Kindsmüller, M.C., Leuchter, S., & Urbas, L. (2003). Design Patterns for Web-Based Experiment Control. In H. Strasser, K. Kluth, H. Rausch, & H. Bubb (Eds.), *Quality of Work and Products of the Future* (pp. 584-587). Stuttgart: Ergonomia Verlag. http://www.safety-critical.de/doc/iec03.pdf.

Design Patterns for Web-Based Experiment Control

Martin Christof KINDSMÜLLER, Sandro LEUCHTER, & Leon URBAS

MoDyS Research Group, Center of Human-Machine-Systems, Berlin University of Technology Berlin, Germany

Abstract. A taxonomy of requirements for advanced experiment control systems (ECS) is derived from basic and application-oriented research scopes independently from implementation considerations. Every demand within the taxonomy is described in detail with an emphasis to its potential impact on the ECS-design. On top of the Java 2 Enterprise Edition architecture for web-enabled scalable business application development the single controller cookie framework (SCCF) is proposed to facilitate development of ECS. The advantages of this framework over other ECS are discussed according to the taxonomy. Other components that can be plugged into the framework to achieve certain functionality (e.g. logging) are mentioned. SCCF and accompanying components form the MoJavEE, the MoDyS Java 2 Enterprise Edition Experimental Environment.

1. Introduction

MoJavEE is a component framework for creating experimental control systems (ECSs) for a wide range of research data gathering applications in ergonomics and psychology. It was initially developed to conduct a series of experiments within the fields of operators' trend interpretation (Kindsmüller et al., 2002) and timing experience (Schulze-Kissing et al., 2002) in the domain of chemical process control. Within this paper we show that MoJavEE is in no way limited to this kind of application but can be seen as a generic ECS architecture that offers some strong advantages compared to other experimental control systems.

MoJavEE is based on the programming language Java and the Java 2 Enterprise Edition application architecture and middleware. Java is an object-oriented language with many advantages over other programming idioms: It offers a comprehensive standard library (API) for many kinds of tasks, it is common, has error preventing programming constructs like exception handling or avoidance of pointer arithmetic. Java programs are portable though byte-code interpretation. J2EE (Java 2 Enterprise Edition) defines a scalable multi-tier architecture and middleware for complex web-based (business) applications on the basis of Java. Access to J2EE applications is often done with web-browsers. The application itself consists of active web-components (JSP & servlet) and separate business logic that are defined in Java. Access to database and other infrastructure can be easily achieved.

MoJavEE is based on the model-view-controller (MVC) design pattern (Singh et al., 2002 p. 348). Generally in object-oriented design it is applied to relate GUI-elements (views) to the underlying, logical structure of data to be displayed (model) and elements to change values (controllers). The advantage of this pattern for development is a clear separation of responsibility in code which leads to side effect free modifications of models, views, and controllers.

2. Requirements for contemporary experimental control systems

Even if we limit our focus to research on ergonomics and psychology there is still an almost infinite number of ways for computer based research data collection. The decision about which way is the right one is often crucial but can only be made in regard to the special type of context situation in which empirical data is produced. To sketch the demands on a contemporary ECS table 1 gives an overview of a taxonomy arising from this context situations by listing the main aspects and the major options.

Table 1: *Taxonomy of demands for experimental control systems.*

Aspect	**Option 1**		**Option 2**
research scope	basic research	…	application-oriented
research approach	experimental	…	pre-experimental
research material	plain text	…	interactive microworlds
number of subjects per run	single	…	all
spatial proximity	in house		remote
investigators involvement	supervised		non supervised
subjects interact with each other	interaction		no interaction
randomized sequences	required		not necessary
sequencing	interaction based		fixed/pre-defined
anonymity	possible		impossible
accessibility	public		closed group
data collected	responses (yes-/no-answers, ratings, errors), response time		

For the aspects in the upper part of the list like research scope, the listed options could be seen as extreme poles on a continuum, i.e. in between those options there is a number of hybrid versions possible that are neither option 1 nor option 2 or both. For other demands such as spatial proximity or anonymity the listed options just exist as antipodes with no continuum in between. Either there is anonymity or not.

