International Research Journal of Computer Science (IRJCS) ISSN: 2393-9842
Issue 4, Volume 2 (April 2015) www.irjcs.com

Towards a generic approach for learner/player tracking in Serious games

Yassine EL Borji*, Mohamed Khaldi
Laboratory of Computer Science Operational Research
and Applied Statistics, Abdelmalek Essaadi University,
B.P.2121, Mhanech II, Tetouan, MOROCCO.

Abstract— There is a consensus that serious games are playing an increasingly important role in the educational sphere, they have reached a certain maturity to become a possible alternative to traditional methods of learning since they have become an essential part of the learning process and one of the main pedagogical tools used for diffusion of educational messages. However, their integration in the learning process still limited since they don't provide efficient features for monitoring and assessing learner (player) interactions and decisions without breaking the nonlinearity of the game in order to show them the consequences of their decisions, which has been proven in a previous comparative study [1].

In this paper we propose a generic approach for learners (players) tracking in order to assess their knowledge, skills, and abilities that may be learned during game play in serious games by taking advantage of in-game interaction such as the user behavior during the game and the interactions performed by learners (players) while playing. Our research work that aims to develop a learner (player) tracking system in serious games composed of different models representing the data sources, and other modules corresponding to the main processes charged with exchanging, analyzing, transforming, registering, and interpreting data based on non-centralized multi-agent architecture. Each module including one or more agents communicating with each other but each one acts according to his behavior independently from other agents. So the collaboration between these models and modules allow for monitoring, guiding and reporting on learner (player) progress during (Dynamic tracking) and after the gameplay (Post tracking) in order to identify their actual tasks and detect some typical errors systematically.

Keywords—Serious games, E-learning, Learner Tracking, multi-agent systems, trace-based systems, Indicators, Dynamic tracking, e-assessment, in-game interactions, Feedback, Trace, Soap.

I. INTRODUCTION

For a long time, educators tended to ignore games as a source of education, today the rapid development of games industry, their immense effect on the new generation and the existence of certain characteristics within them possibly advantageous for educational purposes have led to an increase in the interest shown by the educational community, principally in the last decade, in their use in the field of education. [2]

So thanks to the ubiquity of games, the widespread use of the Internet and the need to create more attractive educational practices have led to the emergence of serious games as a new form of education and training to become an essential part of the learning process and one of the main pedagogical tools used for diffusion of educational messages. Indeed, whatever the contribution and potential of serious games in the education many research questions arise. In particular, how can we assess the knowledge acquired by the learner through the game? And all the more so in video games there is usually no player tracking tools, except possibly an overall score, far enough finely reflect the learning of the player.

This paper focuses on how to facilitate the integration of serious games into educational settings by providing tools and features helping in monitoring and assessing learners' actions based on in-game interactions in order to show them the consequences of their decisions as well as to track and report on their progress in the context of game play. In this regard our proposed approach tries to overcome the limitations of serious games related to learner (player) monitoring, reporting and guiding during their progress in the game environment by providing models to manage learner errors representing the data sources of our system and four modules communicating with each other representing the main processes that exchange, analyze, transform, register, and interpret data. So our system is designed to track learners step by step during their progression to identify their actual tasks and detect some typical errors systematically.

The rest of the paper is organized as follows: section 2 presents the scientific context within which this research work, and highlight the treated issue. Sections 3 introduce a literature review in research work previously done treating the same problematic. The fourth section presents in a detailed way our proposed approach which is in turn divided into 5 sub-sections each of them explains a part of our approach. Finally, Section 5 presents our conclusions.

II. SCIENTIFIC CONTEXT AND PROBLEMATIC

There is a need in the current virtual environments used in educational context to be able to demonstrate evidence of learner progress which used to evaluate the performance of the learner as well as to help guide the learner and adapt the environment depending on the actions of the learner or even improve the design of the environment.

© 2014-15, IRJCS- All Rights Reserved

Tracking learner progress refers also to finding out and recording how learners are progressing in their learning, this type of monitoring is also called formative assessment. However monitoring learner progress is an aspect of assessment that can be difficult to manage well. That is why several studies have addressed the issue of automatic tracking of learner action in virtual environment based on several approaches such as the trace analysis and modeling the actions of the learner, These studies are often used to help understanding the user's behavior in several environments such as Virtual Worlds for Serious Applications, Interactive learning environments and serious games.
For serious games to be considered a viable educational tool sets, they must provide means of testing and monitoring of progress that will be used to evaluate the performance of the learner. This need has been proven in previous work that is interested in contributing to the knowledge of serious games as well as to promote their use in educational field Based on the metadata schema [3] to formally describe serious games and other comparative study. We have presented a Comparative Study to Develop a Tool for the Quality Assessment of Serious Games Intended to be used in Education [1] by examining pedagogical, Playful and Technical criteria of the game. So the need for tracking and assessment tools has been proven by applying the tool on some serious games used to help student learn computer programing such as: Prog&Play, Eeclone, TA Spring, and CoLoBot. Therefore the result showed that each one of these games has its own weak and strong points. But they all have unmet needs in the educational side especially the side that is interested in the assessment method adopted in the game also none of those games does allow tracking the learner during his progress.
But instead of assessing only the game, it is desirable to use the data from the tracking to help progress in the game for example by giving clues to the player. The learner also needs to understand why he / she has lost, what mistakes and how to fix it. These indicators are also useful for the tutor. Furthermore, these elements can also help the designer to improve the game by highlighting errors designs or contribute to a dynamic adaptation of the possible routes based game actions of the learner.

