An Ontology-Based Approach for Sharing and Analyzing Learning Trace Corpora

Hajer Chebil, Jean-Jacques Girardot, Christophe Courtin

An ontology-based approach for sharing and analyzing learning trace corpora

Hajer Chebil, Jean-Jacques Girardot
Laboratory for Information Science and Technology
Henri Fayol Institute, Ecole des Mines de Saint-Étienne
Saint-Etienne, France
chebil@emse.fr, girardot@emse.fr

Christophe Courtin
Syscom research team
Université de Savoie
Le Bourget du Lac, France
christophe.courtin@univ-savoie.fr

***Abstract*—Researchers wishing to analyze Technology Enhanced Learning (TEL) situations usually collect interaction traces produced by TEL environments. This paper addresses the issue of sharing, between researchers using TEL environments, of contextualized interaction trace corpora and analysis tools of these corpora. We present a new ontology-based approach called the "proxy approach" to address this issue of sharing.**

Proxy model; ontology-based approach; sharing; learning interaction trace corpora; trace analysis tools; interoperability; technology enhanced learning

I. INTRODUCTION

Technology enhanced learning (TEL) environments represent a promising alternative to traditional learning methods in both face-to-face and distance learning situations. They benefit from technology advances in software developing and the Internet network. In order to analyze the efficiency of these environments, a common practice consists in collecting interaction traces to record learners' activity when using a learning tool. More generally, researchers often use interaction traces collected by TEL environments to study different research questions related to knowledge acquisition processes, pedagogical scenario designing, usability of learning tools, etc. This research study is a part of the "TEL environment customization" project[1]. Research teams of this project collect and analyze interaction traces to study different research questions such as the adequacy of a pedagogical scenario to a real learning situation, the role of the awareness tools in the use of communication tools, and the mechanisms of knowledge acquisition. Collected traces are heterogeneous because they record interactions with different learning assisting tools. They are also of different natures: numeric logs, audio/video records and human observations. We focus our interest on the needs of sharing interaction trace corpora and post-hoc analysis tools of these corpora between researchers using TEL environments within their research studies and analyzing produced interaction traces. In fact, because building an experiment in real learning conditions may be complex and time-consuming (number of actors, period, technical devices, etc.), it can be interesting for researchers to access shared data related to a previously performed experiment [5]. The availability of experiment data and traces allows researchers to access original data to reproduce published analysis results allowing confirming, refuting or enriching them [2]. Sharing analysis tools between different research teams, interested in analyzing interaction traces produced by TEL environments, facilitates the comparison and complementarity between different analytical methods. Our work consists in proposing a model allowing to construct a platform intended for *researchers* to share contextualized learning interaction trace corpora, and analysis tools. The platform called BEATCORP (Benchmarking platform for Analysis of Trace Corpora) is a kind of a benchmarking platform allowing researchers to perform comparative and cumulative analyses on one or more of the shared corpora and to integrate produced resources to the concerned corpora. The model we propose, entitled "proxy model", is ontology-based and has three principal functions: (1) modeling the contents of a corpus, (2) operationalizing shared corpora querying, and (3) defining a set of concepts generally found in interaction traces allowing to explicitly align these ontology concepts with those of the implicit ontology (existing in the researcher's mind) defining concepts of new shared corpus traces. The research question we are going to deal with in this paper is: "How can an ontology-based model contribute to sharing interaction trace corpora and analyze them using shared analysis tools without having to impose a new interaction trace representation?". The remainder of this paper is organized in four sections: first, we present existing studies related to the issue of sharing trace corpora and analysis tools; second, we explain the objectives of our research work and enumerate the constraints we face; third, we present our ontology-based proxy approach; fourth, we give an application example; and last, we conclude and describe future work.

II. RELATED WORK

We present in this section related works having dealt with the issue of sharing interaction trace corpora and post-hoc analysis tools. We highlight two issues related to the sharing objective: sharing trace corpora and sharing post-hoc analysis tools. Some existing projects address both of these issues, others deal with either one or the other. Sharing data is generally related to a standard representation format shared by

[1] http://liris.cnrs.fr/~clu-eiah/?page_id=151

This work is part of the Research Cluster ISLE project (now ARC6), supported by the "Région Rhône-Alpes" in France.

