Managing, synchronizing, visualizing, analysing and sharing multimodal computer-mediated human interaction data: introducing Tatiana (A Trace Analysis Tool for Interaction Analysts)

Gregory Dyke[*], Kristine Lund[°], Jean-Jacques Girardot[*]
[*]RIM Department, Ecoles des Mines de St. Etienne,158, cours Fauriel F-42023 Saint-Étienne, France
[°]ICAR Research laboratory, CNRS, University of Lyon, 15 parvis René Descartes, 69700 Lyon, France
Email {Gregory.Dyke, Jean-Jacques.Girardot}@emse.fr
Email Kristine.Lund@univ-lyon2.fr

ABSTRACT
This article

1 Introduction

The socio-cognitive study of human computer-mediated interactions can be performed through recordings of these interactive activities, particularly if they are not limited to interaction log files but also include audio and video recordings (Avouris, Fiotakis, Kahrimanis, Margaritis, & Komis 2007). Cox (2007) encourages researchers to use computers and the various techniques they offer (visualisation, data mining etc.) to perform their interaction analyses on what he calls "process data". However, corpora of human interaction, particularly when these interactions are both face-to-face and mediated by computers are difficult to manage from a technological standpoint and complicated to understand and analyse due to the multiplicity and variation of source data. Indeed, it is not enough to look at individual data streams; different media streams must be combined to achieve a global understanding of the interactions that occurred (Goodman, Drury, Gaimari, Kurland, & Zarrella, 2006). Furthermore, it is often necessary to perform analysis as a team, be it in order to validate the analysis method through inter-coder reliability (De Wever, Schellens, Valcke, & Van Keer, 2006), to extend applicability of an analytical method to a new domain of application (Lund, Prudhomme, Cassier, 2007), to spread the workload (Goodman et al., 2006), or to combine the insights of several analysts (Prudhomme, Pourroy & Lund, 2007).

The difficulties described above suggest the necessity of tools which provide not only the means to manage this variety and quantity of data, but also to allow visualisation and analysis within a common framework and in a way that can be shared with other researchers (cf. Reffay, Chanier, Noras, & Betbeder, 2008 for work on structuring learning corpora for sharing purposes).

In this paper, we will present the tool Tatiana (Trace Analysis Tool for Interaction ANAlysts), intended for researchers who wish to analyse CSCL and CSCW situations from an interactional socio-cognitive perspective. We will begin by presenting three of our on-going analytical contexts, then briefly present several tools which are comparable to Tatiana. Next, we will discuss Tatiana's features in relation to these tools and explain the necessity of Tatiana's features with regard to examples from our analyses. Finally we will present the specifics of Tatiana's architecture and show how it supports the iterative creation of artifacts, designed both to exhibit researchers' understanding of their corpora and to allow researchers to further this understanding.

2 In regards to three analytical contexts

In order to motivate some of our choices and requirements for an analysis tool for computer-mediated human interactions that can also be face-to-face, we will draw examples in this paper from three corpora that we are currently analysing along with the research questions we have with regard to these corpora. These examples may also be of use for analysis tool designers who wish to consider the applicability of their tools in the contexts we will describe.

2.1 Shared note-taking during tutorial sessions

Within the LEAD[1] research project (LEAD, 2006), we observed nine dyads over the course of three to five meetings with their teacher for an introductory-level computer-programming project. These meetings took place face-to-face with the assistance of a chat and a real-time shared text editor (both of which the two students and the teacher had access to, on their laptops). The tutorial trilogues[2] were not designed to follow a specific pedagogical scenario. Instead a rough parallel could be drawn between them and industrial project review meetings where the teacher could correspond to either the person who ordered the program, a consultant-pedagogue or a programming expert and the students to the design and development team. Aspects discussed during the meeting included task understanding and choices about program design (e.g. functional specifications, implementation). Variously, during the meeting, students took notes, defined tasks to be accomplished before the next meeting, and prepared the layout of their final report and presentation. The data collected for each session includes multi-track audio (one track for each participant), video, the interaction log-file produced by the shared text editor and the chat (implemented in-house on the DREW[3] collaborative software platform) as well as field notes taken by an observing researcher. Conditions varied among dyads: Two supervising tutors were involved, supervising three different projects. Dyads used one of two versions of the shared text editor. The first had turn-taking regulated by a token; the second had free access.

We are currently exploring three sets of research questions in relation to this data. The first set concerns the distributed use of the different modalities — e.g. How do participants organize communication modes, given their objectives? How do activities (e.g. talk, chat, note-taking) overlap? Can we see any effects of task parallel processing? The second set concerns forms of collaboration as they may relate to tool characteristics — e.g. does the manner of contributing to the notes or their final form change in relation to the type of text editor (free access or turn-regulated)? Finally, the third set concerns pedagogical content — e.g. what are the ways in which students reformulate teacher talk when taking notes? Does cooperative face-to-face note-taking by students make itself felt in the tutor's pedagogical activity? In order to answer these questions, it is essential to be able to view the transcript of multimodal dialogue in combination with the interaction log files (chat, text-editor) and to understand how the text was written in the shared text editor. This latter task is complicated by the form taken by the interaction log data produced by the shared text editor: events are recorded each second (provided changes have been made) and contain the full text, the name of the author who did the changes and the position of their cursor at the time of the change. While this information is useful for sending messages between computers in order to implement the shared text editor, it is not readily assistive to the analyst in that it provides too granular a view and associates too much information with each event. We are exploring different ways to show the analyst how text has changed (e.g. Kollberg, 1996).

2.2 Collaborative design in mechanical engineering

One of the objectives of the COSMOCE[4] project was to study the process of collaborative design in mechanical engineering. In this context, we set up an empirical study similar to what occurs in industry, implicating three university students at the Masters level. The students' main task was to choose an existing solution or construct a new one for building a double pulley system for a ski gondola and they carried this out during three phases:

- a preliminary stage of individual work meant to prepare them for the project review
- a collaborative stage where the project review was carried out and where we expected argumentation to occur
- a final stage of individual work after collective confrontations and decisions.

During the collaborative stage, the students discussed their solutions at a distance using the chat and shared argumentative graph editor of the collaborative platform DREW. When they had come to agreement, they were asked to use the shared text editor to describe the different constraints that their solution took into account. The data collected for this stage is the interaction log data provided by the DREW platform (chat, argument graph

[1] The European project LEAD (*Technology-enhanced learning and problem-solving discussions: Networked learning environments in the classroom*) is funded by the 6th framework *Information Society Technology* LEAD IST-028027. http://www.lead2learning.org/

[2] put the Kerbrat reference here

[3] The Dialogical Reasoning Educational Web tool (DREW) was designed and developed within the SCALE project (Internet-based intelligent tool to Support Collaborative Argumentation-based LEarning in secondary schools, funded by the 5th framework *Information Societies Technology* SCALE IST-1999-10664.

[4] The COSMOCE project (*Conception, Outils, Supports, Médias, Organisation pour la Collaboration des Entreprises*) was funded by the French Rhône-Alpes Region.

and text editor), but also hand-written and drawn documents from phases one and three.

Initial extensive analyses of this corpus illustrated student designers' knowledge dynamics in the form of patterns of argumentation based on the progressive suggestion of solutions and the criteria used to evaluate them (Prudhomme, Pourroy & Lund, 2007). These analyses combined the expertise of three researchers from four different areas: socio-cognitive interaction, argumentation, collaborative design and mechanical engineering. Challenges included combining the different forms of expertise during analysis and agreeing on how implicit mechanical engineering domain knowledge was mobilized by students. Current research questions (Lund, Andriessen, Prudhomme, van Amelsvoort, in preparation) include tracking the ways in which students' individual written knowledge construction (in stages one and three) can be linked to their collective chat and argument graph debate (stage two) and describing interactional phenomena through the construction of temporal and semantic visualisations of argumentation (Lund, Cassier, Prudhomme, in preparation). Although the result on argumentation patterns did not benefit from use of Tatiana, our aim is that the other above described future analyses do.

2.3 Face-to-face computer-mediated collaborative learning through debate

Our partners[5] in the LEAD project are currently running or have collected data on various experiments based on student small group work in classrooms using both face-to-face and computer-mediated communication and problem-solving. Design of these classroom observations varies from partner to partner but they usually produce video and interaction log data from the CoFFEE[6] (De Chiara, Di Matteo, Manno, & Scarano, 2007) collaborative platform (chat, threaded chat, argument graph, text editor, voting mechanism, etc.). Our objective is to assure that Tatiana is able to manage the variety of data and assist the various analytical approaches that will be used by our partners in this project. We faced two related difficulties so far: 1) the initial format of the interaction log data made some of the information necessary for analysis very difficult to access, 2) it is difficult to understand the link between the data imported from the interaction log files and the form taken by the threaded chat or the shared argument graph (because of the non-linear nature of these media).

In addition to the two previously presented research contexts, these situations involving student dialogue, computer mediated interactions and teacher scaffolding are another example of complex multimodal, multimedia human interaction data, stemming from multiple sources.

