Traces Based Reasoning (TBR) Definition, illustration and echoes with story telling

Alain Mille *

August 24, 2005

1 Introduction

The question is wether it is possible to exploit a computer environment to facilitate user's tasks, whatever they are. The goal is not to develop a new kind of "intelligent" systems using a "world representation" to reason about it as the user himself according to some adapted cognitive model. The question is rather to design computer environment able to work "in understanding" with the user. This system's "empathy" for the user cannot be reduced at building a user profile, typifying his/her according to pre-established factors. Actually, the user works in context of an environment which cannot be reduced to the computer environment and it is vain to ask him/her to describe precisely the significant context of each use. Hence, the main idea is to try to progressively learn useful contexts from user interactions which are mediated by the computer environment.

This paper does not address the use of traces as data to mine and analyse in order to build general rules on the "world domain", which is an other research field.

The question is to exploit traces as they are "recognizable" by users to make easier user's tasks when mediated by a computer environment.

The first part of the paper focuses on this approach, trying to define precisely what we propose to call "Traces Based Reasoning"; second part describes shortly the MUSETTE (Modeling USEs and Tasks to Trace Experience) approach common base of several TBR oriented systems. MUSETTE theorizes the notion of "use trace" and its exploitation by an Explained Task Signature (ETS) mechanism; third part presents shortly a TBR oriented application which is an ITS system reusing learner traces to help the learning process [1]. Discussion is an opportunity to place our work in the context of other approaches

*LIRIS UMR CNRS 5205 Universités Lyon1, Lyon2, Insa-Lyon, EC-Lyon : This synthesis leans on research works carried out by the team "Cognition, Experience and Situated Agents" and specifically on works carried out with Pierre-Antoine Champin, Jean-Mathias Héraud et Yannick Prié. This research report is the English translation of a contribution to a French book

[1]PIXED: Projet d'Intégration de l'Expérience dans l'Enseignement à Distance

exploiting use traces. This last part of the paper pinpoints synergies between TBR and story telling engineering.

2 Traces Based Reasoning?

Generally speaking, the meaning of the term "experience" implies that any knowledge would find its source starting from some experience. Reasoning is the result of consistent recall of previous knowledge able to produce some output in the context of some particular goal. Expertise and practice are two different registers of experience depending on wether experience is compiled in explicit symbols or implicitly embodied; experience being behind any knowledge, any reasoning would be "experience based". Expertise is declarative knowledge while practice is not. As practice is not declarative, traces in the environment[2] are the only indirect "records" of non explicited knowledge emerging during concrete action. These records can be very efficient for making easier knowledge emergence if they are available somewhere in the environment during a similar experience.

This idea of "knowledge record" has bin introduced and argumented by Bruno Bachimont [2] who proposed the concept of Knowledge Records Engineering, what could refer to Bernard Stiegler ideas [28] about notion of "primary, secondary and ternary retentions", ternary retentions making possible "externalization of individual experience through traces and as transmission" [free translation of alinéas 10-13, page 23].

In the restricted field of computer environments, "knowledge records" can be the result of an explicit process of knowledge engineering [8]. Such a process searches to represent knowledge under formula which can be exploited by a computer environment. So, representing semantics by formula would allow interpretations corresponding more or less to the so recorded knowledge. There exists a large spectrum of such ways to record knowledge in a computer environment: from very simple inference systems to sophisticated solving problem environments.

2.1 Knowledge Based Systems

Knowledge Based Systems [31] are very ambitious. The main idea is that it is possible, inside a particular domain, to solve problems automatically on the basis of knowledge recorded as formula exploited by inference engines able to demonstrate their results. Artificial Intelligence Techniques exploited this approach with some success but also with some disappointments due to the fact that real situations are dynamic and that it is difficult to encode any possible context. Many works addressed this problem of dynamic situations by improving languages of knowledge representation and corresponding algorithms in order to be able to describe the dynamics in a proper way. These works are yet active and provides a variety of logics allowing to build expressions taking into account time, space, point of view, context, belief, etc. When the question is to represent common sense knowledge, works on epistemic logics try to propose efficient solutions. More expressive is the representation, more important is the

[2]Environment in general including the user himself, not only a computer environment.

knowledge engineering effort to elicit knowledge in the appropriate way. Actually, knowledge engineering research finds its origine in this huge difficulty to elicit useful knowledge in order to represent it in a computer environment.

2.2 Case Based Reasonning

Case Based Reasoning (CBR), has also ambition to be able to solve problems by reusing solved ones. Building a library of solved problems (cases) would be easiest from a knowledge engineering point of view than building equivalent knowledge bases. CBR finds its roots in research works in psychology [25] but computer applications simplified the approach. CBR is less ambitious than KBS because reasoning is analogical with abductive inference: a problem solving episode, called a case, is represented by a problem (problem descriptors) and by its solution (solution descriptors). A new target problem is solved by adapting the solution of a past similar problem. This is an hypothesis which has to be checked to the reality and there is no way to demonstrate that this is really a solution, and that it is the best one. Checking the proposed solution allows to revise and repair the case before adding it to the cases library. Despite its inherent limitations, CBR meets big successes: the more important reason is that a CBR system is apparently a "learning system" able to add new cases and, theoretically, to improve its domain representation. CBR works we carried out show us that users try to use cases for other problems than the ones they were described for: they try to rearrange case descriptors under other problem-solution pairs. Of course, the original CBR system cannot help them for these new kinds of problems. Nevertheless, Cases are viewed as very helpful documents even when they cannot use the abductive inference: they use simple requests on the case library to mine it.

