Towards Real Time Interaction Visualization in NED

Thomas Fischer, Hong Kong Polytechnic University, China

Christiane M. Herr, University of Kassel, Germany

Cristiano Ceccato, Hong Kong Polytechnic University, China

Abstract

Where design education moves from the studio to computer networks, interaction information easily becomes unavailable for pedagogic analysis. In this paper we propose automated learning interaction visualization to solve this problem and show our progress in developing technical tools for this purpose.

1 Introduction

The concept of studio interaction and its traditional classroom predecessor has developed for decades and centuries. One of its advantages is the immediate physical presence of learners and teachers in the teaching scenario. It allows the direct observation and analysis of interaction. This instrument of direct observation will suffer as computer- and network based education will spread in the future: Where educational interaction is not visualized implicitly, as for example with avatars in virtual environments, information gets lost. This information has the potential to enrich learning situations in various ways and – even more importantly – it is the key to student assessment. Therefore we must find ways to adapt traditional studio interaction observation to NED (Networked Education in Design (see Falk et al. 2000)).

2 The Requirement for Ad-Hoc Interaction Analysis in Online Teaching

We regard teaching, as well as design, as a wicked problem. As a consequence, its strategies require constant monitoring and adaptation to dynamically changing problem variables. One central set of learning situation variables arises from interaction, as it is observable in the studio but hidden online. Logging interaction data provides us with a method to review what learners do and how they do it. But log file analysis is a time consuming and error-intensive activity for humans and not practical in the intervals in which teaching strategies should be reviewed. We propose visual representations as a means to solve these problems.

In order to record data on online learning software usage it is easily possible to enhance the software itself in a way that it generates protocols of its own execution. This is actually a common basis for example on the WWW server level of NED applications: log files of document requests and protocol errors are generated by default and can be reviewed if necessary. However, these technically oriented log files provide no means to track educational interaction on the individual and group levels teachers are interested in.

A major advantage of electronic protocols is their automatically persistent storage, which allows retrospective analysis whereas moments in traditional learning setups are fugitive and can only be prevented from getting lost by techniques such as video surveillance. Advantages like this have to be paid for with shortcomings in other regards: Machine-generated protocols typically only collect events which are initially expected to occur, describing them in a plain, non-interpreted sequence of log entries. Individual or group behavior, complex interaction patterns and correlation between them can hardly be identified intuitively in large protocols describing several weeks of course interaction in hundreds of kilobytes of rather cryptic data. In this context, entire or partial two-dimensional or three-dimensional data visualization has already proven to us to be an effective method to visually analyze learning interaction as well as an essential source of information for its assessment.

3 Early Field and Lab Tests

The ‘Virtual Design Company’ online learning application was the first manifestation of the networked approach to design education. Simulating a groupware platform of a design company, coordinating its globally distributed agencies, the application allows project based design education in the studio as well as over the Internet.

Data collected by this system during its first application in late 1999 was the material of our early visualization experimentation. The results included real time generated HTML bar diagrams, separately showing ‘active’ and ‘passive’ system usage (see figure 1) and post-produced graphics generated with MS Excel® processing comma-separated lists of interaction data. These two-dimensional visualizations could not communicate the timeline-oriented quality of our data. The simple HTML bars were used as a basis for assessment but in terms of the data’s obvious potential this output design was as little satisfying as images manually post-produced using Excel.