2.4 Affording for human interaction analysts' work

We are confronted with the difficulty of creating an object of study — from the different sources described in the previous section — that gives the researcher the most complete, faithful and pertinent view possible for the questions being addressed. Harrer et al. (2007) have modelled the analysis process (cf. Figure 1) with a view of basing the design of analysis tools and interoperable formats on this model. Our current understanding of analysis, based on our own experience and that of colleagues in various projects is that the most important part in this model is the iterative loop: researchers who arrive at the interpretation phase and are not satisfied with their results incrementally improve the analysis of their data until they arrive at a satisfactory result which can be reported to the scientific community.

[5] Universities of Bari, Nottingham, Paris, Salerno, Tilburg, and Utrecht.

[6] CoFFEE (Cooperative Face2Face Educational Environment) was designed and developed as part of the LEAD project.

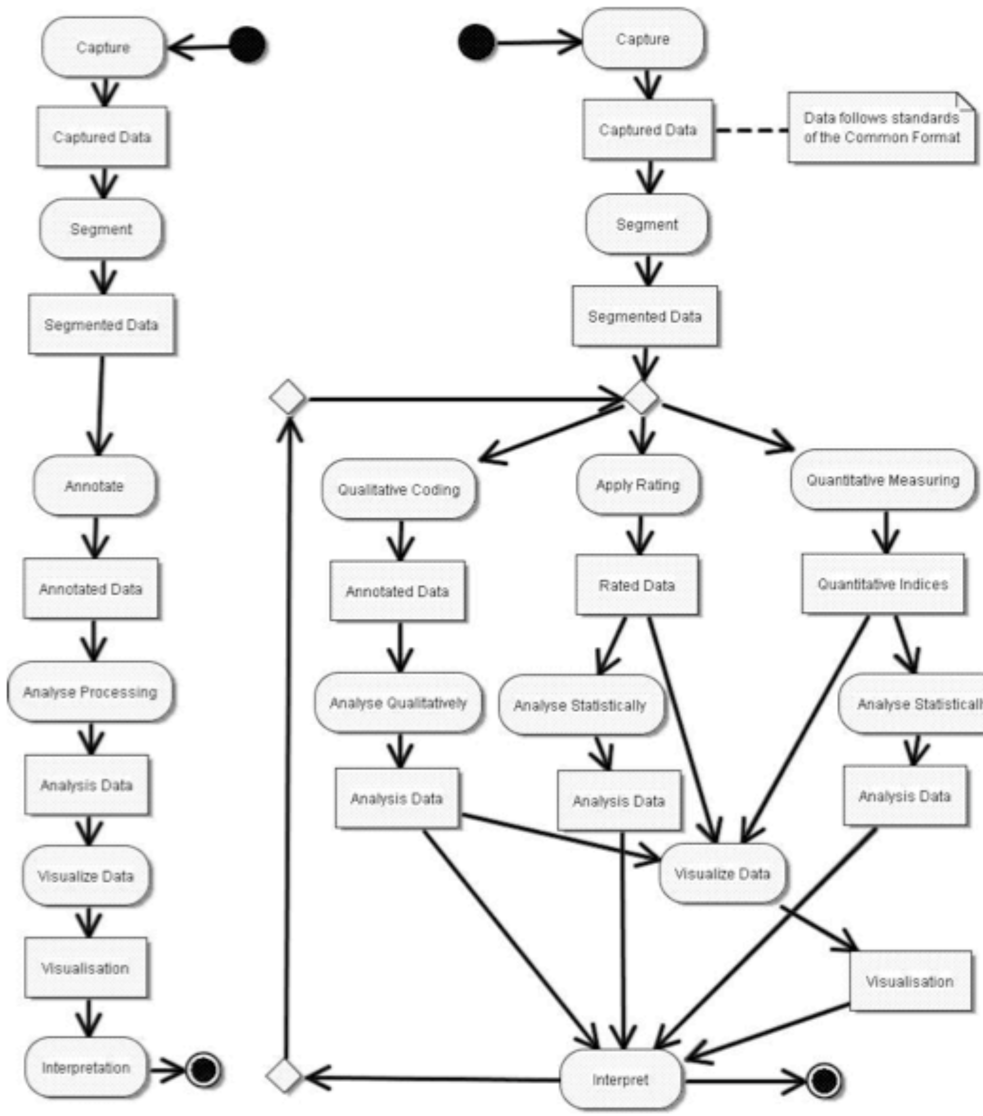


Figure 1. Graphical representation of the Cavicola process model (Harrer, et al., 2007, p. 2)

We have based our design of Tatiana on a similar (but less detailed) model that puts more focus on the iterative nature of analysis (cf. Figure 2). Analysts constantly evaluate whether their current collection of primary data and secondary artefacts is sufficient. If it is not, they create a new artefact that is intended either to further their understanding of the data or to reify their current understanding of the data.

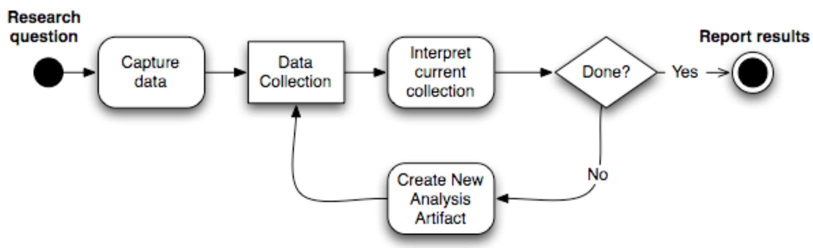


Figure 2. Graphical representation of the Tatiana process model

Sometimes the creation of these artefacts is automated (e.g. transforming data into a new representation, statistical analysis), sometimes it is manual (e.g. creating visualisations) and sometimes it is tool-assisted (e.g. transcription, annotation and coding). These artefacts are frequently representations of data with some kind of temporal dimension: the sequence of events recorded in the observation is made available to the researcher, either by presenting it in a replayer (the researcher uses a remote control to navigate the data, see §Tool replayers) or in some kind of graphical representation (where time is plotted along a vertical or horizontal axis).

We propose that these two kinds of artefacts which retain some notion of ordering of events and interactions in time be termed *replayables*. These are objects that can be replayed, synchronised and analysed. We further propose that analysis consists of the iterative creation of replayables that exhibit researchers' understanding of their data or that allow them to further their understanding by the creation of new replayables. The step from one set of replayables to the next includes transformations such as: transcription, annotation, coding, visualisation, filtering, synchronizing, merging, etc.

In the next section, we will briefly present existing tools that employ some of these transformations, but in alternative frameworks.

3 A selection of analytical tools for understanding human interaction

Tatiana was designed and developed out of real analytical needs (cf. §In regards to three analytical contexts), but also in relation to existing tools. When describing the rationale behind the various features of Tatiana in the coming sections, we will frequently refer to other tools and how they implement the feature in question. Here, we briefly present those tools from which we will be taking example.

Elan (EUDICO Linguistic ANnotator where EUDICO is European DIstributed Corpora Project) (http://www.lat-mpi.eu/tools/elan/) is a tool for performing transcription and annotation of video and audio recordings. It is intended for the analysis of linguistically interesting phenomena.

Videograph® (http://www.ipn.uni-kiel.de/aktuell/videograph/enhtmStart.htm) is a multimedia player with which digitalized videos, e.g. recordings of classroom instruction, can be played and evaluated. The program enables the construction of observation categories and rating scales which the viewer can use as a "measuring instrument" to analyse the contents of the video.

Anvil (http://www.anvil-software.de/) is a free video annotation tool, that offers frame-accurate, hierarchical multi-layered annotation driven by user-defined annotation schemes. An annotation board shows color-coded elements on multiple tracks in time-alignment. Special features are cross-level links, non-temporal objects and a project tool for managing multiple annotations. Originally developed for Gesture Research, Anvil has also proved suitable for research in Human-Computer Interaction, Linguistics, Ethology, Anthropology, Psychotherapy, Embodied Agents, Computer Animation and many other fields.

CLAN (Computerized Language ANalysis) is a program that is designed specifically to analyse data transcribed in the format of the Child Language Data Exchange System (CHILDES). CLAN allows you to perform a large number of automatic analyses on transcript data. The analyses include frequency counts, word searches, co-occurrence analyses, MLU (Mean Length of Utterance) counts, interactional analyses, text changes, and morphosyntactic analysis.

The Nite XML Toolkit or NXT (http://groups.inf.ed.ac.uk/nxt/) is a tool for complex annotation of multimodal, textual or spoken corpora. NXT is in use on a wide range of corpora, representing everything from Biblical text structure to the relationship between deictic expressions and gestures in multimodal referring expressions. Its extensible design allows it to also be used for analysis of interaction log data.

ABSTRACT (Analysis of Behaviour and Situation for menTal Representation Assessment and Cognitive acTivity modelling; Georgeon et al., 2007; Georgeon, Mille, & Bellet, 2006) is a framework designed to assist the understanding of human activity from computer-recordings of these activities. It features a language for querying and transforming the data into increasingly abstract elements and visualising the results in a symbolic graphical way.

