USING WEB-BASED MEME MEDIA TECHNOLOGIES TO CREATE AN INTEGRATED VISUAL ENVIRONMENT FOR CLINICAL TRIALS

J. Sjöbergh **M. N. Kuwahara** **Y. Tanaka**

Meme Media Lab, Hokkaido University, Japan

ABSTRACT

We present a system for designing, carrying out, and analyzing the results of clinical trials, or more specifically cancer treatment trials. This system was built using Webbles, a type of next generation Web intelligent objects designed to make services and functionality as easily reusable as data content is with current technologies. We discuss the benefits of using this technology, some problems we ran into, and some design decisions we made.

1. INTRODUCTION

Clinical trials to evaluate what treatment works best against a certain disease require a lot of work, including data acquisition, management, retrieval, and analysis. There is the planning and design of the trial, what to test and how to test it. There is the actual running of the trial, treating patients and collecting data on how things are going. There is also the analysis of the collected data to understand what treatments should be used or what should be further studied etc.

These steps are of course connected, a good design of the trial and creation of proper tools such as data forms helps the physicians treating patients in the trial to report the wanted data in an easy and unintrusive way. Mistakes in the design phase or lack of cooperation in the running of the trial may mean that the data needed in the analysis step are not available.

The design of clinical trials still often do not take advantage of the advanced data processing capabilities available nowadays. The trial design, e.g. what treatments should be given when to different groups of patients, is still often created by hand on paper.

We have created a system for designing, running, and analyzing clinical trials, more specifically trials concerning cancer in children. The system is highly visual and interactive in nature, and thus gives a good overview of what is going on. The system has been built using Webbles, a type of visual intelligent objects for Web development. Using these visual and interactive intelligent objects it has been fairly easy to develop an advanced system quickly.

In this paper we describe how this intelligent Web object technology was used in a practical situation. We describe the application we created, some difficulties we had, some design decisions we made, and show the benefits of using this technology.

2. RELATED RESEARCH

Research has been done on integrated support systems for clinical trial design and management. Many things go into planning and designing a clinical trial. There are AI based support systems that check various constraints in the trial design stages and warns if the knowledge base indicates that the current trial contains problems. One such system is the *Design-a-Trial* system by Modgil and Hammond (4), which for example warns the trial designer when creating a trial using a not recommended medicine for the control group instead of the current standard medicine.

There are systems such as those by Shankar et al. (5) and Wong and Gibbons (10) that allow graphical visualization of a trial during design, showing the work flow of the trial events, displaying graphical forms for data input. While visualization when showing the trial is common, we have not seen any system except our own that allows graphical interaction in the work flow planning, allows building the trial by dragging and dropping the graphical representations of the trial elements, and provides integrated support throughout the trial using the same visual representation at all times in the workflow, not only in the design phase.

3. WEBBLES AS AN ENABELING PLATFORM ARCHITECTURE

Webbles are the latest generation of the Intelligent Pad system which in turn is a Meme Media system (Tanaka (7), Tanaka et al. (8)). Meme Media technologies are aimed at allowing people to easily reedit and redistribute intellectual resources. The idea is that functionality in services should be as easily taken from one place and reused in another as text or images (data) can be reused today.

Webbles are media objects. A Webble can be many different things, a media player, a text, a wrapper for a database service, a service calculating currency conversions etc. Webbles can communicate with other Webbles through “slots”, which are externally available properties of the Webbles. Slots in different Webbles can be connected, so a value change of a property in one Webble is propagated to another Webble. Communication can be set up to be unidirectional or bidirectional. Webbles can be connected hierarchically in parent-child relationships. When the value of a slot in the parent is changed, all children are notified that something has changed. They can then choose to react by reading the values of slots they are interested in or in any other way that may be appropriate. For details on the inner workings of the Webble platform, see Kuwahara and Tanaka (3).

Webbles exist on the Web, and applications created using Webbles can be run in a normal Web browser. The Webble architecture requires the Microsoft Silverlight plug-in to run. This plug-in is easy to install but not available for all computer platforms. Webbles can make use of anything that is supported in Silverlight, which means that it is easy to make for instance a media player Webble or an image manipulation Webble. Webbles are made to be easy to reuse. When a Webble that does something requiring a lot of implementation work is needed, once this work has been done it can easily be reused in other applications or by other Webbles with very little extra work.

