A Pilot Study into the Usability of a Scientific Workflow Construction Tool

Paul M.K. Gordon, Christoph W. Sensen
University of Calgary
Sun Center of Excellence for Visual Genomics
+1 403 210-9535

{gordonp,csensen}@ucalgary.ca

CPSC Technical Report
2007-874-26

ABSTRACT

We describe a recent pilot study into the usability of the scientific workflow creation and enactment tool called Taverna. Both programmers and non-programmers were used as subjects for a defined programming task. We used a combination of user observation and questionnaires to determine programming efficiency roadblocks in the tool. More generally, differences between the roadblocks encountered by programmers and non-programmers suggest that pilot studies are crucial to inform the proper evaluation of novice programming tools. The study also suggests that there is a high demand for reusable Life Sciences workflows, due to both their ability to facilitate human-human communication about data analysis, and their ability to simplify repetitive operations used by bench scientists. Most roadblocks to Taverna programming are interface related, but a more fundamental issue is related to data input and type enforcement. Despite UI issues, we discovered users' willingness to re-use and modify workflows, which leads us to suggest that programs first be created in simpler tools as a stepping stone in end-user development for the Life Sciences.

Categories and Subject Descriptors

D.1.7 [**Software**]: Programming Techniques – *Visual Programming.* D.2.6 [**Programming Environments**]: Integrated Environments J.3 [**Computer Applications**]: Life and Medical Sciences – *Biology and Genetics.*

General Terms

Experimentation, Human Factors, Languages

Keywords

End-User Development, Visual Programming, Workflows, Empirical Usability Study, Bioinformatics

1.INTRODUCTION

Scientific inquiry often involves *ad hoc* comparison, combination and transformation of data; therefore it is difficult for professional programmers writing support software to foresee all workflows that scientists will require to address their research needs. Barring having a programmer always at their disposal, the scientist must perform various analysis steps manually, or somehow be able to program themselves. The most prominent tool for End-User Development (EUD) in the Life Sciences is Taverna [1]. The pilot study described here is done in the context of attempting to answer three questions:

- Can we meaningfully evaluate the usability of Taverna?
- Do scientists see computerized workflow-based analysis as useful?
- Is there a need to make computerized workflow programming easier for the scientists?

Our research into EUD for Life Science researchers is motivated by the same reason EUD in general: there are many times more end-user programmers than professional programmers [2].

2.RELATED WORK

To date, only one major evaluation of Taverna has been performed [3], which was primarily from a technical perspective, with anecdotal user observations. More generally, user studies in bioinformatics are virtually non-existent (see [4] for a rare example), as are formal attempts at user-centered design [5]. In the wider fields of EUD and Visual Programming, two evaluative frameworks are commonly used for providing insights into programming: the cognitive dimensions [6], and the communicative dimensions [7]. Both are highly relevant to scientific EUD because the former focuses on usability of the programming notation (human-computer interaction, essential to user adoption), while the latter focuses on the ability of the visualization to mediate discussion about the algorithm (human-human interaction, essential in peer-reviewed science).

In addition to the communicative aspect of workflow usefulness, there is of course the pragmatic aspect: is there a net benefit to learning how to build computerized workflows? Blackwell's Attention Investment Model [8] identifies key factors in deciding to take the first steps in programming, which translate roughly into:

- *Cost:* how long would it take to do it manually?
- *Investment:* what is the time required to learn how to program it?
- *Pay-off:* how much time does the automation save?
- *Risk:* what is the probability of failure to write the correct program?

Both dimensional frameworks, and the attention investment model were used in developing the study methodology.

3.STUDY SETUP

3.1The Task

Participants were asked to build and enact a workflow in Taverna (version 1.5.1) that 1) took FastA-formatted sequence (amino acid or DNA) as input, 2) ran any BLAST [9] service of their choosing, and 3) outputted the results. Participants were allowed to ask the author for assistance if they had exhausted their own thoughts on how to perform certain actions in the interface. Figure 1 illustrates a correct solution for the task.

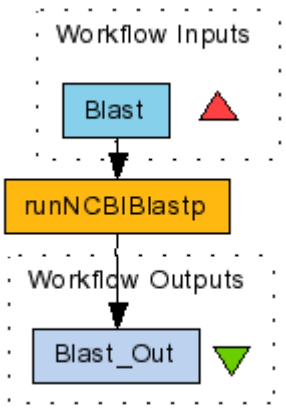

Figure 1 - A workflow constructed by one of the study participants, as visualized by Taverna.