Replayer (Morrison, Tennent, & Chalmers, 2006) is a tool primarily designed to aid the evaluation of mobile applications. It allows the synchronisation of many forms of data such as video and system logs, along with different visualisations of these logs (graphical, tabular, superimposed on a map, etc.)

DRS (Digital Replay System, Greenhalgh, French, Humble, & Tennent, 2007) is a tool aimed at analysis of fields in the general domain of e-social science where data is comprised of video, audio and other sources such as eye-tracking and system logs. Like Replayer, it allows the synchronisation of visualisations of this data. It further provides means of adding various types of annotation to the data.

ActivityLens (Fiotakis, Fidas, & Avouris, 2007) is software primarily intended for usability evaluation of educational software. It synchronises video, field notes, log files and images and provides features for annotation, statistical analysis and activity theory analysis. This tool is an evolution of an earlier tool called ColAT (Collaboration Analysis Toolkit, Avouris et al, 2007).

The Synchronized Analysis Workspace (SAW, Goodman et al., 2006) is an analysis and visualisation tool to assist the navigation and analysis of multiple synchronized data streams It provides (among other features) time-aligned symbolic graphical representations of log events, the ability to filter these events and the capability to share partial or complete results with other analysts.

We will also be interested in tools such as DREW (Corbel, Girardot, & Jaillon, 2002; Corbel et al., 2003), Digalo[7] (Lotan-Kochan, 2006) and CoFFEE (De Chiara et al., 2007) which all feature a tool replayer. A tool replayer is a mode of the software application that reads the interaction log file of a session and replays the internal state of the software, along with its corresponding user-interface elements. Replaying a session is a way for the researcher to follow student activity as it unfolds and also gives the researcher a partial idea of what the students were experiencing at that time.

4 Tatiana Features and Rationale

We can organise the features of Tatiana in five broad categories.

1. Data management: import, export and modelling of data within the software.
2. Synchronisation: time alignment of different data sources and description of available user interface elements used to navigate the temporal dimension.
3. Visualisation: transformation and visualisation of data in order to assist the researcher's analysis.
4. Analysis possibilities offered by the tool.
5. Extensibility – manipulation of new kinds of data, visualisations and analyses.

4.1 Imported and produced data

Any tool which assists data analysis handles two types of data: 1) data that it intends to analyse and 2) data that is produced by the tool. Elan, Anvil and Videograph, for example, mainly focus on video and audio files as input data and create their own file formats to describe transcriptions and annotations. Audio and video formats are standardised, and can generally be handled by widely available libraries. These media are managed in all the tools we have presented, including Tatiana. However, while traces and data are produced by the tools used in computer mediated activities, tools used for analysis must choose how to handle data import and how to create an internal representation of data that is both powerful enough to do what is needed, yet flexible enough to handle many kinds of data.

ActivityLens chooses to import interaction log files that are represented in a common format proposed by the CaviCola (Computer-based Analysis and Visualisation of Collaborative Learning Activities) working group of the CSCL SIG of the Kaleidoscope network (Kahrimanis, Papasalouros, Avouris, & Retalis, 2006). While this format allows ActivityLens to understand such notions as steps, groups, users and tools, it imposes a particular way of organising data. However, this common format remains very generic, allowing a wide scope for additional fields if necessary. ABSTRACT describes the imported data using OWL/RDF ontologies and creates parsers to explain the import of data. This implies that for any data to be imported, a parser must be written and an ontology must be created to describe the data's structure. Unless some kind of kernel ontology is defined, ABSTRACT is not in a position to interpret data. NXT and DRS provide different viewers for different kinds of data. NXT can import any data that is similar to a transcription (this may exclude some types of complex interaction log data). DRS provides a wizard for importing data that is in some way structured in a table and provides the opportunity to write custom classes for importing more complex data.

4.1.1 Data in Tatiana

We choose to follow yet another path. We assume that an analysis tool should be capable of reading any data file (our caveat is that non-audio/video files should be written in XML, with an associated structural description) and that the cost of accepting a new file format should be minimised. We also assume that it is not necessary for the tool to interpret any semantic concepts in the files (save those needed for analysis and synchronisation – see below) and that it is sufficient to name and present information in a way that allows the researcher to add this meaning. To this end we have chosen a solution that is very generic. The internal structure handled by Tatiana (the Tatiana display format) is a sequence of events, each of which has a number of facets. Each facet is named and typed – each event can have an arbitrary number of facets. Import is handled through XQuery (W3C, 2007) scripts. These scripts can be combined in a filter definition language that allows generic filters to be described (we will address these filters in more detail in the section on visualisations). In general, an import script will select a subset of the file to be imported, such as the chat of a DREW interaction log file or a transcription output by the Elan tool and assign the various information known to be contained in each event of the file to distinct facets in the corresponding display format event. A generic merging filter can then be used to combine the selected information. From this representation, export to any format is relatively easy, provided all the necessary information is present (so far, Excel™ is the only widely used format for which we have implemented export as

[7] Digalo, a tool for graphic-based e-discussion and e-argumentation, was designed and developed within the DUNES project (Dialogic and argumentative negotiation educational software), funded by the 5th framework *Information Society Technology* and continues to be used in the ARGUNAUT project (An Intelligent Guide to Support Productive Online Dialogue) funded by the 6th framework *Information Society Technology.*

previous analyses were carried out within this application).

In both of the LEAD corpora described above, it is necessary to create a replayable which combines data from interaction log files and transcriptions of audio and video files. Our method has so far proved successful for importing the data from two very different interaction log-file structures (CoFFEE and DREW) and from data in the Elan file format (Elan having been used for transcription). Owing to the awaited release of the CoFFEE replayer, it has been necessary to provide a temporary way of giving meaning to data produced by the CoFFEE argumentative graph editor. This has been done by importing this data into Tatiana and exporting it to a DREW file, thus verifying that the Tatiana display representation is an acceptable pivot format. In this way, we provide the means to visualise CoFFEE graph editor data, albeit in the DREW replayer.

4.2 Synchronisation of different co-occuring data streams

All the tools described above provide synchronisation between different types of replayables, whether they be simply video files, transcriptions and annotations (Elan) or a wide variety of different data (SAW, Replayer, DRS). The analytical necessity of synchronisation of different data sources and replayables is plain: the observed events — at the time of observation — were temporally situated and must be replaced in this context to be understood. Furthermore, different views on the same data (e.g. video and transcription) complement each other well when they are synchronised. In fact, it is a scientific necessity to have this synchronisation; when making a claim the question that is always asked is: "is there evidence in the data to back up this claim?". Analysts frequently make use of replayables that present the corpus in a way that is more readily understandable, browsable or analysable (compare transcriptions and the video/audio they transcribe). However, when a claim is postulated based on information found in one of these "secondary" artifacts, it is necessary to verify that the original data also evidences the claim that is being made. Synchronisation presents a way to easily refer back to the primary data at any point where confirmation of a claim is needed.

There are two main methods for time-aligning data – the first is to assume that it was correctly timestamped when it was captured and that it is thus naturally aligned and easy to synchronise (e.g. SAW and NXT). The second is to provide time-alignment information when constructing the corpus. DRS and ActivityLens provide graphical interfaces to do this.

Once this data is time-aligned, it is interesting to look at methods of navigation through the corpus. Tools like Replayer (cf. Figure 1) and SAW provide a general "brush and link" functionality (Becker & Cleveland, 1987). Selecting any element or collection of elements in one view of the data selects elements of the corresponding timestamp in other views. We believe this kind of navigation is crucial, particularly when it comes to linking ABSTRACT graphical representations to the data they are based on. Another navigation tool is the metaphor of the video control (see below right in Figure 3), which allows navigation along a timeline and provides such functionalities as play and pause.

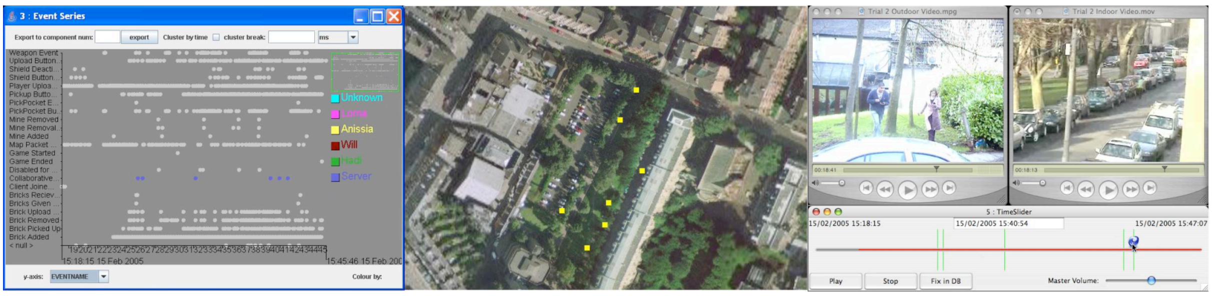


Figure 3. In this example from Replayer, the events selected in the left hand view are also highlighted on the map and in the video timeline on the map (http://www.dcs.gla.ac.uk/%7Emorrisaj/Replayer.html).