III. PREVIOUS WORK

In the last few years the use of games technology for learning has emerged to be based on understanding, independency, learners' empowerment and skills improvement. Thus several studies have addressed the issue of automatic tracking of learner actions in serious games in order to assess their knowledge, skills, and abilities that may be learned during game play.
An example of these studies is the so-called micro-adaptivity approach [4] which aims to track and assesses the player interactions in educational games. An approach that enables an educational game to intelligently monitor and interpret the learner's behavior in the game's virtual world in a non-invasive manner. The approach labeled micro-adaptivity was developed in the context of the ELEKTRA project (Learning Experience and Knowledge Transfer) which uses the Competence-based Knowledge Space Theory (CbKST) to provide a detailed domain model that includes a set of meaningful competence states as well as a set of possible learning paths. So The ELEKTRA game and its successor 80Days4 game tracks the player interactions and uses them to update the competence state represented in the CbKST for the learning domains provided in the games. However the introduced approach to micro adaptivity demands a significant computational load in processing assessment and probability updates. Future research must refine the theoretical and technical approaches to facilitate realtime computing.
Another example in [5] which we present an approach for assessing learner's progress and detect misconceptions in serious games. The learner assessment is done in real time during the game. The approach is based on an "expert" Petri net and domain game action ontology, Petri Nets are used to model the expert rules of the domain and to diagnose the non-compliance of these rules. The ontology represents the domain concepts and their equivalent in terms of game actions. So the Petri net models only game actions and describes the expert behavior in the game allowing the learner to acquire knowledge and then uses the reachability graph of the Petri net to track the learner in order to detect, in real time, the learner's misconceptions, improve learner assessment and provide an accurate feedback for both the learner and the instructor.
In [6] the authors treat the issue of tracking learner in serious games from different angle in which they use the already existing tracking and assessment tools of the LMS by integrating the <e-Adventure> educational gaming platform and the Learning Activity Management System (LAMS). So this study present the integration of games created with the <e-Adventure> educational gaming platform into Learning Activity Management System (LAMS). This integration allows teachers to use the information gathered during a game-play session to conduct the student through different activities of the learning plan or simply to collect more information that can be used for further assessment and tracking purposes. However this approach is interesting but requires the teacher a great effort of interpretation to analyze the provided game indicators.

IV. OUR APPROACH

Generally tracking learner (player) in virtual training environments requires a systematic approach to determine a person's achievements and areas of difficulty. In this paper we discusses tracking issues related to formative assessment in serious games and offers suggestions for a new approach that can be used to help tutors know how learners are progressing. This approach takes into account two main scenarios to track the interactions of the players.

© 2014-15, IRJCS- All Rights Reserved

In a dynamic way in which the learner receives immediate feedback during his progress in the game or through the trace analysis approach that allows to have a global view on the learner interactions as well as to diagnose his strengths, weaknesses and misconceptions by generating a set of indicators that will be analyzed later. ***(Figure1).***

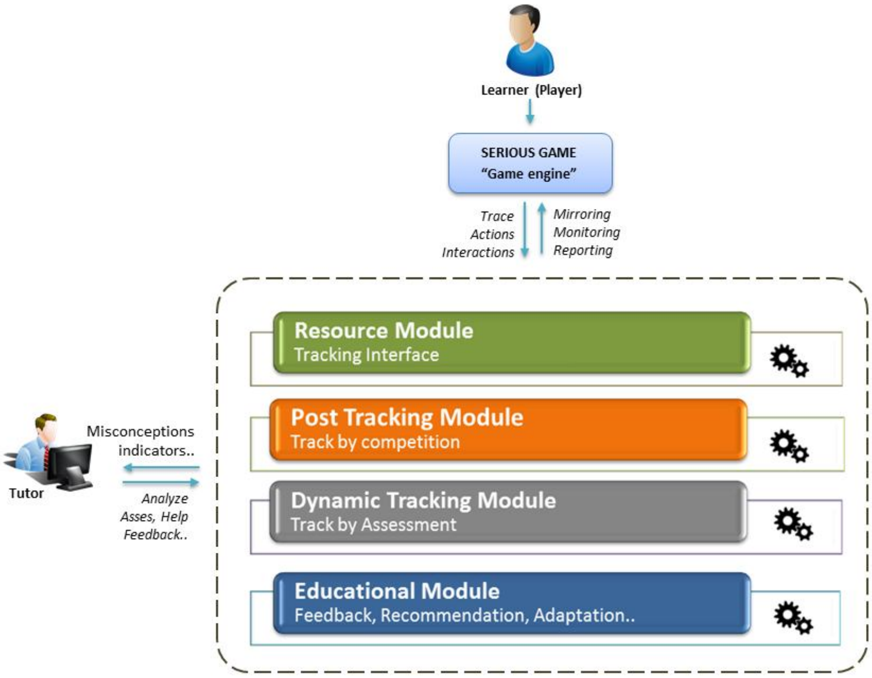

Fig 1: Overview of system modules.

