Share data treatment and analysis processes inTechnology enhanced learning

Denis Bouhineau, Sébastien Lalle, Vanda Luengo, Nadine Mandran, Michael Ortega, Claire Wajeman

To cite this version:

Denis Bouhineau, Sébastien Lalle, Vanda Luengo, Nadine Mandran, Michael Ortega, et al.. Share data treatment and analysis processes inTechnology enhanced learning. Workshop Data Analysis and Interpretation for Learning Environments, 2013, Autrans, France. hal-00948796

HAL Id: hal-00948796

https://hal.inria.fr/hal-00948796

Submitted on 16 Mar 2014

Share data treatments and analysis processes in Technology enhanced learning.

Denis Bouhineau
Université Joseph Fourier
Grenoble
Denis.Bouhineau@imag.fr

Sébastien Lallé
Université Joseph Fourier
Grenoble
Sebastien.Lalle@imag.fr

Vanda Luengo
Université Joseph Fourier
Grenoble
Vanda.Luengo@imag.fr

Nadine Mandran
CNRS
Grenoble
Nadine.Mandran@imag.fr

Michael Ortega
CNRS
Grenoble
Michael.Ortega@imag.fr

Claire Wajeman
Université Joseph Fourier
Grenoble
Claire.Wajeman@imag.fr

ABSTRACT

In the context of our research team (multidisciplinary with numerous and various TEL systems), we have been working during the last three years on the design and implementation of an open platform to collect, save and share experimental data drawn from the interaction with TEL systems, which could build, save and share analysis processes executed on these data. From our point of view both data and analysis processes are worth to be stored and shared, and moreover have to be joined in a unique repository to get the whole picture. This communication presents the analysis processes part of the project.

Sharing analysis processes, i.e. the whole complex process, is rather unusual, whereas contemporary platforms or software already propose generic algorithms to work on data (for instance with a statistical point of view or a data mining point of view). Hence, we attempt to model the main concepts of global treatments for experimental data analysis in order to collect, execute, save and then share them in a platform, dedicated to TEL Systems. The execution part is the most difficult and constraining part of our work. This needs to be implemented with a complex architecture. An important part of the communication is so devoted to the description of the architecture, and to the link between the global point of view of the whole process and the local point of view of elementary or specific algorithms used during the process. A short, but realistic, example of application of our platform is given, with the definition of a global process and the definition of an elementary algorithm used in the global process. The process is executed on real data leading to a graphical display of results, which are then briefly analyzed.

1. INTRODUCTION

The domain of educational data mining and learning analytics has grown in the last decade thanks to the increase accessibility of educational data. Both research domains propose models (like student models), algorithms, methods (like experimental methods) and tools (like visualization tools or data mining tools) built for the treatment and analysis of the data produced during teaching and learning interactions.

In these domains the algorithms have to be reliable. They have to be reusable for other sets of data or experimental situations and more than one set of data could be necessary to validate the algorithm (cross validation). Also the experimental methods have to be validated and replicable. In contrast, sometimes, a given kind of treatment or visualisation can be more justified or more pertinent for a particular situation constrained by the nature of the data.

Besides, the necessity to build tools, methods and/or processes adapted to the educational specificities is recognised [1]. For instance, the generic systems like Weka (for data mining task) or R (for statistical task) are not easy to use or not enough specific for the TEL domain.

In conclusion one way to improve our research domain would be to share not only data but also treatments and analysis processes designed to process and analyse our TEL data. However, like explained in the next section, there currently exist platforms to share specific TEL data on one hand, generic platforms to process this data on the other hand, but there is no TEL open platform which associates both (data and analysis processes specific to TEL).

Thus, our objective is to mutualise experimental research in TEL Systems, i.e. to mutualise data but also treatments and analysis processes in the domain of TEL Systems. In this paper we will focus on the second part, i.e. build and mutualise treatments and analysis processes.

