Model for Recording Learning Experience Data from Remote Laboratories Using xAPI

José Pedro Schardosim Simão[1 2], Lucas Mellos Carlos[2], Hamadou Saliah-Hassane[3],
Juarez Bento da Silva[2] and João Bosco da Mota Alves[2]
[1]Federal Institute of Santa Catarina - Tubarão, Brazil
[2]Federal University of Santa Catarina - Araranguá, Brazil
[3]University TELUQ - Montreal, Canada
jose.simao@ifsc.edu.br, lucas.mellos@posgrad.ufsc.br, hamadou.saliah-hassane@teluq.ca,
juarez.b.silva@ieee.org, joao.bosco.mota.alves@ufsc.br

***Abstract*—This work proposes a model for the recording of learning experiences in experimental environments supported by remote laboratories, based on the xAPI specification. Using the Experience API (xAPI), a learning experience registered in a statement, in which an 'actor performs an action on an object'. Initially, a general and specific vocabulary was defined to record activities in remote laboratories. Then, a prototype was developed using the remote labor management system RELLE, integrating a set of laboratories to xAPI, using the previously defined vocabulary. The data generated by the use of the laboratories is saved in an open source application that validates and saves each generated statement. The model described in this work can be applied in different platforms of remote laboratories, and the generated data used for different purposes, be they pedagogical, administrative or technical.**

***Index Terms*—remote laboratories, learning experience, educational data mining, Experience API**

I. Introduction

Online laboratories, technological tools that allow students to perform experimental activities through computational devices, have often been used as a complement to traditional experimentation practices and in other cases, as distance courses, as an alternative to face-to-face laboratory activities. These capabilities allow students and teachers to conduct laboratory practices in a more flexible and interactive way, offering different experiences considering the various possible configurations, and having the same learning potential as traditional laboratories [1].

Online laboratories are usually classified in (1) virtual laboratories, based on mathematical models and simulation accessed through the internet, (2) remote laboratories, where a real device is controlled by the user through the internet, and (3) hybrid laboratories, which characteristics common to both previous models [2].

The first reports about the development and use of remote laboratories have more than two decades. The first implementation of a robot remotely operable over the internet was made in 1994, part of the Mercury Project [3]. In 1995, a remote robotic arm controlled through the Internet was developed, with a client that included video streaming and collaborative tools [4]. In 1998, a set of experiments from a remote chemical engineering laboratory, which allowed for batch and interactive control of the experiment through a web page, was presented by [5]. Predko [6] presented a remote debugger for the 8051 microcontroller, which was the first remote experiment developed at RExLab in 1997 [7], and described at that time as "one of the most interesting uses of the Internet".

Since then, the use of this technology has become popular and projects using remote laboratories have been developed in primary education [8], secondary [9] and higher education [10], [11], in disciplines related to science, engineering and technology. In a survey carried out by the Go-Lab project, [12] indicate that among the main portals and federations globally, there were about 1565 virtual and remote laboratories in 2014.

Several projects have been carried out with the aim of developing and popularizing support for remote laboratories. Some initiatives in the academic world may be highlighted: iLab [13], LabShare [14], VISIR [15] and WebLab-Deusto [16] and others such as LabsLand [17] and Labicom [18], are companies created to provide services related to remote laboratories. Likewise, other initiatives have focused on the development of repositories that index resources available in different institutions, such as Lab2Go [19] and LiLa [20].

In addition, remote labs were considered as one of the top five changes in 100 years of Engineering Education by the IEEE Proceedings' 100-Year Special Edition, regarding the impact of Information Technology and Communicationso [21].

With the popularization of remote laboratories, a considerable number of educational institutions have used the tool as support for experimentation activities in STEM disciplines. Consequently, a large volume of data is generated from the use of laboratories.

Data collection enables learning systems to implement Learning Analytics (LA), which are useful for measuring, analyzing and reporting on students in order to optimize their learning [22]. In this sense, there is an inclination of industry and entities, such as the IEEE groups: P1876 - Networked Smart Learning Objects for Online Laboratories [1] and ADB - Industry Connections Actionable Data Book [2], to indicate

[1] http://sites.ieee.org/sagroups-edusc/

[2] http://standards.ieee.org/develop/indconn/adb/index.html

the use of the Experience API (xAPI) specification to register experiences when it is mediated by technological tools.