Participants followed a think-aloud protocol while performing the programming task, which was audio recorded. The author also took notes during the sessions. After completing the programming task, the participants were asked to fill out a survey with 11 questions about their computer and bioinformatics knowledge, 14 questions related to the cognitive dimensions, 6 questions related to the communicative dimensions, and 6 questions related to workflows and EUD more generally. Participants were also asked about what they perceived to be Taverna's strongest and weakest points.

3.2The Participants

Four scientists and three professional programmers participated in the programming task. Two additional developers intimately familiar with Taverna participated in the questionnaire alone, to provide a broader sample of views on the evaluative dimensions. Two of the scientists had some Perl programming experience, but rarely use it. One scientist was a self-described technophobe. All participants were familiar with BLAST, and therefore had no problems comprehending the nature of the task set before them.

4.FINDINGS

4.1Task Completion

The time taken to build and enact a correct workflow varied between 28 and 39 minutes, with 2 of the 3 professional programmers completing the task the fastest. While all users were able to enact their workflow, none of the various BLAST services used worked successfully, due to data type issues (actual FastA text input versus expected XML input).

4.2Questionnaire

In addition to the 9 participants, the author also completed the questionnaire prior to performing the study, to help identify and reduce possible bias in the interpretation of the results. One cognitive dimension was evaluated with two different questions, as a form of internal control on the question formulation. Two participants did not answer a few questions, because they did not feel that they understood those questions completely.

The results of the cognitive dimensions evaluation are summarized in Figure 2. While the sample size of the study is relatively small for statistical purposes, definite trends in the evaluation emerged. More users agreed that Taverna did not force *Premature Commitment* (e.g. order in which elements are added to the workflow) while programming, in contrast to the author's original evaluation. All groups of users found Taverna's *Error-Proneness* to be higher than the author, which probably reflect the data type issues encountered during the task.

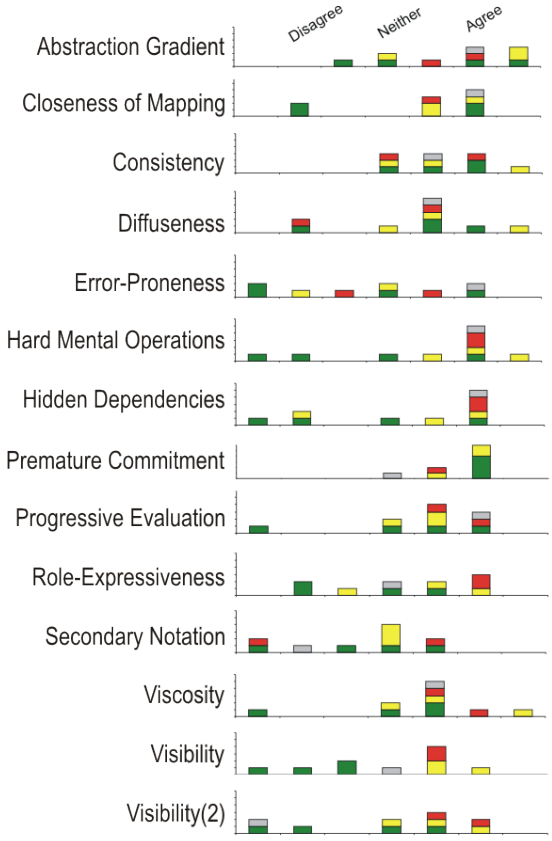

Figure 2 - Histograms of user responses to questions evaluating the cognitive dimensions of Taverna, where "Strongly agree" (rightmost column) reflects agreement with a positive usability statement. Colour key: green – scientists, yellow – programmers, red – Taverna experts, grey – author.

Progressive Evaluation (the ability to see intermediate results while programming), *Viscosity* (ability to modify any part easily) and *Diffuseness* (workflow visibility) were viewed

somewhat positively by all groups of participants, while *Secondary Notation* (ability to add annotations) somewhat negatively. Programmers and non-programmers disagreed on *Hard Mental Operations* (creating and following the workflow logic) and *Visibility* (presentation of relevant information during workflow construction), with non-programmers responding negatively.

The responses to questions regarding communicative dimensions of the workflows created are summarized in Figure 3. All dimensions were viewed positively by all groups, in line with the author's expectations.