Below every demand within the taxonomy is described in detail with an emphasis to its potential impact on the ECS. Research scope ranging from basic to application oriented research is a very prominent category to classify research but has little direct effect on the design decisions for ECSs. The research approach's immediate influence is not that big either and can be mainly located within the ECS's logging module. A pre-experimental approach usually requires a more general logging method (logging of all available data) whereas experimental designs can make it very easy to tailor an ECS to deliver logging data that can be directly fed into statistical packages for immediate hypothesis testing. The richness of research material is usually a real challenge for an ECS, as virtually every kind of media format is a possible research item and should therefore be supported by the ECS. Furthermore all kind of interaction with the aforementioned media formats should be supported as well. Due to limited resources like special hardware devices or especially trained investigators or due to the research setting an ECS has to facilitate a huge variety in the number of subjects examined per run (ranging from a single subject to all subjects of the sample).

Coming to the antipodes within the taxonomy spatial proximity (in house versus remote) can be quit difficult to support by an ECS (especially remote experimenting). Investigators involvement (supervised versus non supervised) in combination with the number of subjects per run is often crucial for efficient experimenting. The necessity of running supervised single subjects experiments makes economic data collection almost impossible. Therefore some effort is put into the attempt of making an ECS "intelligent" enough to either replace the investigator in single subject runs or supporting the investigator to be able to migrate from single subjects runs to groups runs. A large impact on ECS-design is generated by the kind of inter-subject interaction that has to be achieved within the research setting. Controlling the communication between subjects (one-to-one, one-to-many, many-to-many) requires a fundamentally different ECS-architecture than a mere subject-machine-interaction. Randomization of sequences and interaction based (in opposite to fixed) sequencing of steps are common demands and should be easily achieved within the ECS. Subjects being sensitive to the "collection of personal data" as well as institutional or legal needs may demand that subjects should be able to interact with the system in complete anonymity. This should be supported by an

ECS without mixing up data from different subjects. The innate structure of the research setting or demands by the customer can make it necessary to control the accessibility of the system. The ECS should make it easy for subjects to gain access to the system whilst prohibiting abuse by blocking unauthorized access. The last aspect in the taxonomy, the data collected, is again a crucial one for the ECS-design. Depending on the research design all kinds of response types like yes-/no-answers, rating decision, text input, etc… and response times with different requirements on resolution could be necessary.

Looking at the taxonomy as a whole it becomes clear that neither all combinations of aspects make sense nor all of them are possible within all feasible research contexts. While some of the aspects are easy to combine by every ECS others are still making a lot of sense but are not easy to combine like “being in a closed group” and “staying anonymous” at the same time.

3. The MoJavEE component framework approach

Bearing in mind the demands mentioned in the last paragraph it becomes evident that it is difficult if not impossible to fulfill every demand within a single ECS. Therefore MoJavEE was implemented using a component based framework approach instead of a monolithic architecture. Taking into account that web-based experimental research techniques gained some acceptance over the last years (Reips, 2002) and that they can be supplemented by browser-based components to be successfully used in lab settings (Polkehn & Wandke, 1999), we have chosen web-based technology like the HTTP and HTML standard to act as an implementation base for MoJavEE. In the remainder of this section we focus on how certain aspects of this concept are implemented within MoJavEE, especially features that are known as flaws of either conventional ECSs or other web-experimenting environments. A more general disquisition on the advantages of web-experimenting can be found in Reips (2002).

As mentioned before a conventional ECS usually has difficulties in supporting all media formats that are possibly of interest in experiments and surveys in ergonomics. Using a HTTP/HTML-based client-server approach leaves the presentation part to a common web-browser (i.e. Netscape or Microsoft Internet Explorer), therefore every media format that is accepted by the web-browser can be used. Beside HTML-documents a huge variety of image, audio, and video formats is supported. Over and above a static presentation of the aforementioned media formats there can be almost any kind of interaction with this elements. This can be achieved either by enriching static HTML-documents with scripting code (i.e. JavaScript) or by embedded Java applets. Not quite as complicate as the data delivery part but still complex is the data gathering part. Responses can range from binary decisions, ratings to free text answers. For some subclass of experiments timing information is very important. Server side timing information is very easy to access but not accurate enough for all kind of applications. Therefore MoJavEE supports client side timing by using Java applets or JavaScript methods to measure response time with a sufficiently high accuracy.