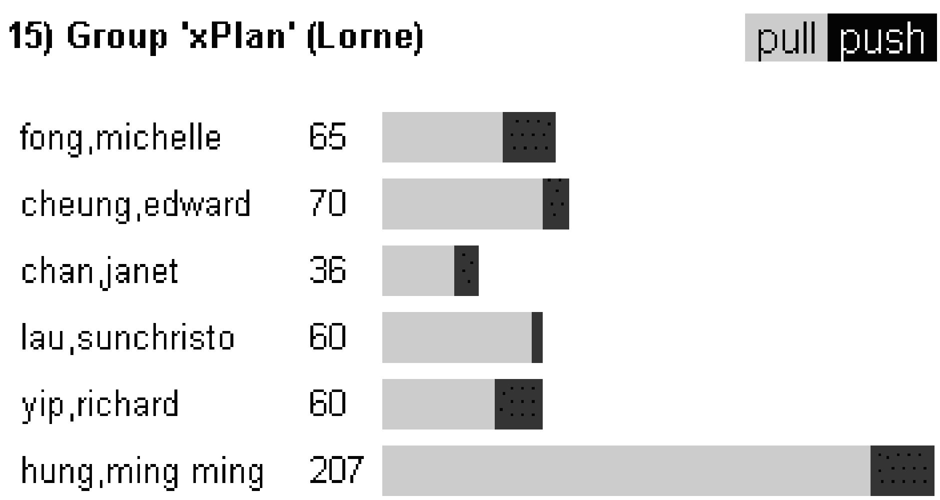

Figure 1. HTML bar diagram.

Our investigation in visual data interpretation tools for didactic contexts was reinforced by several ad-hoc findings we made reviewing the first bar diagram shown in figure 1. This group shows a typical characteristic, which practically all learner groups of this course share in a more or less obvious way: One student of each group shows a significantly stronger performance than her or his group partners. Interestingly, these individuals are not necessarily those we know as particularly strong students. But in all cases these are the students who have initiated the respective group and therefore share a higher level of group awareness. This correlation is only one example of how visual activity representation allows didactic conclusions of a quality, which previously was available to physical classroom observation only.

In later experimentation, the CGI-based strategy of HTML generation was adapted to dynamic VRML generation in order to exploit the third dimension to communicate the ‘depth’ of logged information. The amount of VRML data generated even from small data segments easily grew up to several megabytes which is too much to be handled by VRML browsers on current desktop PCs. While making these experiences we also understood that teaching contexts require highly extendable learning systems which is why we need a generic geometric solution for interaction visualization which itself is extendable to initially unforeseen requirements. To develop this generic geometry and generation procedures we chose 3DStudio MAX® as our (interim) prototyping platform.

4 Extendable Geometry for Extendable Applications

Teaching means dealing with unforeseen events, methods, content and media because throughout courses, teachers keep on monitoring learners, their activity and maintain assumptions on their individual levels of skills and knowledge. When new information results in changes of these assumptions, changes of initial didactic planning are likely to occur. For this purpose, learning software is best prepared if it is developed by its users themselves and full source code control is given. But once new forms of interaction are supported, the log protocol specification has to be extended accordingly to handle new aspects of interaction. Therefore, it has to be specified in an extendable fashion in the first place. One integer number associated with every entry identifies the individual interacting with the system. The second entry to be specified for a log protocol specification is the encoding for the actual interaction aspects. These aspects are directly selected by didactic analysis from interaction options supported by the application. In case of the *Virtual Design Company* application these options contained login procedures, the usage of supported TCP-IP-like services such as email, FTP, chat and news board postings as well as individual statements of student self-reflection. In log protocols we express these aspects with unique integer numbers. A third entry is optionally used where attributes for the respective interaction aspect are required.

As we implemented the interface between the protocol database and 3DStudio MAX® as a text parser written in Max Script, we produced a flat file version from the original log protocol database. A sample entry in this plain text version looks like the following:

```
1072869,13,¤ý£l
```

The first integer number in this like is the project second since the beginning of the course. The second one is the interaction aspect, which in this case (13) is the code for an email sent to a group. The third field is an ASCII-interpretation of a bitwise encoding of a subset of the course participants. This flat file database carries activities of individuals in dedicated files: It is person-oriented whereas the initial log protocol was timeline-oriented. We found person-oriented databases to be evaluated much easier and faster, especially when subsets of the entire course interaction have to be visualized. On the log specification level, the specification described above guarantees extendibility in all contained dimensions: The number of persons involved in a didactic situation, the number of interaction aspects supported by a learning software, the course timeline and the attribute encoding.