4. THE TRIAL OUTLINE BUILDER

The Trial Outline Builder (TOB), our system for designing, running, and analyzing clinical trials, was built as part of the EU FP6 integrated project ACGT, Advancing Clinico Genomic Trials on Cancer. This is a large European Union co-funded project, very interdisciplinary, and researchers from many countries participate. Several systems have been created in the ACGT project, e.g. the Oncosimulator simulating tumor growth by Stamatakos et al. (6) and the Obtima system for cancer treatment trials by Weiler et al. (9).

The TOB system is run as a component inside the Obtima system. The TOB is a graphical system for designing clinical trials. It has three different modes, first the trial design mode where the overall structure and other details of the trial are created. Then there is the local physician mode, where the physicians actually treating patients in the trial can see what the trial flow specifies they should do, and where they can enter data on what they have done so far and what results that gave. Finally, there is the analysis mode, where the results of the whole trial can be analyzed.

While a graphical tool is intuitive and easy to use, the addition of this type of tool to a trial management system is new. There are many general workflow authoring software tools available, see Barker and van Hemert (1) for an overview, but as far as we know the TOB is the first system of this kind for clinical trials. Below, we give an overview of the three different modes of the TOB. Then, we discuss some design decisions and problems we had when using the Webble technology to create the TOB.

4.1. Trial Design Mode

The trial design mode is where the trial chairman designs the outline of the trial. Figure 1 shows an example screenshot from the trial design mode. A trial is built up by different medical events, for instance chemo therapy events, diagnosis events, stratification events (partitioning patients into different treatment groups based on some factor, e.g. location), or randomization events (partitioning patients using an element of chance to determine the group).

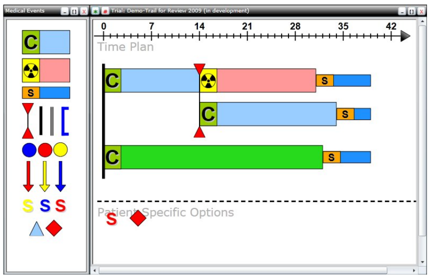


Figure 1: The trial design mode.

In the design mode, there is a medical event repository, where all available medical event types are displayed. There is also a work board for the current trial. Events can be dragged from the repository and dropped onto the work board to add events of this type to the trial. The events connect themselves to the events already in the trial to make a treatment plan. The events can then be dragged around, deleted, duplicated, or have their properties changed. The work board also displays a timeline, showing how much time the current treatment plan takes, and what event will take place after how many days in the trial.

The trial chairman can also attach CRFs (Case Report Forms) to the medical events in this mode. These are forms for data input, that are to be used by the physicians treating patients in the trial to report what they have done and what the results were, to store the information in the trial database. The CRFs may ask for the size of the tumor (diagnosis events), the dosage and medicine used (chemo therapy events), the age of the patient (registration events), or many other things.

The trial structure, the properties of the medical events in the trial, the CRFs used, etc. are all stored in a database. The TOB automatically saves any changes to the trial to the database as soon as they occur, so there is no need to press any buttons or enter any menus for saving data, and there is thus no risk of losing data by forgetting to save changes.

4.2. Local Physician Mode

The local physician mode is used by the physicians treating the patients in the trial. Here they can enter data such as which path in the trial the patient is following, what treatment the patient has received, and how the patient has responded to the treatment. Figure 2 shows an example screenshot.

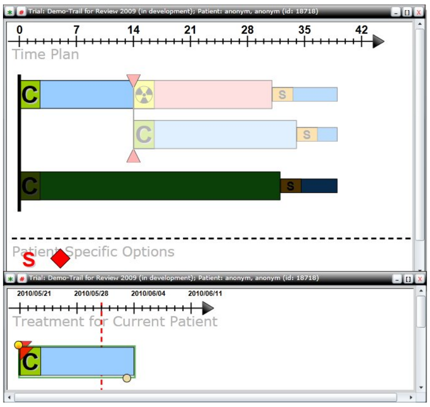
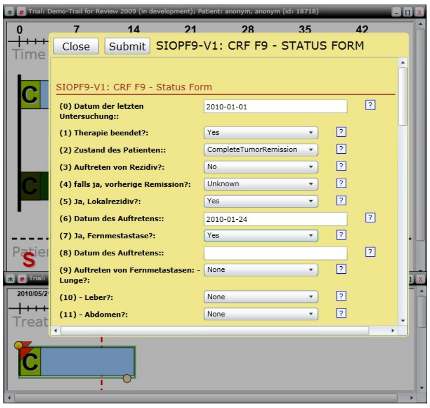

