A Cloud Computing for the learner's usage tracks analysis

Mariem Chaabouni
INSAT/ISI
Tunis, Tunisia
mariem_chaabouni@hotmail.com

Mona Laroussi
INSAT
Tunis, Tunisia
Mona.laroussi@insat.rnu.tn

***Abstract*—We present in this paper the definition of a collaborative and cooperative platform, exploited through the Cloud, for analyzing learners tracks and managing indicators in educational scenarios. This paper describes the architecture and the design proposed for the platform, then it evocates the related security aspect. Finally, a test scenario is described to demonstrate the platform functionalities.**

Keywords-CEHL (Computer Environment of Human Learning); pedagogical indicators; tracks analysis; Cloud Computing; web services

I. INTRODUCTION

In the context of distance learning, the assessment of the learner teaching activity becomes difficult due to the lack of feedback to the tutor. The analysis of the learner usage tracks generated by learning tools during training sessions is a way for supervising and monitoring the distant learners.

The tracks analysis is a process executed on multiple steps: (i) Collection of the observation data; (ii) Treatment of these data; (iii) Interpretation of the obtained data. The treatment and interpretation of the learner tracks can be performed by generating pedagogical indicators which have as main objective the improvement of the learner activity perception. These indicators would help the teacher to easily interpret the individual or the group educational situation and evaluate performed sessions. Indicators are dedicated to assist tutors in reengineering of their pedagogical scenarios. Actually, this reengineering means the improvement, the control and the reworking of the learning process according to learner's requirements and learning environment variations. Indicators are susceptible to give pertinent information about the pedagogical scenario execution.

Examples of indicators in literature are the following: the collaboration level indicator [1], the division of labor indicator [2], interests of a page indicator [3], and many others proposed by multiple works. Researchers operating in this domain have dealt with several aspects related to this type of indicators. These researchers are particularly interested by the aspects of reuse and capitalization of educational indicators. In order to implement these aspects, works (e.g. [1], [4] and [5]) have introduced the design patterns concept to describe the pedagogical indicator in a unified way by proposing a model reusable by others (tutors and institutional designers). So the design pattern approach comes to respond to the needs of sharing and reusing of the followings: knowledge, skills and expertise related to tracks analysis generating indicators in education and learning context.

Our research operates in this context precisely in the pedagogical indicators engineering which becomes an active research field. Indeed, a large number of indicators have been proposed in the literature (cognitive, social, educational, technical, etc. [5]). Each indicator, requiring expertise and knowledge in many areas that affect learning, is defined in different environments. Additionally, systems performing trucks analysis and indicators calculation require significant resources in terms of storage (a large amount of indicators), power calculation (a frequent and sometimes complex indicator's calculation) and cost (storage and computing resources, maintenance, etc.).

To respond to the above problematic, the idea was to regroup and capitalize these indicators into a unique shared platform available in the Cloud. This provides a way to exploit indicators with a collaborative and intelligent manner.

Our objective is to define a sharable platform for managing pedagogical indicators in the Cloud. It is a collaborative and a cooperative platform for sharing a set of indicators defined and integrated by several participants. This platform offers an indicators database integrated and exploited by several designers/tutors operating in different learning environments. A tool for managing this database is also provided by the platform.

In this paper, we present in the first part our approach proposed for the platform managing pedagogical indicators on the Cloud. Then we present the architecture and design for the platform, we evocate the related security aspect and we end by a test scenario demonstrating the platform functionalities.

II. TOWARDS A COLLABORATIVE AND COOPERATIVE PLATFORM FOR INDICATORS MANAGEMENT SHARED ON THE CLOUD

With the use increase of the Cloud architecture and its introduction into various areas, some educational organizations begin to migrate to this architecture. This is related to the fact that many schools or institutes do not have resources and infrastructure to integrate advanced e-learning solutions. Cloud computing is the basic environment and platform of the future e-learning [7]. It has become a high technology because of its scalability, availability, extensibility and efficient use of resources. "Blackboard" and "Moodle" which represent the biggest actors in E-learning world have now some application versions oriented Cloud [8].

The objectives of the migration of E-learning environments to Cloud architecture are to facilitate the learning, encourage openness, share and reuse the educational resources. In our case, the shared educational resources are the pedagogical indicators design patterns. So we propose to deploy a platform offering a database of indicator patterns and a tool for managing this database. These are accessible by a large number of users in different learning environments (CEHL[1]) as illustrated in "Fig. 1".

