

Designing authoring tools for the creation of on-line behavioural interventions

Adrian Osmond¹, Jonathan Hare¹, Joe Price¹, Ashley Smith¹, Mark J. Weal¹, Gary Wills¹, Yang Yang¹, Lucy Yardley² and David De Roure¹

¹University of Southampton, Department of Electronics and Computer Science

{aco | jsh2 | jtp | ads04r | mjw | gbw | yy4 | dder}@ecs.soton.ac.uk

²University of Southampton, School of Psychology

l.yardley@soton.ac.uk

Abstract. Behavioural interventions are used by social scientists to effect change in a person's behaviour. The LifeGuide project is developing tools to enable the easy creation, deployment and trialling of Internet-based behavioural interventions. The use of on-line behavioural interventions is appealing as it can be more cost effective than face-to-face interventions, can deliver tailored advice at times that suit the participants, and can provide detailed statistical information that can be used to better understand behaviour or demonstrate the efficacy of the interventions themselves. The problem however is that developing on-line interventions is a complex, time-consuming task that often has involved high levels of specialist computing support in construction and delivery. The LifeGuide project is looking to put tools into the hands of domain specialists (psychologists, social scientists, health professionals, etc.) that enable them to easily construct their own behavioural interventions and deploy them on the Internet. This paper looks at the authoring tools currently being developed by the project, assesses their usability through case studies of interventions developed so far, and suggests where the project will look in the future to continue to improve the tools to meet the needs of the wide range of intervention authors.

1. Introduction

There are many times where behaviour change could be beneficial, both to an individual and to society as a whole. If more people could be convinced to stop smoking, for example, this would have a positive impact on their long-term health. Behavioural interventions are usually created in conjunction with experts in a given field, such as medicine in the case of the smoking example. An intervention will typically take place over a period of time, during which tailored advice will be given to a subject. The efficacy of the intervention is measured by evaluating whether or not the subject changes their behaviour in response to the advice.

The Internet provides a platform from which it is possible to reach millions of people. As websites can be available to people anywhere in the world at all times of the day and night, intervention researchers have been taking advantage of these factors and moving their interventions on-line (Jennett, 2003 and Griffiths, 2006). Internet-based behavioural interventions typically take the form of an interactive website, which asks questions to a participant, stores their responses and gives them tailored advice depending on how they respond. This ability to tailor advice to a particular situation may help to make an on-line

intervention more persuasive (Kreuter, 2000 and De Vries, 1999). These Internet-based behavioural interventions are typically run as trials, which may include non-standard website behaviour such as randomisation and stratification of participants into groups, with each group being shown a different version of the intervention.

Since Internet-based interventions are not as straightforward as standard websites, building an on-line intervention gives a researcher a choice of one of two options. The first is that they study how to use a given technology and build the intervention for themselves. This is fine if the researcher has some background in the technology or a lot of time and an eagerness to learn. However, this option will not suit most people who wish to author interventions - we should not expect domain experts to also be technology experts. The second option is to hire people who already have such expertise. Obviously this adds extra cost to the process and also leaves the researcher fully reliant on the software engineers. Change in an on-line intervention is a common occurrence. Interventions will often need to pass an ethics committee and have quantitative and qualitative pilot studies, all of which can produce changes to the intervention. Making these changes will result in the extra cost of rehiring the programmers, assuming that they have the time and the willingness to take on such work.

The LifeGuide project is a collaboration between computer scientists and behavioural psychologists. It aims to put tools in the hands of intervention authors to enable them to create Internet-based behavioural interventions directly, without the mediation of software engineers. The tools developed by the project also enable people with minimal technical expertise to author, edit, deploy and trial interventions on the Internet, gathering the associated data for analysis.