The same flexibility has to be supported by a generic geometry based on this data specification. To fulfill this requirement we have developed the architecture shown in the visualization of the Virtual Design Company data shown in figure 2.

This geometry exploits space for free visual extendibility in three dimensions: On the x-axis it handles an arbitrary number of individuals, on the y-axis it shows the course timeline and on the z-axis it handles the interaction aspects. The six large T-shaped forms are representations of course milestones such as course start, interim presentations and final presentations. This arrangement allows direct quantitative comparison between the performances of individuals because through iterative upwards moving, individual activity representations appear as a bar-diagram in the front view. Figure 3 shows a front view of the above visualization with representations of quantitatively comparable representations of chat room, news board and FTP server usage.

In future applications, when visualizations like this are generated in real time either as VRML using PERL or as static jpeg images using 3DStudio Max, an HTML GUI can allow teachers to select certain time frames, subsets of learners and subsets of interaction aspects to get isolated visualizations of interaction details. Therefore it is possible to selectively emphasize course interac-

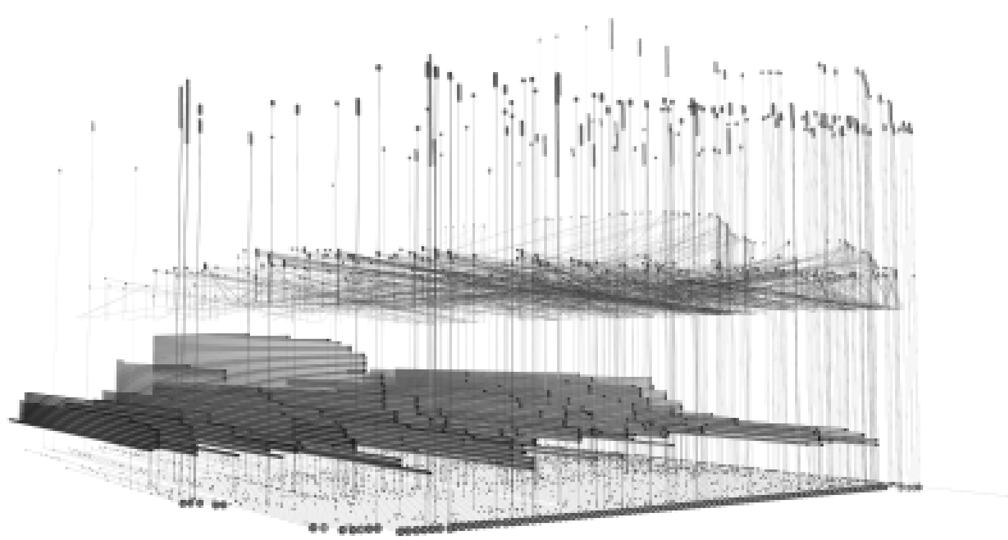

Figure 2. Overall Course Visualization.

tion elements while maintaining graphic linearity whereas other non-linear strategies like fish-eye views cause difficulties in terms of visual performance comparison.

