Interactive trace visualization in Trace-based systems

Julien Laflaquière
Université de Lyon, CNRS
Université Lyon 1, LIRIS,
UMR5205, F-69622, France
Julien.laflaquière@liris.cnrs.fr

Yannick Prié
Université de Lyon, CNRS
Université Lyon 1, LIRIS,
UMR5205, F-69622, France
yannick.prie@liris.cnrs.fr

CHI 2009, April 4 – 9, 2009, Boston, MA, USA
ACM 978-1-60558-246-7/09/04.

Abstract
We first introduce the notions of modeled-traces (M-traces) and trace-based systems. We then present activity analysis and activity support as the two main uses of M-traces, and we discuss interactive visualization challenges related to active reading of traces.

Keywords
Activity analysis, activity support, modeled-traces, trace-based systems, trace visualization

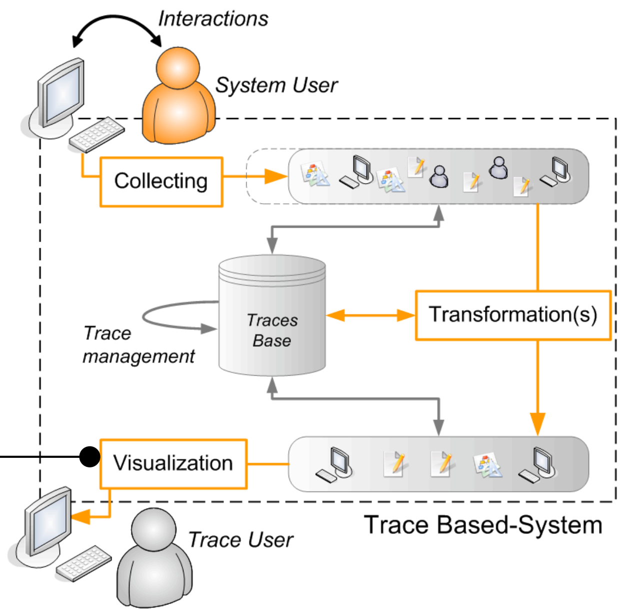
ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous. See [3] for help using the ACM Classification system.

Introduction
Numerous research works have investigated the general issue of *use traces*. Such temporal data, sometimes called interaction history or user data, are collected during the interaction between a user and a digital environment. The collected traces are most of the time composed of *log files* related to the system's functionalities. Being usually too complex for human direct interpretation, statistical treatments are usually needed for their rendering, aiming at converting

quantitative data into qualitative visual insights that characterize the usage, the user or the activity. In that case, visualization techniques are applied directly on raw data so as to make remarkable patterns emerge [2].

Trace-based systems

The main goal of our approach is to be able to manage and present digital traces that are qualitative representation of observed activities. We argue that a means to reach such goal is to consider *explicit* and *activity-oriented* modeling of traces. 'Explicit' here opposes to the implicit modeling of most of the log files, while 'activity-oriented' opposes to their function-oriented (or conception-oriented) modeling.



In this paper we deal with the visualization of M-traces that have been transformed so as to be used in particular situations.

Figure 1: Trace-based system global architecture

Our approach at LIRIS-SILEX team is conceptualized in the *Trace-Based System* framework [1]. At the core of this framework is the notion of *modeled-trace* (or M-trace) defined as the association of a *collection of temporally situated observed elements structured by relations* with an *explicit model of these observed elements and relations*. Such explicit trace models can be considered as ontologies of M-traces that describe both their constitutive elements (entities or events) and the complex relationships that can be expressed between them.

Though it can easily represent simple event log files, the notion of M-trace can also tackle complex application use histories. Moreover, being explicitly modeled with knowledge representation techniques, traces can be managed in generic and flexible ways. They are considered as first class objects that can be manipulated independently of the particular environment that was involved in the activity they are traces of.

A trace-based system is then a system dedicated to the management of modeled-traces, from their collecting to their use. It offers services for *trace transformations* such as fusion, filtering and rewriting (figure 1). Such transformations can help abstracting traces from low-level *a priori*, log-like modeling to higher-level *a posteriori*, activity-oriented modeling. Abstracting can be important for reaching adequate level of interpretability, especially in the case of visualization.

Interactive visualization of M-traces

In this article we focus on two main situations for M-trace exploitation: *activity analysis* and *activity support*.

Activity analysis
In a situation also known as *Exploratory Sequential Data Analysis* [3], an analyst who was not involved in the traced activity transforms a M-trace so as to characterize the investigated activity. It is the case in the *Abstract* approach [4] where a TBS was implemented for facilitating the analysis of behavioral data in a vehicle-driving situation. The analysis is carried off-line, after the action, thanks to a visual interface (figure 2). On the contrary, in the *eLycée* project [5], the M-trace is displayed during the activity. Students work in a virtual classroom with a tutor while their activities is traced. Such traces can be exploited after the class session but also by the tutor directly *during* the session, so as to manage the group activity online. The challenge here is to provide a M-trace that describes not just what is going on but *how* student are or were behaving, during or after the time of their activity.