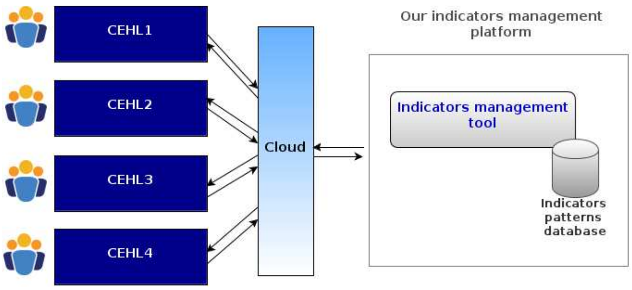

Figure 1. Approach for sharing the indicators management platform.

The indicators management platform is available on Cloud with SaaS mode, which means that it is managed and hosted in distant servers and its interfaces are available on the client side. The proposed tool is a browser-based application used as a service over the Internet, running on a flexible infrastructure.

This approach would support the aspect of reuse and sharing provided by the design patterns of educational indicators, and come out with other benefits. The main aspects we aim to attempt are the following:

1) *Share and reuse:* Ensure that the ressources of the platform are shared among multiple users in different CEHL by defining clear criteria for reuse. The shared ressources include the software (i.e. a tool for exploiting indicators and an indicator patterns database) and the hardware (i.e. computing power, large and secure storage, etc.).

2) *Interoperability and standardization:* Ensure that the platform components are interoperable. That means assigning the ability to function and to communicate with other systems by presenting uniform and standardized interfaces.

3) *Evolutivity and extensibility:* Define an architecture allowing the evolution and the extensibility of the platform. This encompasses the ability to easily extend the indicators database and also the ability to incorporate new features into the indicators management tool.

4) *High availability of resources and no servers maintenance:* Ensure the availability of the indicators database and the calculation tool to provide a good service quality. In addition the maintenance of servers is dedicated to the Cloud host.

5) *Simplicity and speed of access:* Provide speed and easy access to different features of the tool.

III. ARCHITECTURE AND DESIGN PROPOSAL FOR INDICATORS MANAGEMENT PLATFORM

We propose for our platform an open and a modular architecture shared on the Cloud. This ensures its reuse, interoperability, a high availability of resources and an easy way of use.

The architecture of the indicators management platform, shown in "Fig. 2", is composed of the indicator patterns database and the tool managing database, both deployed on the Cloud.

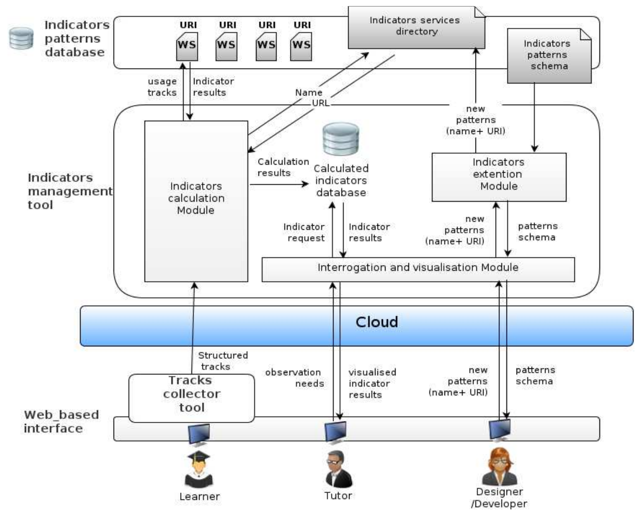

Figure 2. Detailed architecture for indicators management platform.

The platform's main objective is to exploit educational indicators facilitating the tutor perception of the situation of the monitored learner. This platform allows the indicators calculation and also the integration of new indicator patterns in an intelligent and collaborative way.

Three actors interact with the platform:

- *The Tutor:* transmits its observation needs to the tool which sends required indicator results;
- *The designer/developer:* adds new indicators to the platform according to a defined schema (the tutor and the designer/developer can be physically the same person);
- *The learner:* provides usage traces used by the tool (no access rights are granted to this actor).

The raw traces of the learner go through the "tracks collector tool" proposed in an existing work [9]. This tool is

[1] Computer Environment of Human Learning

responsible for collecting traces from educational devices and structuring them in the standardised format IMS-LIP[2].

A layer of RESTful Web Services (WS) is available on the Cloud forming the indicator patterns database. This consolidates the sharing, the extensibility of the database and its interoperability with other systems. The database contains the indicators managed by the platform stored as executable and capitalized patterns. So, a web service of an indicator pattern is an independent Cloud-based application able to execute the associated indicator function.