data owners. In fact, following a standard adopted by a community makes it easier for a researcher that doesn't know a corpus to understand its contents. However, such standard doesn't exist. This absence motivated researchers to propose formats for sharing data. Furthermore, sharing analysis tools will be straightforward if they accept the same input data format. This need of sharing a set of analysis tools supports the relevance of the idea of a shared standard representation. If such a standard exists, analysis tools will be adapted to accept input data following that standard. This will increase the usability of analysis tools by facilitating the interoperability between learning environments and analysis tools. Such a consensual standard has, unfortunately, not yet been reached. We will now present some existing work which dealt with sharing trace corpora and trace analysis tools. The Multimodal Learning and teaching Corpora Exchange project (MULCE) [5] focused on the importance of sharing Learning and Teaching Corpora (LETEC) between researchers. Shared corpora contain interaction data collected during collective experimental learning situations. This work puts emphasis on the necessity of contextualizing the shared interaction data. A LETEC structure was thus proposed recommending the use of existent standard formats and decomposing a LETEC into four parts: (1) the context composed of the pedagogical scenario and the research protocol, (2) the interaction data, (3) the license, and (4) the analyses. This project led to a repository [12] for the online sharing of LETEC corpora making it possible to (1) perform queries on the repository content based on classification criteria of shared data, (2) browse corpora elements, and (3) download shared corpora. The PSLC Datashop project [3] [10] offers a web-based platform providing a repository of interaction trace datasets and a suite of tools to perform exploratory analyses and visualizations on those data. In order for the tools to be usable by researchers, interaction data have to be formatted in a particular format proposed within the project and called the "tutor message format" [15]. This format is specific to interactive learning environments such as intelligent tutoring systems (ITS). In addition to interaction data, related publications, files and presentations can be stored in a dataset. To make it possible to store datasets respecting the "tutor message format" and further benefit of the analysis tools, the project developed logging application programming interfaces for ITS developers to directly log interaction data in the proposed format. If a researcher works on an ITS with a different logging format, he has to convert his data in order to analyze them with the shared set of analysis tools. The Interaction Analysis (IA) JEIRP Kaleidoscope [4] project aimed at offering a shared library of interaction analysis tools for the Technology Enhanced Learning (TEL) community. The project rested on seven interaction analysis tools which were strongly coupled with specific learning environments. The project objective was to share these tools to analyze interactions independently of the learning environments that produced them. An interaction description format called "the common format" has then been proposed, and used to uniformly represent interaction data of different learning environments. The project members emphasize the complexity of proposing a common format. In fact, a trade-off has to be reached between: (1) a very generic format which enables representing a multitude of data but which may cause losses in certain data semantics (which can be useful in automating some processes by the analysis tools), and (2) a more specific format which allows the implementation of automatic features but restrains the multitude of data to be represented. The solution chosen in this project was a sort of trade-off by choosing to represent recurrent elements (which are either required or optional) and giving the possibility to add additional information. Although the initiative of this project was interesting and promising, it has not really led to an available library of shared interaction analysis tools. The CALICO [9] project deals with sharing and analyzing discussion forum traces collected in professionalizing training sessions which can be either distant or blended (both distant and face to face training). The extended use of discussion forums makes it possible to propose a generic representation format because of the similarity of the data handled. Relatively simple processing can convert data from one specific format to the one chosen by the project. Once the data are expressed in the proposed format, it becomes possible to share and analyze them using the tools shared by the platform developed within the project. In this project, proposing a shared format for representing discussion forums interaction data is realistic because of the specificity of the considered interactions.

We have noticed the difficulty of proposing a standard representation of interaction data, which would facilitate interaction trace corpora and analysis tool sharing, and which covers all trace modeling needs. We propose a new approach to these issues of sharing called the "proxy approach" and based on an ontology-based model. The main idea is to avoid imposing a new model to represent interaction traces. Having introduced this model in [1], we will focus in this paper on the role of the ontology in the "proxy approach" and give an example of application. Following sections present the objectives of our work, the constraints to which we are confronted, and our new approach that we claim realistic and pragmatic in the absence of a standard representation format adopted by the TEL community.

III. OBJECTIVES AND CONSTRAINTS

This section presents the objectives of our research work and enumerates the constraints that we face.

A. *Objectives*

This work aims to propose a model which underlies the development of a platform, intended for *researchers* using TEL environments, allowing to (1) share interaction trace corpora; (2) share analysis tools allowing to analyze the shared corpora; and (3) integrate and link the performed analytical work to the corresponding corpora. It should be noted that we focus our interest on post-hoc analysis of interaction trace corpora.

B. *Constraints*

We identified eight constraints we have to deal with in order to achieve our research work objectives.

1) Trace heterogeneity: Traces record interactions between a human and TEL environment during a learning situation. Data recorded in the traces (called observed

elements [6]) depend on the used learning environments. Indeed, tracing the use of a communication tool like a chat or a forum follows a model different from that of an ITS. It is also worth noting that tracing is related to observation choices related to researchers' analysis needs. Moreover, traces can sometimes be specific to particular domains when using specialized learning tools (for example ITS for learning mathematics).

2) Different natures of traces: We distinguish, because of their different natures, two types of traces. The first is a log trace (raw or enriched) which records the actions performed on a computer. The second corresponds to a trace non-directly interpretable by a machine and needs a human intervention to understand its content (it is typically an audio/video recording or manually collected observations by a human). The first trace type has an extra property that the second does not have, which is the ability to implement automatic processing of the trace to perform calculations and automatic transformations without needing human intervention. Such automatic processing needs a prior work of transcription when working on a video trace.

3) Traces with different levels of granularity: Traces collected within TEL environments have different levels of granularity. Indeed, traces aren't always collected in a level easily understandable by a human (even a computing expert). They can be of very low level and associated with low-level events related to the used hardware device (e.g. mouse clicks, eye movements' coordinates recorded by an eye-tracker, coordinates of a PHANTOM Omni movements, etc.). In our work, the traces that are likely to be analyzed using shared analysis tools have to be in an abstraction level associated to meaningful events of an instrumented learning activity (e.g. send message, reply to message, draw an object, etc.). We can thus imagine that the traces stored in a shared corpus result from previous transformations performed outside the platform on traces having lower abstraction levels.