2. FRAMEWORK AND RELATED WORKS

The research in TEL is often multidisciplinary (science education, psychology, computer science, statistics, etc...) and the treatment and analysis methods are numerous (symbolic, numeric,..). In the context of our research team, we collect data in classrooms and use it to conduct some research projects. For example, we have collected data in secondary schools using microworlds (TPELEC[1] or Aplusix [2]). These data are used to analyse the learner behaviour [3], to test a diagnostic algorithm [4] or to test a teacher monitoring system [5].

Besides, because these experimental researches are time-consuming and difficult to organise, we would like to capitalise the derived treatments and analysis processes, for reusing, improving and validating them. One example is the research project Listen [6], which starts in 1996. Since one decade this project collects data. It uses it to improve the proposed system and to understand the student behaviour. The data structures have evolved and the functionalities to browse them also. However their analysis are domain-oriented (reading English). A platform, for browsing and mining temporal and educational data, has also been designed, but as there is no constraint on the format of data, analysis processes can be hardly generalized and reused in other domains or data.

[1] tpelec.imag.fr/index.html

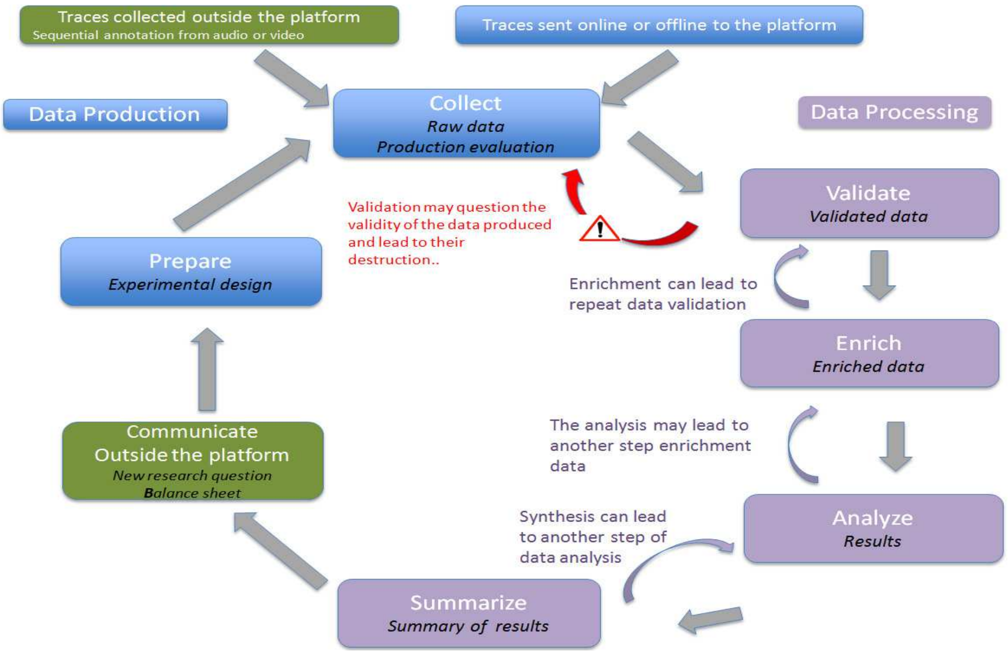
Traces collected outside the platform
Sequential annotation from audio or video
Traces sent online or offline to the platform
Data Production
Collect
Raw data
Production evaluation
Data Processing
Validate
Validated data
Validation may question the validity of the data produced and lead to their destruction..
Prepare
Experimental design
Enrichment can lead to repeat data validation
Enrich
Enriched data
The analysis may lead to another step enrichment data
Communicate
Outside the platform
New research question
Balance sheet
Analyze
Results
Synthesis can lead to another step of data analysis
Summarize
Summary of results

Figure 1. Data Lifecycle

In our case we are interested, for example, in the validation of one hypothesis related to the nature of student knowledge across several domains or in the validation of one kind of learner behaviour monitoring (find contradictions in students' behaviour during problem solving activities), for different learning domains. Thus, we are interested to reuse all or a part of one data processes (i.e. data treatment and analysis process) in order to improve and make more efficient our experimental researches in TEL.