II. Experience API

Experience API provides a standard for collecting data on events linked to the learning experience. It is designed to support the standardization and collection of distributed learning activities, both formal and informal, allowing easy discovery of learning behavior, and making it possible to formalize, store and retrieve learning experiences in virtual environments [23]–[25].

The development of xAPI was funded and supported by ADL (Advanced Distributed Learning Initiative), also responsible for its predecessor SCORM, a standard designed to ensure data interoperability between learning management systems (LMSs) and learning objects. The specification was developed by Rustici Software as part of the TinCan [3] project, which aimed to: (i) improve interoperability between e-learning systems that collect and exchange data of student learning, and (ii) overcoming the limitations of SCORM [25].

The stored data can be used by teachers to monitor students' progress in general subjects or specific skills, facilitating their evaluation and evaluation of the educational object. [26] presents the xAPI stakeholders as:

- **Activity Providers (AP):** create data in the xAPI format and send it to LRS. APs are systems and applications where learning activities and events occur, such as content and learning portals, applications, and other systems. Multiple Activity Providers may be sending data to an LRS at the same time.
- **Learning Record Stores (LRS):** databases that verify if the entry corresponds to the xAPI specification, storing all valid data for recovery by Activity Consumers, or by administrative users who wish to access the data for analysis.
- **Activity Consumers (AC):** systems similar to Activity Providers (an AP can behave as AC depending on the context) as they are generally systems and applications that modify the user experience based on xAPI data. This can be an LMS that ”controls” a complete learning activity, as this activity appears in the LRS, or something more complex, such as a rating table, a badges system, or learning content.

There are more than 150 applications that already export data in the xAPI format natively or using third part extensions. These include LMSs (such as Moodle, Blackboard and Totara), Content Management Systems (Wordpress and Drupal for example), authoring tools (such as Storyline, Captivate and Reader), social learning systems (such as Curatr), mobile applications (such as Tappestry or the Learning Locker Acapp app), performance support tools (such as Red Panda, the Trek Learning experience system, or xAPI Apps), assessment platforms (such as TAO or Question Mark), among others [26].

A. Statements

A learning experience in xAPI is tracked and formatted in a statement, in which an actor performs an action on an object (actor + verb + activity + additional properties) [27]. The figure 1 presents a simplified schema of a statement.

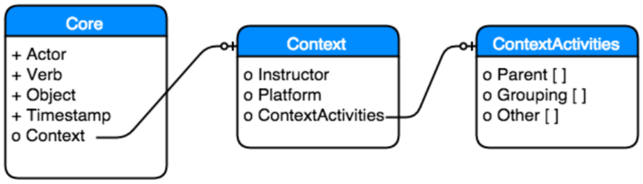


Fig. 1. Simplified schema of a xAPI statement [25]

The actor is the agent responsible for performing the task to be registered, and may represent an individual agent (person or system) or a group, which in a statement is represented by a Functional Inverse Identifier (IFI) - usually an address email or OpenID URL [27].

The verb describes the action taken during the learning experience. The xAPI used to specify a list of verbs, which became part of the initial versions of the specification; however, now the process of creating verbs defined so that communities of practice can establish meaningful verbs for their members and make them available for use by anyone. The only exception is the reserved verb “voided”, used to invalidate a statement [27].

The object defines or in which the action was performed - an activity (“George wrote a writing about football”), an agent/group (“Isabel interviewed Joe”), a substatement, George wrote an essay on ’Isabel interviewed Joe’ ”).

The specification describes the format for representing learning activities described in JSON. The data format of the xAPI statement is based on WC3 Activity Streams 1.0, with changes made to include results and context of an activity [25].

The specification presents also a RDF-based data model (Fig. 2) the constraints between the data are specified in natural language. In addition to the common actor, verb and object information, a statement can also detail the outcome of the event or the conditions under which it was performed by means of the result and context properties, include attachments that are part of the learning record, and the property of authority, which is used to identify the agent or group that is claiming that this statement is true. Four other properties can be used: uuid, timestamp, stored and version, are data properties that describe the UUID identifier assigned to the statement, the moment the event occurred, the time it was stored in LRS, and the xAPI Version, respectively.

This statement is then stored in LRS, and in order to avoid data problems being tampered with or altered after they have been sent, the Statement API is immutable, that is, it is not possible to edit or delete a statement after it has been sent, only be invalidated by means of a statement with the verb “voided” [26].