4) Analysis tools strongly coupled to TEL environments: It is common that a research team working on a particular learning environment develops an analysis tool to meet its analysis needs. An analysis tool development is then often strongly coupled to a learning environment, and is designed to accept the trace format generated by the learning environment. However, some researches provide more generic analysis tools (e.g. visualization tools of messages exchanged in forums [9]). Defining an input format for the analysis tool remains nevertheless necessary.

5) Lack of a standard representation of interaction traces: As already mentioned, we noticed the absence of a standard model and format for the representation of learning interaction traces. The existence of such a standard would have solved the problem of sharing a common data formalism for interaction traces produced by learning environments and accepted as input of analysis tools.

6) Necessity of contextualizing interaction traces: It is a complex task for a researcher to understand and use traces resulting from an experiment in which he did not participate especially if the used learning tools are unknown to him. Hence we point out the importance of contextualizing the shared interaction traces as in [5].

7) Necessity of capitalizing upon analyses performed on the shared trace corpora: In order to allow consulting, reproducing and enriching previous analyses performed on one or more shared corpora, it is necessary to link the resources produced by an analysis work to related corpora and to keep as much information as possible allowing to contextualize, understand and reproduce results (e.g. analysis work description, interpretation model used in the classification of interaction trace events, the data on which the analysis was performed, etc.).

8) Access rights and anonymity: sharing interaction trace corpora can cause the divulgation of personal data of the participants to an experiment having consented to be tracked. It is then important to take into account aspects relative to the anonymity of the shared data and the access rights to it.

IV. THE ONTOLOGY-BASED PROXY APPROACH

The lack of a standard representation model for learning interaction traces motivated projects dealing with the issue of sharing corpora and analysis tools to propose a new trace representation format trying to make it as generic as possible. Noticing that, in addition to the first three constraints we identified in the previous section (trace heterogeneity, different natures of traces and traces with different levels of granularity), we choose to avoid imposing a new model to represent traces which will not necessarily cover all trace-modeling needs. Alternatively, we propose a new approach called "proxy approach" which is based on an ontology. This section presents the proposed ontology and its different functions.

A. Ontology functions

The proposed ontology we call "BeatcorpOnto" is designed using the open source ontology editor Protégé [14] and expressed using the ontology web language OWL [13]. BeatcorpOnto defines three different models with different objectives: the first one models a shared corpus by defining a set of concepts for its description (metadata and contents), we call it "Corpus Model"; the second, we call "Trace Concept Model", defines a set of concepts identified as frequently found in collected interaction traces and used by a researcher to explicitly map a sub-set of these concepts with concepts present in his collected traces; and finally, the third models the operational aspect allowing to query corpora and to achieve interoperability between shared corpora and analysis tools, we call it "Operational Model". The last two models represent the core of the originality of the proposed approach.

1) Corpus Model: We highlighted in the sixth constraint identified in the previous section and entitled "necessity of contextualizing interaction traces" (cf. III.B.6), the need to contextualize shared interaction traces. This corpus model tries to satisfy this constraint by defining concepts that describe a shared interaction trace corpus. A shared corpus in the BEATCORP platform is composed of physical resources and a

description defined by the "Corpus Model" concepts specified by the ontology. These concepts concern: (1) metadata allowing to give general information about the corpus and the studied learning situation; (2) describing physical resources composing the corpus; and (3) describing analytical work performed on the corpus.

Our objective being to share learning interaction trace corpora, analyze them and to share analysis results, we differentiate between two classes of corpora: initial corpus and analysis corpus.

a) Corpora types: An initial corpus results from the observation of a TEL experiment, and is constructed in the BEATCORP platform by collecting the resources used and produced during the experiment (e.g. activity description given to the participants, production of a participant, log traces collected, etc) and filling the corpus description (cf. below). An initial corpus constructed in the BEATCORP platform can contain descriptions and resources related to analysis works performed on the corpus outside the platform.

Apart from analyses that are realized by analysis tools outside of the platform and integrated to initial corpora, it is possible to use the analysis tools shared within the platform to perform analysis works on one or more shared corpora. Resources used and produced during such analyses are integrated to analysis corpora. An analysis corpus is constructed in the BEATCORP platform in order to study a particular research question interesting to a researcher or a research team. There is no constraint regarding the number of analysis works to be integrated to an analysis corpus and related to the studied research question. An analysis work can be performed on one or more shared corpora. When describing the analysis work, references to the corpora from which analyzed data are extracted should be kept. Let's note that we consider the possibility that a researcher can be interested in capitalizing on a previous analysis, or to use a resource belonging to a previously-constructed analysis corpus. An analysis corpus can then refer to either initial or analysis shared corpora.