The indicators management tool is based on modules assisting actors in the exploitation of the shared indicators database. It allows accomplishing two main processes: The new indicator integration process and the indicator calculation process.

A. New indicator integration process

For the integration process of a new indicator pattern, the designer/developer begins by implementing the corresponding web service with any chosen programming language. He must follow the required indicator patterns schema provided by the platform. This schema describes exactly the format of requests and responses to WS that must be met by each new indicator pattern. This will enable the pattern to be integrated and used by the platform. After that, the designer deploys this web service on the Cloud and provides its name / URI [3] to the “indicators extension module” via the tool web interface. An “indicator services directory” accompanies the database for referencing available WS.

B. Indicator calculation process

For the process of calculating an indicator, the “Indicator calculation module” retrieves the structured traces provided by “traces collector tool”, and needed for the calculation process. These traces are sent to the relevant WS which performs the treatments. Then, the module retrieves the returned results and stores them in the “Calculated indicators database”. A tutor interrogates this database to obtain the indicators results.

C. Indicator patterns schema

The indicator patterns database is generic and can be grafted on any platform to generate educational indicators.

The “Fig. 3” models the "Indicator_pattern_webservice" interface and its related classes which describe the contract of an indicator pattern web service. This interface represents the schema that must be considered while implementing an indicator pattern WS.

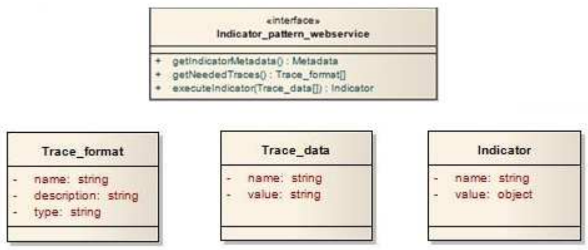

Figure 3. "Indicator_pattern_webservice" interface and related classes.

An indicator pattern modeled by a web service is called by a HTTP request containing the name of the method to invoke and the required parameters. After performing treatment related to the received request, the Web service can return a HTTP response containing:

- The metadata associated to the indicator pattern ***(getIndicatorMetadata():Metadata)***.
- The format of the learner usage traces needed to calculate the indicator ***(getNeededTraces():Trace_format[])***.
- The results of the indicator based on the trace data received on input ***(executeIndicator(Trace_data[]): Indicator)***.

Metadata, Trace_format, Trace_data and Indicator are a set of classes used by the interface.

1) Supporting indicator patterns proposed in existing works: The proposed platform is mainly characterized by openness and collaboration aspects. It is therefore conceived in a way allowing the acceptance of integration of different indicator patterns proposed by existing works. In this paper, we present an example which integrates the Reusable Indicator Patterns (PIR) proposed by Diagne [5].

Reusable Indicator Patterns are defined by an indicator function f() and a set of metadata [5]. We use these metadata to define the class *Metadata* (shown in "Fig. 4") forming the return type of the function *getIndicatorMetadata()*.

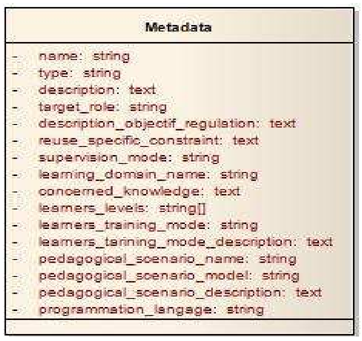

Figure 4. Metadata class based on Reusable Indicator Patterns.

2 IMS Learner Information Package: a standard based on XML, for exchanging learner’s data between several systems.

3 Uniform Resource Identifier.

Other indicator patterns can be considered in the platform such as UTL patterns [10] and the collaboration indicator patterns [1]. Therefore our work is restricted to the PIR pattern.

IV. INDICATORS MANAGEMENT PLATFORM AND SECURITY

The choice to deploy our indicators management platform on Cloud led us to deal with the data security aspect. This is why we mainly focus on the data confidentiality.

The Cloud applications require strong confidentiality in the communication protocol used to access Web resources. Traces learners must be secured for reasons of confidentiality and protection of learner privacy.

The RESTful web services receive learner usage tracks in order to calculate the corresponding indicators. So it is necessary to ensure the confidentiality of these transmitted tracks. Many studies have worked on the security aspect of the RESTful web services. We propose to use in our approach the "REST security protocol" defined by Serme [11] which is a protocol designed to secure the RESTful services communications.