[3] https://scorm.com/tincan/

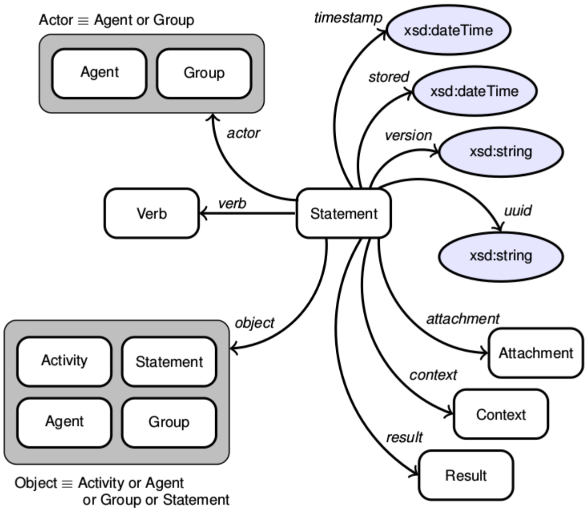


Fig. 2. Semantic network of a statement model [28]

B. Vocabulary

A vocabulary is a list or collection of terms that are used by a community of practices in order to label or categorize information in a particular domain.

Using a vocabulary ensures that everyone uses the same word to mean the same thing. Vocabulary in xAPI can use a single list (dataset) or multiple lists of specific terms selected for use. Vocabulary data sets should be cured and organized according to groups of verbs and/or activity types [30]

Communities of practice may publish new sets of vocabulary data or terms of reference from existing vocabulary sources. Each of the terms in a vocabulary data set must have or refer to a unique IRI. For example, the verb ”satisfied” is identified by IRI ‘https://w3id.org/xapi/adl/verbs/satisfied. ’Each vocabulary data set must also have a unique IRI and generally follows a design pattern consistent with the IRI path of the vocabulary terms it contains [30].

III. Proposed Model

This model proposes to record data on learning experiences in remote laboratories using the Experience API, in order to store the interactions between users and laboratories, considering their multi-directional flow.

Records of student sessions in laboratories accurately and contextually punctuate the moment the user accessed the lab, which commands he sent, what the laboratory returned, and the conditions under which the user logged out. These data are described in statements that are validated and stored in an LRS, from which they can later be retrieved.

The RLMS (Remote Laboratory Management System) acts as a mediator, since communication is not normally done directly between the user and the laboratory. In addition to the scenarios where he RLMS intermediates the communication between laboratory and LRS, are considered scenarios where smart devices based remote laboratories also concentrate the sending of statements to the LRS. The components of the model are shown in Figure 3.

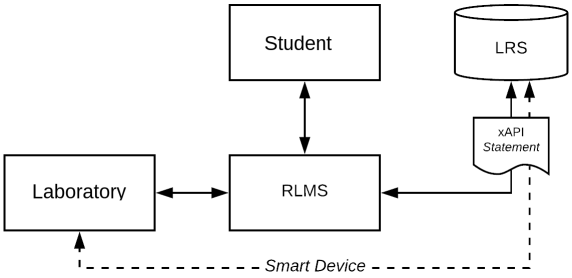


Fig. 3. Architecture of the Model

IV. Application Scenarios

A. Generic Remote Laboratories

Firstly, we consider a scenario where a generic lab is managed by an RLMS, as described in Figure 4. When a user accesses the lab, the first step is their authentication, which allows the RLMS to identify the student, and provide data for the statements in which he is the actor.

