2009.