The "REST security protocol" ensures data confidentiality by encrypting their content. This protocol operates at the message level by adding HTTP-headers to transmit metadata.

V. FIRST TEST SCENARIO

We present in this section a test scenario of our indicators management platform in order to demonstrate the above mentioned features. At the moment of writing, our platform is still on the development phase.
For the implementation and the deployment of the platform in the Cloud, we use the “Google App Engine”[4].

The scenario consists of implementing a new indicator pattern as a web service deployed on the Cloud, integrating it into the indicators management platform and then using it by the tutor. To perform this scenario, we opt for the indicator “Participation Percentage (PART)” proposed by Dimitrakopoulou [2]. The PART function is mentioned in “Eq. (1)”.

$$PART\ (ti) = Agents\ (ti) / TotalAgents \quad (1)$$

In our case, *PART(ti)* measures the participation level of a selected group of learners. *Agents(ti)* represents the number of different learners of a group have posted at least one message during ti time slot. *TotalAgents* represents the total number of learners collaborating on the group.

The “Fig. 5” illustrates the proposed process of the integration and the use of a new indicator pattern. This process matches the different phases of the scenario.

[4] A platform offering users the ability to build and host web applications on Google's infrastructure

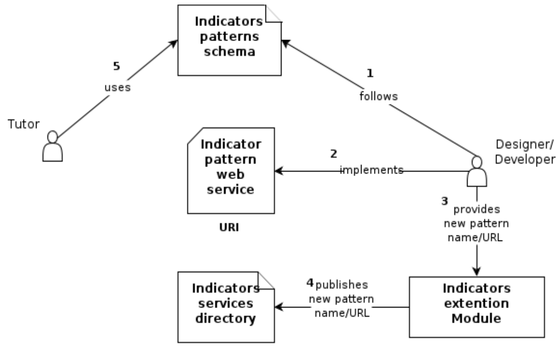

Figure 5. Process of the implementation and the integration of a new indicator pattern.

Phase 1 (follows): A designer/developer, detecting a new observation need on the participation of groups of learners, decides to implement a new PART indicator pattern. He begins by consulting the schema patterns of indicators that must be followed in the Web service implementation. This pattern is accessible via the web interface available for the designers. As shown in “Fig. 6”, we choose to write the indicator patterns schema in SMD (Service Mapping Description) [12]. The SMD, which consists of a notation proposal based on JSON (JavaScript Object Notation), can be used for describing REST web services.

```
{
    "SDMVersion": "2.0",
    "transport": "REST",
    "envelope": "URL",
    "parameters": [...],
    "services": {
        "getIndicatorMetadata": {
            "type": "method",
            "transport": "GET",
            "_comment": "Returns the metadata of the indicator",
            "parameters": [ ],
            "returns": {
                "type": {
                    "name": {"type": "string"},
                    ...
                    "description ": {"type": "text"},
                    "domain": {"type": "string"},
                    ...
                }}
        },
        "getNeededTraces": {
            ...
        },
        "executeIndicator": {
            ...
           "parameters": [
                {
                    "type": {
                        "name": {"type": "string"},
                        "value": {"type": "string"
                    }}}
            ],
            "returns": {
                "type": {
                    "name": {"type": "string"},
                    "value": {"type": "JSONObject"}
          }}}}}
```

Figure 6. Extract of the indicator patterns schema.

Phase 2 (implements): According to the provided schema, the designer/developer implements the RESTful web service representing the new indicator with the preferred language. Subsequently, he deploys the developed web service on Cloud in the preferred host and keeps the service URI.

For instance, the following URI: **[http://part-indicator-webservice.appspot.com/rest/part/executeIndicator/20/1]** returns the following JSON result: **{name: PARTICIPATION PERCENTAGE (PART), value : 0.6}**

This result represents the value of the indicator PART equals to *0.6*. The URI returns also the equivalent result on XML. These various returned formats favorites interoperability.

Phase 3 (provides new pattern): To integrate this new indicator in the platform, the designer/developer adds the deployed web service URI through the available web interface (see “Fig. 7”).