2.3 Knowledge management / Knowledge Engineering

Knowledge management [22] has not the ambition to solve problems automatically; the question is to use a computer environment for documenting at best the knowledge of an institution, a group, or individuals. The target is any document recording knowledge in some way and the challenge is to develop organization and its corresponding computer environment to collect, to organize knowledge records with the hope to allow its future availability in a proper way for the institution, the group or individuals.

The main goals of Knowledge Management are reusing and sharing these knowledge records in such a way that they could make sense in the situation they will be re-exploited. The computer environment is not an automatic interpreter of the knowledge records but a confortable way to index them.

The general idea is then to gather a documentary recording of human knowledge with some formal description of the contents allowing to make some inferences to link together different knowledge pieces. Most of the time, the inference mechanism is classification based on some ontology. Description logics, for example, are well suited for this kind of description. Usually, computer environments use logically organized terms in order to improve the requests on documents by ontological inferences.

Knowledge engineering found a new and important development with the knowledge management challenge. Actually, it is easy to understand that describing logically the contents of knowledge records, could be seen as representing the knowledge itself in some way. Researches in this direction are in the same field than knowledge based systems, and many researches try to add more expressivity at description logics for example. This effort is specifically important in the Semantic Web community [3] trying to offer powerful inference mechanisms in what is called the "middleware" of computer environments. Qualities and drawbacks of knowledge representation are still there, and it is really difficult to build robust symbolic descriptions of the world with a shared semantic.

2.4 Story telling and Traces Based Reasoning

Story telling approach for knowledge management has a low inferential ambition too. The question is to develop computer environments to collect, manage and restitute stories as powerful knowledge containers for who are telling them, who are listening them and who are reusing them. This kind of knowledge record is particularly adapted for knowledge sharing and, hence, is very useful in situations of human learning. Once more, exploiting stories through computer environments, needs some efficient description of their contents. That description can be done by cutting down stories in short story components. These short story components are easier to describe in some robust way.

Regarding other knowledge recording supports, narration has intrinsic properties pointed out by Schank for example. One of these remarkable properties is that narration traces a sequence of events considered as relevant to express the underlying process. Actually, temporality is consubstantial to narrativity. Moreover, one does not tell a story by accident; the story is there to illustrate at less a situation with some salient event which will do the "wit" of the story.

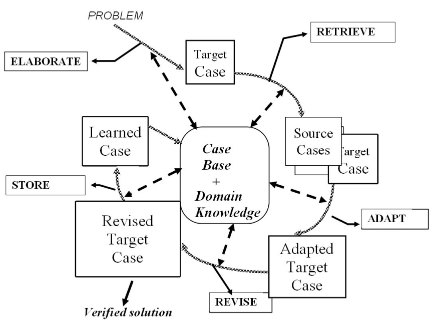
2.5 Traces Based Reasoning

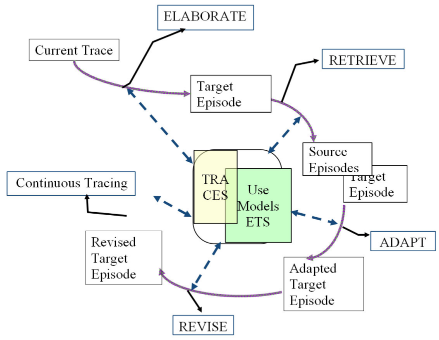
We share the idea that human experience, temporally situated by definition, is well represented by a temporal record or trace describing an implicit underlying process. CBR claims also that property by addressing problem solving episodes, even if, de facto, CBR systems exploiting the temporal dimension of cases are not so numerous; case descriptors are not compulsorily time stamped. Moreover, a problem solving episode is considered independently of the different "stories" (contexts) where these episode occurred. A case is described with a fixed granularity, in a specific temporality and contains intangible description terms.

For our part, we proposed to exploit use traces of a computer environment as possible indirect records of knowledge which emerged while the user did his/her tasks with the help of the computer environment. We propose a theory defining what we call "trace", how it can be represented and which kind of computations can be done in order to retrieve useful past sequences for new uses.

When traces are exploited on the basis of pattern similarities allowing some adaptation to new situations, we propose to call this kind of computation "Traces Based Reasoning" (TBR). TBR is a kind of generalization of CBR principles.

By analogy with CBR, the TBR cycle would be illustrated as figured in table 1.





CBR Cycle	TBR cycle

Table 1: The CBR cycle handles cases, which are stored in a case base, under a predefined form; the TBR cycle elaborates dynamically episodes which could be potentially useful in available traces according to some "task signature"; the target episode is built with the help of other proposed episodes under the user control. The target episode belongs to the current trace, it will be stored in it without particular indexing. Stored traces are containers of potential episodes which will be revealed in new situations.

As for CBR, we consider that most of the reasoning cycle steps can be realized by the computer environment or/and by the user himself. In the next parts, we present what we propose to call "trace theory" with the MUSETTE model as starting point. Then, we illustrate TBR by an Intelligent Tutoring System where traces are used to help the learning process.

3 Une théorie de la trace : l'approche Musette

It is a triteness to say that computer environments are largely used for more and more numerous and variate tasks with the help of organization tools, storing tools, communication tools, information research and sharing tools, ...

Computer environments allows some user personalization of their interfaces in order to fit his/her practices, his/her uses and, generally speaking his/her needs. To go further with this strong trend, it is necessary to catch information in order to be able to describe somehow the tasks realized with the computer environment; tracing uses corresponding to these tasks is the challenge; the traces must be represented in such a way that they could be re-exploited later in order to answer unanticipated questions.