4.2.1 Synchronisation in Tatiana

Tatiana chooses the second method of synchronisation – that of providing time-alignment information when constructing the corpus. Tatiana's display format events thus have a compulsory facet: corpora are described with the beginning and end dates in milliseconds along with the time difference necessary to translate this into a "unix timestamp" (milliseconds since January 1st 1970). It follows that importing scripts must be written with this in mind, thus enabling us for example, to merge data imported both from interaction log files and transcriptions with relatively little effort. Tatiana also uses a brush and link strategy: when an analyst clicks on the naming of an argument graph element in the area where the chronological list of events are displayed, the replayer (see §Tool replayers) positions itself to the moment in time where the box was named in the argument graph. We also aim to link researcher constructed representations with the original data from which they were built (e.g. the patterns of argumentation found in the COSMOCE corpus). The video control metaphor — especially with fast-forward — is very useful for getting a quick overview of a long session in order to identify interesting periods. For example, in the case of argument graph construction, one can identify periods of high or low activity, interface problems, constructions that generated a lot of dialogue, etc.

4.3 Visualisation of multimodal human interaction data

The goal of recording data of a session for a later analysis is threefold. Firstly, too many things happen at once to make real-time analysis possible (although field notes often provide a lot of insight that might otherwise have been missed). Secondly, the data is overwhelming in complexity and quantity. Thirdly, the analysis process requires that data be both a working artifact (e.g. a document or video that can be annotated) and evidence that can be referred back to, so that claims can be independently verified. Indeed, making CSCL and CSCW corpora easier to share and independently analyse is a scientific necessity (Betbeder, Ciekanski, Greffier, Reffay, & Chanier, 2008). The cost of this is that we are left with corpora that are not only overwhelming in the sheer amount of data that is presented, but also whose link with what was happened during data collection is difficult to establish, particularly in the case of interaction log files (video poses other problems e.g. angle of shots). An immediate reflex when presented with this problem is to create replayables which filter out a lot of data, reducing both the information overload and the difficulty in understanding the link to what happened. This is dangerous: there is a risk of misinterpreting this data through lack of context. The challenge is thus to produce a combination (through synchronization) of replayables that individually only show a restricted amount of information but together paint a global picture which always allows the researcher to have the context readily at hand. We will look at different possible kinds of visualisations of these replayables and at how replayables can be constructed, transformed and formatted before being visualised.

4.3.1 Existing visualisations

One of the simplest CSCL tools is the chat. The logs it produces can be easily visualised as tabular data with one row per event and columns for the different data of that event: timestamp, user, content. As a consequence, tools such as Excel™ are often used. This is also the main type of visualisation adopted by ActivityLens and one of those proposed by Elan, DRS and others. Another view, common in video analysers such as Elan and also present, for example, in DRS (cf. Figure 4) is the horizontal time-line on which events are presented as a duration and completed with textual descriptions.

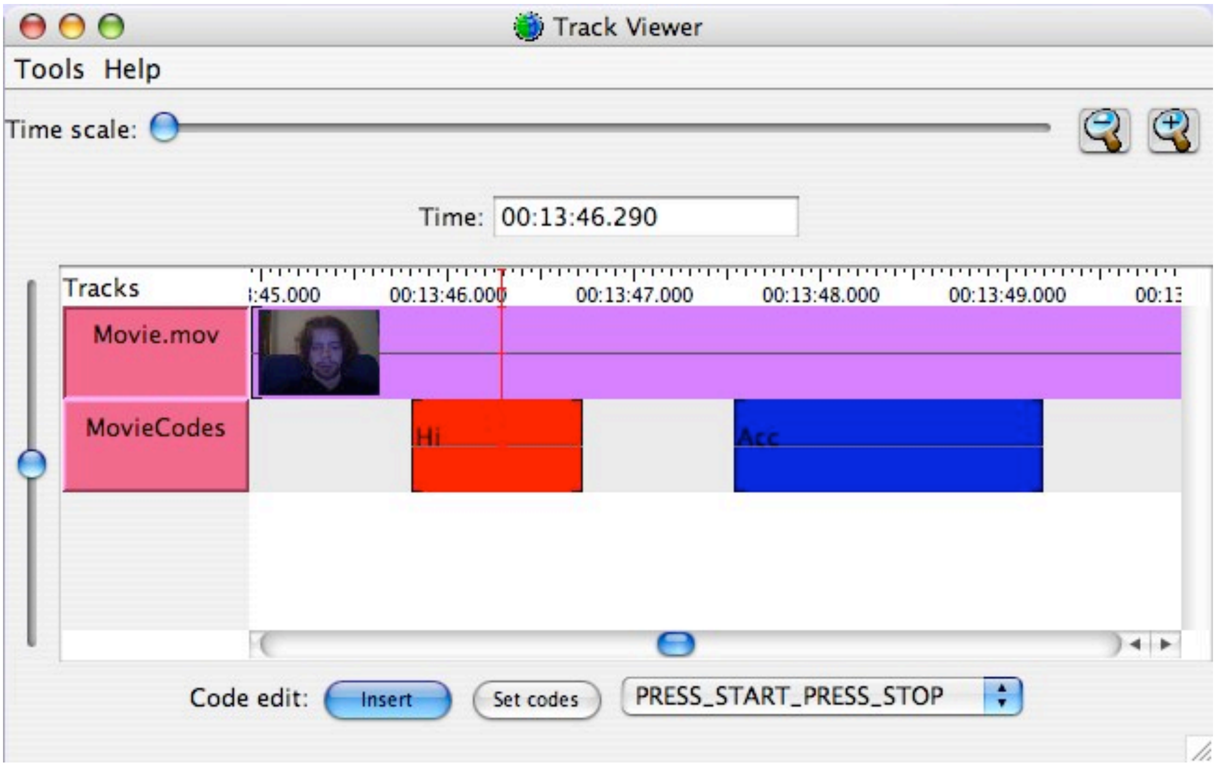


Figure 4. DRS, most types of data can be viewed in the track viewer. Here the transcription is present, combined with the colour-codes of the categorisations assigned to this transcription. (http://www.mrl.nott.ac.uk/research/projects/dress/software/DRS/Home.html)

A very similar form of visualisation is a symbolic graphical representation of events on a horizontal timeline. This feature is pervasive, being present in SAW, ABSTRACT (cf. Figure 5), Replayer, DRS and others. Our main criticism of these visualisations is the current lack of configurability. While they provide valuable information, the ability to tweak the presentation, particularly the graphical layout of these events is, to our knowledge, lacking (except through programmatic intervention). Other forms of visualisation include presentation of data in a graph (with time on the horizontal dimension and some other information such as speed or heart-rate on the vertical dimension) and presentation of GPS data on a map or overhead picture (cf. Figure 3).

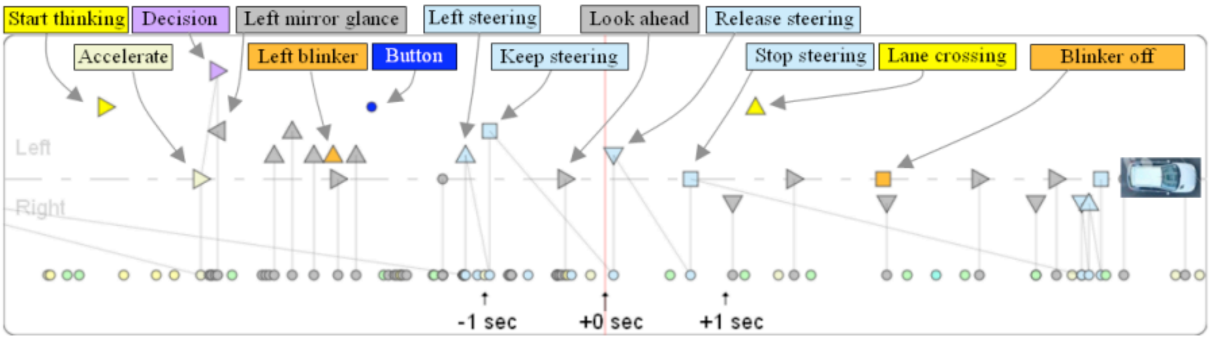


Figure 5. Symbolic graphical representation of events recorded during a lane change on a motorway, viewed in ABSTRACT (http://liris.cnrs.fr/abstract/)

4.3.2 Tool replayers

A final method for making sense of interaction log data is to synchronise one visualisation (e.g. tabular) with an on screen representation of what the subjects under observation had on *their* screen. This situation is similar to that of linking a transcription to a video, with the exception that tools already provide their own transcript. While one way of doing this is to use screen capture, we argue for the use of external tool replayers either in addition to or as a replacement for screen capture. Tool replayers read the interaction log file that they previously produced and interpret each message almost as they would if they were an additional client during the observation. The disadvantage of this is that the interaction log is almost certainly a subset of the events that really happened such as scrolling, moving the mouse, selecting, clicking and typing. As such, there is never a guarantee that what the analyst sees on his or her screen is what the user saw on theirs (in fact this would be highly unlikely). There will, however, be a large similarity – hopefully enough for the researcher to gain the necessary insight to understand what a given line in the log means and how it fits into the wider context.