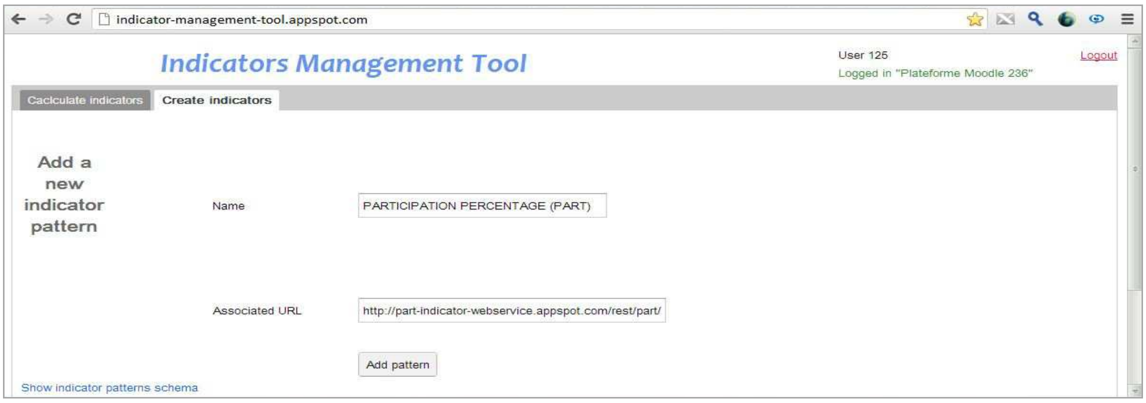

Figure 7. Web interface overview for adding new indicator.

Phase 4 (publishes new pattern): The “calculating module” publishes the Name/URI of the new indicator pattern in the services directory in order to be visible by the platform. An example of a services directory is shown in “Fig. 8”.


```
{"Entries": [
        {
            "name": "Collaboration level Indicator",
            "uri": "http://collaboration-level-indicator-
webservice.appspot.com/"
        },
        {
            "name": "Participation percentage PART ",
            "uri": "http://part-indicator-
webservice.appspot.com/rest/part/"
        },
        {
            "name": "Division of labor Indicator",
            "uri": "http://division-of-labor-indicator-
webservice.appspot.com/"
        }
    ]
}
```


Figure 8. Example of a services directory.

The JSON object in the red box shown in “Fig. 8” represents the entry of our added indicator. Except from the PART indicator URI that is useful, others are purely fictitious and URIs are added for demonstration purpose. The mentioned indicators names are extracted from an existing work [2].

Phase 5 (uses): A tutor can consult the list of available indicator patterns in the platform displayed in a web interface accessible via Internet (see “Fig. 9”). He selects the indicator PART from the list and provides the parameters needed for the calculation process. In our case, the parameter is the group of learners the tutor wants to observe.

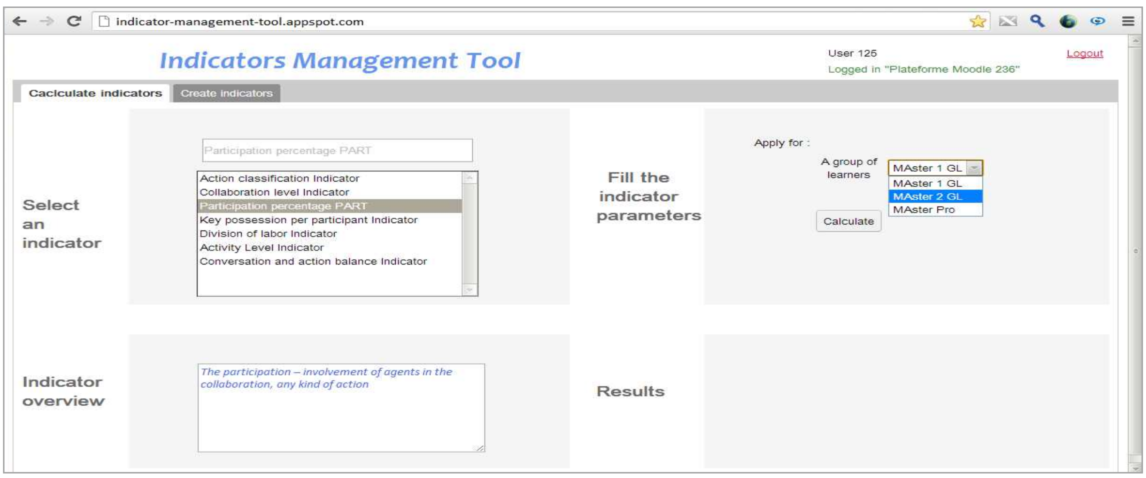

Figure 9. Web interface overview of the indicators search and calculation.

VI. CONCLUSION AND PERSPECTIVES

This work operates in the CEHL domain and particularly in learners tracks analysis performed through generating indicators. It aims to help designers/tutors in reengineering their pedagogical scenarios.