3.1 General presentation

Figure 1 presents the MUSETTE general framework. A user interacts with the computer environments which modifies it i n some way. An observer agent records interaction traces according to some observation modalities depending on environment instrumentation possibilities. The observer produces what we call a "primitive trace" according to a so called "use model" describing what is "object of interest" for the user. This primitive task can then be analyzed in order to provide "use episodes" according to some "Explained Task Signature

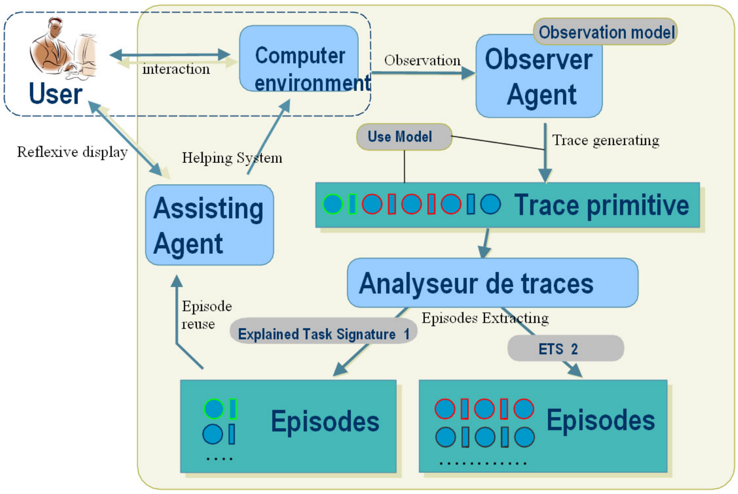


Figure 1: General architecture of a computer environment exploiting use traces for user assistance

(ETS)". Potentially,these episodes constitute explained experience knowledge containers which can be exploited as such. For example, a facilitator agent could present them to the user, in similar situations, as inspiration for his/her current tasks (reflexive use of traces). Hence, and this is of particular interest when launching new computer tools environments, one can observe and analyse user practices by the way of relevant traces and ETS: ethical questions raise there and have to be solved in order to guarantee personal data privacy. In each case, the question is to express at best interactions contents with the computer environment and the "use model" presented above has to be elicited with this goal.

3.2 Use Model and Traces

In order to build a primitive trace fitting to the user point of view, it is necessary to capture interactions, which can be done by the way of instrumentation tools of the computer interface.

These tools are developed rather to "spy" users than to provide users raw information where to find "objects of interest" as they are manipulated during interactions. Confidentiality questions and privacy area protection are discussed at the end of the paper. The observer agent must be able to detect objets of interest for the user starting from direct or indirect interaction events on them. The primitive traces is constituted of so flagged objects of interest which are recorded from the tracing flow. Primitive trace is the first level of trace "readable" by a user[3].

MUSETTE proposes a high level ontotogy to define what are objects of interest, as illustrated by figure 2.

[3]Readability is relative: as any detail is present, it could be boring "to read" such a primitive trace except for an external analyst willing to miss nothing...

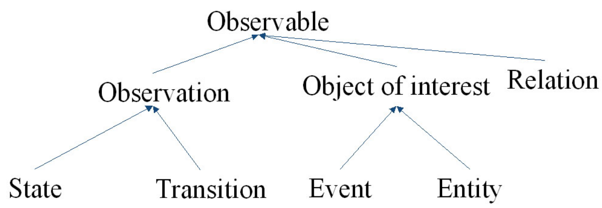


Figure 2: Ontology of basic types in MUSETTE approach

From the user point of view, an "observable" is an object of interest either "event" or "entity" depending on his feeling that this object occurs during the interaction or was there at the beginning of the interaction but playing a role during it. Of course, an observation of one interaction is itself an "observable" since it is the basic sequencing entity in a trace. One interaction is characterized by one state and one transition which have to be observed as such. There exists two type of observation: "state" and "transition". Finally, the general notion of "relation" between observables has been introduced in order to make easier relationships expression during interaction (as opposed to permanent relationships which would exist between object of interest and can be described in some ontology for example).

The use model is a specialization of the MUSETTE high level ontology: it is used to describe different types of entity, event, relations which are meaningful for the user and which are observable by the observer agent through some interface instrumentation. An intrinsic constraints have to be satisfied: traces must be constituted by strict sequences of states and transitions; states are exclusively constituted of entities; transitions are exclusively constituted of events. Over extrinsic constraints can be added to the use model provided that the language expressing the model allows to express them in a proper way. Extrinsic constraints expression form is not defined in MUSETTE.

Figure 3 illustrates an excerpt of such a use trace (internet navigator observation).

3.3 Explained Task Signatures and Episodes

An episode is a sequence in a trace flagged for its capacity to express interactions in the framework of a particular user task and which could be useful for experience reusing or sharing for example. "Explained Task Signature" (ETS, see figure4) is proposed mechanism to express the way of triggering and flagging such episodes in a trace. ETS is a pattern constituted of observables in such a way that users expected to find them again in its task context by approximately matching it in the current task tracing flow. This task signature is "explained" by annotations which will be used as explanations of observed matches during restitution to the user. This is specifically important when the question is to share experience with others. These annotations could be constituted of simple terms, ontologically situated terms or of stories components if, as we think it interesting, the Musette approach is used in the framework of a story telling system: this point will be discussed at the end of the paper. The annotation formalism does not take place in Musette definition and depends on