There are some projects offering to share data in TEL domains, like the DataShop project [7], in which the analysis is mainly to find and test models about the learner behaviour or learner knowledge in the context of the interaction with tutors belonging to one learning paradigm (cognitive tutors). In this platform the authors propose one learning curve constructing algorithm (learning curves allow measuring the students' acquisition of skills over time), which works with only one kind of formatted data.

Another project, called Mulce (MUltimodal contextualized Learner Corpus Exchange) [8], shares experimental corpuses. They defined the Learning & Teaching Corpus (LETEC) as a package containing data issued from a online course, contextual information, and metadata necessary to make these data visible, shareable and reusable. Their objective is to better describe the data collect context. In Mulce, analysis tools are described but externalised.

In these two platforms, most of the analysis algorithms or visualisation tools are not included in the platform. Both authors' platforms emphasize the necessity of making connection with analysis tools.

Concerning algorithms, we find implementations in generic platforms (like Weka or R) or implementations of TEL analysis algorithms made available by researchers (Bayes Net Toolbox for Student Modeling [9], EDM tools[2], a set of data mining algorithms used in ill-defined domains[3] or Tatiana[4], a set of visualisation algorithms to analyse collaborative data). In a review of specific educational data mining [1], statistic and visualization tools are presented. In all cases, some manuals steps are necessary to integrate both data and processes for allowing analysis.

Our objective is to propose an open platform that would be interoperable, flexible and where each tools used, during the data treatment and analysis process, would be reusable. In next sections we describe the foundations of the platform and the architecture used to build and capitalise processes. Before conclusion, an example is detailed.

3. DATA TREATMENTS AND ANALYSIS PROCESSES DESCRIPTION

The treatment and analysis of activity traces, from a global point of view, cannot be reduced to the use of an unique 'advanced' algorithm (SPC, Bayes, ...). It is a complex process where humans are involved and which can include complex algorithms but also, and quite often, small and simple algorithms such as those which clean or anonymize data. These simple algorithms are usually introduced at the beginning of the process. At the end of the process, we can also observe that there are very often important efforts done to give a graphical presentation, in order to make human interpretation of the results easier. We want to stress here that it is the whole process that we aim to describe, manage, save and share.

In this context we defined a data lifecycle to take into account all possible steps during the treatment and analysis of activity traces.

[2] http://users.wpi.edu/~rsbaker/edmtools.html

[3] http://www.philippe-fournier-viger.com/spmf/index.php

[4] http://code.google.com/p/tatiana/

The whole cycle is described with seven phases: preparation, collection, validation, enrichment, analysis, summarization and communication (figure 1). These phases are organized in a sequential order, but going back to a previous phase is often necessary. However not all phases are mandatory. For instance the enrichment phase could be optional. Each product (transformed data, results, processes ...) of each phase has to be capitalized and could be reused.

In the following, we focus on the data processing step, which is made up of the 4 phases: validation, enrichment, analysis and summarization.

Data processing starts with the validation phase (called data preprocessing in [1]), and is executed on the raw data obtained from the collection phase. This phase will necessarily generate a new data set, since the raw data must never be directly altered. The scientific analysis will be performed from these validated data. The validation phase includes both checking and correction of the raw data. The validation phase of the raw data is essential to guarantee the meaning given to results and to interpretations [10]. It is frequently composed of small actions necessary to check the variables names, format or value, if data are missing, to identify and correct duplications, remove noise and unlikely data... It also includes the checking of consistency between collected data and what has been planned a priori and the accuracy of values of variables, etc. There are various strategies for performing these checking, like listing and comparing values, cross-tabulation, study of distribution of variables or verification using graphical representations. It can be specific strategies that are defined by the experimental design. This checking phase is usually followed by the appropriate data correction or data improvement operations. In all cases all the operations performed and the results have to be reported in the experimental design. If troubles are identified and no remediation is possible, the researcher must consider how to upgrade the research questions and the experimental design in order to ensure consistency with the available data.