The backend side of MoJavEE is based on J2EE JavaBeans, JSPs and servlets and utilizes the MVC design pattern. The model (implemented as JavaBean) consists of the experiment object which is a representation of the experimental design. Experimental designs of varying complexity can be mapped onto this experiment object using a hierarchical structure of experimental chapters, items and passes. The experiment object is created at the start of a subject’s experimental session and represents the subject’s progression within the experimental procedure. The life cycle of the experimental object ends by the termination of the subject’s experimental session. The view within MoJavEEs MVC pattern is by no means limited to JSPs, but can virtually be of every entity a common web-browser can process (see above).

This is achieved by the single servlet controller design (every view is called by the experimentor servlet and gives back control to the same servlet) and J2EE's session tracking concept (MoJavEE uses cookies instead of URL rewriting to facilitate the use of static HTML documents and other media formats as views). This architecture will be further referred as the single controller cookie framework (SCCF).

When a subject requests the experimentor servlet for the first time the experiment object is created, which is instantiated with static information about the experimental design from a serialized object, a property file or a database. Treatment, sequence and item assignments can be taken directly from this static information or computed on the fly at creation time or even at any time later within the experiment. The SCCF makes it very easy to keep track of the subject's progression within the experiment and to rearrange the sequence for every subject based on prior behavioral data. Keeping the views simple and multifunctional because they do not need to contain any sequence information is another advantage of the SCCF. Thus the SCCF satisfies all kind of sequencing needs discussed in section 2. The anonymity aspect is implicitly handled as subject IDs can be automatically assigned and kept internal, therefore it is simple to conduct anonymous or double blind experiments. The support of multiple subjects per run was another issue discussed in section 2. MoJavEE offers multi user capability at no extra cost as it is inherent to the SCCF. In addition to mere multi user capability the system supports central data logging (plain text, XML, SQL database) and zero installation experimenting (on any TCP/IP-networked computer platform). In conjunction with the automatic subject ID assignment the multi user capability allows for large groups of subjects being examined in parallel by low assignment of investigators. Last not least one large benefit that is hard to achieve by other ECSs is the fact that due to the client/server architecture MoJavEE has online access to all data of all subject under examination. Having instantaneous access to this data and a powerful system for processing it on the back-end side, enables investigators to run e.g. matched samples experiments by automatically establishing experimental twins. Equally well supported are experimental settings with controlled subject inter-communication or controlled shared knowledge. MoJavEE so far is not an out-of-the-box solution but versatile, powerful, stable and fairly easy to tailor to the investigators needs.

This work has been supported by Volkswagen Foundation in the program "Junior Research Groups at German Universities".

4. References

1. Kindsmüller, M.C. & Urbas, L., 2002. Der Einfluss von Modellwissen auf die Interpretation von Trenddarstellungen bei der Steuerung prozesstechnischer Anlagen. In: Grandt, M. & Gärtner, K.-P. (Eds.), *Situation Awareness in der Fahrzeug- und Prozessführung*. DGLR, Bonn. 131-152.
2. Polkehn, K. & Wandke, H., 1999. Web Supported Experiments. In: Reips, U.-D., Batinic, B., Bandilla, W., Bosnjak, M., Gräf, L., Moser, K. & Werner, A. (Eds.), *Current Internet science - trends, techniques, results*. Online Press, Zürich.
3. Reips, U.-D., 2002. Standards for Internet-based experimenting. *Experimental Psychology, 49* (4), 243-256.
4. Schulze-Kissing, D., Urbas, L., van der Meer, E. & Leuchter, S., 2002. Der Einfluss kognitiver Beanspruchung auf zeitgesteuertes Verhalten in Mensch-Maschine-Systemen. In E. van der Meer, H. Hagendorf, R. Beyer, F. Krüger, A. Nuthmann, S. Schulz (Eds.), *43. Kongress der Deutschen Gesellschaft für Psychologie. Humboldt Universität zu Berlin,* Programm, Abstracts. Pabst, Lengerich. 136-137.
5. Singh, I., Stearns, B., Johnson, M. & the Enterprise Team, 2002. *Designing Enterprise Applications with the J2EE Platform, Second Edition*. Addison-Wesley, Boston.