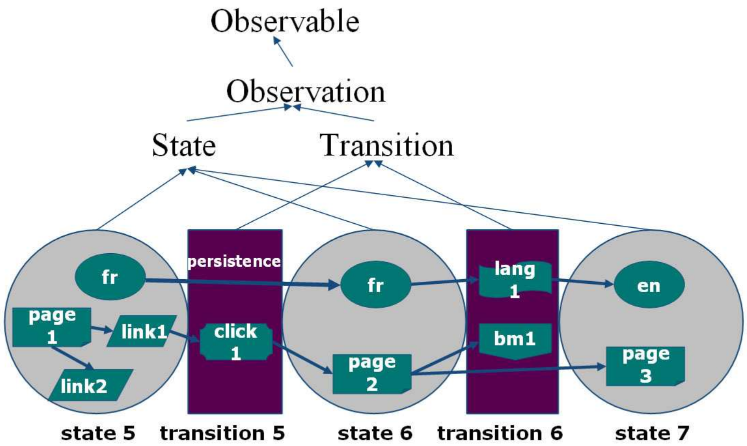


Figure 3: One excerpt of a navigator use trace: `state-5` captures the fact that page `page-1`, having 2 links `link-1` and `link-2`, is displayed with the language context `fr` (French); a click on the link `llink-1` has been considered as an event in the transition `transition-5` which leads to state `state-6` displaying page `page-2` in the context `fr`; persistency relation has been added by the observer agent (meaning that language context does not change); bookmarking `bm1` and language context change are considered as events triggering transition `transition-6` which leads to state `state-7` displaying page `page-3` (resulting of the language processing)in language context `en` (English), and so on...

the implementation of the ETS management system.

The approximately matching process between an ETS and potential corresponding sequences in the trace obeys to a graph matching problem between the ETS graph and trace corresponding subgraphs. According to expressivity of the chosen language for observables representation, the matching complexity can vary in great proportions: from simple surface matching to fine matching taking into account observable structures of properties. It is worth remembering that trace is a very special graph and that the matching process can take advantage of that: complexity reveals as manageable in an interactive exploitation ([4] [7],[6]. The matching is approximate because it is a similarity measure which is used to classify potential corresponding episodes to a given ETS. When exploited for reuse, this similarity measure should be exploited to guide some adaptation of the past episode to the current one.

3.4 MUSETTE discussion

MUSETTE approach is currently exploited for several various appications. As task facilitators, we can cite: helping new designs from past ones [5], providing an experience based companion for CAD [24], facilitating project cooperation between professionals [29], experience based guiding for ITS learners (see section 4), facilitating audiovisual contents indexation and in context retrieval [11],

[4]Possible episodes are given back to the user in real time.

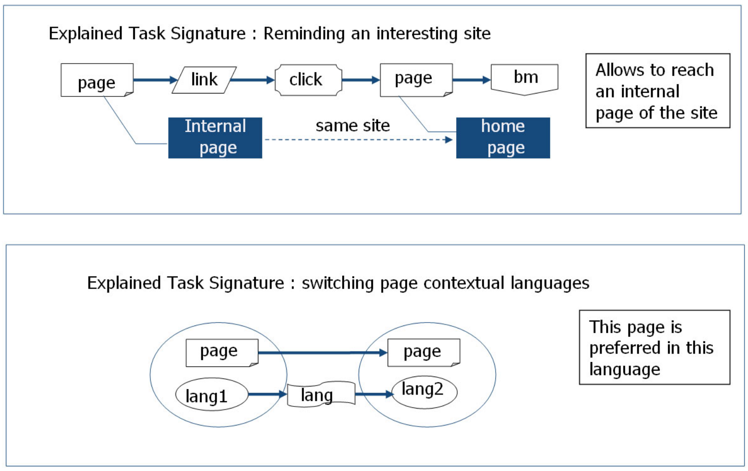


Figure 4: Two ETS examples: The pattern with sequence constraints is expressed through observables as available in the use model; annotations are linked to the ETS; in the bookmarking (of an interesting page) task example, a graphical annotation shows that the goal is to reach an internal page of a web site: it completes a textual annotation describing succinctly the signed underlying task.

facilitating experience sharing between learners during a cooperative learning process [23]; several works are ongoing in order to use the MUSETTE framework as an efficient way to describe and to manage user tasks analysis in order to design adapted computer environments and we can cite: understanding car drivers' behavior understanding [14], ITS learner behavior understanding [18]. The common base of all these applications of MUSETTE approach is that users are considered as co-designers of the computer environments potential meanings. The user point of view guides the way to exploit users' experience of computer environment according to their way of working for such or such task.

4 Un projet d'intégration de l'expérience pour l'enseignement à distance : PIXED

We illustrate shortly how Trace Based Reasoning has been concretly" used in [17] during a PhD work in our research team; a prototype named PIXED (Projet d'Intégration de l'EXpérience dans l'Enseignement à Distance[5]) has been developed.

Distance Learning gets a new dimension and a big dynamic with networking and internet facilities. The problematic core moved from "teaching" to "learning", which is well represented by the term "e-learning". We will keep here the term "distance learning" to speak about this particular learning modality.

[5]Project Integrating EXperience in Distance Learning.

4.1 General presentation

In the general framework of distance learning, we adopt a constructivist attitude according to which the learner builds his knowledge by interacting with learning environments, trying to exploit at best available educational activities. Learning is viewed as exploiting (beyond other skills) the ability to solve the problem: "What is the best way to follow in order to master this notion I have to learn?".

PIXED proposed the learner a convenient path inside the course hyperspace. The proposed model to reach this goal partly fits PAPI specifications [1] and integrates what we call [20] an annotated notional network (ANN). Network nodes are notions defined as learning goals and links are constituted by precedence relations with a defined threshold and by sufficiency relations having a threshold too and a a contribution value expressing the level of contribution of the source notion to the target notion. These relationships express didactic issues.

PIXED allows teachers as learners to annotate parts of or whole education activities when they are simple hypermedia documents (html documents).

Annotations are convenient supports of communication between teachers, between teachers and learners and between learners; they constitute the base for dynamic adaptation of contents presentation (details in [19]).