Facing the problem of learner tracking in serious games has some challenges especially when it comes to dynamic tracking of learner interactions during his progress. Therefore, this paper discusses an enhanced and generic approach which provides means of testing and monitoring of progress that will be used to evaluate the performance of the learner. In the following we describe in detail the components of the main modules constituting our system and how they work.

A. Resource Module

This module acts as an intermediary between serious games and other modules. It represents a listener system that handles events and transmits messages between other modules and the game environment. So it retrieves what happens in the game environment by exchanging messages with it. Also in order to handles the communication between the tracking engine and game engine an interface is provided and used to call the tracking engine methods. The service is described using the Web Services Description language (WDSL) and uses the Simple Object Access Protocol (SOAP) for messages communication and transport ***(Figure2)***. For each soap message we should determine its elements such as SOAP:Envelope (Defines the start and the end of the message) / SOAP:Header (Contains any optional attributes of the message used in processing the message, either at an intermediary point or at the ultimate end point) / SOAP:Body (Contains the XML data comprising the message being sent) [7] as well as the message type if it's an action message (observable action) or a state message (a change in a state of the object in the game environment) and call the "tracking engine" in order to track and evaluate those actions based on predefined tracking rules in the" tracking model" and provides the pre-defined feedback associated to those tracking rules dynamically to the learner.

B. Dynamic tracking and Feedback

Serious games should have the possibility to track and monitor players progress during the game in order to assess their interactions and decisions without breaking the nonlinearity of the game. Moreover, serious games should provide valuable feedback.
As part of our approach, we suggest the dynamic tracking module which aims to provide instant feedback on the player's interactions in order to guide them during their progress in the game. We consider different aspects of the feedback in serious games by answering the following questions: when? What information? Which format? Since we consider our tracking approach related to formative assessment process, feedback should be immediate and possibly punctuated with explanations to help the learner understand the causes of his error. Feedback can also be carried out at the end of a level or the whole game to allow enough freedom to the player and not interrupting immersion in the game.

© 2014-15, IRJCS- All Rights Reserved

International Research Journal of Computer Science (IRJCS) ISSN: 2393-9842
Issue 4, Volume 2 (April 2015) www.irjcs.com

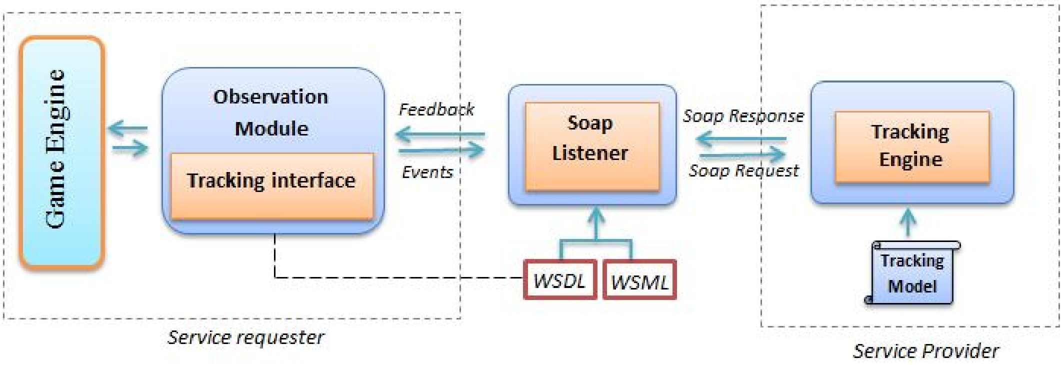

Fig 2: Representation of the Dynamic tracking module.

So the dynamic tracking module has been given the job to handle actions and events coming from players interactions and calls the "tracking engine" in order to assess those actions based on the predefined "tracking model" which is represented as XML based description in which the educator defines tracking rules based on the possible player actions and associated consequences as well as conditional matches by providing feedback - messages or actions - to the player within the game engine after detecting specific pattern by the tracking engine. For example, if the student fails to make one simple tasks that are part of a complex task, it reminds the student that all simple tasks to be exercised. Or if the student wants to open a door that has been opened (repeat error), we should explains to the learner that action has already been performed. Hollangel's [8] classification of erroneous actions can provide a more structured approach than simply creating a set of patterns that are adequate to create all possible mistakes. It contains a generic classification of errors types that may be committed by learners. This erroneous actions classification or phenotypes are structured in two classes: simple and complex phenotypes. Simple phenotypes represent differences (deviations) from a specified ideal path. The complex phenotypes are constructed from simple phenotypes. ***(Figure3)***