2. The current landscape for on-line interventions

Traditionally, behavioural interventions have been conducted face to face as this has many benefits. It allows an expert to get to know the people they are helping and make the advice that they give completely personalised to each individual. However, there are two main limiting factors to this approach - an expert's time can be very expensive and it is not infinite, restricting the number of people that they can see. The Internet, however, is an easily accessible platform that is always on, with vast quantities of potential participants for experiments. This has led to an increase in desire from behavioural scientists to develop Internet-based behavioural interventions (Murray, 2005 and Portnoy, 2008).

2.1 Current development of interventions

Interventions can undergo a lot of change from their conception to their final version. They are often piloted, shown to other experts for critiques and any changes made may have to be cleared ethically. Even once data gathered from the final study has been analysed, another hypothesis can be made, forming another study and the cycle begins again. This cycle is shown in figure 1. Throughout this process, authors of on-line interventions want large amounts of control over how their interventions behave and look in order to make them as effective as possible. Since on-line interventions aim to replace an expert with a computer that delivers the advice of the expert, questions lie at their core. The expert sets these questions and advice is tailored for the subject based on the responses that are given. On top of authoring questions, creating an Internet-based intervention will likely include many of the following:

- Creating pages with text, images, videos and interactive elements, all of which may need to be precisely positioned and styled
- Deciding whether or not to view a page and in what order to view pages based on answers given to questions
- Managing user accounts for participants, so that they can log in and take part in the intervention over multiple sessions and have their data recorded
- Running trials of the intervention including techniques such as randomisation and stratification of participants into groups and making all the data available to the author for download

These requirements currently leave people who want to author interventions no choice but to use bespoke software with their interventions. Most behavioural scientists will not be able to create such software, meaning that they have to turn to software engineers to help them to develop their intervention, as well as make changes to it.

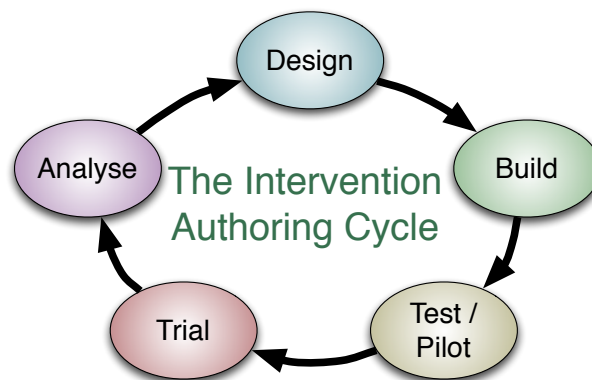


Figure 1. A diagram of the cycle for authoring an intervention

2.2 Lack of existing software

It is possible that some Internet-based interventions could be created using an HTML editor such as Front Page, Dreamweaver or iWeb. These are excellent at creating web pages, however, they are not designed for making interactive web pages – pages that change their content depending on different conditions. Systems designed for creating adaptive hypermedia such as AHA! (De Bra, 2003) would be too technical for most intervention authors. A further problem of authoring using an HTML editor is that the pages that are created will not be set up to record the data generated by participants without browsing server logs, which even many technologically savvy users would shy away from.

Other on-line interventions could be constructed using web-applications for conducting surveys online. There are many of these tools – SurveyMonkey, Opinio and Zoomerang are three such examples. At a first glance many of them look like they would be suitable for producing Internet based behavioural interventions and in the simplest case they might be. They allow the creation of questions of many kinds, showing questions (or not) based on responses to previous questions and even the dynamic alteration of text inside a page. They also log responses and allow them to be downloaded by authors. They fail to meet the requirements for more complex interventions however, as none of them allow for the laying out of the whole page that a tool like an HTML editor does. On-line survey tools are also not

designed with a framework for running trials and do not usually allow for repeat visits, a necessary feature for many interventions. Nor do they offer randomisation and stratification of participants, or make use of data gathered so far from other users, which authors may want to use to deliver statements such as “Your alcohol intake is above average.”