In this context, this paper presents a shared platform in the Cloud provided to the designers/tutors acting in various learning environments. This platform allows the designers/tutors to firstly gather a large number of indicators in a reusable, extensible and interoperable database, and to secondly equip them by a tool for managing this database. In other words, it is a collaborative and cooperative platform allowing the exploitation of a set of educational indicators defined and integrated by several participants.

Our proposal is a scalable and an open architecture for integrating indicator patterns by several designers/tutors in different educational environments. These indicator patterns are deployed in the Cloud and designed as reusable web services. This allows sharing experience, knowledge and expertise of the designers in various fields including computer literacy, education, psychology, etc.

The choice of Cloud architecture for the proposed platform provides the following benefits: (1) centralization of reusable indicators, (2) sharing and reuse of the hardware and software resources of the platform, (3) scalability and extensibility of the platform, (4) high availability of the platform resources, and (5) a quick and easy use of the indicators by tutors desiring to have a feedback on their pedagogical scenarios execution.

As perspectives of this work, we can envisage a set of improvements related to our indicators management platform like following:

- Integrate other indicator patterns proposed in the literature (e.g. UTL patterns [10], collaboration indicator patterns [1]).
- Provide advanced search functionalities for the indicators apart from the search by name. For instance, we can add a search by: type of indicators, operating domain, on-line publishing date, etc. These search functionalities are provided to the tutor through the web interface.

REFERENCES

[1] É. Gendron, “Framework for the elaboration of collaboration indicators from activity traces,” PhD thesis, University Claude Bernard Lyon 1, 2010.

[2] A. Dimitrakopoulou, “State of the art on interaction and collaboration analysis,” 2004.

[3] N. Bousbia, and J.M. Labat, “Perception de l’activité de l’apprenant dans un environnement de formation sémantique du parcours de l’apprenant,” in Actes de la conférence EIAH 2007, 2007.

[4] D.P.T. Ngoc, “Spécification et conception de services d’analyse de l’utilisation d’un environnement informatique pour l’apprentissage humain,” PhD thesis, Université du Maine, 2011.

[5] F. Diagne, “Instrumentation de la supervision par la réutilisation d’indicateurs : Modèles et Architecture,” PhD thesis, Université Joseph-Fourier- Grenoble I, 2009.

[6] S. Iksal, and C. Choquet, “Usage analysis driven by models in a pedagogical context,” in Workshop on Usage Analysis in Learning Systems, In 12th International Conference on Artificial Intelligence in Education (AIED 2005), Amsterdam 2005., pages 49–56, 2005.

[7] H. Ma, Z. Zheng, F. Ye, and S. Tong, “The applied research of cloud computing in the construction of collaborative learning platform under e-learning environment,” in System Science, Engineering Design and Manufacturing Informatization (ICSEM), 2010 International Conference on, volume 1, pages 190–192. IEEE, 2010.

[8] P. Pocatilu, F. Alecu, and M. Vetrici, “Measuring the efficiency of cloud computing for e-learning systems,” WSEAS Transactions on Computers, 9(1) :42–51, 2010.

[9] M. Ben Sassi and M. Laroussi, “ Vers une modélisation standardisée des traces des apprenants,” in frantice.net, Numéro 5 - September 2012.

[10] D.P.T. Ngoc, S. Iksal, and C. Choquet, “Re-engineering of pedagogical scenarios using the data combination language and usage tracking language,” in Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on, pages 506–510. IEEE, 2010.

[11] G. Serme, A.S. De Oliveira, Y. Roudier and J. Massiera, “Enabling message security for RESTful services,” in ICWS 2012, 19th IEEE International Conference on Web Services, June 24-29, 2012, Honolulu, Hawaii, USA, Honolulu, ÉTATS-UNIS, 06 2012. [Online]. Available : http://www.eurecom.fr/publication/3739

[12] M. A. Chatti, M. Jarke, and M. Specht, “PLEF: a conceptual framework for mashup personal learning environments,” IEEE Learning Technology Newsletter, vol. 11, no. 3, 2009.

[13] C. Choquet and S. Iksal, “Modélisation et construction de traces d’utilisation d’une activité d’apprentissage : une approche langage pour la réingénierie d’un eiah,” Revue des Sciences et Technologies de l’Information et de la Communication pour l’Education et la Formation (STICEF), 14, 2007.

[14] M.A.H. Masud, and X. Huang, “An e-learning system architecture based on cloud computing,” system, 10 :11.

[15] A. Mikroyannidis, “A semantic framework for cloud learning environments,” 2012.