b) Corpus description metadata: In order to document a shared corpus within the BEATCORP platform, we propose to use a set of metadata. These metadata provide interested researchers with relevant information on the corpus content but can also be used as query characteristics when querying the shared corpora database. A part of this metadata set is inspired from the Dublin Core Metadata Initiative (DCMI) standard [11], which proposes a generic metadata set to describe digital or physical resources and is a domain-independent metadata standard. To the chosen DCMI elements, we propose adding some other elements meaningful to our corpus description issue. The additional metadata elements (except "research question") are exclusively used in describing initial corpora because of their relation with the learning experiment aspect of the described corpus. The ontology defines the "corpus" concept which has two sub-classes "initial corpus" and "analysis corpus" corresponding to the corpora types to be dealt with within the BEATCORP platform. Metadata which are common for the two types of copora handled by the platform are defined as OWL properties describing the "corpus" concept. Metadata which are specific to initial corpora are defined as OWL properties describing the "learning corpus" concept. The metadata properties inspired from DCMI elements for the description of a corpus are: (1) "title": name given to the corpus chosen by the researcher and which can correspond to a project name; (2) "description": gives a summary description of the studied TEL experiment in case of initial corpus, or a description of the corpus content in case of analysis corpus; (3) "creator": entity -person or organization- that created the corpus, it is possible to have multiple creators; (4) "contributor": entity -person or organization- that contributed in the corpus (but with less importance than a creator), it is possible to have multiple contributors (5) "subject": significant keywords or expressions describing very briefly the corpus; (6) "publisher": entity responsible of the availability of the corpus; (7) "creation date": creation date of a corpus within the platform; and (8) "licence": licence defining access rights to a corpus. To these general metadata inspired from the DCMI standard, we added another important element in interaction trace corpora description which is "research question". This element is important because our platform is essentially dedicated to researchers which can be interested in checking the research question that motivated the experiment. This element is useful when describing either an initial or analysis corpus. As we mentioned above, the creation of a new analysis corpus is determined by a new research question to study. The above metadata elements describe a corpus, they can then be applied to both types of corpora. We further defined six metadata elements specific to initial corpora: (1) "collection date": corresponds to the first collection of the interaction traces during the experiment. We can imagine the case that a corpus collected ten years ago (collection date) is reconstructed this year (creation date) as an initial corpus in the BEATCORP platform. We consider that this information can be useful to a researcher because it gives him or her an idea about the actual period of the experiment; (2) "learning objective": this element informs a researcher browsing the corpus on the learning objective of the studied experiment (it is possible to have multiple learning objectives), this information can help a researcher choosing an interesting corpus to analyze; (3) "learning type": this element indicates whether the observed learning is individual or collective. In fact a researcher may be interested in studying a particular learning type; (4) "learning mode space": this element gives the researcher an idea of the learning mode with respect to space, which can be face-to-face learning or distance learning; (5) "learning mode time": this element gives an idea of the learning mode with respect to time, which can refer to synchronous activities and asynchronous activities; and (6) "learning tool": this element refers to one or more learning tools used during the experiment described by the corpus.

c) Corpus resource description: We identified different types of resources that can be shared within a corpus. We insist at this level that we do not impose specific formats for the physical resources shared in a corpus. We distinguished

five types of potential shared resources within an initial corpus: (1) pedagogical resources, a pedagogical resource can be either (1.1) teaching-oriented, which means offered by the learning environment to the learner during his learning activity (e.g. a problem statement, a course material), or (1.2) learning-oriented, which means produced by the learner (e.g. a dissertation); (2) trace resources, we distinguished two types of trace resources (cf. paragraph III.B.2); (3) analysis resources, which can be: (3.1) imported resource, a complementary resource which is needed by the researcher to analyze the interaction traces (e.g. an interpretation model to annotate interaction events), (3.2) produced resource, a resource produced by an analysis tool used by the researcher during his analysis work, such a resource makes it possible for another researcher to consult the performed work and to eventually enrich it, an analysis tool does not necessarily save the results of analytical work, this is due to the fact that sharing is not necessarily an objective of the researcher, and (3.3) interpretation resource, a resource produced by the researcher during his analysis work to interpret the results; (4) publication resources, any publication presenting results of a research work has to be integrated to the corpus; and (5) documentation resources which document the corpus description (e.g. experimentation description, analysis work description).

As far as the resources types to be shared within an analysis corpus are concerned, these can only be (1) analysis resources, (2) publication resources, and (3) documentation resources.
Resources are described by metadata giving different information about them (e.g. creator, creation date, format, etc.). We will not detail these metadata in this paper for the sake of space.