As paying for software engineers to develop an intervention and introducing a go-between into the process is not desirable and no software tools exist that would allow intervention authors to create their own Internet-based interventions, there is clearly the need to create a solution to these problems. It is the aim of this project to make LifeGuide that solution.

3. Developing LifeGuide

Two separate tools are needed to perform the tasks set out in the requirements – server side software to deliver the interventions to participants and log their data and an authoring tool to be used by authors to create these interventions in the first place. During the analysis of the requirements of the authoring tool it became apparent that this tool would also need to be thought of as two different sections, which, although they would be integrated into the same piece of software, could be thought of as distinct from each other. The first was the page author, which would act like an HTML editor, allowing easy creation of content. The second was the logic editor, a tool to allow an author to decide which pages (and parts of pages) a participant should see and in which order. This means an author wishing to create an intervention would first use the page author to create the pages, populating them with questions and other content. They would then use the logic editor to create the logic behind the intervention. The authoring tool itself gathers the outputs of both the page author and the logic editor when the author saves their work and combines them to create the finished pages of the intervention. This intervention is exported to a single file and sent over the Internet to the server, which stores it in its intervention store, and “plays” it for participants. This architecture can be seen in figure 2.

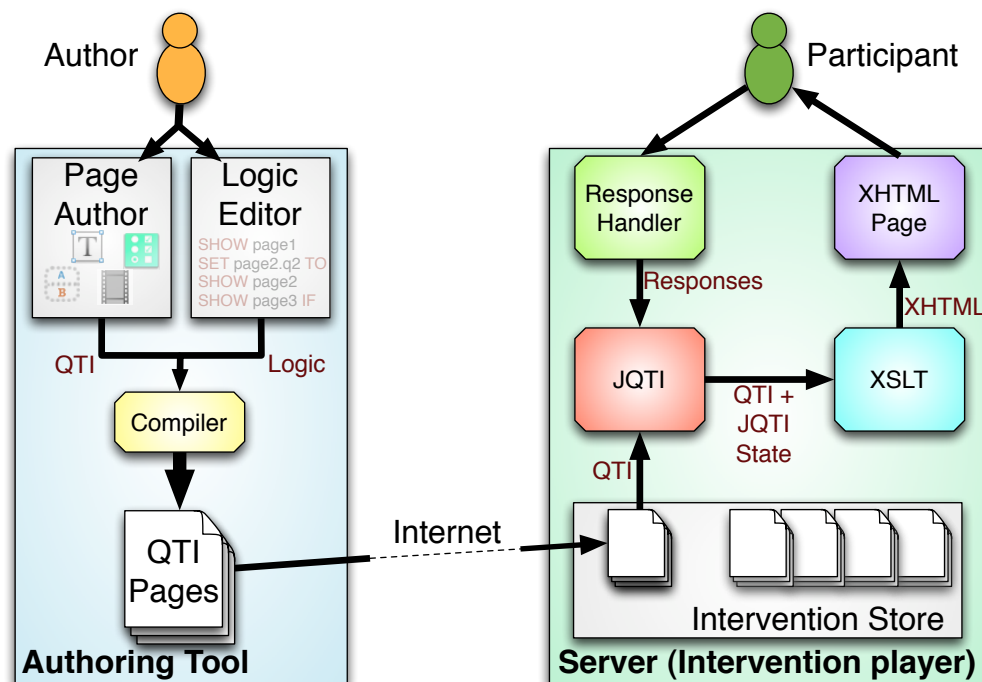


Figure 2. LifeGuide architecture diagram

3.1 Designing LifeGuide

Question and Test Interoperability (QTI) is an Extensible Markup Language (XML) specification for creating questions and tests, originally designed for e-Assessment, that has been used by a number of projects at the University of Southampton (Wills, 2008). A more technical description of how LifeGuide uses QTI can be found in the paper by Hare (2009). QTI forms the foundation for LifeGuide and, using tools built around QTI, provides logging of the data required by intervention authors. The server also has a Web front-end - a website that allows authors to upload their interventions, set up trials, view and download detailed data about how participants have used the intervention and communicate with other authors and researchers (see figure 3).