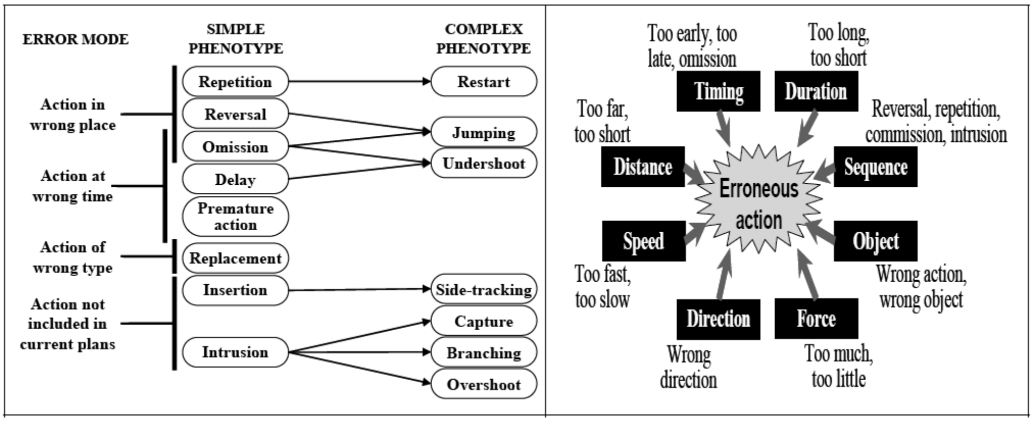

Fig 3: The classification of erroneous actions (right) & links between simple and complex phenotypes (left) according to [Hollnagel 1993b, 1998] [8]

The communication between the game engine and the tracking engine is managed through a web service described using the Web Services Description language (WDSL) and uses the Simple Object Access Protocol (SOAP) [7] for messages communication and transport in order to promote interoperability between modules and components of the system ***(Figure2):***

- The service requester (SOAP Client) will initiate the process by making a SOAP request. In this process, client will refer the WSDL file which resides in the SOAP server, to form a valid SOAP request.
- SOAP Listener will receive the SOAP request from the client and make sure the request adheres to the schema defined in WSDL
- Once the request is validated, listener will figure out appropriate method to call in the tracking engine with the help of WSML file.
- Tracking engine method will be called.
- The result from the tracking engine method will be packaged into SOAP XML response in accordance with the scheme defined in WSDL.

© 2014-15, IRJCS- All Rights Reserved

ISSN: 2393-9842
www.irjcs.com

C. Post tracking based on TBS

This module provides a qualitative approach to track and monitor learners progress in serious games based on their interactions. The users of any virtual training environments may leave traces which concern all their activities.
Several definitions of the concept of trace exist in the literature. These definitions differ according to its role and its use as in [9]. In our research, we are interested mainly in the digital traces of learner interactions that had no previous transformation or interpretation, this is called raw traces, primary or primitive. These traces undergo treatments to provide "the highest level" of information we call indicators. In games context, these traces are very voluminous and very heterogeneous. They are the results of various interactions between the learners (players) and the game environment. So our approach is based on the trace analysis of learners interactions stored in a log file.
For video games or serious games the log file is created for each play of the game it contains information about the role being played; the difficulty level selected; the time it takes for the user to make each decision; the number and type of actions taken in the game (such as AI action, security, political, or construction); and the resulting score [10]. The output from these logs is our main data source for post tracking in order to provide report based on player behavior and performance within the game as well as to analyze and track their progress in the game.
So in order to analyze traces collected from learners interactions in the game we will use a system based trace (TBS) as a generic solution to the problem of modeling and handling traces. We consider a trace based system (TBS) any system that implements the following three functions [11] ***(Figure4)***:

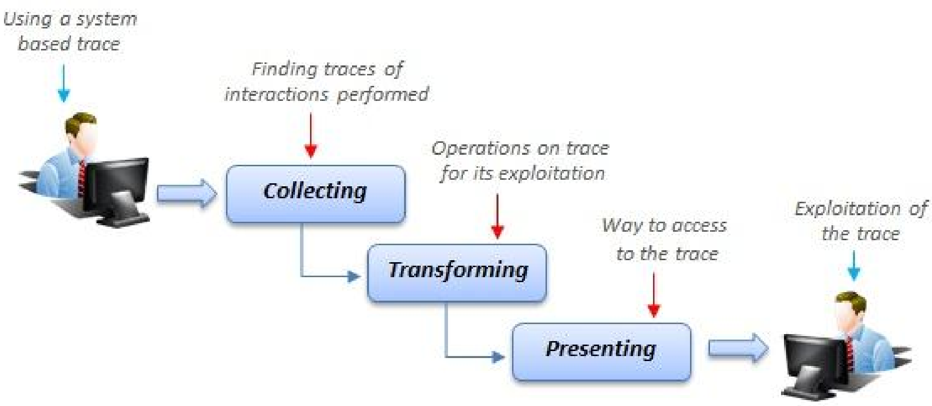

Fig 4: Generalization of trace based system

- ✓ ***Collection function:*** The collection of traces is an essential phase before any use of such traces. This phase will collect the traces resulting from the activities of learners. It is done immediately after the completion of each play of the game. These traces are recorded in the trace database ***(See Figure6).***
- ✓ ***Traces Transformation***: This generally consists of preparing the data to be reused as a trace. That means it allows the reformulation of modeled Traces already collected in order to bring the modeled trace to present a level of abstraction appropriate to the operation to be made of such as : application for further transformation, present the modeled trace to the user or calculation of indicators. There are three kinds of automatic transformations: selection-type transformations, Transformations rewrite type patterns, transformations by temporal fusion.
- ✓ ***Presentation Function*** The presentation of the trace allows a user to access the modeled traces and / or to order automatic processes and manual transformations. Thus the term presentation includes the features of visualization and interaction modeled traces to present. It gives practical access to the trace for its exploitation (a visualization, and if necessary fixed ways of interacting with the displayed trace) or provides as a service, traces to another application (assistants agents or independent display system for example).

Every system that implements these functions will necessarily introduce a number of storage features, consultation, calculation and handling traces. It will be in this case considered as a management system for a Database Trace.
Beyond the generalization that we just mentioned, the approach that we adopt propose particularly a formalization called modeled traces-based systems [12] ***(Figure 6)*** which provides a formal framework to implement the idea that the digital traces must be explicitly defined a priori in a trace model, So this approach is based on two key concepts the modeled trace and the trace model. The modeled trace which is the main is defined as the combination of a collection of observed to an explicit model of the collection called trace model, In other words a modeled trace is the combination of a digital traces and trace model to interpret the observed [12]. The trace model does not over determine the modeled trace of it, but sets its "vocabulary". ***(Figure5)***

© 2014-15, IRJCS- All Rights Reserved

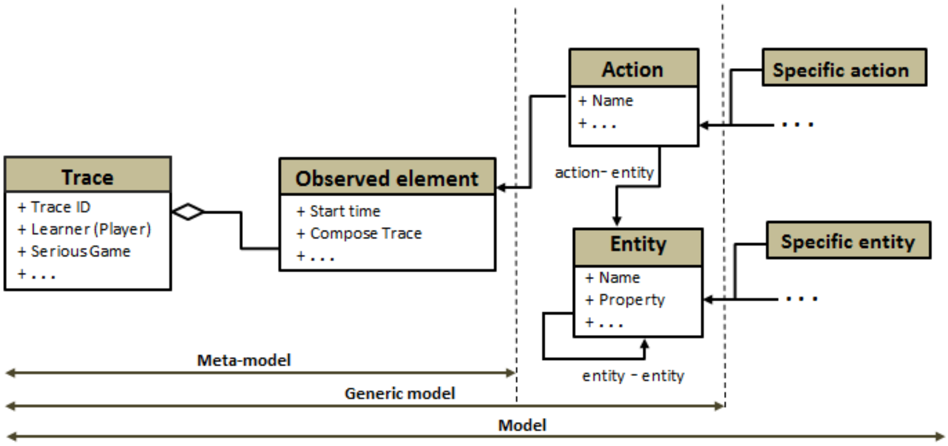

Fig 5: Our trace model (Require a specialization for each SG)

The proposed trace model is based on the TBS activity trace meta-model which has been refined into a generic trace model by considering actions performed within the activity as the main observed elements in the activity trace. We consider that the trace concerns a particular Learner and it is composed of a set of actions, whereas each action may manipulate a set of entities (objects or resources in the game environment). Furthermore, in order to be able to specialize the proposed trace generic model for all serious games, we consider that actions and entities may be organized in hierarchies and that structural relations are possible between entities (for instance, one resource may be part of another like the relationship between a file and its folder) [13] . So In order to enrich the content of the produced activity traces (log file) and enhance their presentation and visualization we should develop a semantic model for activity traces as an OWL-ontology that provides a specialization of our generic trace model.
Thus as shown in the figure below ***(Figure 6)*** the approach of the modeled traces that we adopt uses the general functions of a trace based system but with the means to manage the modeled traces associated with their models.