4.2 Pixed Use Model

Objects of interest are constituted by objects with which the learner interacts while learning. **educational activities** (**digital documents** and **Quiz**) are associated to nodes of the **Annotated Notional Network** (ANN). The **course** is initially built on the base of an ANN designed by teachers on a didactic base. Each learner navigates in the course thanks to the ANN and his/her progress is represented through a subgraph of this ANN which he/she can enrich by annotating it with notions he/her create and/or with educational activities he/she exploits from his/her own. Building an ANN is not trivial. Educational activities are integrated as digital documents with their description. Educational activities are annotated by **notions** they explain, or illustrate, or make reference for, and so forth..., specifying which notions are useful, mandatory, sufficient, and so on ... to exploit them. So, a set of notions is defined. Notions being connected together trough **precedence relations** and **sufficiency relations** to describe a course, they are used to automatically build an initial ANN. An important point: it is possible to build several ANN (several different courses) from the same set of notions representing the domain to teach. Building the use model is the responsibility of knowledge engineering techniques while semantics of the relations (precedence, sufficiency) is clearly oriented by didactic objectives.

Navigating in a course thanks to an ANN can be difficult for a learner who, by definition, does not master proposed notions to learn. He has to find a path beyond different possible navigations to reach the "target notion" (Starting the course, the notion supported by the course itself is the notion to reach). Pixed integrates an original process to build a relevant path under the form of a tree representing the estimated effort to reach the current target notion (tree branches lengths and orientations figures analogically the learning effort). Actually, semantics of precedence and sufficiency relations allow to build a **rec-**

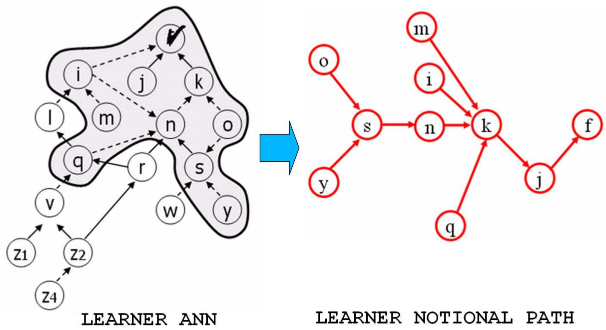


Figure 5: Translation of an ANN representing the current mastering of an ANN in a proposal of a notional path given as a learning guide.

ommendation tree to follow a path according to an adapted didactic progress. In the example of figure 5, non acquired notions and in dependency relation with target (f) are in the grey shape, whereas acquired notions are external. The notional path can be read in the following manner: "If you want to reach notion (f), then you have to learn (k) then (j). If you meet difficulties while learning (k), you can make it easier by learning first (m) or (i) or (n) or(q). If you chose (n) to prepare (k), begin learning by (s)".

4.3 What do we call "experience" in PIXED?

Experience is considered, from the system point of view, as recorded in a trace of the learning interactions observed during the computer environment use. As a first approximation, we can say that any annotation an any choice to continue the course (educational activities choice), will be traced (including interactions resulting from Quizs). What is made inside a particular educational activity is not traced in this version of Pixed. As numerous other ITS, it is possible to exploit traces to sum up learning progress by displaying a synthetic view of the learning process.

[10] reported a survey of such systems, providing above all statistics on traced predefined events and does not provide a lot about the single learning process. Our goal is to send back to the learner some possibility to exploit learning traces as a knowledge container to guide him/her in his/her learning progress. The question is "Knowing what I did, what I learnt, what I wish to learn, what is the relevant path and what are the relevant education activities on this path?". Solution is notional path adaptation with alternate educational activities which, according to similar experience, could be useful in the current situation. Learning episodes can be of various lengths as well in their "problem" part (the current learning state plus the target notion) as in their "solution" part (a specific notional path).

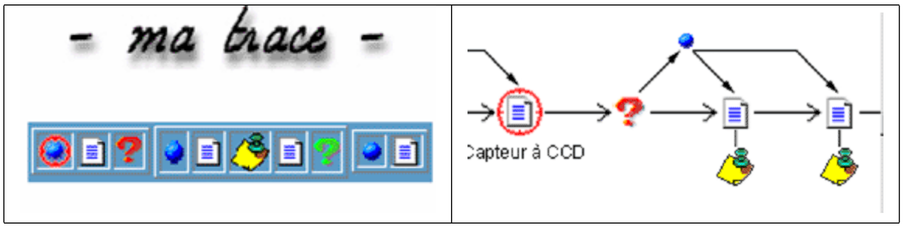


Table 2: Displayed Pixed trace (left) and displayed Experience Path (right); learner can navigate through these diagrams

4.4 Exploiting traces to assist a specific learning task: self orienting to progress in his/her learning process after failure of the current notion learning process

When the learner navigates in the course through the corresponding ANN, Pixed traces learning interactions. A current learning episode is constituted by the learning trace starting from the first login with the course goal as target notion (final notion to master). On the base of dissimilarity measures and of different success "potentials", Pixed selects similar episodes to the current one for which some help is asked. With the help of these past episodes, Pixed proposes to adapt the current one in order to maximise its success potential. Actually, this episode provides the learner an "experience path" (kind of method to continue the current episode) which he could personalize and exploit. Hence, the learner can navigate in this "experience path", can choose intermediate notions, can consult annotations written by others or can exploit educational activities, anything which has been of some help for learners in similar learning situation. As the tracing process continues, the current episode is automatically included in the trace base.

Adaptation is based on the set of episodes which maximize the learning potential for the target notion of the current episode providing some alternates to it. An ordered list of episodes with high learning potential is built and the first one is proposed to the learner as a starting point.

4.5 PIXED discussion

Pixed is a prototype developed while MUSETTE was theorized in the research team. Hence, it does not profit of all possibilities of the MUSETTE approach. Limitations concern abilities to edit the use model and to create new Explained Task Signatures. MUSETTE is a general framework while Pixed is a specific application.