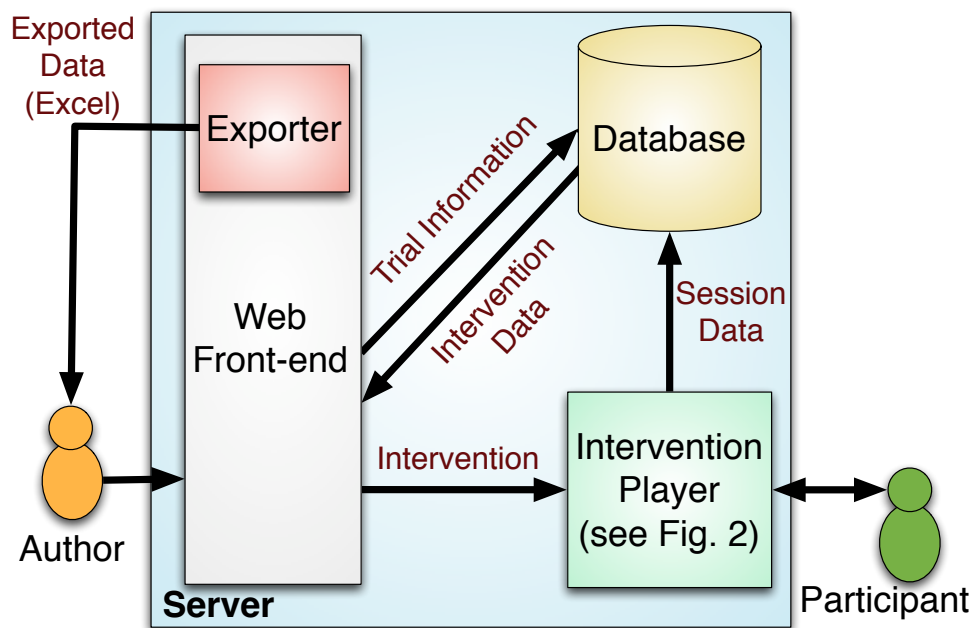


Figure 3. The Web front-end and data in LifeGuide

In many respects the logic editor is the most crucial part of the authoring tool. It is what sets LifeGuide apart from standard HTML editing software and allows an intervention to be more than a series of static web pages. It is in enabling authors to express this logic in a way that is natural and that doesn't require specialist training that the real challenge of the project lies. Initially the idea of a graphical interface for the logic was suggested, with pages in the intervention being represented as nodes and arrows linking pages showing the possibility of moving from one page to another. This is fine on very small and simple interventions, but it does not scale well. When translated to a medium sized intervention, which has many different conditions for reaching certain pages, it becomes more problematic. The requirement for an early working tool for logic creation was essential, so it was decided to adopt a textual approach in the first instance, with a view to development of graphical methods for logic authoring at a later date. The work by Ryder (2005) in this area will provide a good foundation for this.

When the methods that the behavioural psychologists used for noting how pages should link together and conditions for displaying certain pieces of information were analysed, it appeared that using English in note form was quite common practise. As a result of a series of co-design meetings, it was suggested that a scripting language be used that mimicked plain English as closely as possible. When the outline of this language was presented it was more

warmly received. Of course programming languages are designed to be unambiguous unlike freeform language. As we move away from this we run the risk of introducing ambiguity.

We have seen great improvements in Web applications with the increase in prevalence of JavaScript. Applications such as the Google Docs suite of tools demonstrate that it is now possible to do some very powerful things in Web browsers. We decided that the authoring tool should have a Web browser embedded in it, which would use JavaScript to create pages, as this would offer true “What You See is What You Get” (WYSIWYG) authoring. As JavaScript can also manipulate and transform XML documents, it would be possible to author pages directly into QTI using this approach, with the author only ever seeing a webpage. The contents of this browser make up the page author, which can be seen in figure 4.