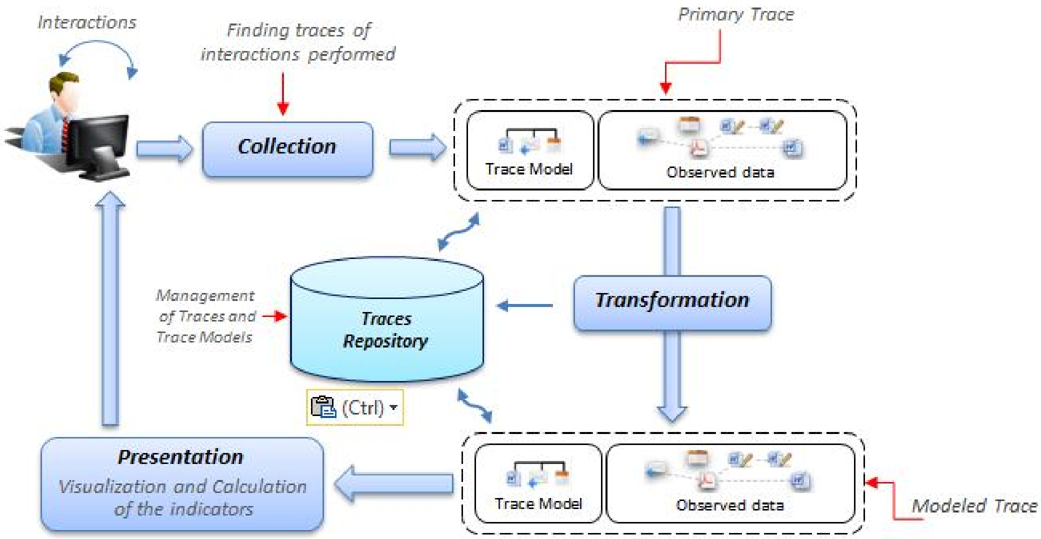

Fig 6: General architecture of M-TBS

The approach of the modeled trace based-system that we have adopted allows us in the first place the collection of the traces of the learner's interactions from logs that are generated after each part of the game. Handling and transforming these traces lead us to have a modeled trace based on a predetermined trace model that defines explicitly priori the elements of the trace. This process will culminate with the calculation of a set of indicators that will be used later in the educational module.

D. Educational Module

The educational module is an important component in the proposed approach, it occupies at supporting the learner during his progress in the game since it plays the role of a virtual tutor accompanying the learner in his training. In other words this module's main objective is to help and assist the learner (propose action to perform, display help, correct...) in order to show him the consequences of his decisions.

© 2014-15, IRJCS- All Rights Reserved

International Research Journal of Computer Science (IRJCS) ISSN: 2393-9842
Issue 4, Volume 2 (April 2015) www.irjcs.com

As we have previously explained, the traces must be transformed into data "highest level" so that their interpretation would be possible. These data are referred to as indicators. According to [14] we define the concept of Interaction Analysis Indicator as " a generally calculated variable or set using observed data, reflecting the mode of the process or quality interaction ". Several works were made to produce indicators based on the transformation of traces [11] others use data-mining techniques [15]. Some consider query languages as a tool to calculate simple indicators [16]. In the current work we use the transformation of traces stored in logs to produce indicators. For the calculation and modeling of indicators we will use the new version of the language UTL (Usage Tracking Language) presented in [17] to model the indicators in a capitalizable form, automated and reusable to provide relevant indicators for "tutor / instructional designer. This indicator calculation process is performed automatically and requires intermediate data (primary-datum, derived-datum) to generate the values of the indicators.
As shown in the ***(Figure 7)*** ***w***e can represent an indicator according to his information model defined in UTL [17] with three facet of the DGU model (*Defining* - *Getting* - *Using*):

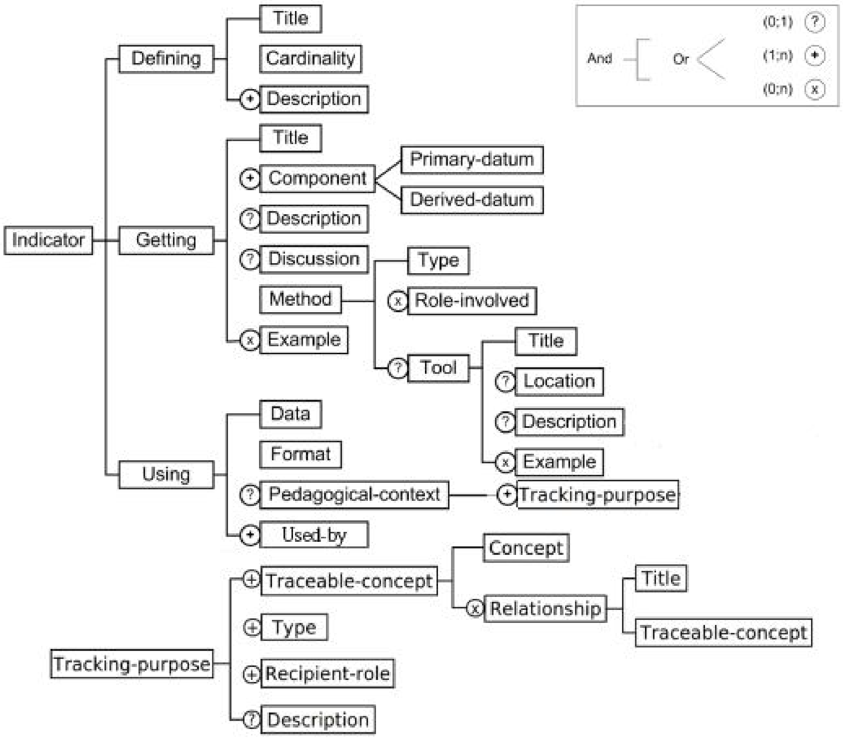

Fig 7: Information model of Indicator in UTL (Choquet, 2007) [17].

