

Pimp my questionnaire!

Moreto, a Toolkit for Online Research

Willem van Bergen B.Sc.
Dirkjan Bussink B.Sc.

Monito

Enschede, July 5, 2006



Management Summary

The Moreto toolkit has been developed as a system that can aid a developer at Monito in developing online questionnaire systems. The goal of the toolkit is to provide a system that can be easily adapted to the specific needs of certain questionnaires. The target market of the toolkit is therefore also the market in which the off-the-shelf systems for creating online questionnaires do not suffice, because of for example the need for a more flexible navigation.

The assignment consists of three parts, first an analysis of what the actual requirements are for the toolkit, both from Monito's point of view as well as the customer's view. The second part is the actual design of the system and the third part consists of building a prototype that explores the actual practical solutions needed for the envisioned system. These three steps resemble the first three steps of the engineering cycle, and therefore this approach was adopted. These first three steps are the problem investigation, the solution design and the solution validation.

The problem investigation started with a domain analysis. This analysis consisted of five major parts, first the stakeholders that are important have been analysed. The most important ones here are Monito themselves, both in the role of management and the role of developers, Monito's customers, the designers of online questionnaires, the questionnaire respondents and also Monito's hosting provider on whose platform the system has to run. Each of these stakeholders has their own goals, which the toolkit tries to satisfy. For example, customers want high quality products with few costs, while the developer's main goal is the ease of use of the toolkit. The entire goal decomposition tree with both the stakeholders and their goals can be found in figure 2.3.

The second part is an analysis of the environment. This environment mainly consists of the hosting platform of Byte, Monito's hosting partner. The third part includes a comparison with existing off-the-shelf questionnaire applications. These usually allow the user to interactively compose a questionnaire using predefined components such as a multiple choice question. This makes is very easy for the user, but also limits them in their design of the questionnaire.

In the fourth part, we looked at the existing systems that Monito has built for their customers. These are more complex questionnaires that are custom made for the specific questionnaire. One of the goals of the toolkit is to implement these kind of applications with the toolkit, so the features of these existing systems are very important. Some of these aspects are a flexible ordering of questions or custom navigation in a questionnaire. A detailed overview of these properties can be found in appendix A. The final part looks at some domains that are similar to that of online questionnaires and states that it is important to see whether the toolkit could be used in those domains.

This domain analysis resulted in a large list of requirements for the toolkit, some of which are more important than the other. These requirements have been divided into different categories, namely *Core*, *Common* and *Custom*. The first category includes all requirements that must be met, the second category denotes functionality that would be often useful and the last category includes functionality that should be possible to implement with the toolkit. The complete list of requirements can be found in chapter 3.

During the design of the toolkit, several general design aspects have been taken into account. The toolkit should be very extensible, easy to use and productive for a developer. The first principle resulted in the extensive use of subclassing and also in making sure that the different parts of the toolkit are loosely coupled. The second principle resulted in the design of an API that a programmer can use in conjunction with appropriate documentation. The final principle resulted in the use of existing software libraries that can greatly improve productivity by not having to create cumbersome code over and over again.

On the central parts of the toolkit is the part that manages the actual data that is obtained during a research. This part is called the *data manager* and provides a complete API that makes managing this data very easy. The concept that lies at the heart of the data manager, is the concept of a matrix. Each row in this matrix represents the data for a certain subject, while each column represent a certain answer or property belonging to that specific subject. This part of the system has been formally specified. This specification can be found in appendix B.

The second important part is the actual structures of questionnaires. A questionnaire is represented by a series of pages where each page can contain so-called items. Such an item can for example be a question, but also a piece of static content. This model allows for very flexible questionnaires, but also creates the necessary structure. One of the important aspects of the toolkit is that it should support a lot of different question types. This part therefore got special attention and it has been made very easy to do. A developer can simply extend an item with the necessary functionality for a certain question.

The navigation through this questionnaire structure is also made very easy. A developer can implement jumps to different pages separately for a certain questionnaire, for example based on a previous answer from a respondent. Also answers are remembered when navigating back and forward, as well as meta information on when the respondent started, where he or she is currently and when they finish.

The third part is how respondents participate. The toolkit should support anonymous participation, but also invited participations. These invitations could for example be sent by e-mail with a specific authentication code. The system also supports the generation of authentication codes separately, so these codes can be sent by a letter to the targeted respondent group.

All of these efforts are of course useless if the obtained data can not be exported or analysed. Therefore this is also an important part. Results can not only be obtained afterwards, also during a questionnaire, so Monito's customers can see how the questionnaire is progressing. The data manager API provides basic functionality that can be used to do some analysis, but the main usage will be a simple dump of the data so Monito's customers can analyse the data themselves using data analysis software, e.g. SPSS or Excel.

The implementation of the prototype satisfies most of the requirements. The ones that are not met are all still implementable in the current system. These are simply not yet implemented because of time constraints. The toolkit was also tested with a real life test case. This was a relatively simple project, but both the customer and the developer were very satisfied with the toolkit. Most of their comments were on additional functionality they would like to see and some improved documentation for developers. Especially more documentation on how to get started would be useful for developers when they use the toolkit for the very first time.

One of the bigger issues is the performance in the hosting environment. This problem did not occur on the development environment, because of the different setup and less load on the web server and the database. Future improvements of the hosting platform could easily fix these problems, but this of course depends on whether the hosting partner will do this.

Most of the features of existing custom made systems have been included, or can easily be included into the toolkit. Here the general result is that a lot is possible, but some development needs to be done, much like the currently not satisfied requirements.

The modular design resulted in a system where different components could be used separately and possibly even for applications in other domains. Especially the data manager is very usable on its own, it is designed as a single system that does not depend on for example the questionnaire structure. As for application in other domains, this is possible when the application collects data that fits the matrix model. Collecting preferences for a planning system would for example be more useful than using the toolkit for creating an online web shop.

The overall conclusion is that toolkit satisfies most of the requirements and goals of the different stakeholders. What is not satisfied can be easily implemented given more time. The toolkit can be seen as a rough diamond that needs more polishing to create a very good solution. This prototype is a good base to work further on. Some of the work that needs to be done is promoting the toolkit to customers, some internal design changes in the invitation process and more functional improvements that have not been implemented because of time constraints.