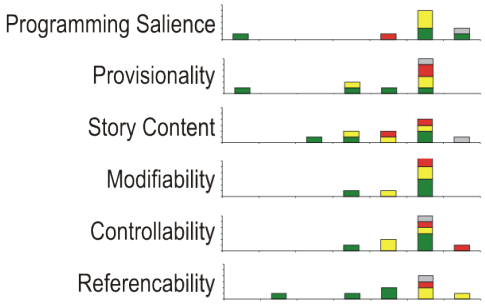

Figure 3 - Histogram of users' evaluation of their workflows' communicative dimensions. The values and colour keys are the same as used in Figure 2.

The positive communicative aspects of the workflow clearly influenced the interest of the participants in workflows, as did the prospect of being able to re-use and modify them (see Figure 4). This interest lies starkly is contrast to their general interest in end-user development (see Figure 5).

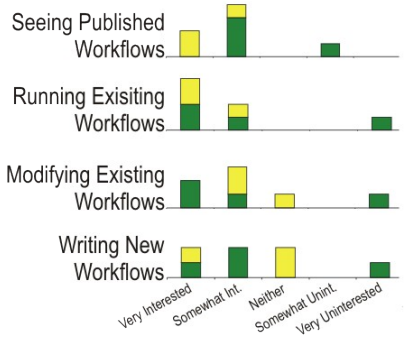

Figure 4 - Histogram of the 7 task participants' interest in the various aspects of Taverna workflow use in science.

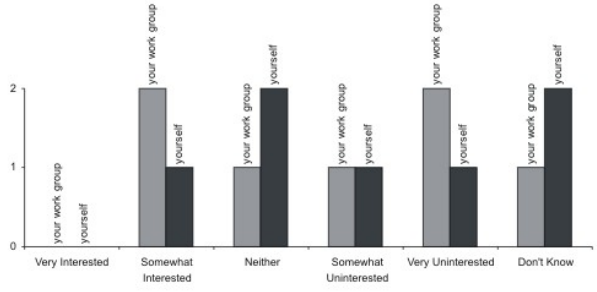

Figure 5 - General interest of the task participants and their research groups in EUD.

4.3Observations

4.3.1Common to all participants

In the Attention Investment Model, the user's *perception* of the difficulties and benefits of the programming effort is critical in them deciding whether to pursue it. As such, it was important to elicit these perceptions from the participants, and they are summarized in Table 1. It is noteworthy that positive perceptions were reported more than negative, but were obviously not all of the same cost/benefit value.

Table 1. Participants' Perceived Strengths and Weaknesses of Taverna

Group	Strength	Weakness
Programmers	Focus on *What* rather than *How*	Inconsistency of interface elements
	Flexibility to add/del/link services	
Both	Number of services	Documentation/Help
End-users	Understandability of graphical model	Difficulty getting started & Navigating the interface
	Task Automation	Readability of service search results
	Lack of programming syntax	

Additional observations about interface issues were made by the author during the course of the sessions, and the most salient examples common to both programmers and non-programmers alike are highlighted in Figure 6. The failure to get appropriately formatted data into the workflow frustrated the users. Feelings of personal inadequacy in enactment failure were noted during the sessions, with comments from the scientists such as "My head hurts!" and "That made me feel dumb".

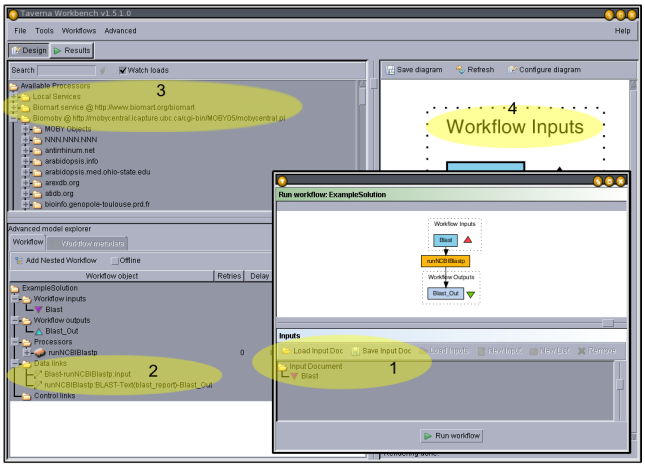

Figure 6 - Major Taverna interface roadblocks observed in creating the workflow: 1) incompatible and incomprehensible data formats/types, 2) difficulty linking workflow elements, 3) poor visibility, searchability and typing of processors, 4) no direct manipulation of workflow diagram by default.