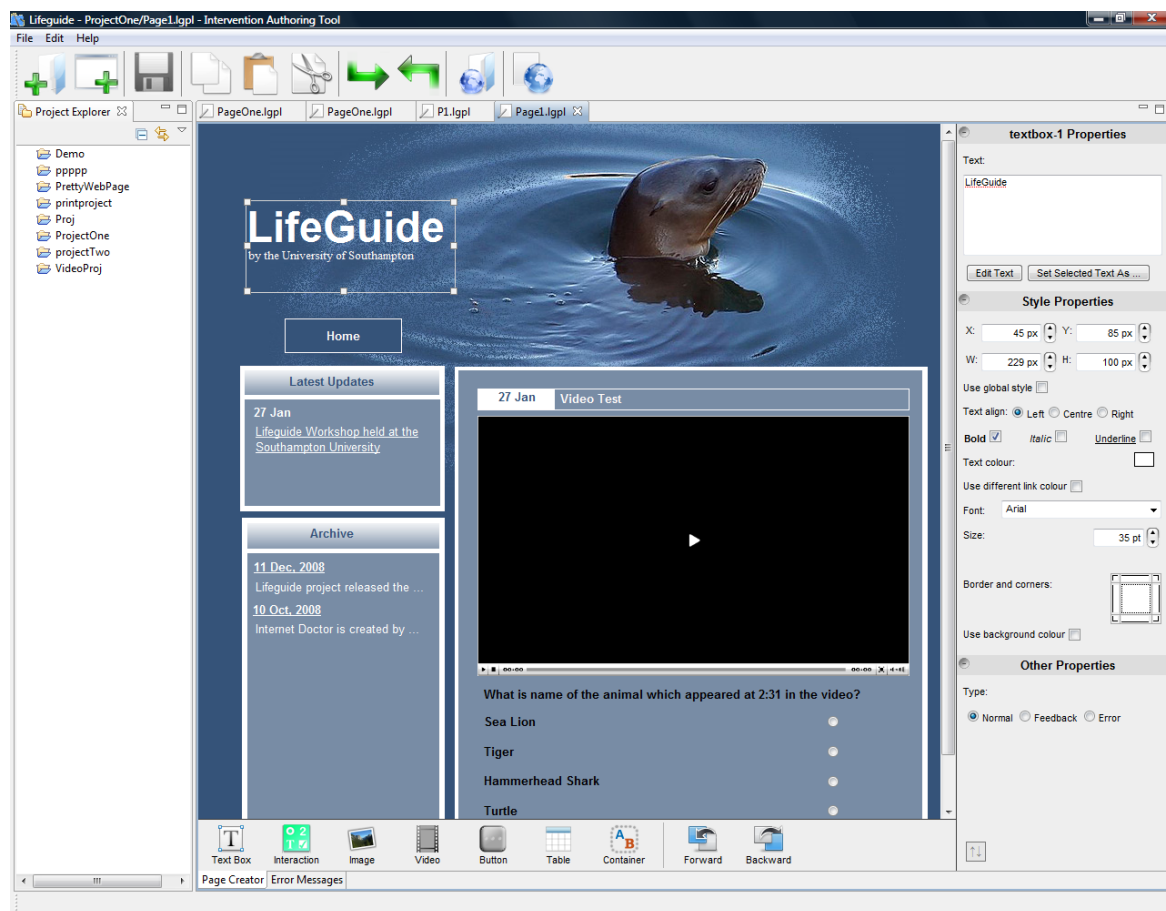


Figure 4. The page author editing a page inside the authoring tool

The script in the logic editor is compiled into QTI. This XML is then merged with the XML from the page author; resulting in a series of XML documents that the authoring tool can create into an IMS content package, ready to upload to the server.