The enrichment phase creates new data for analysis, and more specifically new variables. For example patterns of students' actions are created in this step. Various types of variables may be created using different techniques. For instance recoding of data by grouping modalities creates data with the same data type than the initial data and the data format is not modified. On the contrary, quantitative data that are aggregated from sequential data would provide new data type and new format.

The analysis phase leads to results that support scientific interpretation. It consists in analyzing data, therefore in designing analysis processes, in order to interpret results according to research questions. The processes can be complex and composed of a chain of algorithms (i.e. algorithms executed in a sequential way). For example, in the 1980s, [11] proposed to analyze data with chains of treatments called "*Filières*". Algorithms of this type have been implemented in statistical software like SPAD, SAS, TANAGRA. A "*Filière*" is a workflow of algorithms: a statistical treatment may produce new data; from these data a new complementary treatment will be made. Algorithms can perform filtering, apply logical rules, compute descriptive statistics, deliver inferential statistics, produce classification, model, give graphical representations.

The summarization phase allows to extract and to organize the relevant results in order to discuss the research questions formalized *a priori* or to raise new research questions or hypotheses. This phase may include the construction of summary tables, of clear and intelligible graphics… The summarization process has to be described as well as all other processes.

During the validation phase, the enrichment phase or the analysis phase, regarding the results of processed data obtained the researchers can be lead to add other steps in order to explore the data. Thus, the organization of a processing chain can be elaborated in an empirical manner depending on on-oing results. Furthermore the algorithms used to enrich and analyze the data may follow different analysis paths, i.e. build several chains explored in parallel. Some of these paths are complementary. Some will be abandoned. A conclusive path is the one that gives interesting results about the research question. In each conclusive path, relevant results are kept and reshaped to be used in the summarization phase. Even in a "non-empirical" manner, research questions are generally addressed in several ways, and the exploration of various paths can be planned from the very beginning of the experimental process. As the researcher can freely create, explore and merge several paths during the analysis, then the overall process can be viewed as a directed graph.

Finally some kind of tools could be involved in several phases. For example, visualization tools such as histograms or methods of visual data mining will be implemented for each treatment phases (validation, enrichment, analysis and summary).

The seven phases of the complete described and validated process can be reused on other data or by other researchers. The process could be understood like a cycle where the results of the summarize phase could lead to a new analysis of the same kind of data, which is validated in the platform, or could lead to a new experimental process with a new research question.

In the following, we will distinguish two levels in our modeling: first, a global level concerning the whole process and, secondly, a details level associated with each step of the overall process where each algorithm, each treatment or tool will be described. Our objective is to capitalize both: the details steps and the whole process. Thus it is necessary to propose an architecture which allows the description and the archiving for the global and the details level. Because the process design could be empirical, i.e. we test more than one way to address the problem, the platform has to be flexible.

4. UNDERTRACKS PLATFORM: COLLECT AND EXECUTE PROCESSES

Each level, the global level and the details level, requires a dedicated implementation for collecting, editing, managing and executing the data processes. We call the concrete part of our work: the "UnderTracks" platform.

First, the global level (the design of the analysis process) needs a specific application to build a process (represented as a graph) and to execute it. This application assists the user in defining the associations between either: the raw data (stored in database's tables) and the algorithms involved in the process, or two algorithms in sequence, meaning that the second one uses the results of the first one. These associations allow a flexible use of algorithms. As our goal is to share and reuse both the algorithms and the processes (processes can be considered as sequences of algorithms), we have to guarantee that associations between data tables and algorithms can be expressed in a generic way, i.e. do not depend on a particular experiment or data. For example, an algorithm designed for sorting elements in a data table should be reusable for sorting students as well as actions, exercises…

Finally, concerning the details level, all algorithms must be developed in a way so that they can be really executed. All these notions are described in more details in the following sections.