Tests in classes make evidence of the fact that availability of traces allow learners as tutors or teachers to "tell" what happens during the learning process with the help of traces. Hence, sharing his/her experience with an other learner is largely facilitated with the concrete support of a "narration" based on the "experience path" as it is proposed in PIXED.

Nevertheless, if it was necessary to support other things, it would be necessary to create new ETS according to the corresponding type of Pixed episodes. These episodes would become new reflexive supports for sharing such or such

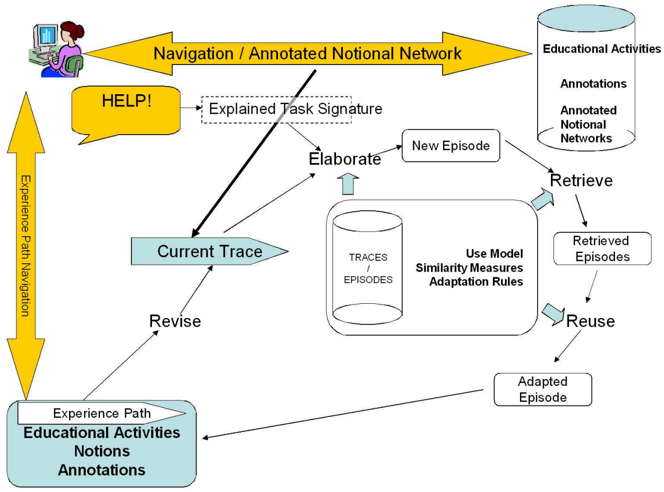


Figure 6: Pixed Cycle: during navigation, the learner asks for assistance to be guided: the learning situation is represented by an ETS according to the general task "how can I continue the learning process"; corresponding current episode is elaborated according to this ETS and available traces are exploited to find similar episodes as good candidate for help; a way to continue the learning process is proposed on the base of the best candidate episode, adapted and customized by the learner; PIXED keeps trace of this new episode as for others.

learning practice. Many other exploitations of the MUSETTE approach could be of interest for an application as PIXED.

5 Discussion RAPET / Story Telling

5.1 About the use of Trace Based Reasoning

A long time before modern computers and Internet, [4] described a system named MEMEX (figure 7) for capturing anything a scientist was searching for, what he annotated, what he used, and so forth, and so creating a trace that other scientists (and himself!) could use later. [15] displayed effects of multiple uses on digital objects by the way of wear marks on these objects as book pages are creased by multiple uses. More recently, [33] developed a tool allowing users to display paths used by other users inside a web site. The idea of exploiting computer environment interactions traces is not really new but preliminary works focused on what could be directly displayed to the user as possible path to follow for his/her proper needs; needs were not possible to describe and there was no mechanism for retrieving interesting traces according to some specific context. Hence, several works concentrated on the user task in order to "put in context" interesting paths to follow [12] whereas others like [9], [21] captured and reused navigation episodes by searching for predefined patterns; in an other direction, [30] used past procedure cases for assisting a specific application. These works are mostly in the domain of CBR and need precise descriptions of cases by forms

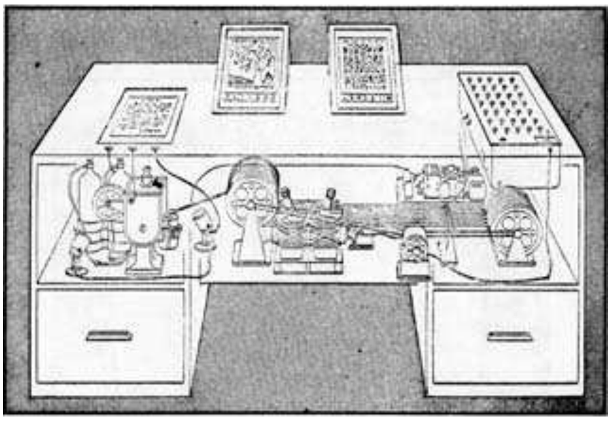


Figure 7: The system MEMEX as pictured by Vanevar Bush for exploiting automatically microfilms

or by conversational inputs while case structure has to be well specified and a regular case base has to be available.

Trace Based Reasoning allows to relax constraints of predefined requests at which system could answer by using collected experience of a computer environment for particular tasks. Nevertheless, the need of some task description is not canceled but has not to be complete at the beginning of the use process of a computer environment. An initial period of "learning" is needed by the computer environment with the help of "pilot users" in order to build use model and initial ETS while using the system itself in the context of various tasks: during this "learning" period, users work as designers of the TBR system. The modeling effort is moderate because the computer environment exists and has not to be designed itself and that the question is just to identify classes of behavior by the way of the Use Model starting from real observed uses. Hence, building Explained Task Signatures does not imply the same effort than for modeling tasks to build a knowledge based system [26] or even to document tasks in a knowledge management framework [16]. Building ETS by "playing an example" allows users to do it relatively easily. Moreover, these ETS could be used as a preliminary step to elicit and document how tasks are done for example, and it could be of big help for knowledge engineering for example.

The relative "weakness" of expression of a task in an ETS becomes a strength when the goal is to "facilitate" a task by providing users efficient means to re-exploit situated experiences as meaningful knowledge containers. Actually, it is a concrete way to manage the variability of similar situations and to take into account implicitly of the context of each situation, the user being the final "designer" of facilitators he needs specifically.

Trace Based Reasoning has to be understood as "distributed" between computer agents and users with variables levels of delegation according to agent's abilities to exploit such experience traces. Various "facilitators" can then be invented as it was illustrated through the PIXED example in this paper.