3.2 Software development

A co-design process, as discussed by Pagliari (2007), was adopted with behavioural psychologists, which has ensured valuable early feedback on design and implementation and allowed regular reviews and refinements to be made to both. The development of the software has followed the agile methodology. Records of the tasks are kept both digitally and physically (the physical log of the state of progress takes the form of a wall divided into 3 sections – completed, being developed and to be developed.) Reassessments of the priority of

tasks usually come due to the needs of one of the intervention authors that we have aiding us in our development. These authors are creating real on-line interventions and have decided to use LifeGuide to help them do this. When they find parts in their interventions that cannot be created with LifeGuide these are discussed and the solution is usually added to the development plan. Having real users from an early stage has given the development team insights into the process of intervention authoring, what interactive elements are required by intervention authors and also where future authors are likely to struggle with LifeGuide.

3.3 Support for intervention authors

The early users of LifeGuide have been given direct access to the developers, so that when they encounter problems they can find solutions to them as quickly as possible. In certain important or urgent cases we have offered these authors the chance to come and work in close proximity to the development team. When this has happened it has resulted in rapid progress from the author but it can slow software development speed. Whilst this approach works well when there are only a few authors using the system, it will not scale when that number grows. In order to provide proper support for future intervention authors who wish to use LifeGuide, the development team have been logging problems that the users of the beta software have had. These problems form the basis for the help documents that are under constant development. Help is also provided in the form of several downloadable demo interventions, which show how to perform certain commonly occurring tasks in LifeGuide, such as creating sign up and log in pages, allowing interventions to take place over multiple sessions. The logic in these demos has comments throughout to explain how it works. We have also held several workshops in which attendees have been given a chance to learn how to use the tool to create a short intervention under the supervision of the LifeGuide team. These have served as a useful tuition tool and more workshops are planned in the future with a focus on teaching existing LifeGuide users new skills.

Whilst having systems of support in place is essential, we strive to create software that requires as little support as possible. For this reason, a number of common problems experienced by our early users have been identified and addressed.

3.4 Difficulties with developing LifeGuide interventions

Graphic design has proved slightly problematic for our intervention authors, with LifeGuide authored interventions sometimes being criticised as unattractive. Whilst it is entirely possible to make attractive looking interventions in LifeGuide, authors often struggle with making their work graphically striking because they are not graphic designers and so should not necessarily be expected to possess such skills. Some HTML editing experience also helps improve an author's ability to create more attractive interventions, as through such experience certain techniques and tricks can be learned which would also apply to LifeGuide; however, this experience cannot be assumed either. In order to help authors to create more visually appealing interventions we have taken the following steps. Firstly and most importantly, we inform authors of the importance of appearance and encourage them where possible to get a graphic designer to help them in creating the visual style for their intervention. Secondly, LifeGuide handles the rendering of elements such as questions for the author, offering set visual styles that should fit in with most interventions. Finally, LifeGuide will soon allow the creation of themes and templates for interventions, with some standard ones provided with the software. These will work in the way that PowerPoint offers themes for its slides.

In some cases interventions have required specific components that have not been easily achievable in LifeGuide. On the occasions when this has happened there has usually been one of two outcomes. The most common outcome is that, after some co-design meetings, a compromise is reached which allows the behavioural psychologists to achieve the overall effect that they were aiming for, but without the need to make time-consuming extensions to LifeGuide. Occasionally, if there is no way to achieve what is required, or if numerous authors will require the functionality in the future, the software is extended.

The server currently offers data export in Excel 2007 form, with variables forming the columns and each user session taking up a row. However, as the export converts raw QTI into Excel, it currently includes large numbers of variables that come from QTI's inner mechanics. This has confused some users, as they find it difficult to know which variables are useful to them. Also, Excel 2007 is not suitable to everyone – some people do not use Microsoft Office and others do not have the latest version. Using an older version of the Excel file format is not plausible as they limit the number of columns to 256, which may not be enough for larger interventions. To help alleviate these problems, we intend to do two things. Firstly, make the data available in more formats such as comma-separated values (CSV). We also want to allow users to select the columns that they would like to see in their exported data, potentially with suggestions from the server as to which might be the most useful to them.