Trial: Demo-Trail for Review 2009 (in development); Patient: anonym, anonym (id: 18718)
0
7
14
21
28
35
42
Time Plan
C
S
C
S
C
S
Patient Specific Options
S
Trial: Demo-Trail for Review 2009 (in development); Patient: anonym, anonym (id: 18718)
2010/05/21
2010/05/28
2010/06/04
2010/06/11
Treatment for Current Patient
C

Figure 2: The local physician mode.

This mode shows an overview of the whole trial treatment plan. When a specific path has been chosen for the patient, the parts of the trial that can no longer be reached are shaded out in black. Events that the patient has not yet reached but are in the future plan of this path are shaded out in white.


Trial: Demo-Trail for Review 2009 (in development); Patient: anonym, anonym (id: 18718)
0
7
14
21
28
35
42
Time
C
C
Patien
S
Trial:
2010/05/2
Treat
C
Close
Submit

SIOPF9-V1: CRF F9 - STATUS FORM

SIOPF9-V1: CRF F9 - Status Form

(0) Datum der letzten Untersuchung::	2010-01-01
(1) Therapie beendet?:	Yes
(2) Zustand des Patienten::	CompleteTumorRemission
(3) Auftreten von Rezidiv?:	No
(4) falls ja, vorherige Remission?:	Unknown
(5) Ja, Lokalrezidiv?:	Yes
(6) Datum des Auftretens::	2010-01-24
(7) Ja, Fernmestastase?:	Yes
(8) Datum des Auftretens::	
(9) Auftreten von Fernmetastasen: - Lunge?:	None
(10) - Leber?:	None
(11) - Abdomen?:	None

Figure 3: Inputting data in a Case Report Form (CRF).

The events that the patient has already entered are shown in a separate window, showing this patient's individual treatment path. Here the local physician can open the CRFs prepared by the trial chairman by simply clicking on the related event. When reporting data from the chemo therapy event in the example screenshot, clicking this event in the individual treatment plan will thus open the CRF as a data input form, as in Figure 3. The local physician can also edit the properties of the medical events in the individual treatment plan for the patient, to reflect any changes that may have been necessary.

4.3. Analysis Mode

In the analysis mode, researchers can analyze the data collected in a trial. Figure 4 shows a screenshot of the analysis mode. Again, the treatment plan of the trial is shown. Here, the researcher can select what data to look at by clicking the medical events in the treatment plan. Clicking an event opens the CRFs connected to this event and the researcher can select which questions or data fields in the forms to include in the interactive visual querying and analysis.

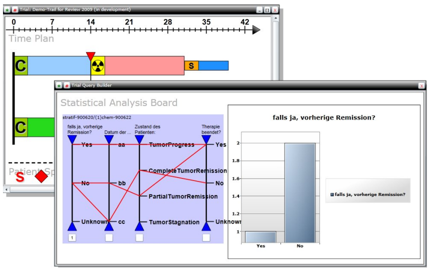


Figure 4: The analysis mode.

The data selected to be included is shown in a separate analysis window. In this window, the different data fields are treated as parallel coordinates (Inselberg and Dimsdale (2)). Each patient is represented by a line going from the position on the coordinate axis for one data field representing the data for this patient to the patient's position on the next coordinate axis, representing another data field selected from (possibly) some other CRF etc. Parallel coordinates representing data fields from CRFs that come from different paths in the trial will end up in different groupings of parallel coordinates, while those coming from the same treatment path will be grouped together.

Parallel coordinates representing derived attributes can also be used. Coordinates for properties computed from some CRF fields can be created by selecting groups of coordinates and specifying how to combine their values. It is thus possible to create a coordinate for the age of the patient by calculating this from the birth date information from one CRF and the date of entry into the trial from (possibly) another CRF. It is also possible to generate charts by selecting what parallel coordinates to base the chart on. The type of chart wanted can also easily be specified.

5. DESIGNING WITH WEBBLES

As previously mentioned, the TOB system was built using the Webble technology. Here we detail some of the design decisions we made, some benefits of using the Webble technology and some problems we had.