d) Analysis work description: It is essential to share the description of the analytical work realized on a corpus. This description represents a kind of tracing of the researcher's activity when performing analytical work, and thus represents a very important support to other researchers to reproduce and compare published results. The concepts defined by BeatcorpOnto depict an analysis work performed outside the BEATCORP platform and integrated to an initial corpus, or an analysis work performed within the platform and integrated to an analysis corpus dealing with the studied research question. An analysis work is described by: (1) its begin date and end date; (2) its description; (3) its analysis objective(s); and (4) reference(s) to the researcher(s) that created and/or contributed to the analysis work. An analysis work can be linked to publication resource(s) that describe it. Assuming that an analysis work can involve more than one tool, analysis is described for every analysis tool used. An analysis by tool is described by (1) the analysis tool used to perform analysis, (2) the date of extraction of the interaction traces to be analyzed, this information ensures the reproducibility of the analysis results because it would enable the retrieval of the same interaction trace data even if the corpus were enriched later; (3) the description of the analysis performed by the tool; (4) the complementary resources imported to be used by the analysis tool in performing analysis; and (5) the resources, if any, produced by the analysis tool as a result of the analysis. A researcher using an analysis tool to perform an analysis can choose to use resources coming from one or more initial or analysis shared corpora. So, when describing an analysis by tool, for each corpus having resources used by the analysis, the performed scripts, (cf. the following paragraph) to extract, filter and format the corpus data to be analyzed, have to be referenced.

2) Trace Concept Model: This part of the model concerns the semantics of data collected within an interaction trace. The approach consists in defining a set of concepts that can be found in learning interaction traces. Each of these concepts is given a textual definition having a semantic level close to that of the researcher. A researcher, wishing to share a corpus in the platform in order to further analyze it using shared analysis tools, checks the concepts we define in the ontology and compares them to those present in his corpus' interaction traces. This can be seen as an explicit mapping that the researcher performs between a sub-set of the concepts of our ontology and the implicit ontology that defines the concepts of his traces. Indeed, we consider that even if the researcher doesn't define an explicit ontology to model his interaction traces, such an ontology exists in his mind and is expressed through the trace format. Thus, the researcher identifies among the concepts defined in the ontology those that exist in his interaction trace corpus. In practice, the mapping is achieved between a specific format of interaction traces and the ontology concepts we define. Identifying an ontology concept as being present in a corpus is related to the trace format produced by the learning tool(s) used to produce the corpus. A particular mapping is then closely related to a particular trace format. Mapping an ontology concept to a particular trace format will be done by writing a script (cf. following paragraph) which extracts the data corresponding to the concept from trace resources having that format and by linking the script to the ontology concept. A script is therefore specific to a couple of a specific trace format and a particular ontology concept. From the analysis point of view, a shared analysis tool accepts an input format which defines the input data it is able to analyze and their formatting. As for the outputs of learning tools, we state that an implicit ontology defines the concepts to be present in the input of an analysis tool, which are defined in the tool input format. The researcher, who wishes to use an analysis tool on the traces of a shared corpus, identifies within the ontology concepts those that are necessary for his analysis work and verifies if a mapping exists between those concepts and the trace format. The following paragraph explains in detail how to query a corpus and use an analysis tool to analyze the extracted data. As already mentioned, interaction traces can contain different information depending on the observer needs which means that it is possible to have more or less information to represent the same kind of interaction traces. Concepts defined within the ontology may be missing and then not cover every information that can be retrieved within interaction traces.

Among these information, some can be very specific to a particular domain and others can be relatively generic and related to a category of tools (e.g. communication tools, production tools). We are specifically interested in the second type of concepts (i.e. generic concepts). If a researcher wishing to share a corpus notices that a relatively generic concept is present in his interaction traces but not in the ontology, it will be possible to add that concept to the ontology. We don't claim that the ontology we propose is complete. Therefore, it can be enriched as new corpora and analysis tools are shared within the platform. This enrichment will be straightforward because of the simplicity of the concept class definition within the ontology. A concept is represented as an OWL class and can be classified depending to its specificity. We identified four types of concept classes that could be further developed. Indeed we worked on a "limited" set of concepts as a proof of concept. An application example will be presented in section V. The four concepts categories we identified are: (1) time concepts, which give information about time indicator within an interaction trace (e.g. begin date, duration); (2) generic concepts, identified as frequently found in interaction traces independently of used tools within learning environments (e.g. user, tool); (3) communication concepts, related to communication tools that are frequently used within TEL environments (e.g. message, recipient); and (4) production concepts, related to production tools (e.g. produced object, activity evaluation). These categories will certainly be enriched to be able to represent the concepts that are interesting to researchers. Furthermore, as already mentioned, a researcher may be interested in querying resources produced by analyses. One or more categories can then be added to represent concepts useful for the semantic representation of analyses that enrich interaction traces. We identified two types of concepts that can be defined within the ontology: (1) a simple concept, which is related to simple information extracted from interaction traces (e.g. user performing action); (2) a complex concept, which is composed of other simple or complex concepts (e.g. chat interaction composed of other concepts like "user", "begin date", "message").