The advantages, on the other hand, are numerous and in our view they outweigh this disadvantage. The first is in terms of memory usage: video files are cumbersome and screen captures are not always easy to collect. Furthermore, a screen capture would only show what one student saw, not showing elements that are not in the current window or scroll frame and necessitating either the collection of screen captures on all machines involved or the similar caveat that the analyst only sees what one student saw with no guarantee that other students saw the same thing. Through strategic logging and replay of user interface of events, this problem can also be circumvented. For example, scrolling could be captured and the replayer would show a scrollbar as a navigational aid for the researcher and a series of "shadow" scrollbars representing the current view of each user (cf. Figure 6). Replayers in general can allow the analyst to manipulate the original user interface, preserving quality when windows are resized (unlike video which can become very blurred) and navigating to elements that are out of view (such as new messages that appear outside of a users currently scrolled view).

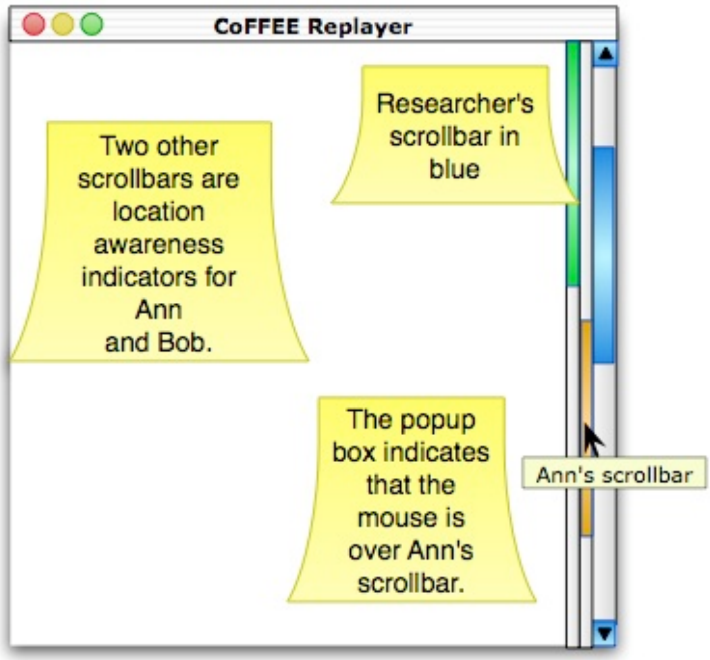


Figure 6. The researcher's scrollbar is shown in blue. She can use it to freely navigate the workspace. However, at any given point during the experiment, Ann and Bob were only looking at a subset of that workspace. If the researcher were to bring her scrollbar parallel to Ann's scrollbar, for example, she would see what Ann could see at that moment in time.

The main and crucial advantage of tool replayers is the possibility of benefiting from the internal software state that the replayer reproduces, in order to show information to the researcher that was not present in the original user-interface or to show it in a different manner. To illustrate this possibility, we have two examples drawn from our own corpora. In the DREW shared text editor, newly contributed text is briefly displayed in the user's colour (for awareness purposes) before fading to black. In the corresponding replayer, however, this text keeps its colour, thus making it easier for the researcher to understand who wrote what (cf. Figure 7, taken from the experiment described in Lund, Rossetti, and Metz, 2007).

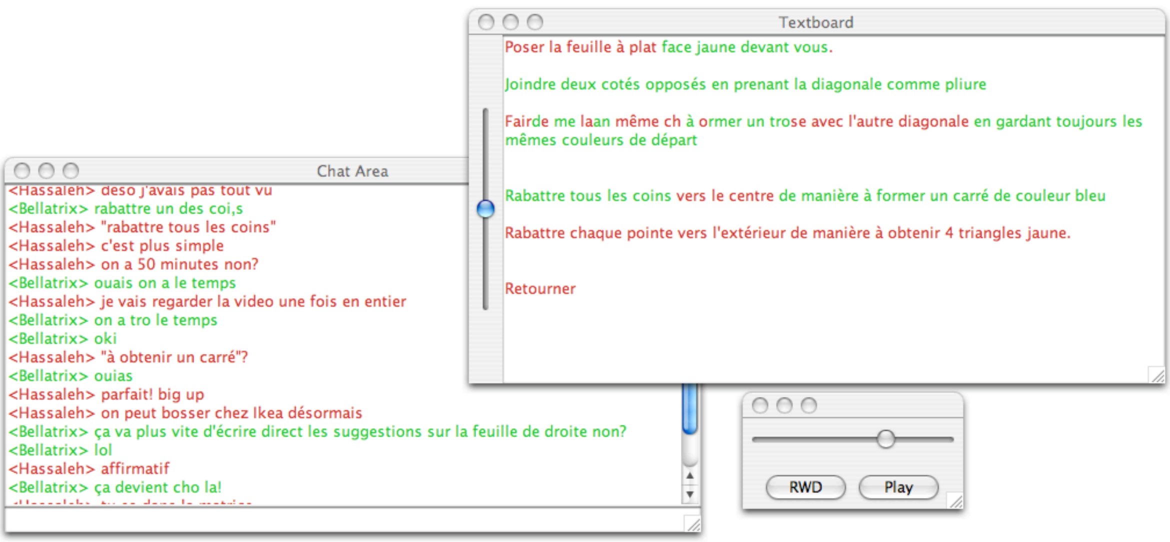


Figure 7. Replay of the DREW chat and collaborative text editor: while the chat conserves the different colours that were seen by the users, the users' text editor had only black text. The replayer colours this text to facilitate the comparison of the origins of text editor and chat interventions.

The corpora which contain use of the DREW and/or CoFFEE argumentative graph editors are made difficult to analyse by the constant manipulation of already created objects. A box might be created, later edited, then moved again and finally deleted. All along, a system needs to be provided so that the researcher can understand that this refers to the same object and know what the current properties of that object are. As we will see when discussing formatting (cf. §Extensibility), we choose to number these objects and refer to them by number. But which box is number 3? As explained above, we were able to use Tatiana to transform a CoFFEE trace into a DREW trace, pending the development of a CoFFEE replayer. We produced a trace[8] showing both the contents and the unique identification of each box (cf. Figure 8), bearing in mind that this is not what the students saw in their interface. Tatiana is designed to interact with external tool replayers so as to synchronise them with the rest of the data through a relatively simple XML-RPC protocol.

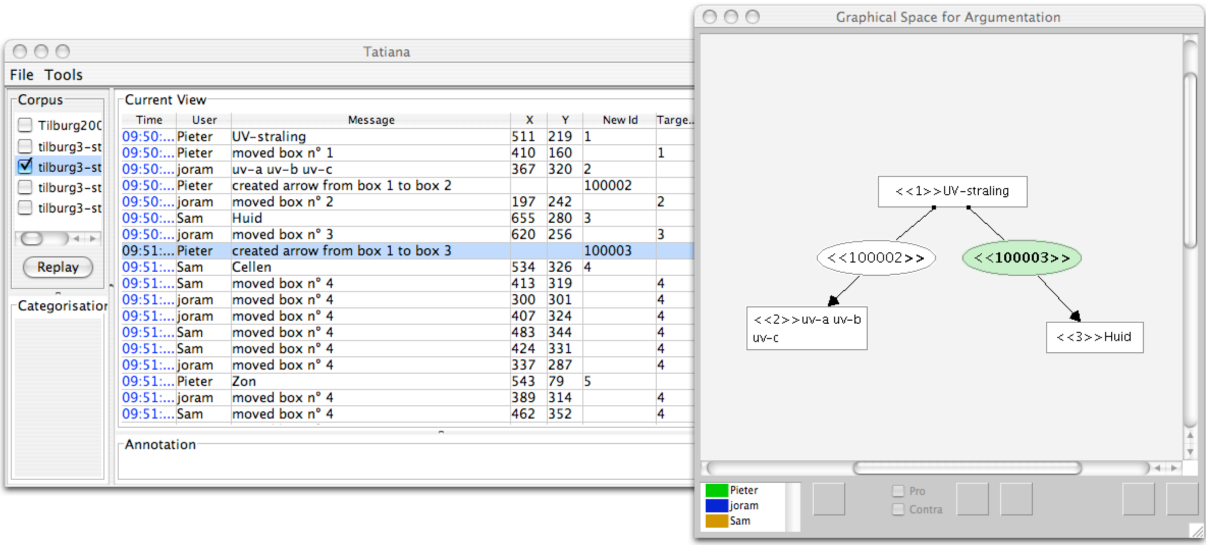


Figure 8. When replaying the CoFFEE graph editor in the DREW replayer, we show both the content that the users could see (e.g. "UV-straling") and the id of that box as seen by the researcher in Tatiana. Above, Pieter has just created a link between boxes 1 and 3 identified as arrow 100003.