4.1 Architecture Overview

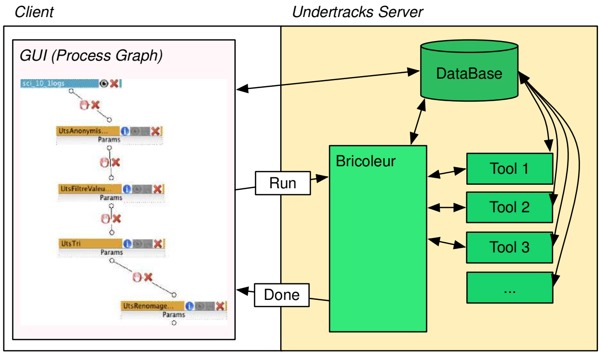

Figure 2. Architecture of the Process part of UnderTracks (collect and execute global processes)

The Process part of UnderTracks (for collecting and executing processes) is composed of four main entities (see Figure 2):
- A Graphical User Interface (GUI). This entity allows user to create, load, call the process runner, and then see and recover results. A process is represented by a workflow, which can be considered as a directed acyclic graph. It quite often starts with data from UnderTracks databases (raw data from an experimental study) and leads through sequenced operations to final results.
- A process runner: called "*Bricoleur*". The process description, which is a graph created with the GUI, is sent to the Bricoleur for execution. This program runs through the graph and delivers data to the different algorithms that compose the process.
- Bricks. A brick is an entity of the process's graph. It can be data or algorithm.
-A DataBase. This entity temporally stores data for each bricks, and it's used for data travelling along the graph. It also stores results and processes.

Next sections present bricks and the GUI in details.

4.2 Bricks

Bricks are nodes of the process's workflow. There are three kinds of bricks: data bricks (DB), also called Bricoleur's "raw material", and tool bricks (TB) corresponding to algorithms. Lastly, visualization bricks (VB) aim to present results in a graphical way. DB contains data coming from UnderTracks database, or from previous processes. While creating a process, user usually starts from a DB and plugs TB for adding treatments to it. The TB thus modifies data from the DB. Other DB can be plugged along the workflow, like in Figure 3, until VB. DBs and VBs are natural boundaries of the graph (DBs as sources, VBs as sinks). TBs are natural internal nodes of the graph. Some TB can also be sinks of the graph.

Once the process's workflow is created with the GUI, it is sent to the *Bricoleur* program, which is in charge of running it. This consists of travelling data from DBs through TBs and VBs. The Bricoleur stores each step's results of the described process into the database.

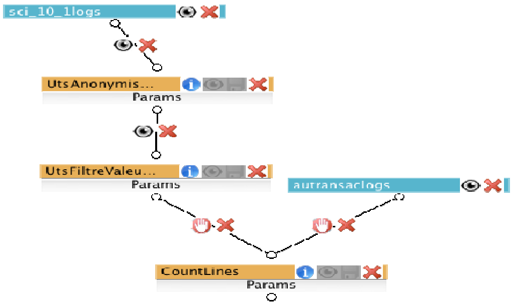

Figure 3 Example of a process's workflow. Data bricks (DB) are in blue and tool bricks (TB) in yellow

4.2.1 Tool Bricks Description

TBs can be sequenced into the GUI for building a process. They are simple or complex algorithms, encapsulated in a common description (specific to UnderTracks) for allowing their use and their sequential execution by the *Bricoleur*. Entry data of TBs can be divided into two categories:
- The table-like data on which the algorithm acts. This data category can be considered as the flow, which is modified and which travels along the workflow. This category of data can come from a DB, directly connected upstream of the TB, or from another TB's result, also connected upstream of the current TB.
- The algorithm tuning values. These are the parameters that describe the algorithm *way of acting* on table-like data. For instance, a tool that applies a threshold can have one parameter: the threshold value.

By using these two data descriptions, the *Bricoleur* is able to execute the TB's algorithm and thus able to compute the treatment result. The latter is re-injected into the following bricks and temporally stored into the database. Two kinds of results can be obtained:
- Table-like result. Here data is similar to the first kind of entry data, for allowing the re-injection into the following TB. Notice that TB can either modify/extend the input tables, or create an entirely new data table.
- File result. If anticipated by the developer, the TB algorithm can provide a file result. This file is not re-injected into the workflow, but can be an aside result, with its own format

VBs and TBs share the same format for the entry. For the results, VBs produces only a file result, with its own graphical format like, for instance, an HTML page with graphical representation of the results.