3) Operational Model: This model defines the operational mechanisms that allow querying, extracting and formatting shared corpus data to be analyzed within a shared analysis tool. For this purpose, we defined five types of scripts.

a) Concept querying script: this kind of script ensures the mapping between ontology concepts and a particular interaction trace format. It is frequent that TEL environments record interaction traces in an XML format, or in a format which can easily be exported or converted to XML. A concept querying script is then, in our first implementation, an XQuery (the XML querying language) [16] script that searches within an XML interaction trace document the information which corresponds to a particular concept. If the queried concept is complex, such a script calls querying scripts which correspond to the composing concepts. A concept querying script is performed on a particular trace document and for a particular position within it (e.g. the date of the tenth interaction within the interaction trace document having some path).

b) Data type converting script: Data extracted from interaction traces can be expressed in a particular data type that is different from the one expected by an analysis tool. A conversion can then be needed to express extracted data in the convenient data type. A typical example is the data type used to represent date. Indeed some systems represent date as a Unix timestamp expressed as integer value, others represent it as a date/time type. A data type converting script is then defined for a couple of an input data type and an output data type.

c) Extracting script: A researcher interested in analyzing interaction traces of one or more corpora chooses an ontology complex concept that he wishes to query. That concept should of course have been mapped with the corpus interaction traces. In other words, querying a concept supposes the existence of a concept querying script associated to that concept. An extracting script calls a complex concept querying script and performs it on a whole trace document (e.g. extracting all chat interactions). A particular extracting script is then related to a particular trace format and a complex concept defined by the ontology. Extracting script extracts all interaction traces corresponding to a complex concept without imposing any constraint.

d) Filtering script: Data extracted using an extracting script may need to be filtered in order to meet analyzer needs in terms of the data that have to be extracted in order to be analyzed by a shared analysis tool. A filtering script processes the output of an extracting script to prepare data necessary for the input of an analysis tool. A particular filtering script is then related to a particular couple of formats: a trace format and an analysis tool input format. We identified two types of filters that can be applied within a filtering script. The first filtering type is identifying a set of projection concepts. A filtering script is executed on an extracting script which is in turn related to an ontology complex concept. Projection concepts are chosen among the concepts composing the queried complex concept (e.g. the complex concept chat interaction is composed of "user", "date", "message", projection concepts are "user" and "message"). The second filtering type is the definition of selection conditions that should be verified within the filtering script output. A selection condition is related to a concept that composes the queried complex concept. It is usually defined by a comparison operator (e.g. equal, not equal, greater than) and the expected selection value to which the extracted data will be compared. An example of a selection condition could be: the chat interactions of user "*user1*" (complex concept: chat interaction; selection condition concept: user; operator : equal; selection value: user1). Furthermore, a filtering script can call, if needed, a data type converting script to completely prepare all extracted data to be analyzed.

e) Formatting script: This last script is performed on the output of an extracting script or a filtering script (if filtering is

needed) in order to format extracted data to be directly analyzable by a shared analysis tool. A formatting script is then related to a particular couple of formats: a trace format and an analysis tool input format.

B. Specificities of the approach

The proposed approach is different from existing ones because it avoids imposing an interaction trace representation. Instead, it is an open approach which tries to be adaptable to different sharing needs. This approach needs a minimal effort of integration because it allows to accept any interaction trace resources and contextualization resources without having to convert them to a particular format (the only technical constraint, in our first implementation, is that interaction traces have to be expressed in XML to be queried in XQuery). The proposed approach is incremental and needs a minimal integration effort. Indeed, a researcher who shares a corpus and wants to analyze it using a shared analysis tool isn't expected to do all the work of mapping all concepts defined within the interaction trace format, instead, he only needs to map those that have to be extracted for the analysis. Mapping performing and script implementing are then developed in a participatory manner. In fact, TEL environments usually offer a multiple set of functionalities that are not necessarily all used. Mappings and scripts can then be added when needed by a researcher and shared in order to be used by other researchers.

Concerning the shared interaction trace anonymization issue, shared data should have been anonymized before their sharing within the platform. The anonymization feature is out of the scope of our research work. As far as access rights to the platform shared corpora and analysis tools are concerned, we will use a simple access managing system inspired from the Unix operating system. We can define access rights on platform objects differently for (1) owners, (2) authorized users, and (3) other users. We will not further detail this aspect of the platform for sake of space.

V. APPLICATION EXAMPLE

In the very first implementation of the BEATCORP platform, we worked on an application example of the proposed approach. We created a first initial corpus containing interaction traces produced by a TEL environment called DREW (Dialogical Reasoning Educational Web tool) [7] which is dedicated to collaborative learning activities. DREW offers a set of tools including a chat that produced the interaction traces which we are interested in. We then created a second initial corpus produced by the course management system Moodle [17] which offers multiple tools including a chat tool which interaction traces have been collected when creating the corpus. Note that the first corpus contains traces produced during an authentic learning situation corresponding to a students' supervising session for a C language project, while the second corpus contains experimental chat session traces not significant from learning point of view. Although an initial corpus, as we have already stated, may contain contextual resources about the learning situation, the two corpora we created contain only trace resources. This is understandable for the second corpus because it doesn't represent an authentic learning situation. But, for the first corpus, this illustrates the lack of contextualization due to the absence of the sharing intention. We used the eXist-db [18], an open source native XML database management system which offers an embedded XQuery querying engine, as an infrastructure for sharing and querying corpora. For each corpus, we imported available trace resources and filled a little description (metadata like creator, date, used tool, etc.). Each of the TEL environments used in the creation of the corpora is described and their trace format resources had been imported. These resources are useful for understanding interaction trace contents, in particular those relative to chat interaction in our example, and thus for mapping ontology concepts to interaction trace contents. Studying the interaction trace representation formats relative to DREW and Moodle allowed us to identify ontology concepts relative to chat interactions that exist within chat interaction traces of these two learning environments.