4.3.3 Visualisations in Tatiana

Visualisations in Tatiana are tabular views (a row for each event, different information in different columns) and a horizontal timeline symbolic graphical view (currently under development). Tatiana displays data that is structured according to its Display format. The tabular view is parametered by a description of columns saying what columns should be present and the facets from which this information should be drawn. This enables us to avoid having too many columns that are almost always empty at a small cost to the researcher. For example it is useful to have the unique identification of a new box and that of a new link between boxes in the same column, in spite of these facets having different names (cf. Figure 8). The researcher can also modify this arrangement if it is not convenient. For the graphical view, the graphical object representing each event has several properties: Position, shape, size and colour. The object's length and position on the horizontal axis is defined by the timestamp and duration of the event. Its other parameters are calculated according to various facets in rules defined by the researcher within Tatiana. The necessity for this kind of configuration came about from our painstakingly creating a prototype visualisation (cf. Figure 9) of the distributed use of the different communication modalities in our shared note-taking situation only to realise that we had grouped modalities by user when it would have been clearer had we grouped users by modality (i.e. having the lines for dialogue of Student A, Student B and Teacher all grouped together under a common dialogue heading, rather than having them under different user headings). We think it is important for researchers to be able to create these kinds of visualisations in a way that is easy to iteratively improve, and to have them synchronized with the rest of the corpus (Lund, et al. 2008). We are currently working on a way to add links between graphical objects into this kind of visualisation.

[8] The data shown here was gathered by the Tilburg university partner in the LEAD project.

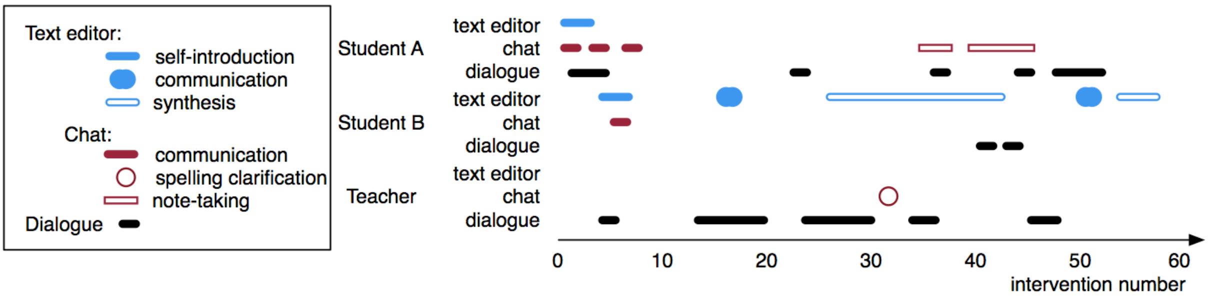


Figure 9. Distribution of communication across media and modalities.

In regards to the COSMOCE corpus, researchers created new argument graphs that rearranged the students contributions in the argument graph editor and added in contributions from the chat as well as their explicit or implicit links to other elements of the argument graph We think it will be possible to modify our graphical visualisation to cater for this kind of analysis activity while preserving brush and link functionalities (cf. Figure 10). This would allow researchers to visualise the temporal logic behind a graph that is not temporally organised.

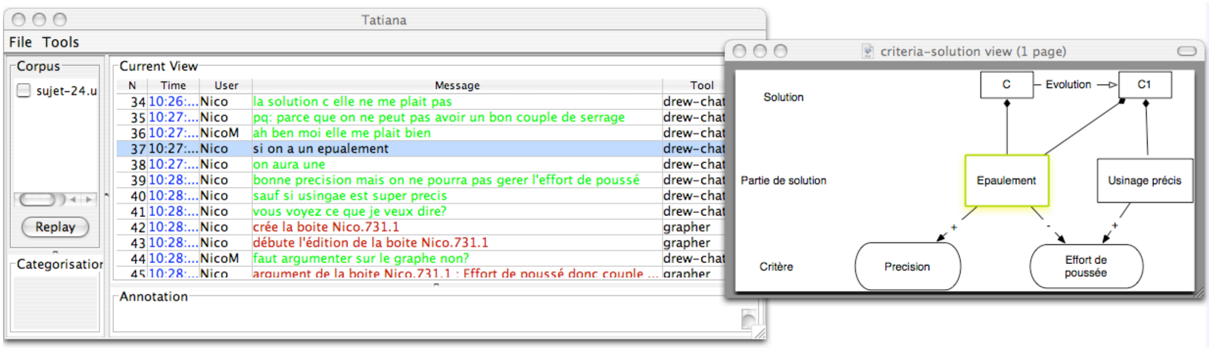
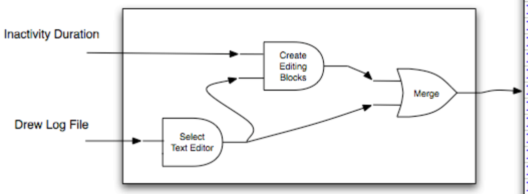


Figure 10. Mockup of synchronisation between a graph created by a researcher describing a solution proposed in a mechanical engineering observation and the actual interaction data collected.

4.3.4 Data transformation for visualisation in Tatiana

In order to prepare data for visualisation in Tatiana, we again make use of XQuery scripts and the generic filters which combine these scripts. For example, the script which extracts information such as the new position of a moved box in the DREW argument graph editor also creates a facet for that event with a human readable content of the type "Ann moved box n°3 to coordinates 456,324". Another script could rewrite this information to "Ann moved box 3 up and left", if this information was deemed more pertinent for the researcher. A Tatiana filter could then combine these two scripts together, the first serving to extract data from DREW's argument graph editor and the second serving to rewrite information present in a Tatiana Display file. In a similar way, scripts can also be used to filter data (such as only showing information from a single group in CoFFEE) or creating new events or grouping events together.

As an example of this, in our shared note-taking corpus, we only had limited funds available for transcription and wished to have a preliminary transcription done only of events that occurred during periods of typing. Because of the nature of the shared text editor interaction log files, identifying periods of writing is not easy, short of watching the video and marking down when it seemed when a student was typing. In order to do this, we created a script that identifies periods of inactivity in a given tool and creates a replayable (in the Tatiana display representation) with events of beginning and end of writing blocks (a block being defined by a surrounding period of inactivity). The combination of these scripts produces a re-usable filter (cf. Figure 11) for identifying inactivity/activity in the DREW shared text editor. This enabled us to easily summarise a session containing hundreds of events indicating the modification of the contents of the shared text editor into just a few blocks of activity with at least 60s of inactivity between blocks. We then only transcribed periods of activity. We will be using this filter as preliminary work for manually grouping edition events (produced every time a character is edited) into more meaningful blocks in terms of text edition. We are currently working on a graphical interface for the creation and edition of these filters by the researcher, allowing the researcher to combine pre-written scripts in new ways to suit his or her current needs.



Time	User	Message	Tool
14:59:03	francois	begin editing block. 06s until next 20s break	textboard-begin-write
14:59:04	francois	1 chars added near munauté) -	textboard
14:59:05	francois	3 chars added near auté) -> a	textboard
14:59:06	francois	3 chars added near é) -> adap	textboard
14:59:07	francois	6 chars added near adapter no	textboard
14:59:08	francois	4 chars added near ter notre	textboard
14:59:09	francois	5 chars added near otre forma	textboard
14:59:10	francois	1 chars added near tre format	textboard
14:59:10	francois	end editing block. 5mn39s until next writing block	textboard-end-write
15:04:49	Armand	1 chars added near adjacentes	textboard
15:04:49	Armand	begin editing block. 6mn05s until next 20s break	textboard-begin-write
15:10:55	Armand	3 chars added near format	textboard
15:10:55	Armand	begin editing block. 1mn26s until next 20s break	textboard-begin-write
15:10:56	Armand	3 chars added near rmat	textboard

Figure 11. A Tatiana filter combining three filters which takes as parameters a DREW interaction log file and an inactivity duration and produces a replayable containing three kinds of events: *textboard-begin-write*, *textboard-end-write* and *textboard*. This replayable is then visualised in a table view.

5 Analysis of multimodal human interaction data

Once the data has been transformed into replayables that can be viewed and initially understood, the analysis process will consist of creating new replayables which further enhance understanding by adding extra meaning and abstraction in order to describe what happened during the observation and make claims of a quantitative or qualitative nature. We will base our description of analytical requirements on existing tools (which one assumes satisfies a variety of analytical necessities) and on our own experience.

5.1 Analysis practices and corresponding tool requirements

Nearly all the tools presented that afford analysis (NXT, DRS, SAW, Elan, ActivityLens) allow this analysis to take the form of free textual annotations either of segments of a corpus (described by a beginning and end, but not related to a specific media) or of singular events. Even though the practices of annotating are distinct from the practices of categorization, they can be implemented with the same technology: annotations can be restricted to a set (either open or closed) of keywords that in essence afford categorisation. In order to annotate segments, it is necessary to create segments. Transcription can also be technologically equated to segmentation and annotation. The case of annotation drawn from a set of keywords (i.e. categorisation) is of particular interest: in the NXT, for example, these categories can be described in complex ontologies in order to use these categories as a basis for adjacency pair searching.