5.1. Webbles in the System

There are eight different types of Webbles in the system. There are the *Medical Events*, the *Repository Window*, the *Work Boards*, the *Timeline*, the *Query Board*, the *Parallel Coordinates*, the *Charts*, and the *Database Wrapper*. The *Medical Event* Webbles each represent one event in the treatment plan of the trial. They can be randomization events, surgery events, chemo therapy events, etc. The *Repository Window* Webble is where all the different types of *Medical Event* Webbles are available for dragging into the treatment plan when building a new clinical trial. The *Timeline Webble* shows what day or point in time the graphical positions on the screen correspond to. The *Work Board* Webbles are where the treatment plan is shown. There are two *Work Boards*, one for the complete treatment plan of the whole trial, where the trial is designed etc., and another for showing the individual treatment plans of specific patients.

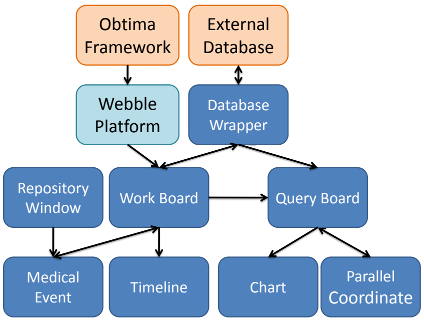


Figure 5: The Webbles and the connections in the system.

The *Query Board* Webble is very similar to the work boards, except that it shows *Parallel Coordinates* and *Charts* instead of *Medical Events*, for the analysis of trial results. The *Parallel Coordinate* Webbles represent a coordinate axis for each question selected from a CRF for analysis on the *Query Board*. They keep track of which values to include or filter out in the analysis. The *Chart* Webbles display statistical data in charts. Finally, the *Database Wrapper* Webble handles communication with the external database.

The connections between the Webbles in the system are shown in Figure 5. The *Medical Event* Webbles are connected to the *Work Board* Webble they reside in. The *Work Boards* listen to any property changes in their *Medical Events*. The *Medical Events* are not connected to each other. The *Timeline* Webble is told by the *Work Board* it belongs to how long the current trial is. The main *Work Board* is also connected to the *Query Board*, and sends information on what parallel coordinates to display using slots. The *Query Board* listens to property changes in the *Parallel Coordinate* Webbles and also sends information to the *Parallel Coordinates* and the *Chart* Webbles using their slots. Information from the surrounding system, e.g. which user is currently logged in, goes through the Webble platform. Communication with the external database goes through the *Database Wrapper*.

5.2. Designing the Webbles

Some Webbles in the system are very straight forward to build. The *Repository Window* is just a generic window, the *Chart* Webble is just a generic graphical chart object, the *Timeline* is a very passive object just growing in length depending on the duration of the *Medical Events* in the treatment plan. This duration information comes to the *Timeline* from the *Work Board*, which keeps track of all the *Medical Events* added to it. These simple Webbles were easy to build because they just use the template for building a Webble to wrap something that Silverlight already supports, such as a graphical window or charting.

The *Medical Events* are mainly graphical objects for displaying the properties of events in the trial in a visual and easy-to-understand way. They have some intelligence, for instance enabling new events to be dragged from the *Repository Window* without the original disappearing. If their parent is the *Repository Window*, the *Medical Events* copy themselves and leave the copy behind in the original position when dragged. In most other senses they are passive objects, just notifying the *Work Board* when their properties have changed and database need to be updated.

The *Work Board* Webbles handle many things. They handle the layout of the *Medical Event* Webbles, instructing them where on the work board they should move when new events are added or where in the treatment plan they should attach themselves when dropped by the user on some part of the work board. When the layout of the trial changes, the *Work Boards* keep track of what information needs to be sent to the database and communicate this to the *Database Wrapper*. The *Work Boards* also handle the CRF related business, allowing users to enter data into a CRF (and sending this data to the database) and by communicating to the *Query Board* that the user wants to view certain CRF questions as parallel coordinates.

The *Query Board* Webble is very similar to the work board. It reads the trial result data from the database and then handles *Parallel Coordinate* and *Chart* Webbles for visualizing this data. While the *Parallel Coordinate* Webbles handle filtering and grouping on their specific axis, and keep track of which data values correspond to which screen positions, they do not keep track of the other *Parallel Coordinates*. The drawing of lines in the parallel coordinate system is thus done by the *Query Board* that has a complete view of all the coordinates that are used and the order in which they currently appear on the screen etc.