The *Defining* facet of the indicator is characterized by three elements: Title (Title), cardinality (Cardinality) representing indicator values and a detailed description (Description) of the indicator. The facet *Getting* can be also composed of a description and discussion to clarify the nature of the desired data and must include an element Method which can be human (indicate the Role-involved), semi-automatic or automatic (indicate the support-tool). The facet using is characterized by four elements: data element (Data) stores the value of the indicator, once calculated or determined, the pedagogical context element (Pedagogical-context) defines the observational objectives (tracking-purpose) of the indicator.

The educational module is called directly after the calculation and generation of indicators provided after trace treatments in order to analyze and select essential indicators that help in monitoring and guiding learners. These selected indicators will also be posted at the end of each part of the game – reporting on progress made – showing the consequences of decisions taken by the learner and all statistics concerning status of game progress.

In other words this module provides intelligent insights with reports and statistics on the progress of the learner. With these reporting and monitoring features, it provides detailed reports such as the timing (too early, too late, omission), Duration (Too long, too short), directions taken or movement made, chosen objects (Near, similar Object, Object unrelated) number of attempts to reach the solution, statistics on the status of game progress... Therefore the tutor can easily track and identify the progress made by each learner.

This module is also called by the dynamic tracking module when the tracking engine detects a new action that has no matches or predefined consequences in the tracking model, so the educational module is called in order to update the tracking model by defining a new behavior-pattern for the new action performed by specifying the action parameters (comparison type, manipulated object, specified value...).

© 2014-15, IRJCS- All Rights Reserved

Then the educator will be notified to define tracking rules based on possible interactions of the player. Moreover, feedback can be defined in the meta-description of the tracking rule as part of the consequences and the conditional matches.

E. General Architecture

The ***(Figure 8)*** shows a sequence diagram that emphasizes communication between the various modules in the context of access to game data and the production of information. The non-centralized architecture of our system is based on four modules communicating with the game environment: The Resource Module (RM), the dynamic tracking module (DTM), the post tracking module (PTM), the educational module (EM) in which each module acts according to his behavior independently from other modules.

The process is triggered from the resource module interface when it sends a request to track learner (player) interactions. There is two ways to track learner's interactions: dynamically in which we call the DTM that handles events and actions coming from game environment by generating the appropriate feedback for each action based on the predefined rule in tracking model or he calls the EM to update the model in the case of a new non predefined action. Secondly there is the PTM which allow treating traces of learners stored in logs and therefore generates indicators giving an overall view of the game play through tracking reports and statistic showing the progress made by learner. These tracking reports are generated by the EM based on indicators calculated by PTM, which will then be communicated to game environment through the RM module interface.

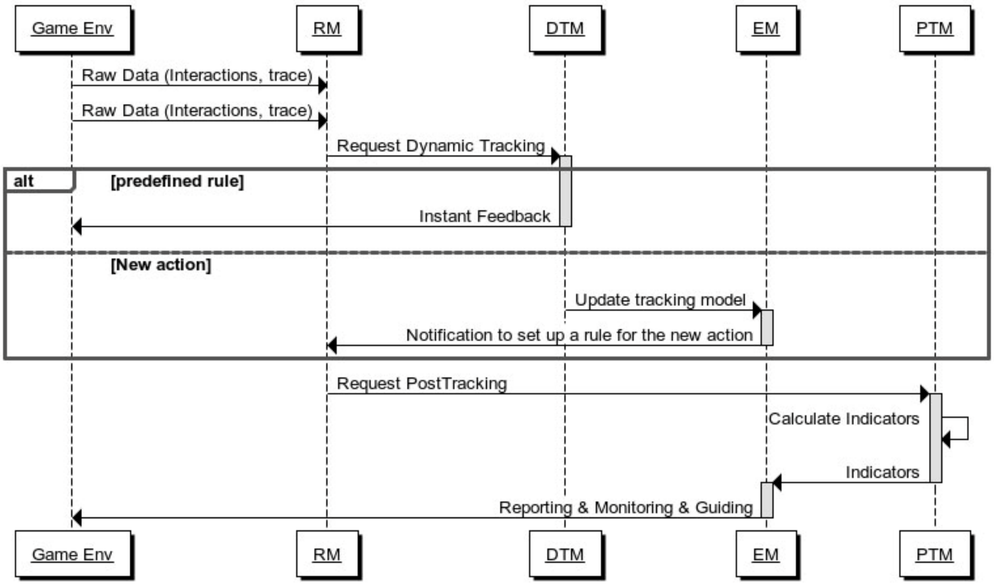

Fig 8: Sequence diagram illustrating the communication between the different modules.