5.2 Trace Based Reasoning / Story Based Reasoning synergies

It is in the general framework of voluntary sharing of experience that we consider strong synergies between TBR and Story Based Reasoning. We can remind the "Story Based Reasoning" cycle as defined by Eddie Soulier in [27]: 1) Producing a narration of the task pinpointing the "events driven" situations, then condensing the narration in a set of events as they are sequentially perceived and storing them in memory as indexes; 2) Formatting this experience as a narration in order to remind and to share it; 3) The storyteller tells his/her experience whereas his/her interlocutor tries to extract story indexes in order to compare with his/her proper indexes and then to his/her proper stories for taking some possible lessons from this shared experience.

We can take the different steps of this reasoning schema and project them in the context of interactions with a computer environment: 1) Collecting events coming from computer environment interactions during user's activity and recording them as a sequence of states and transitions; 2) Re-exploiting this collected experience by the way of Explained Task Signatures as indexes to mine stored traces according to situational events of the current activity; 3) The "story teller" (a user willing to share his/her experience) provides episodes to the "interlocutor" (a user who would like to exploit experience of the storyteller) who exploit corresponding Explained Task Signatures as indexes to retrieve his/her proper experiences to compare and adapt them if it is useful. This is a TBR simplified cycle as it is presented in previous sections considered here in the specific context of supporting experience sharing.

The event sequence capture is systematic in the framework of TBR while Story Based Reasoning process organizes stories collect around well identified "events driven" situations. We could think about a "sorting" of experience sequences according to some "events oriented" situations and this difference will disappear.

Hence, we can formulate Pixed illustration from an experience sharing point of view as stated in the above cycle: 1) The learner exploits the computer environment for learning; he/she does not succeed to master a notion with the help of educational activities as they are proposed by the tutor; finally, he/she finds alternate useful education activities (on Internet for example) allowing him/her to succeed the Quiz of the current notion. 2) This learner (requested by an other learner) describe the situation by an explained task signature exhibiting the interactions pattern showing how he/she did to succeed after a failure sequence. This signature is explained by a short story part (his/her textual explanation of the way he/she managed the situation), is used to remind the learning trace where an episode is considered as very similar. 3) This learning trace "enhanced" by the Explained Task Signature is provided to the "interlocutor" who can first remind his/her proper learning traces on the same Explained Task Signature and, second, adatp to his/her context the "winning sequence" of the provided learning trace in order to master the notion they were both searching to learn.

An other useful exploitation of the MUSETTE approach could be to support building, analysis and re-building of events oriented narrations when largely mediated by a computer environment. This kind of exploitation seems to be a very promising perspective. Building new Explained Task Signatures would be the opportunity to "mark" the narration by specific instants (patterns) which can be retrieved in the trace flow. The ETS annotation can come from the cutting down of the story in shorter mini-stories and, reciprocally short mini-stories can be illustrated by corresponding interactive behavior episodes. ITS environments are probably a very promising application field, especially when the question is to learn know-how or know-how to behave (knowing how to behave expresses itself through more an more communication tools supported by computer environments).

Finally, using the mini-stories/explanations as basic elements for displaying use traces could be an interesting perspective since users has a usually a very efficient abilities to "interpret" things with the help of mini-stories. If the computer environment allows users to select a set of Explained Task Signatures which make sense for some activity he wants to track, we could perhaps even get something similar to a narration?

6 Conclusion

If implicit individual experience is unreachable since embodied and, consequently, if the question is how can we try to express our knowledge by the way of records reachable for us and others, then we could consider that a user who is tracing his/her interactions aiming to share his/her experience, proceeds also from knowledge recording. This is the TBR approach as presented in the first part of the paper. The second part presents a theorical approach of "traces" in a computer environment with the MUSETTE model. Then, the MUSETTE model is illustrated through PIXED which is an ITS environment including a TBR mechanism helping learner to orientate himself/herself during a learning progress. Finally, we discuss interesting synergies between Trace Based Reasoning and what we call Story Based Reasoning to support experience sharing. Discussion draws promising perspectives of other TBR / Story Telling synergies through a virtuous spiral in which Explained Task Signatures (with corresponding episodes) explain mini-stories and in which mini-stories explain episodes (with corresponding Explained Task Signature).

Tracing experience pose immediately the problem of intrusion in private areas of anybody using a computer environment. Research community have sharp consciousness of this point as reported in [13] and privacy protection becomes a big research issue for data mining community for example [32]. We want to stress on a fondamental point concerning Trace Based Reasoning: first of all, it is a "reflexive" reasoning process where a user wants to exploit his/her proper traces for some task. Traces are his/her property and this has to be clear for him/her in such a way that traces will be manageable only by him/her. It is only by an explicit exportation operation that a user can allow somebody else to exploit his/her private traces. It is not sufficient to recommend to develop TBR environment respecting these rules, it is important to study different ways to give the user an absolute control on his/her traces independently of the TBR environment, by using encrypted data for example, satisfying international standards about privacy protection.

References

[1] Draft standard for learning technology - public and private information (papi) for learners. IEEE P14842.7/D7 by IEEE.

[2] Bruno Bachimont. Pourquoi n'y-a-t-il pas d'expérience en ingénierie des connaissances. In Nada Matta, editor, *Actes de IC'2004*, pages 53–64, LYON, mai 2004. Presses Universitaires de Grenoble.

[3] Tim Berners-Lee. *Weaving the Web*. Harper San Francisco, 1999.

[4] Vanevar Bush. As we may think. *Atlantic Monthly*, 1(176):101–108, 1945.

[5] Pierre-Antoine Champin. *Modéliser l'expérience pour en assister la réutilisation : de la Conception Assistée par Ordinateur au Web Sémantique*. Thèse de doctorat en informatique, Université Claude Bernard - Lyon 1, Villeurbanne (FR), 2001.