4.2.2 Adding Tool Bricks into UnderTracks

TBs and VBs can be developed by anyone, and added to the current UnderTracks platform. A web interface is proposed. It first asks for the TB's or VB's code and description (description of the entry data, of the algorithm, and of the data output). Next, both code and description consistency is checked. If no errors are detected, the code is encapsulated into a specific descriptor for enabling communication between the new TB, VB, the Bricoleur and the database. Finally, both, encapsulated code and description are added into the UnderTracks platform for being used into the GUI.

4.3 Building the process

The GUI program is executed on the client side, i.e. on the user's computer. It allows him to construct the process workflow by drag-and-dropping bricks and connecting them (see Figure 3). The connection consists in matching the data coming out from upstream TBs or DBs with the entry of the current TB or VB. Outcoming and incoming data descriptions are table-like data. Then, connecting two bricks consists in linking outcoming and incoming data columns. For instance, Fig 4 shows the connection between two TBs. Four columns of the upstream brick's result are connected to the five entry columns of the following TB.

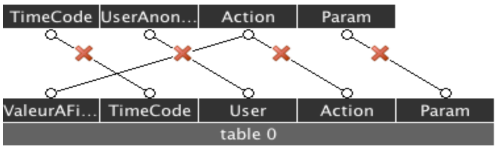

Figure 4 Example of connection between two bricks

4.4 Technologies

UnderTracks processes part uses different programming languages:

- GUI has to be launch on the user's computer. It's then developed in JAVA, in order to been *portable*, i.e. usable on all Operating System.
- The "Bricoleur" program and the tools are placed into the UnderTracks server, and so developed in C++ for being efficient in terms of computing performances.
- Communication between the GUI and the Bricoleur is not direct. Php scripts are used in between, and allow communication via the web, for security and log aspects.
- PostGreSQL is used for the database. This technology seems more adapted to our needs (security, data consistency, strong support of SQL functionalities, …)
- For now, algorithms in the TBs and VBs can be developed in C++ or JAVA, according to developer needs.

```
public class UTObjectIn {
    public  int     nbTables;
    public  String  [][][]table;
    public  int     nbParams;
    public  String  []params;}

public class UTObjectOut {
    public  String   [][]table;
    public  String  file;}

public static UTObjectOut tool(UTObjectIn uIn) {
    UTObjectOut uOut = new UTObjectOut();

     [ … ] /* tools code */

    return uOut;}
```

Figure 5 Tools profile in JAVA

4.5 Example

4.5.1 *Add one brick*

The implementation of the details level is done with an emphasis on Java and C++. Each tool or algorithm must be written in one of these languages with a standard header for the definition of the main function where parameters meets table, see Fig. 5.

The main function of tools has a standard and simple signature, using input and output objects: respectively UTObjectIn and UTObjectOut (Fig 5). The UTObjectIn class contains the incoming data tables represented as a three-dimensional array of string, and a list of parameters. The three-dimensional array allows to store a list of incoming two dimensional data tables (the incoming flow), while the parameters list is for storing the tuning parameters of the algorithm. The UTObjectOut contains one output data table that can be considered as the outcoming flow, and a file if needed.

Fig. 6 shows the code for a renaming tool which parses a table and changes its third column. The values are processed using a list of keywords, given in the parameter uIn.params[1]. In that case, the renaming process replace the found keyword with the value of a given parameter uIn.params[0].

```
public static UTObjectOut rename(UTObjectIn uIn) {
UTObjectOut uOut = new UTObjectOut();
ArrayList<Integer> bufOut=new ArrayList<Integer>();
HashSet<String> keywords = new HashSet<String>();
String parametres []= uIn.params[1].split(" ");
for(int i=0;i<parametres.length;i++) {
  keywords.add(parametres[i]);}
uOut.table=new String[uIn.table[0].length][uIn.table[0][0
].length];
for (int i =0;i<uIn.table[0].length;i++) {
  for(int j=0;j<uIn.table[0][0].length;j++) {
    uOut.table[i][j] = uIn.table[0][i][j];}
  if (keywords.contains(uIn.table[0][i][2])) {
    uOut.table[i][2] = uIn.params[0];}}
return uOut;}
```