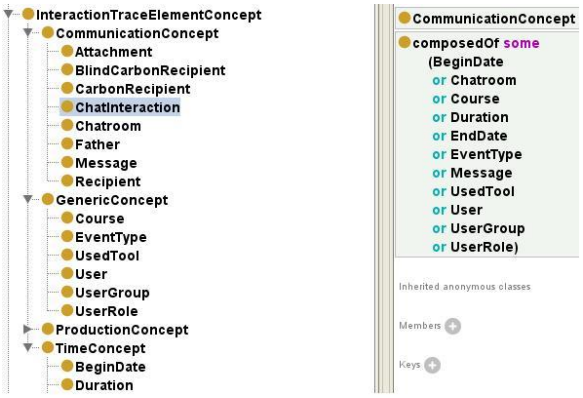

Figure 1. Sub-set of BEATCORP Ontology concepts relative to interaction traces, examples of simple concepts, and the ChatInteraction complex concept

DREW

Simple concept querying script corresponding to Message concept

```
declare function drew:chatMessage($doc as xs:string, $position as xs:integer) as
xs:string
{
    let $msg := (doc($doc)//chat)[$position]/text/text()
    let $empty := ""
    return
    if (empty($msg))
    then $empty
    else
    $msg
};
```

Complex concept querying script corresponding to ChatInteraction concept

```
declare function drew:chatInteraction($doc as xs:string, $position as xs:integer) as
node()
{
    <ChatInteraction>
    {
    <User>{ drew:chatUser($doc, $position) }</User>,
    <BeginDate>{ drew:chatBeginDate($doc, $position)}</BeginDate>,
    if(not(empty(drew:chatDuration($doc, $position))))
    then <Duration>{ drew:chatDuration($doc, $position)}</Duration>
    else (),
    <Message>{ drew:chatMessage($doc, $position)}</Message>
    }
    </ChatInteraction>
};
```

Exrtracting script retrieving all chat interactions, calls ChatInteraction querying script

```
declare function drew:allChatInteractions($doc as xs:string) as node()*
{
    let $nbChatInteractions := drew:nbChatInteractions ($doc)
    for $i in (1 to $nbChatInteractions)
    return
    drew:chatInteraction($doc, $i)
};
```

Filtering script retrieving all chat interactions corresponding to sending messages, executed on the output of allChatInteraction extracting script

```
declare function drew:allChatMessageSendingInteractions( $allChatInteractions as
node()* ) as node()*
{
        for $i at $j in $allChatInteractions
        return
        if(empty($i/Message/text()))
        then ()
        else $allChatInteractions[$j]
};
```

Figure 2. Examples of (1) simple and complex concept querying script, (2) extracting script, and (3) filtering script, relative to DREW chat interaction traces

Moodle

Simple concept querying script corresponding to Message concept

```
declare function moodle:chatMessage($doc as xs:string, $position as xs:integer) as
xs:string
{
    let $msg := (doc($doc)//chatInteraction)[$position]/message/text()
    return
    if($msg = "enter" or $msg = "exit")
    then ""
    else $msg
};
```

Complex concept querying script corresponding to ChatInteraction concept

```
declare function moodle:chatInteraction($doc as xs:string, $position as xs:integer) as
node()
{
    <ChatInteraction>
    {
        <User>{ moodle:chatUser($doc, $position)}</User>,
        <UserRole>{ moodle:chatUserRole($doc, $position)}</UserRole>,
        <UserGroup>{moodle:chatUserGroup($doc,$position)}</UserGroup>,
        <BeginDate>{moodle:chatBeginDate($doc,$position)}</BeginDate>,
        <Chatroom>{moodle:chatroom($doc, $position)}</Chatroom>,
        <Course>{moodle:chatCourse($doc, $position)}</Course>,
        <EventType>{moodle:chatEventType($doc,$position)}</EventType>,
        <Message>{moodle:chatMessage($doc, $position)}</Message>
    }
    </ChatInteraction>
};
```

Exrtracting script retrieving all chat interactions, calls ChatInteraction querying script

```
declare function moodle:allChatInteractions($doc as xs:string) as node()*
{
    let $nbChatInteractions := moodle:nbChatInteractions ($doc)
    for $i in (1 to $nbChatInteractions)
    return
    moodle:chatInteraction($doc, $i)
};
```

Filtering script retrieving all chat interactions corresponding to sending messages, executed on the output of allChatInteraction extracting script

```
declare function moodle:allChatMessageSendingInteractions( $allChatInteractions as
node()*) as node()*
{
        for $i at $j in $allChatInteractions
        return
        if(empty($i/Message/text()))
        then ()
        else
        $allChatInteractions[$j]
};
```

Figure 3. Examples of (1) simple and complex concept querying script, (2) extracting script, and (3) filtering script, relative to Moodle chat interaction traces