[6] Pierre-Antoine Champin and Yannick Prié. Musette: uses-based annotation for the semantic web. In S. Handschuh and S. Staab, editors, *Annotation for the Semantic Web*, pages 180–190. IOS Press, Amsterdam (NL), 2003.

[7] Pierre-Antoine Champin, Yannick Prié, and Alain Mille. Musette: Modelling uses and tasks for tracing experience. In Fuchs, editor, *Workshop From structured cases to unstructured problem solving episodes - WS 5 of ICCBR'03, Trondheim (NO).*

[8] Jean Charlet. L'ingénierie des connaissances,entre science de l'information et science de la gestion. Working Paper sic00000805, CNRS, France, 2004. http://archivesic.ccsd.cnrs.fr.

[9] Françoise Corvaisier, Alain Mille, and Jean-Marie Pinon. Information retrieval on the www using a decision making system. In *Proceedings: RIAO'97*, pages 284–295, 1997.

[10] Lise Desmarais, Lise Duquette, Delphine Renié, and Michel Laurier. Evaluating learning and interactions in a multimedia environment. *Computer and the Humanities*, 31:327–349, 1998.

[11] Elöd Egyed-Zsigmond. *Gestion des connaissances dans une base de documents multimédias*. Phd, INSA de Lyon, 2003.

[12] Robert Farrell, Peter Fairweather, and Eric BrumBrumer. A task-based architecture for application-aware adjuncts. In *Proceedings of International Conference on Intelligent User Interfaces*, pages 82–85. ACM Press, 2000.

[13] Gabriel Ganascia. Atelier information, communication et connaissance. Rapport de recherche SIC 00001080, CNRS, Septembre 2004. archivesic.ccsd.cnrs.fr.

[14] Olivier Georgeon, Thierry Bellet, Alain Mille, Daniel Tisserand, and Robert Martin. Driver behaviour modelling and cognitive tools development in order to assess driver situation awareness. Soumis à Humanist workshop on Modelling Driver Behaviour in Automotive Environments. Ispra., 2005.

[15] William C. Hill, James D. Hollan, Dave Wrobleweski, and Tim Mc Candless. Edit wear and read wear. In *Proceedings of ACM Conference on Human Factors in Computing Systems, CHI'92*, page 3, 1992.

[16] Harald Holz, Anne Könnecker, and Frank Maurer. Task-specific knowledge management in a process centred see. In K.D. Althoff, R.L. Feldmann, and W. Müller, editors, *Advances in Learning Software Organizations, LSO 2001*, volume * of *LNCS*, 2001.

[17] Jean-Mathias Héraud. *PIXED : une approche collaborative de l'expérience et l'expertise pour guider l'adaptation hypermédia en enseignement à distance*. Thèse de doctorat en informatique, Université Claude Bernard Lyon1, 2002.

[18] Jean-Mathias Héraud, Jean-Charles Marty, Laure France, and Thierry Carron. Une aide à l'interprétation de traces : Application à l'amélioration de scénarios pédagogiques, 2005. Soumis à EIAH05, Environnements Informatiques pour l'Apprentissage Humain.

[19] Jean-Mathias Héraud and Alain Mille. Pixed : assister l'apprentissage à distance par la réutilisation de l'expérience. In Jean Lieber, editor, *Proceedings de l'Atelier Raisonnement à Partir de Cas, Plateforme AFIA'03 Laval*, 2003.

[20] Jean-Mathias Héraud, Alain Mille, and Jean-Michel Jolion. Les réseaux notionnels : un outil pour guider la navigation dans un cours hypermédia. In *Proceedings 3ème Colloque Jeunes Chercheurs en Sciences Cognitives*, pages 116–121, Soulac, France, 1999.

[21] Michel Jaczinkski and Brigitte Trousse. WWW assisted browsing by reusing past navigations of a group of users. In *Advances in cases Based-Reasoning EWCBR'98*, 1998.

[22] Ikujiro Nonaka and Hirotaka Takeuchi. *La connaissance créatrice, la dynamique de l'entreprise apprenante*. De Boeck, 1997.

[23] Magali Ollagnier-Beldame. Sharing experience in human learning. 2005.

[24] Mick Philippon. Cad/cam companion. 2005.

[25] Roger C. Schank. *Dynamic Memory: A theory of reminding and learning in computers and people*. Cambridge University Press, 1982.

[26] Guus Schreiber, Hans Hakkermans, Anjo Anjewierden, Robert de Hoog, Niqel Shadbolt, Walter Van de Welde, and Bob Wielinga. *Knowledge Engineering and Management: The CommonKADS methodology*. The MIT Press, 1999.

[27] Eddie Soulier. *Techniques de STORYTELLING pour le partage de connaissances dans les communautés de pratique*. Thése de doctorat en informatique, Université de Paris VI, 2004.

[28] Bernard Stiegler. Désir et connaissance. le mort saisi par le vif. eléments pour une organologie de la libido. *Revue d'intelligence artificielle : Numéro spécial*, 19(1-2):13–29, 2005.

[29] Arnaud Stuber. Management of sense emergence in experience sharing. 2005.

[30] Akira Takano, Yuko Yurugi, and Atsushi Kaenaegami. Procedure based help desk system. In *Proceedings: ACM IUI 2000*, pages 264–272, New Orleans, LA USA, 2000.

[31] Jean-Michel Truong, Alain Bonnet, and Jean-Paul Haton. *Systèmes Experts*. InterEditions, Paris, 1986.

[32] Vassilios S. Verykios, Elisa Bertino, Igor Nai Fovino, Loredana Parasiliti Provenza, Yucel Saygin, and Yannis Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Record*, 33(1):50–57, 2004.

[33] Alan Wexelblat and Pattie Maes. Footprints: History-rich web browsing. In *Proceedings of RIAO 97*, pages 75–84, 1997.