Figure 6 JAVA code of the renaming tool

The list of keywords to be renamed, the name of the replacement word, and the incoming table are given by the user while constructing the process with the GUI. This tool is used in the process presented in next section. It is used for grouping some data (actions), semantically roughly equivalent.

4.5.2 *Build a Process*

The following process is based on simple activity traces with 4 fields: time, user, action, and action parameters.

The analysis process aims to extract student's behavior from data. To do so, we build the sequence of actions for each student, then we extract couples of successive actions depending on what the researcher want to study. For instance some students consult documents before doing an action whereas others use the trial and error approach.

In more details, the overall process (see Fig. 7, on the left) presents three distinct phases:

– The validation, with bricks for cleaning superfluous spaces from the data, for filtering the data with observed actions, and for sorting the data by user and date-time,

– The enrichment, with bricks for adding data (construct couples of successive actions for one user) as well as bricks with enrichment tools, which regroup and rename the couples of successive actions according to some more semantic information, (consult document or consult result then make some action),

– The results display, with brick which provides some facilities to explore the data (see Fig. 7, right)

The algorithms are quite simple, each code fits on one page (see Fig 6). These tools, all but one, have an input table and parameters, and provide as output another table. The processes main actions are: cleaning, sorting, filtering (the parameter is the list of actions to be removed), and renaming (one parameter is the list of actions to be renamed, the other is the new name to be given, see fig. 6 for the code). The last tool, for the graphical display, has an input table but no output table, its only output is an html page, produced by the tool, with a graphical representation of the data. All these tools are not specific to the definition of this current process and can be used for other analysis.

The result of the process is a graphical representation of the student's behavior (on right side of Fig. 7) which allows to identify different kind of learners: learners which loop lengthily on control (each control followed by another control is

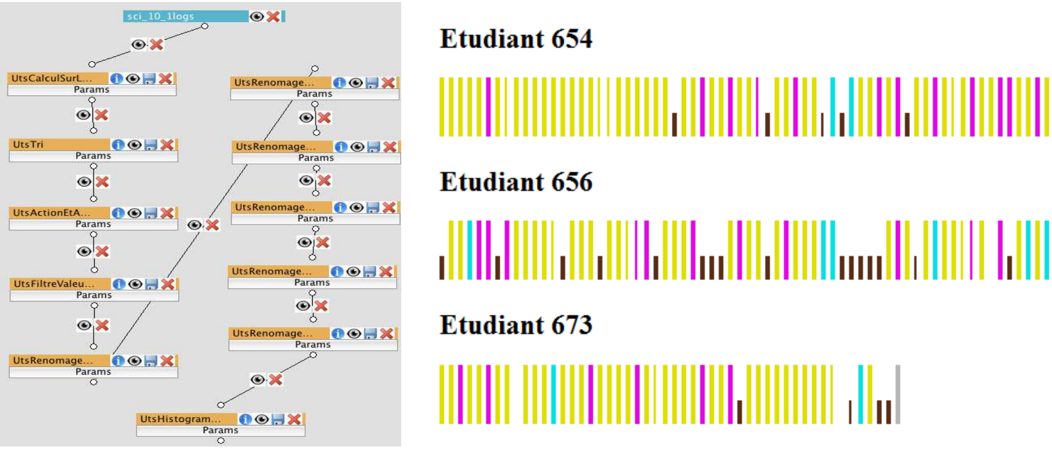

Figure 7. Execution of a process in order to represent graphically a sequence of action. On left, the application used to define the control graph and in charge of the execution of the process. On right the graphical result for the last step.