Categorisation is also used as a prelude to quantitative analysis – in this case, the segmentation function becomes more important as it is necessary, when counting, to count objects of the same nature. The ability to re-segment a corpus – for example by grouping elements together – would allow us, for example in the shared note-taking corpus, to group the events of the interaction log data together into meaningful "chunks of writing". The log produced by the DREW argument graph editor is also verbose: the act of creating a box produces a "create" event and a "move" event which is actually the initial placing of the box – this may be counted as one or two events, depending on the analysts' perspective and the type of "actions" they wish to count. ABSTRACT provides a way of grouping events together automatically: the analyst can write queries which describe rules for deriving increasingly abstract and semantically rich entities – data captured from a steering wheel can be combined into "turn the car" events which and in turn be converted to "change lane" events, which can then be combined with data from gaze-tracking and the car indicators in order to identify "anticipated lane change" and "unexpected lane change".

An analysis activity which none of the tools we have encountered seem to support is that of describing relationships between events. In the previous sections, we have shown how it is desirable to render explicit certain implicit relationships between objects or events: for example, it is useful when analysing our collaborative design corpus to show how one solution is in fact the evolution of another that retains some of the previous solution's aspects while incorporating modifications to make the solution acceptable for additional criteria (cf. Figure 10). These relations can then be used to create graphs or to look for different kinds of adjacency pairs – for example two chat interventions of type question where one is described as being a rephrasing of the other.

A final analytical necessity is that of being able to save and share both corpora and analyses based on these corpora for the reasons already mentioned in the introduction.

5.2 Implementation of analysis in Tatiana

We have thus identified three broad technological classes of analysis activity: annotation, segmentation and creation of links. Tatiana supports these actions through a system similar to that used by the NXT : an anchor is a specific point in a corpus and analysis files describe each analysis action as a series of links to various anchors (Corbel, Girardot, & Lund, 2006). In this case:

- Annotation is a link associated with some text whereas categorisation is two links (one to the category, the other to the anchor being categorised) — for example, an event of creating a box in the graphical tool could be annotated "This is the second time user has created this box". The link would be to an anchor in the interaction log file (the XML element of that event). Applying the category "affirming criteria" to the creation of a link in the argumentation graph tool would create two associated links: one to an anchor in the interaction log file and one to the category in a file enumerating the categories.

- Segmentation/grouping of events is a set of links — grouping five events of textboard events together (because they constitute some kind of "unit of edition") is creating a set of links to the anchors of each of these events.
- Creating a relation is associating two links — claiming that one message in the chat is a reply to the creation of a box in the argument graph is to create two associated links: one to the chat event in the interaction log file; the other to the graph event. Identifying this relation as a "reply" could either be done by annotation, through associating some text to these two links or by categorisation, through associating a third link to the category itself.

Using this information, an analysis file can be applied to a representation in the Tatiana Display format. The result of this action is to import the analysis information as supplementary facets that can, in turn, be used for visualisation.

As Tatiana Display representations are subject to change or re-generation (with a different filter, by mixing in different elements or to present data in a different way), it is important that a link be universal across a corpus. To this end, an import script will create a "source anchor" facet to each event. This system is constrained such that any event will be represented by the same source anchor by any importing script and that any subsequent transformation (filtering, merging) conserves source anchor values. In other words, for every event in the corpus, a “universal” source anchor will refer to it unambiguously.

Analysis as it is currently implemented in Tatiana is limited to categorisation and annotation. Categorisation is fairly rudimentary as we have no interface to specify categorisation ontologies. We hope to work on this in the future along with interfaces for segmentation and creation of relationships. We are, however, confident that the current specification of Tatiana analysis files is sufficient to perform the types of analyses we have described and for sharing these analyses with other researchers using Tatiana (or any other tool that can be interoperated with Tatiana).

6 Extensibility

Different researchers have different needs. While it is impossible to develop a tool which will help any researcher with any kind of research question, we have identified some aspects for which it is useful to make an analysis tool extendable: data import and export, data transformation and formatting, data visualisation and data analysis. DRS, Replayer and NXT provide various views of the data and the means to create new views. They also provide the means to add scripts for importing data from various formats, usually in conjunction with a new view. None of the tools, however, seem to provide an easy way to manipulate the data in ways that were not originally intended. Furthermore, it is important to decide what is meant by an extension. Tatiana is released under an open source license and can, as such, be modified by anybody who has a copy of the software. Setting up the build environment for such complex software is necessarily time-consuming and this model can lead to many incompatible forks of the same project. We prefer to work with a plugin extensibility architecture that allows third party contributions to be added without recompiling Tatiana.

The planned architecture for Tatiana (cf. Figure 12) provides several areas for extension by third parties in order to allow them to manipulate replayables within Tatiana in novel ways. Bearing in mind that researchers use a wide variety of tools, import scripts can easily be added by coding an additional XQuery script. New scripts can also be written for any kind of data transformation. The scripts in the Tatiana library can be combined to form reusable filters. Finally these scripts can also be used to export data to any XML format. Adding a new script to Tatiana is as simple as adding the file to Tatiana's scripts directory. Three kinds of scripts can be used with Tatiana:

1. Import scripts produce XML in the Display format. For each event, they create a "time" and "source anchor" facet.
2. Data transformation scripts read from and write to the Tatiana Display format. They preserve or aggregate the two compulsory facets.
3. Export scripts read data in the Display format and produce XML.

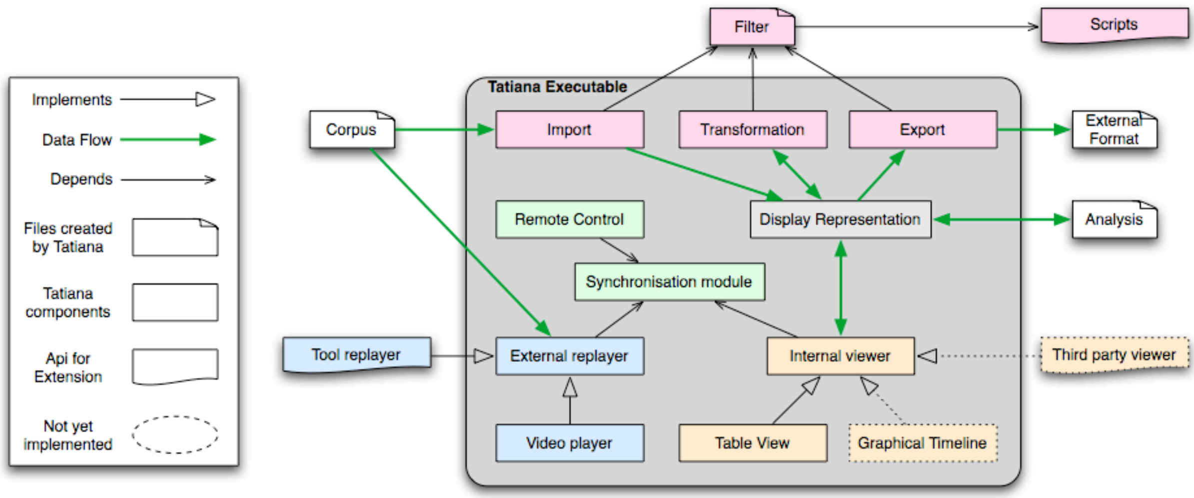


Figure 12. Tatiana architecture showing 1) dependencies between components, 2) components designed with extensibility in mind and 3) future developments.

As described in the section on visualisation, Tatiana provides an API[9] for external tool replayers. Such a replayer must, upon being launched, create an XML-RPC server which accepts the following requests:

- OpenFile(filename) – open the specified corpus file.
- Goto(timestamp) – position the replayer at the specified timestamp.
- Mark(timestamp) – mark the beginning of a selection.
- End() – close the replayer.
- Accepts(command) – asks the replayer whether it implements the given command.
- GetEvents() – return a list of timestamps matching the events present in the file.

With this communication protocol, Tatiana is able to pilot an external replayer and synchronise it with the rest of the data by sending Goto requests when it is necessary to do so (based on the list of events returned by GetEvents). Currently, there are two replayers which implement this protocol: the DREW replayer and a proof of concept version of the CoFFEE replayer (a first stable release is scheduled in the coming months).

Finally, Tatiana is currently being adapted to the Eclipse RCP platform (REF) in order to take advantage of Eclipse's plugin architecture (REF). In order to install a compiled eclipse plugin, all that is necessary is to copy it in the plugins folder and restart the Eclipse RCP based application. A future move will be to define the interface necessary for implementing third party viewers. A first part of this interface concerns synchronisation and will be similar to that of tool replayers. The interface for analysis, however, is not yet defined. Once this is done, we will be able to implement the table view and the graphical timeline view as plugins in order to validate our API.

By providing these three extension possibilities (cf. Figure 12: scripts, replayers and visualisation plugins), we hope to maximise the variety of research questions Tatiana can be applied to while keeping to a reasonably simple framework that solves researchers problems with minimal overhead.