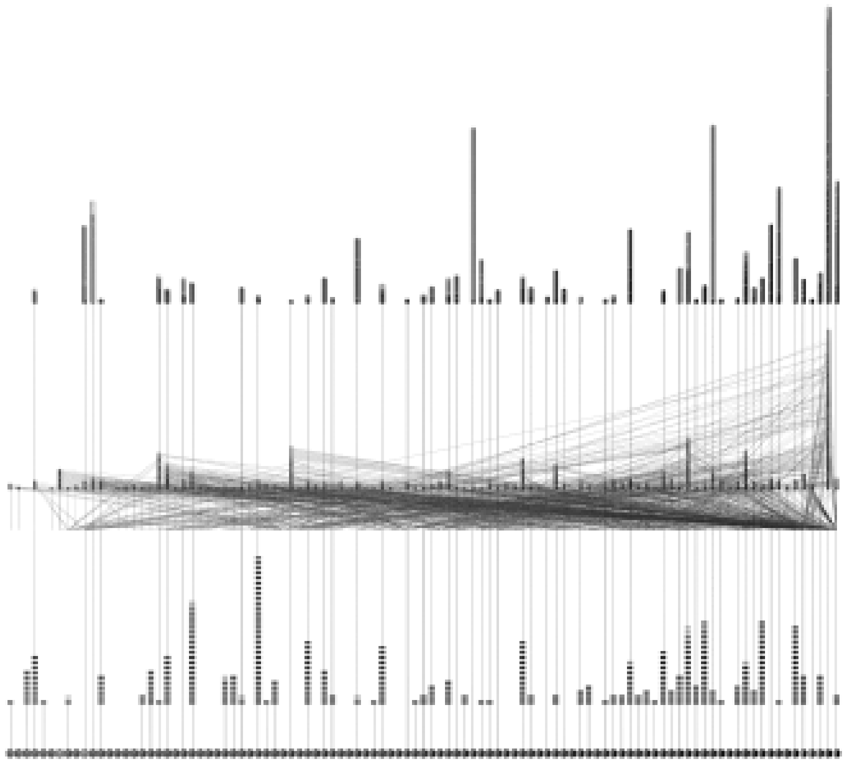

Figure 3. Front view (interaction modules).

As figure 4 shows, interaction aspects are 'piled up' along the z-axis as separate modules (chat contributions, modification of the 'message of the day', email, posting and reading news board messages, FTP up- and download). While course participants and time in x-axis and y-axis can be easily displayed as linear and homogenous data, even when only small segments are visualized, the interaction modules on the z-axis are highly variable in size and order depending on subset selections. To handle this problem of potential inconsistency, we have implemented the 'activity modules' as separate *functions*: The main program is called with one argument that is the path to the flat file database. Then the module functions are called successively, generating geometry and then returning the used, dynamic z distances as their function values. These values are added by the main program in order to call each further function with a z coordinate from which to start.

5 Conclusions

Though interaction visualization superficially appears to be a last additional step in learning software development, it plays a very central role from a pedagogical point of view as well as from a technical perspective. In this context, the log database specification appears to be a critical link between a learning system with its extendible user interface and real time visualization. This fact has to be considered in the early stages of a learning system's design.

As in our experimentation the presented geometry shows no problems to handle any didactic interaction data we assume it has a reliable generic quality and can be applied to various contexts in the future. Furthermore, its level of abstraction suggests the development of application-independent interactivity visualization tools such as authoring system add-ons that are the long-term aim of this work in progress.

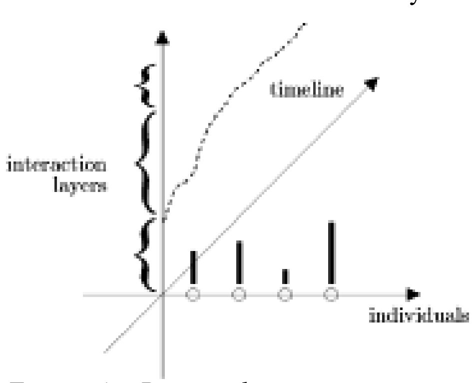

Figure 4. Proposed geometry.

References

Card, S.K., J.D. MacKinlay and Sneiderman, B. (1999). *Readings in Information Visualization. Using Vision to Think.* Morgan Kaufmann Publishers, San Francisco California

Falk, L., C. Ceccato, C. Hu, P. Wong, and T. Fischer. (2000). Towards a Networked Education in Design. A First Manifestation through the "Virtual Design Company" Studio In *CAADRIA2000 Conference Proceedings*, ed. T. Beng-Kiang, 157 - 167. National University of Singapore

Rittel, H. and M. Webber (1973). Dilemmas in a General Theory of Planning. *DMG-DRS Journal* 8(1):31-39. University of California: Institute of Urban & Regional Development.