represented with a bright big yellow bar; consecutives yellow bars mean multiple control, …), learner with unique control during action phases, or reading phases (each control followed by an action is represented with a dark small bar, or a red big bar for readings; two consecutive small dark bars mean one, and only one, control then one or more actions, done twice, id. for readings), learner with small control loop leading to readings, … The researcher could investigate this result and find some more kind of learners. The researcher could continue and tune the parameters of the process to obtain a better vision of the learners' behaviors. Depending on the results, he can also add new bricks in the process, for instance to extract recurrent or unusual students' behaviors, classify students' behaviors, do pairwise comparison between couples of actions…

5. DISCUSSION

Research in TEL is a multidisciplinary area including several systems (web, adaptive hypermedia, tutoring systems, microworlds, simulators, LMS,...), and rooted in several learning paradigms and in several interactions paradigms, as well as in several computing paradigms. For example, the field of Educational Data Mining uses data mining and several computational paradigms: decision tree construction, rule induction, Bayesian learning, logic programming, statistical algorithms ... Processing and analysis of TEL data need specific requirements that are not present in other domains. So, it is necessary to specialize and adapt works in the TEL domain in order to obtain better results [1].

The flexibility and the reusability are the two key features we are looking for. The reuse of data, of algorithms and of analysis processes should improve the efficiency, the reliability and the replicability of our research in the TEL domain. Still, we have to prove that the data collected from several systems, corresponding to various learning paradigms and the algorithms used in our analysis processes, and the analysis processes themselves can be reused in different researches.

To conclude, we propose an architecture and a platform (Undertracks) in order to improve experimental researches in the TEL area. The usage of the platform is shown here with a short real example. This platform aims to (1) obtain, improve and share data and analysis processes for TEL, (2) propose specific processes for TEL systems, able to combine several computer science paradigms and (3) prepare a structure able to evaluate data, algorithms and processes in our domain.

6. REFERENCES

[1] Romero C. and Ventura S. Educational Data Mining: A Survey from 1995 to 2005. *Expert Systems with Applications*, 33(1), 135-146, 2007.

[2] Bouhineau D., Nicaud J.-F., Chaachoua H., Bittar M. , and Bronner A. (2005). Two Years Of Use Of The Aplusix System. In *8th IFIP WCCE* 2005.

[3] Croset M-C., Trgalova J., Nicaud J-F.(2008) Student's Algebraic Knowledge Modelling: Algebraic Context as Cause of Student's Action. Ib Studies in Computational Intelligence,Springer Verlag, pp. 75-98.

[4] Michelet S., Luengo V., Adam J.M., Mandran N. (2010) Experimentation and results for calibrating automatic diagnosis belief linked to problem solving modalities: a case study in electricity. In *ECTEL 2010.*

[5] Guéraud V. Adam J.M., Lejeune A., Dubois M, Mandran N. (2009) Teachers need support too: FORMID-Observer, a Flexible Environment for Supervising Simulation-Based Learning Situations. In WS ISEE*,* AIED 2009.

[6] Mostow J., Beck J. (2006). Some useful tactics to modify, map and mine data from intelligent tutoring systems. In *Special Issue on Educational applications of Journal Natural Language Engineering*, v2, pp 195-208.

[7] Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J. (2010). A Data Repository for the EDM community: In *Handbook of Educational Data Mining*. Boca Raton, FL: CRC Press.

[8] Reffay C., Betbeder M. and Chanier T. (2012). Multimodal learning and teaching corpora exchange: lessons learned in five years by the Mulce project. In *Int. Journal Technology Enhanced Learning*, Vol. 4, Nos. 1/2, pp.11–30.

[9] Chang, K., Beck, J., Mostow, J., & Corbett, A. (2006, June 26-30). A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, Jhongli, Taiwan, 104-113.

[10] L. Berti-Equille (2006) Modelling and measuring data quality for qualityawareness in data mining. *Quality Measures in Data Mining, Studies in Computational Intelligence*, F. Guillet and H. Hamilton (eds), Springer.

[11] Lebart L., Morineau A. & Fénelon J.P., (1981). *Traitement des données statistiques*. Dunod. Paris.