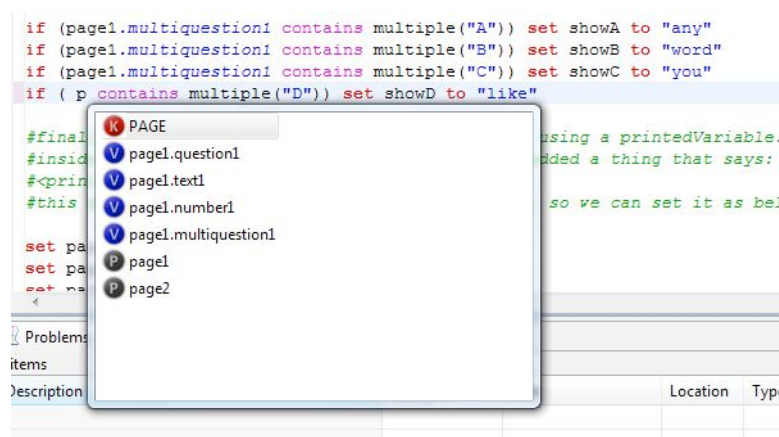


Figure 5. The logic editor's syntax highlighting and auto-completion of a demo intervention

Logic authoring has been the source of most problems for authors, though we had always predicted that this would be the case. Authors who are not used to any kind of programming or scripting have found this new experience to be challenging. This is not helped by the fact that most of the error messages that are presented to them at the moment are QTI error messages, which have little meaning to people who do not know how QTI works. Consequently, a simple mistake like misspelling a page name can result in a message like "Error when loading referenced item" and more unusual errors in the logic often generate even more cryptic messages. In order to help spot errors and to stop them from occurring in the first place, syntax highlighting and auto-completion of code were added to the logic editor. Figure 5 demonstrates these features. The idea of syntax highlighting is that words that the compiler recognises change to predetermined colours, giving an idea of which text it does not understand. Auto completion of code pops up a box when an author starts typing with words that they might want to type. In the case of LifeGuide, the logic editor knows the names of keywords and functions used in the scripting language, page names in the current intervention and the names of the interactive elements on every page. This decreases the chance that the author will make spelling errors, as many of the things that they are likely to

type will be automatically presented to them. In order to make the programming experience easier for authors, the compiler is being rewritten to improve its ability to identify the sources of errors, and therefore give more meaningful, plain English error messages. Even these measures, however, are not likely to help people who struggle to get into the mindset of programming. We see making LifeGuide a tool that all can use as extremely important, as it was the less technical people that LifeGuide was conceived for in the first place. The area of simple to use logic editing, therefore, represents our biggest challenge in the future. It cannot be understated however, that logic authoring is intrinsically complex and the task itself (even on paper) is often a stumbling block irrespective of the technology.

4. Conclusions and future work

Whilst there are many improvements that could be made to LifeGuide, it works well as a platform for developing interventions. When the features that are currently being developed have been finished, LifeGuide will be a polished set of tools suitable for use throughout the intervention authoring process. It is in this ability to develop and manage an intervention, from authoring pages all the way to running trials, that the appeal of LifeGuide lies. The co-design process means LifeGuide has been built with the end user in mind. We believe that, as a tool available to all intervention authors, it must be an attractive alternative to hiring software engineers. Although users of beta versions have not always had a smooth time with the software and have needed support from the software developers, the development team have seen the amount of help needed decrease with each iteration of LifeGuide.