7 Conclusions and Future work

In this paper, we presented the tool Tatiana (Trace Analysis Tool for Interaction Analysts), intended for researchers of CSCL and CSCW situations who wish to analyse the process of human interaction. We noted that Tatiana was designed in direct response to a set of analytical needs and specific research questions in regards to corpora of computer-mediated face-to-face interactions. We described the main features of this tool with regard to data management, synchronisation, visualisation and analysis, along with the ways in which Tatiana can be extended to meet needs that are not currently catered for. In comparing these features with features of similar tools and in showing how these features could be used to answer specific research questions, we showed the innovations brought by Tatiana and the justification for these innovations.

Tatiana responds to the inherently iterative and diverse nature of CSCL and CSCW analysis by providing flexible data transformations and visualisations and by providing several points at which Tatiana can be extended in order to meet new needs. The difficulty in recreating what participants experienced during corpus collection on the basis of the recorded data is answered through multiple synchronised visualisations of the data, particularly through the integrated use of external tool replayers. Finally, through the ability to save and share analyses, Tatiana enables researchers in the human and social sciences to work as a team and to integrate and compare their analyses.

---

[9] Application Programming Interface

Our future work will involve simultaneously bettering our understanding of researcher's analysis methodologies and further developing Tatiana as an environment for managing replayables (and possibly other analysis artifacts). Tatiana currently enables the automated creation of replayables through filters and analysis. What remains is to make this framework more coherent, to give researchers the possibility to save and share replayables and to provide additional views for the creation and iterative improvement of replayables (such as that suggested by the mockup in Figure 10).

We plan to develop Tatiana as a tool for managing the variety of artifacts researchers might wish to create during their analysis. We hope that doing so will enable researchers to make better sense of their corpora and to be easily able to share the meaning-making they have created with other researchers. We also hope that this will enable us to gain more insight into the process of socio-cognitive interaction analysis, making it easier to evaluate and share methodologies throughout the CSCL community.

Acknowledgements

Jdjdjd

References

Avouris N., Fiotakis G., Kahrimanis G., Margaritis M., & Komis V. (2007). Beyond logging of fingertip actions: analysis of collaborative learning using multiple sources of data. *Journal of Interactive Learning Research JILR*. vol. 18(2), pp.231-250.

Becker R. A., & Cleveland W. S. (1987). Brushing scatterplots. *Technometrics*, 29:127-142.

Betbeder, M.-L., Ciekanski, M., Greffier, F., Reffay, C. & Chanier, T (forthcoming, 2008). Interactions. Interactions multimodales synchrones issues de formations en ligne : problématiques, méthodologie et analyses [Multimodal synchronous interactions issued from on-line learning : motivation of approach, methodology and analyses]. In Basque, J. & Reffay, C. (dirs), *numéro spécial EPAL (échanger pour apprendre en ligne)*, *Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF),* vol. 15.

Cox, Richard (2007). Technology-enhanced research: educational ICT systems as research instruments. *Technology, Pedagogy and Education*, 16 (3), 337-356.

Corbel, A., Girardot, J.J., & Jaillon, P. (2002). DREW: A Dialogical Reasoning Web Tool*, ICTE2002, International Conference on ICT's in Education. Badajoz*, Spain.

Corbel, A., Jaillon, P., Serpaggi, X., Baker, M., Quignard, M., Lund, K., & Séjourné, A., (2003). DREW : Un outil Internet pour créer des situations d'apprentissage coopérant [DREW : An internet tool for creating cooperative learning situations]. In Desmoulins, Marquet & Bouhineau (Eds.) EIAH2003 *Environnements Informatiques pour l'Apprentissage Humain, Actes de la conférence EIAH* 2003, Strasbourg, 15-17 avril 2003, Paris : INRP, p. 109-113.

Corbel, A, Girardot, J.J., & Lund K. (2006). A method for capitalizing upon and synthesizing analyses of human interactions, In (eds) W. van Diggelen & V. Scarano, Workshop proceedings Exploring the potentials of networked-computing support for face-to-face collaborative learning. *EC-TEL 2006 First European Conference on Technology Enhanced Learning*, Crete, pp. 38-4

De Chiara, R., Di Matteo, A., Manno, I., Scarano, V. (2007). CoFFEE: Cooperative Face2Face Educational Environment. *Proceedings of the 3rd International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2007)*, New York, USA.

De Wever, B., Schellens, T., Valcke, M. & Van Keer, H. (2006). Content analysis schemes to analyse transcripts of online asynchronous discussion groups: A review. *Computers & Education*, 46(1) 6-28.

Georgeon, O, Mille, A, & Bellet, B.T. (2006). ABSTRACT: un outil et une méthodologie pour analyser une activité humaine médiée par un artefact technique complexe. *Ingéniérie des Connaissances. Semaine de la connaissance.* Rémi Lehn, Mounira Harzallah, Nathalie Aussenac-Gilles, Jean Charlet ed. Nantes, France.

Georgeon, O., Mathern, B., Mille A., Bellet, T., Bonnard, A., Henning & N., Trémaux, J.-M. (2007). *ABSTRACT Analysis of Behavior and Situation for menTal Representation Assessment and Cognitive acTivity modelling*, Retrieved April 10, 2008 from http://liris.cnrs.fr/abstract/abstract.html.

Fiotakis, G., Fidas, C. & Avouris, N (2007). Comparative usability evaluation of web systems through ActivityLens, *Proc. PCI 2007*, Patras, Greece

Goodman, B. A., Drury, J. Gaimari, R. D., Kurland, L., & Zarrella, J (2006) Applying User Models to Improve

Team Decision Making. Retrieved April 10, 2008 from http://mitre.org/work/tech_papers/tech_papers_07/06_1351/

Greenhalgh, C., French, A., Humble, J. and Tennent, P. (2007) Engineering a replay application based on RDF and OWL, *Online Proceedings of the UK e-Science All Hands Meeting 2007*, September 10-13, Nottingham: NeSC/JISC.

Harrer, A. Zeini, S. Kahrimanis, G. Avouris, N. Marcos, J. A. Martinez-Mones, A. Meier, A. Rummel, & N. Spada, H (2007) Towards a Flexible Model for Computer-based Analysis and Visualisation of Collaborative Learning Activities, *Proc. CSCL 2007*, New Jersey, USA

Kahrimanis G., Papasalouros A., Avouris N., & Retalis S. (2006), A Model for Interoperability in Computer-Supported Collaborative Learning. *Proc. ICALT 2006 - The 6th IEEE International Conference on Advanced Learning Technologies*. IEEE Publ., July 5-7, 2006 – Kerkrade , Netherlands, p. 51-55.

Kollberg P. (1996). S-notation as a tool for analysing the episodic structure of revisions. *European writing conferences*, Barcelona.

LEAD (2006). *Problem solving through face to face networked interaction in the classroom*, Retreived April 10, 2008 from http://www.lead2learning.org

Lotan-Kochan E. (2006). Analysing Graphic-Based Electronic Discussions: Evaluation of Students' Activity on Digalo. *EC-TEL 2006*, Crete, Greece: 652-659.

Lund, K., Rossetti, C., & Metz, S. (2007). Do internal factors of cooperation influence computer-mediated distance activity? *Proceedings of the international conference CSCL 2007*, July 16-21, New Brunswick, New Jersey, USA.

Lund, K., van Diggelen, W., Dyke, G., Overdijk, M., Girardot, J.J., & Corbel, A (forthcoming, 2008). A researcher perspective on the analysis and presentation of interaction log files from CSCL situations within the LEAD project. *ICLS 2008*, Utrecht, The Netherlands.

Morrison A., Tennent P., & Chalmers M. (2006). Coordinated Visualisation of Video and System Log Data. Proceedings of 4th International Conference on Coordinated & Multiple Views in Exploratory Visualisation. London, 91-102.

Prudhomme, G., Pourroy, F., & Lund, K. (2007). An empirical study of engineering knowledge dynamics in a design situation. *Journal of Design Research, 3,* 333-358.

Reffay, C, Chanier, T., Noras, M. & Betbeder, M.-L. (forthcoming 2008). Contribution à la structuration de corpus d'apprentissage pur un meilleur partage en recherche [Contribution to the structuring of learning corpora for better sharing of research]. In Basque, J. & Reffay, C. (dirs), *numéro spécial EPAL (échanger pour apprendre en ligne)*, *Sciences et* Technologies *de l'Information et de la Communication pour l'Education et la Formation (STICEF),* vol. 15, forthcoming

W3C (2007). *XQuery 1.0: An XML Query Language*, Retrieved April 10, 2008 from http://www.w3.org/TR/xquery/

BERTELSEN O.W., BODKER S. (2003). Activity Theory. In J. M Carroll (Ed.), *HCI Models, Theories and Frameworks*, San Francisco, CA, USA: Morgan Kaufmann.

DYKE G., LUND K. (2007). Implications d'un modèle de coopération pour la conception d'outils collaboratifs. *Colloque EPAL : Echanger pour apprendre en ligne*. 6-8 Juin, Grenoble, France. (à paraître)

DYKE G., GIRARDOT J.-J., LUND K., CORBEL A. (2007). Analysing Face to Face Computer-mediated Interactions. *EARLI '07*. Budapest, Hongrie. (à paraître)