The *Parallel Coordinate* Webbles as mentioned keeps track of which value ranges to group together for their respective axes and what value ranges to show or filter out. They also know which screen position corresponds to a certain value on their axis. They do not know which is the next or previous axis, and reordering of the axis on the screen is an important visual aid with parallel coordinates, so the line drawing intelligence is handled by the *Query Board* Webble.

The *Database Wrapper* Webble provides a simple slot based interface for Webbles that want to read from or write to the database, thus hiding for instance all the network related things that need to be taken care of.

5.3. Discussion

The TOB system designed using Webbles has, as mentioned, been developed as a part of the EU FP6 integrated project ACGT (Advancing Clinico Genomic Trials on Cancer). It has received very positive reactions from professionals in the project review group. Strengths of the system from a user perspective like theirs include: the system is very visual, which makes it easy to get an overview of what is going on and what has been done so far. The interface is easy to use, dragging and dropping events graphically on the screen is quite intuitive. It is easy to get to the parts of the system that you are looking for, e.g. it is easy to find the correct CRF to use when all you need to do is click on the graphical representation of where in the trial you are now and there find all the related CRFs.

The strengths of the system from a developers view, the strengths of using Webbles for building the system, include: advanced graphical objects are very easy to make. The development time is low, not only can already created Webbles easily be reused, it is also easy to use the many advanced features available in Silverlight, or to wrap other unrelated services. Building the system as presented here took only about 2.5 man months of development work. The system runs in a web browser, which means that it is available on any computer with a modern web browser and Internet connectivity. Internet connectivity is not necessary if all the services are run on the local computer.

Adding new features or functionality to the system by modifying existing Webbles or adding more Webbles is very easy. When some interesting ideas came up in discussions on what to do next in the system it was very quick work to add even advanced components, showing the strength of using Webble technology to reuse functionality.

In hindsight, there are some Webble design decisions that could have been done differently. The thing in the current system that most goes against the Webble design philosophy is the *Work Board* Webble, which does too much. This Webble could have been separated into several more specialized Webbles that by their nature of doing only one thing would be easier to reuse in other projects too. Separating this complex Webble into for instance a Webble for CRF rendering (perhaps even by wrapping the CRF rendering part of the Obtima system with a Webble interface) and a Webble for being the work board to put the *Medical Event* Webbles (or in a more general sense, any work flow events) would have been a cleaner design. Similarly, for the *Query Board* Webble, it would have been more useful to separate out the functionality of aggregating parallel coordinates and drawing the lines in the parallel coordinate system into a *Parallel Coordinate Container* Webble from the TOB specific functionality. Such a container Webble would be useful in many other applications, and we are currently about to rewrite the *Query Board* in this way.

We could also have made each patient and its dataset in the analysis view a separate Webble. A Webble that draws itself on top of the *Parallel Coordinate* Webbles etc. We decided not to go this way since for large trials with thousands of patients, having thousands of Webbles communicating with each other would likely create some performance problems.

In a general sense, the current problem is that many Webbles have low reuse potential, since they are tailored too much towards solving the current specific problem. It would have been nicer if the system was written as a general *Workflow Editor* Webble, pluggable *Workflow Event* Webbles, etc.

That the created Webbles (in our case), Intelligent Pads, or other Meme Media Objects end up being written to solve a specific problem at hand instead of in a more generic way is not a new problem. A part of this problem is likely that there are not that many Webbles available for use yet. If more components already existed, the idea of building new applications by reusing the already available parts would probably work well. The time spent on creating whatever new components may be necessary could also be used to make these more generic since more time could be spent on each new component when there are not as many new components that need to be created. When, as now, there are few components available to use, the pressure to produce a working system quickly often leads to the components being written to do precisely what is needed right now, with less thought for the general applicability of the Webble. Also, it is probably easier to see a future possibility for reuse of a Webble you create if reusing existing components is the norm. Now that the system is already doing most of what it should do, and the requirements for what the system is expected to do have become more stable, we are rewriting parts of the system to be more generic and thus more reusable.