Many participants added multiple components to the workflow, then attempted to connect them direct manipulation of the

workflow diagram. After realizing themselves or being told that the diagram was not interactive, they had difficulty learning how to connect the workflow elements using the Advanced Model Explorer. Users did not determine which element 'ports' to connect, and which ports had the desired connection options (users tried to link the input *from* BLAST, but the connection must be made from the input ports *to* BLAST).

Another major source of inefficiency in workflow construction for all participants was navigating and searching the "Processor" list. No users discovered the search function for at least several minutes, despite the fact that it is always on the screen. While this invisibility is an impediment for first time use, a more serious difficulty was browsing the search results. Search results include the whole sub-trees of the processor tree that contain matches, rather than just the matching nodes. Users often scrolled by relevant items in the long lists, or became disoriented, not recalling what the parent nodes were for a given processor being considered (the parent nodes were off-screen).

4.3.2Programmer-specific

An unusual phenomenon observed in all the programmers' session is that they first attempted to load the FastA formatted data, before building the workflow. While they would certainly never hardcode the input to programs in Perl or Java, it is possible that they viewed Taverna's GUI more as an application than as a programming environment. All users were told before starting that they were to build a workflow program, yet only the end-users intuitively dived directly into creating the workflow elements. Other minor issues specific to programmers were that they often used element labels like variable or object references in an attempt to link elements, and that they trusted the processor names, without fetching the metadata descriptions to see if they were appropriate to use.

4.3.3End-user specific

End-users displayed poor comprehension of Taverna's Processor types. Several end-users searched through "LocalServices" for some time, either assuming that Taverna itself provides the BLAST services, or that it was preconfigured:

> Participant A: *"So, someone's already designed these different outflows. I'll look at these local services then…"*

Another comprehension issue for end-users was that the distinction between Moby Objects (data) and Moby Services (methods) [10] was unclear, which lead in several instances to major delays in actually finding BLAST services:

> Participant B: *"BlastJob would, in English, sound like what I need."*

Although all participants had difficulty with data input for enactment, the programmers realized that they just didn't have the right data. The end-users did not understand the requirements, or the errors:

> Participant A: *"…sorry, since we're talking about FastA format…So then I'd open [looks at the 'Input Doc' button]…I don't know what this is."*
>
> Participant C: *"And now we've got a jDOM error… whatever that means."*

5.THREATS TO VALIDITY

5.1Construct Validity

The evaluative frameworks used to guide the study's data acquisition and observations are well established in the literature.

5.2Internal Validity

For the first three, non-programmer session, the participants were asked to download and install Taverna on their own computer. This proved fraught with many difficulties, nearly doubling the duration of the exercise, and therefore Taverna was re-installed and opened for the remaining sessions. This change was not seen as overly deleterious to the study, because as a pilot study part of the mandate is to determine such feasibility issues. Every attempt was made to remain consistent in providing a minimal level of detail in assisting participants stuck on certain steps in the programming process.

5.3External Validity

BLAST is commonly use software in bioinformatics, and FastA is a prevalent data format, therefore the task was both non-trivial and realistic. Although participants were specifically chosen due to their diverse computer proficiencies, because of the relatively small sample size, the results should be interpreted as indicators for further research, rather than definitive evaluations.

6.CONCLUSIONS

Based on the observations made in this pilot study, a full study could be executed with better documentation about the Taverna interface available for participants. Documentation could be specifically tailored to address end-user-specific comprehension issues discovered here. The distribution of responses by different groups to some cognitive aspects of the programming could act as a guide for questionnaires and interview questions in a larger study, especially when they differ from the researcher's original expectations. The pilot study provides some clear trend data, but is even more important in challenging the researcher's invalid assumptions (e.g. error proneness, premature commitment) about the programming environment.

We do not suppose that end-users cannot learn Taverna on their own, but the question is would they bother trying? The perceived benefits must outweigh the risk of failure, and the cost of learning. Both end-users and programmers perceived value in the communicative aspects of the workflow, and we are beginning to see such workflows in literature [11][12]. Perceived risk of failure and learning costs are higher in less computer-oriented users, and therefore EUD use of Taverna for life science researchers may have a limited sized audience: those with larger task automation requirements. This is somewhat unfortunate, because the communicative benefits (and other beyond the scope of this paper, such as auditing) are lost to the larger audience.