V. CONCLUSION

In this article, we introduced our approach to set up a learner (player) tracking system for serious games which aims to monitor, guide and report on their progress during the gameplay step by step in order to identify their actual tasks and detect some typical errors systematically. Our system is composed of models representing source of data and modules dealing with two tracking scenarios. The first scenario can be dynamic in which the learner receives an instant feedback - messages or actions - after detecting specific behavior-pattern. The other possible tracking scenario is the post tracking which is based on the analysis of traces stored in logs and generates indicators showing the progress made by learner/player during the game. Our system is based on a non-centralized multi-agent architecture in which each module comprises one or more agents that communicate with each other and act independently.

The proposed approach builds on a pedagogical framework to provide tracking features for learners/players and feedback apart from the game engine giving educator more flexibility on defining tracking rules based on the learning objectives, the game context, and targeted audience. Also it gives more flexibility of having non-linear and intuitive learning paths within the game scenarios.

We intend in the future to experiment this approach with different serious games in order to evaluate the flexibility and reliability of the architecture as well as to enhance the tracking model to optimize the evaluation of players behavior. Among our perspectives is to use the mechanisms of plan recognition (PR) approach of El-Kechaïto [18] to improve errors recognition by generating a list of candidate plans and their associated errors used to interpret the behavior of the learner.

© 2014-15, IRJCS- All Rights Reserved

REFERENCES

[1] El Borji, Y., & Khaldi, M. (2014). Comparative Study to Develop a Tool for the Quality Assessment of Serious Games Intended to be used in Education. *International Journal of Emerging Technologies in Learning (iJET)*, *9*(9), pp-50.
[2] Ma, M., Oikonomou, A., & Jain, L. C. (2011). *Serious games and edutainment applications*. Berlin: Springer.
[3] El Borji, Y., & Khaldi, M. (2014). An IEEE LOM Application Profile to Describe Serious Games «SG-LOM». *International Journal of Computer Applications*,*86*(13), 1-8.
[4] Kickmeier-Rust, M. D., & Albert, D. (2010). Micro-adaptivity: protecting immersion in didactically adaptive digital educational games. *Journal of Computer Assisted Learning*, *26*(2), 95-105.
[5] Yessad, A., Thomas, P., Capdevila, B., & Labat, J. M. (2010). Using the Petri Nets for the Learner Assessment in Serious Games. In *Advances in Web-Based Learning–ICWL 2010* (pp. 339-348). Springer Berlin Heidelberg.
[6] Del Blanco, Á., Torrente, J., Marchiori, E. J., Martínez-Ortiz, I., Moreno-Ger, P., & Fernández-Manjón, B. (2010, October). Easing Assessment of Game-based Learning with and LAMS. In *Proceedings of the second ACM international workshop on Multimedia technologies for distance leaning* (pp. 25-30). ACM.
[7] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H. F. & Winer, D. (2000). Simple object access protocol (SOAP) 1.1.
[8] Hollnagel, E. (1993). The phenotype of erroneous actions. *International Journal of Man-Machine Studies*, *39*(1), 1-32.
[9] CRAM D., JOUVIN D., MILLE A. (2007). " Visualisation interactive de traces et réflexivité : application à l'EIAH collaboratif synchrone eMédiathèque ", Revue STICEF, Vol 14.
[10] Cannon-Bowers, J. (Ed.). (2010). Serious Game Design and Development: Technologies for Training and Learning: Technologies for Training and Learning. IGI Global.
[11] Settouti, L. S., Prié, Y., Champin, P. A., Marty, J. C., & Mille, A. (2009). A trace-based systems framework: Models, languages and semantics.
[12] Settouti, L. (2011). Systèmes à base de traces modélisées: modèles et langages pour l'exploitation des traces d'interactions (Doctoral dissertation, Lyon 1).
[13] Yahiaoui, L., Prié, Y., Boufaida, Z., & Champin, P. A. (2011). Redocumenting computer-mediated activity from its traces: a model-based approach for narrative construction. *Journal of Digital Information*, *12*(3).
[14] Dimitrakopoulou, A. (2004). State of the art on interaction and collaboration analysis.
[15] Hoppe, H. U. (2003). A web-based tutoring tool with mining facilities to improve learning and teaching. *Artificial intelligence in education: Shaping the future of learning through intelligent technologies*, *97*, 201.
[16] Merceron, A., & Yacef, K. (2004). Mining student data captured from a web-based tutoring tool: Initial exploration and results. *Journal of Interactive Learning Research*, *15*(4), 319-346.
[17] CHOQUET, C., IKSAL, S. (2007). Modélisation et construction de traces d'utilisation d'une activité d'apprentissage : une approche langage pour la réingénierie d'un EIAH. Revue STICEF numéro spécial, Vol. 14.
[18] El-Kechaï, N. (2007). Suivi et Assistance des apprenants dans les environnements virtuels de formation (Doctoral dissertation, Le Mans).

© 2014-15, IRJCS- All Rights Reserved