There were also some other problems of different types, for instance problems with interacting with systems written using different technologies. One example is that it was difficult to get the secure communication with the back end database system working since the server side secure communication technology was not supported by Silverlight or C sharp. A working solution was created without that much work, though.

6. FUTURE WORK

In the future, we want to add advanced functionality to the system by reusing already existing software. Most analysis of clinical trials is for instance done using the statistics package R. We would like to write a Webble wrapper for the whole R system, so other Webbles can communicate with the R system and make use of its power. We also want to do a thorough evaluation by having professionals use the system in real trials to see what works well and what could be done better. We are also working on rewriting some of the system Webbles to be more generic and thus reusable in other applications.

7. CONCLUSIONS

We showed an application of Webbles, intelligent media objects that are meant to simplify reuse of not only data but also reuse of services and functions. Using Webbles we created a system for designing, running, and analyzing clinical trials. Using the Webble technology the system was easy to implement quickly, and the resulting system is very visual and has an intuitive interface that appeals to professionals in the target audience. The biggest problem was that the created Webbles tended to be too specific, solving the problem at hand but not being very reusable despite the problem solved is one that could be solved with more generic Webbles. This is likely caused by the low number of available Webbles. With more generic components available we believe that new Webbles can be written to be more generic too, since more time would be available for the development of each new component if less components needed to be created.

The biggest advantage of using the Webble technology was that adding new advanced functionality was easy. The development time was low, around 2.5 man months for the parts of the system described here, and it was possible to quickly prototype and try different design choices. Reusing functionality not only by reusing previously developed Webble components but also by for instance wrapping other systems with a Webble interface or by taking advantage of functionality already available in the Silverlight framework is easy. The fact that the system runs in a web browser also means that there is no need to install new software and the system can be run from different computers without extra work.

8. REFERENCES

[1] A. Barker and J. van Hemert. Scientific workflow: A survey and research directions. In *Parallel Processing and Applied Mathematics*, pages 746–753, Gdansk, Poland, 2007.

[2] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *VIS'90: Proceedings of the 1st conference on Visualization '90*, pages 361–378, Los Alamitos, CA, USA, 1990.

[3] M. N. Kuwahara and Y. Tanaka. Webble world – a Web-based knowledge federation framework for programmable and customizable Meme Media objects. In *Proceedings of FlexWeb 2010*, Taichung, Taiwan, 2010.

[4] S. Modgil and P. Hammond. Decision support tools for clinical trial design. *Artificial Intelligence in Medicine*, 27(2):181–200, 2003.

[5] R. D. Shankar, S. B. Martins, M. J. O'Connor, D. B. Parrish, and A. K. Das. Epoch: an ontological framework to support clinical trials management. In *HIKM'06: Proceedings of the international workshop on Healthcare information and knowledge management*, pages 25–32, Arlington, Virginia, USA, 2006.

[6] G. S. Stamatakos, D. D. Dionysiou, N. Graf, N. A. Sofra, C. Desmedt, A. Hoppe, N. K. Uzunoglu, and M. Tsiknakis. The "Oncosimulator": a multilevel, clinically oriented simulation system of tumour growth and response to therapeutic schemes. towards clinical evaluation of in silico oncology. In *Proceedings of the 29th Annual International Conference of the IEEE EMBS*, pages 6628–6631, Cité Internationale, Lyon, France, 2007.

[7] Y. Tanaka. *Meme Media and Meme Market Architecture*. IEEE Press, Piscataway; NJ; USA, 2003.

[8] Y. Tanaka, J. Fujima, and M. Kuwahara. Meme media and knowledge federation. In *KI 2008: Advances in Artificial Intelligence, Proceedings of the 31st Annual German Conference on AI*, pages 2–21, Kaiserslautern, Germany, 2008.

[9] G. Weiler, M. Brochhausen, N. Graf, A. Hoppe, F. Schera, and S. Kiefer. Ontology based data management systems for post-genomic clinical trials within an European grid infrastructure for cancer research. In *Proceedings of the 29th Annual International Conference of the IEEE EMBS*, pages 6434–6437, Cité Internationale, Lyon, France, 2007.

[10] P. Y. Wong and J. Gibbons. On specifying and visualising long-running empirical studies. In *ICMT'08: Proceedings of the 1st international conference on Theory and Practice of Model Transformations*, pages 76–90, Zurich, Switzerland, 2008.