Many of the problems in Taverna are related to the user interface, or program installation, and therefore could be addressed by motivated developers. The major, more fundamental issue is data input types/formatting/connecting, which were showstoppers for users in this study. Taverna is fundamentally data format agnostic. Given users' reported willingness to re-use and modify analysis workflows, it may make more sense to provide end-users with existing workflows

that accept and enforce semantically typed data. These workflows could then be enacted or modified as needed by the end user. The BioMOBY Semantic Web Services protocol provides data type enforcement, but its Taverna plug-in [13] cannot. This leads us to suggest that a larger audience of Life Science researchers may adopt workflow programming as cognitively simpler Moby environments [14][15] reach a wider audience. These hypertext-based tools fulfill users' immediate analysis needs, *and* their ability to export Taverna-workflows (based on recorded user browsing) can be used as stepping stones for programming requiring more attention investment.

7.REFERENCES

[1] Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M. R., Wipat, A., and Li, P. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics, 20*, 17 (Nov. 2004), 3045-3054.

[2] Myers, B. A., Ko, A. J., and Burnett, M. M. Invited research overview: end-user programming. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22-27, 2006). ACM Press, New York, NY, 2006, 75-80.

[3] Oinn, T.., Greenwood, M., Addis, M., Alpdemir, M.N., Ferris, J., Glover, K., Goble, C., Goderis, A., Hull, D., Marvin, D., Li, P., Lord, P., Pocock, M.R., Senger, M., Stevens, R., Wipat, A. and Wroe, C. Taverna: lessons in creating a workflow environment for the life sciences. *Concurrency and Computation: Practice and Experience, 18,* 10 (Aug. 2006), 1067-1100.

[4] Kulyk, O. and Wassink, I. H. C. Getting to know Bioinformaticians: Results of an exploratory user study, In *Proc. of the BCS HCI'06 International Workshop on Combining Visualisation and Interaction to Facilitate Scientific Exploration and Discovery* (London, UK, September 11-15, 2006). ACM Press, New York, NY, 2006, 30-37.

[5] Javahery, H., Seffah, A., and Radhakrishnan, T. Beyond power: making bioinformatics tools user-centered. *Commun. ACM, 47*, 11 (Nov. 2004), 58-63.

[6] Green, T.R.G. and Petre, M. Usability Analysis of Visual Programming Environments: A Cognitive Dimensions Framework. *Journal of Visual Languages and Computing, 7*, 2 (June 1996), 131-174.

[7] Hundhausen, C. D. and Douglas, S. A. 2001. Communicative Dimensions of End-User Environments. In *Proceedings of the IEEE 2001 Symposia on Human Centric Computing Languages and Environments (Hcc'01)* (September 05 - 07, 2001). IEEE Computer Society, Washington, DC, 127.

[8] Blackwell, A.F. First steps in programming: a rationale for attention investment models. In *Human Centric Computing Languages and Environments, Proceedings. IEEE 002 Symposia on* (Arlington, VI, Sept 3-6, 2002). IEEE Computer Society, Washington, DC, 2-10.

[9] Altschul, S.F., Madden , T.L., Schaffer , A.A., Zhang , J., Zhang , Z., Miller , W. and Lipman, D.J. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research 25*, 17 (Sep. 1997), 3389-3402.

[10] Wilkinson, M.D., Links, M. BioMOBY: An open source biological web services proposal. *Briefing in Bioinformatics*, *3*, 4 (Dec. 2002), 331-341.

[11] Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M.R., Li , P. and Oinn, T. Taverna: a tool for building and running workflows of services *Nucleic Acids Research, 34* (Jan. 2006), W729-W732.

[12] Song, Y.C., Kawas, E., Good, B.M., Wilkinson, M.D. and Tebbutt, S.J. DataBiNS: a BioMoby-based data-mining workflow for biological pathways and non-synonymous SNPs. *Bioinformatics, 23*, 6 (Mar. 2007), 780-782.

[13] Kawas, E., Senger, M. and Wilkinson, M.D. BioMoby extensions to the Taverna workflow management and enactment software. *BMC Bioinformatics, 7* (Nov. 2006), 523.

[14] Wilkinson, M. Gbrowse Moby: a Web-based browser for BioMoby Services. *Source Code for Biology and Medicine, 1* (Oct. 2006), 4.

[15] Gordon, P. M. K. and Sensen, C.W. Seahawk: Moving Beyond HTML in Web-based Bioinformatics Analysis. *BMC Bioinformatics, 8* (Jun. 2007), 208.