The LifeGuide authoring tool was built on Eclipse, a popular piece of software for programming, which has given many positives to LifeGuide, but there are a few negatives associated with this decision too. Eclipse was designed with programmers in mind and some of the aspects of its user interface are quite technical. LifeGuide, on the other hand, has been designed for people with no programming experience, who may not understand all of Eclipse's errors, dialogue boxes and wizards. Creating LifeGuide specific text and interfaces for all of the parts of Eclipse that are used in LifeGuide would be a time consuming task and other work has been seen as more of a priority up to this point. However, making the LifeGuide authoring tool feel less like Eclipse might help increase usability for less technical users. On top of these changes, we also plan to improve on the Eclipse "project explorer", a pane allowing the author to select which page they want to edit, to give it previews of pages in the same way that PowerPoint gives previews of slides on its left-hand side. We also plan to improve the user experience by changing the method used for uploading interventions to the server. At the moment, in order to put an intervention on the server it first has to be exported, then uploaded from a Web browser. In the future the authoring tool will be able to log an author into the site and upload an intervention automatically without the need to open a browser.

The main area that we wish to focus our future design efforts on is that of logic editing as this is seen as an area where large benefits can be gained from simplifying the process. We intend to look into the possibilities of building a graphical editing tool for LifeGuide logic. Whilst text based logic editing has provided a good start for LifeGuide, as the first stage of development draws to a close, the near future seems like a good time to research how this could better be achieved.

References

- Jennett, P., Affleck, H., Hailey, D., Ohinmaa, A., Anderson, C., Thomas, R., Young, B., Lorenzetti, D. and Scott, R. (2003): 'The socio-economic impact of telehealth: a systematic review.' *Journal of Telemedicine & Telecare*, vol. 9, pp. 311-320.
- Griffiths, F., Lindenmeyer, A., Powell, J., Lowe, P. and Thorogood, M. (2006): 'Why are health care interventions delivered over the internet? A systematic review of the published literature.' *Journal of Medical Internet Research*, vol. 8, p. e10.
- Kreuter, M., Farrell, D., Olevitch, L. and Brennan, L. (2000): *Tailoring health messages: customizing communication with computer technology*. Erlbaum, London.
- De Vries, H. and Brug, J. (1999): 'Computer-tailored interventions motivating people to adopt health promoting behaviours: introduction to a new approach.' *Patient Education and Counseling*, vol. 36, pp. 99-105.
- Murray, E., Burns, J., See, S., Lai, R., and Nazareth, I. (2005): 'Interactive health communication applications for people with chronic disease.' Cochrane Database of Systematic Reviews, Issue 4, Art. No.: CD004274.
- Portnoy, D., Scott-Sheldon, L., Johnson, B., and Carey, M. (2008): 'Computer-delivered interventions for health promotion and behavioral risk reduction: a meta-analysis of 75 randomized controlled trials, 1988-2007'. *Preventive Medicine*, vol. 47, pp. 3-16.
- De Bra, P., Aerts, A., Berden, B., de Lange, B., Rousseau, B., Santic, T., Smits, D. and Stash, N. (2003): 'AHA! The Adaptive Hypermedia Architecture' Conference on Hypertext and Hypermedia, 26-30 August 2003, Nottingham, UK
- Wills, G., Hare, J., Kajaba, J., Argles, D., Gilbert, L. and Millard, D. (2008): 'Assessment Delivery Engine for QTiv2 Tests' International CAA Conference, 8-9 July 2008, Loughborough UK.
- Hare, J., Osmond, A., Yang, Y., Wills, G., Weal, M., De Roure, D., Joseph, J. and Yardley, L. (2009): 'LifeGuide: A platform for performing web-based behavioural interventions' WebSci'09: Society On-Line, 18-20 March 2009, Athens, Greece.
- Ryder, B., Soffa, M. and Burnett, M. (2005): 'The Impact of Software Engineering Research on Modern Programming Languages'. *ACM Transactions on Software Engineering and Methodology*, October 2005, pp. 431-477.
- Pagliari C. (2007): 'Design and evaluation in eHealth: challenges and implications for an interdisciplinary field.' *Journal of Medical Internet Research*, vol. 9, p. e15