Formatting script for Tatiana analysis tool

```
declare function tatiana:convertInteractionsToTatianaFormat($input as node()*) as node()*
{
    <display>
    {
    for $i in $input
        return
        <item>
            <info name="time">
                <time>
                    <date>{$i/BeginDate/text()}</date>
                    {if(not(empty($i/Duration/text())))
                    then
                    <duration>{$i/Duration/text()}</duration>
                    else ()}
                </time>
            </info>
          {  for $info in $i/*
           let $node-name := xs:string(node-name($info))
           return
           if(not(empty($info/text())) and not($node-name="Duration") and not($node-name="BeginDate"
           then
           element info{attribute name {$node-name}, $info/text()}

           else ()
           }
        </item>
    }
    </display>
};
```

Figure 4. Example of formatting script for preparing input of the Tatiana analysis tool

Fig. 1 illustrates a sub-set of interaction trace concepts defined by the BEATCORP ontology. We can see simple concepts like "Message" and "User", and complex concepts like "ChatInteraction". The "ChatInteraction" concept is defined as being composed of a number of simple concepts. It should be noted that a specific chat interaction trace format can contain data mapped with "ChatInteraction" concept but which is mapped to a subset of the composing simple concepts. The composing simple concepts are defined widely to cover as many chat interaction traces as possible. Fig. 2 and Fig.3 respectively present examples of different types of scripts necessary in querying shared interaction traces produced by DREW and Moodle in order to analyze them. They present examples of: (1) a simple concept querying script relative to the "Message" concept; (2) a complex concept querying script relative to the "ChatInteraction" concept; (3) an extracting script that returns all chat interactions of a trace resource, this script calls the previous one; and (4) a filtering script allowing to filter the output of chat interactions extracting script by keeping only the traces relative to sending message events. Fig. 4 presents an example of a formatting script that formats the results of a filtering script in order to be analyzed by the Tatiana analysis tool [8].

VI. CONCLUSION AND FUTURE WORK

This paper presents the "proxy approach", an ontology-based approach for sharing contextualized interaction trace corpora and analysis tools to analyze these corpora. This approach avoids, in contrast to existing approaches, imposing a trace representation. Alternatively, it is based on an ontology that defines three models to support sharing: the corpus model, the trace concept model and the operational model. Finally we presented an application example as a proof of concept for our approach. Next step will be to work on new different examples to test the applicability level of the approach.

REFERENCES

[1] H. Chebil, C. Courtin, and J.-J. Girardot, "The proxy model: A new approach to sharing and analyzing learning traces corpora" International Conference on Knowledge and Education Technology (ICKET 2012), in press.

[2] A. Corbel, J.-J. Girardot, K. Lund, "A method for capitalizing upon and synthesizing analyses of human interactions", First European Conference on technology Enhanced Learning, workshop Exploring the potentials of networked-computing support for face-to-face collaborative learning, Crete, Greece, pp. 38–47, october 2006.

[3] K.-R. Koedinger, K. Cunningham, A. Skogsholm, B. Leber, "An open repository and analysis tools for fine-grained, longitudinal learner data", First International Conference on Educational Data Mining, Montreal, Quebec, Canada, pp. 157–166, 2008.

[4] A. Martínez, A. Harrer, B. Barros, "Library of interaction analysis methods", ICALTS JEIRP project livrable, 2005.

[5] C. Reffay, M.-L. Betbeder, "Sharing corpora and tools to improve interaction analysis", Fourth European Conference on Technology Enhanced Learning, Nice, France, pp. 196–210, 2009.

[6] L.-S. Settouti, Y. Prié, P.-A. Champin, J.-C. Marty, A. Mille, "A traceèbased system for technology-enhanced learning systems personnalisation", Ninth IEEE International Conference on Advanced Learning Technologies, Riga, Lativa, july 2009.

[7] A. Corbel, P. Jaillon, X. Serpaggi, M. Baker, M. Quignard, K. Lund, "Un outil Internet pour créer des situations d'apprentissage coopérant", in Desmoulins, Marquet, & Bouhineau, Eds. EIAH 2003, Strasbourg, pp. 109–113.

[8] G. Dyke. "A model for managing and capitalising on the analyses of traces of activity in collaborative interaction", Phd thesis, Ecole des Mines de Saint-Etienne, 2009.

[9] Calico, 2012, http://woops.crashdump.net/calico/.

[10] Datashop, 2012, https://pslcdatashop.web.cmu.edu/.

[11] DCMI, 2012, http://dublincore.org/.

[12] MULCE, 2012, http://mulce.univbpclermont.fr:8080/PlateFormeMulce/.

[13] OWL, 2012, http://www.w3.org/TR/2004/REC-owl-guide-20040210/.

[14] Protégé, 2012, http://protege.stanford.edu.

[15] Tutor Message Format, 2012,
https://pslcdatashop.web.cmu.edu/dtd/guide/tutor_message_dtd_guide_v4.pdf.

[16] XQuery, 2012, http://www.w3.org/TR/xquery/.

[17] Moodle, 2012, http://moodle.org/.

[18] eXist-db, 2012, exist-db.org