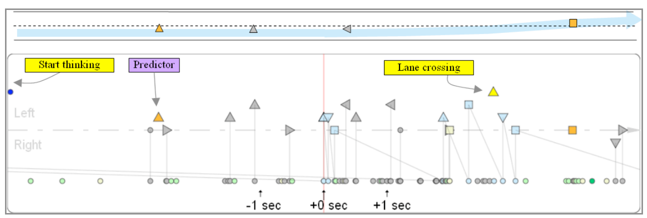


Figure 2: Three abstraction levels of driving-related M-traces in the *Abstract* system [4].

Activity support
M-traces are here directly used *within* the activity, the user of the M-traces being the user of the environment they are collected from. In that case, traces are used for facilitating the activity by visually representing how it takes or took place be it in real time or afterwards. We are here interested in complex activities the unfolding of which cannot be fully planned in advance, such as video active reading with Advene [6] or the production of multimedia learning material with a dedicated software (*Emulsion*).

Figure 3: Using MIT SIMILE timeline for *Emulsion*.

In such cases, we try to facilitate the activity by providing the user with a rich representation of her own activity 1/ so that she can get a sense of her current situation; 2/ for contextualizing the elements she manipulates; or 3/ for reusing past significant episodes so as to produce new activity phases.

Some interaction challenges

Though a M-trace is modeled specifically for a dedicated type of activity so as to 'make sense' relatively to it, no *a priori* pre-determined visualization can pretend to match all its interpretation needs. This is true both for activity analysts and users observing their own activity, and it appears to us that the solution consists in interactive visualization of traces. We pretend that the interpretation itself must be constructed along such interactive visualization, by *trace active reading*, which poses several challenges.

Model-based presentation of traces
M-traces are traces whose elements and relations are described abstractly in knowledge models. Such models can be used as inputs for model-based trace visualization interfaces that give user rich access both to abstract (model) and concrete (actual trace(s)) knowledge structures.

Time-based and structural presentation of elements
Most interfaces for traces visualization are currently based on temporal dimension, emphasizing mainly the temporal relationships between observed elements. Horizontal or vertical timelines present punctual or durative observed elements, often neglecting the structural relationships between them. A challenge here is to design interactive interfaces that smoothly articulate and present at the same time temporal *and* structural relationships between elements.

Panoptic organization vs. rich elements representation
Any timeline design for trace visualization needs to cope with two contradictory needs. First the need for a synoptic vision of a slice of activity, hence the need for a light visualization of elements within their time range. Second the need for a rich contextualization of elements that are isolated within the trace, hence the need for a rich visualization of elements. An important challenge here concerns interfaces suited to adapting temporal elements visualizations to temporal range visualization, while providing users with means to control and tailor such adaptation.

Interfaces for trace filtering and rewriting
Interactive visualization of traces means providing the user with the full control of the representation of her activity. Here TBS generic trace transformation services can help her construct traces that match her representation needs. *Filtering* offers choice of what elements of the trace model (actions, events, resources…) are to be presented. *Rewriting* provides the user with the ability to create her own trace model, by determining *new* types of observed elements, and how they are to be constructed from existing ones. Numerous challenges are related to so defined trace active reading: How to interact with a trace or a trace fragment and specify its filtering or rewriting? How to manage one's multiple traces and transformations? How to elicit new relationships within a trace?

References

[1] Laflaquière, J., Settouti, L.-S., Prié Y. and Mille, A. A trace-based System Framework for Experience Management and Engineering. *2nd Int Workshop on Experience Management and Engineering* in conjunction with KES (2006), 8 pp.

[2] Chen, J., Sun, L., Zaiane, O. R. and Goebel, R. Visualizing and discovering web navigational patterns. *7th Int. Workshop on the Web and Databases, ACM SIGMOD/PODS 2004 (2004),* 13–18.

[3] Fisher, C. and Sanderson, P. Exploratory Sequential Data Analysis: Traditions, Techniques and Tools. *ACM SIGCHI Bulletin*, 25, 1 (1993), 34-40.

[4] Georgeon,O., Mille, A. and Bellet, T. Analyzing behavioral data for refining cognitive models of operator. *Philosophies and Methodologies for Knowledge Discovery*, *7th Int. Workshop on Database and Expert Systems Applications* (2005), 588-592.

[5] Cram, D., Jouvin, D. and Mille, A. Visualizing Interaction Traces to improve Reflexivity in Synchronous Collaborative e-Learning Activities, *6th European Conference on e-Learning* (2007), 147-158.

[6] Aubert, O., Champin, P.-A., Prié, Y. and Richard, B. Canonical processes in active reading and hypervideo production. *Multimedia Syst.*, 14, 6 (2008), 427–433.