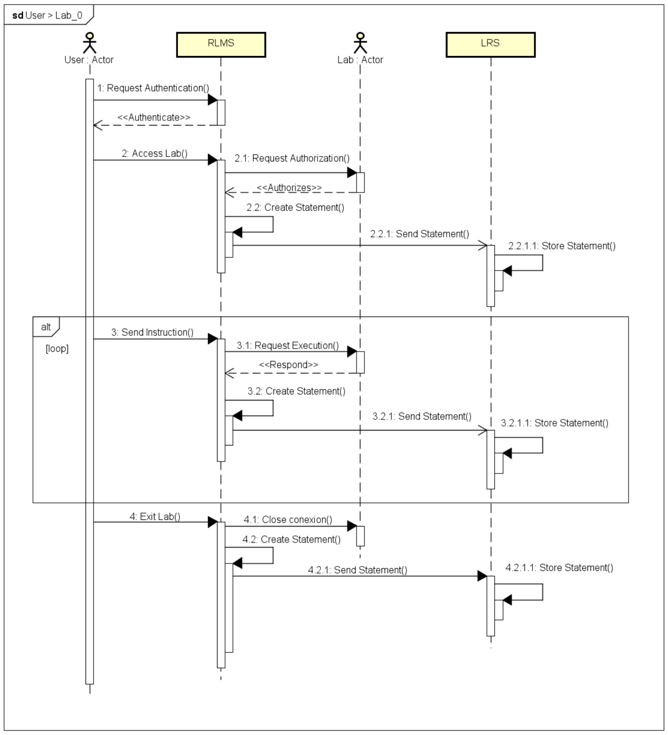


Fig. 4. Sequence Diagram of ’Access to a Laboratory’

Once authenticated, the user accesses the lab page/interface, generating a statement, and awaits authorization. From the issued authorization and the beginning of the student’s session in the laboratory, a statement is sent describing the beginning of the session.

From the beginning of the session, the interactions between student and laboratory occur in a loop and asynchronously, generating a statement for each interaction. When sending a command to the laboratory, a statement is recorded where the student is the actor, and when the laboratory sends some data in response, there is a new record, this time indicating the laboratory as an actor.

At the end of a session a statement is sent to the LRS, informing if the user has finished or left the session in the laboratory. This and other statements in this scenario are constructed in RLMS using the vocabulary and structures proposed in this model, and then sent to LRS.

B. Smart Devices

The Smart Device specification provides remote laboratory interfaces to clients and external services through well defined services and internal functionalities, dissociating client and server. A service represents, for example, a sensor or actuator exposed to the external world (for example, a client) through an API, being completely described through metadata, so that a client can use it without further explanation. A feature is internal behavior of the Smart Device, and there may be communication between internal functionalities and applications of external clients or services through intelligent device services [31].

Considering the ability of laboratories developed using this specification to provide such services and description of their components, the record of their statements can be made directly from the laboratory, without the intermediary of an RLMS. This model ensures greater accuracy in the time recorded in the statement while decreasing, but it would make it difficult to record activities performed in laboratories that are shared by multiple clients.

V. Data Model

In this section are presented the main properties used to describe the learning experiences conducted in remote laboratories.

A. Actors

The 'actor' property must be defined according to the xAPI specification. When a lab is accessed, RLMS must provide its data (with the exception of laboratories that use the smart device specification, which provides its own metadata), as well as authenticate and provide user data as well. The description of an actor is shown in Figure V-A.

```
"actor": {
"name": "Jose Simao",
"mbox": "mailto:josepedrosimao@gmail.com",
"objectType": "Agent"
}
```

Fig. 5. Example of the description of an actor in a statement

In this model, both students (individuals or groups) and online laboratories will be considered as authors. The description of a laboratory as an actor is shown in Figure V-A.

```
"actor": {
"name": "Means of Heat Propagation",
"openid": "http://relle.ufsc.br/labs/5",
"objectType": "Agent"
}
```

Fig. 6. Example of lab as an actor

B. Verbs

The verbs specified in this model are used to describe the actions of each statement and are listed in Table V-B.

TABLE I
Verbs used in the model

Vero	Use	IRI
Launched	Mandatory	http://adlnet.gov/expapi/verbs/launched
Initialized	Mandatory	http://adlnet.gov/expapi/verbs/initialized
Viewed	Optional	http://id.tincanapi.com/verb/viewed
Set	Optional	https://example.com/gtmre/verbs/set
Moved	Optional	https://example.com/gtmre/verbs/moved
Upload	Optional	https://example.com/gtmre/verbs/uploaded
Compiled	Optional	https://example.com/gtmre/verbs/compiled
Exported	Optional	http://activitystrea.ms/schema/1.0/return
Returned	Optional	http://activitystrea.ms/schema/1.0/return
Abandoned	Optional	http://adlnet.gov/expapi/verbs/abandoned
Exited	Mandatory	http://adlnet.gov/expapi/verbs/exited

Some verbs of this model describe general actions related to access and session control (initialized, exited, exported), and others directly reflect the interactions between laboratory and student, describing exactly the actions related to specific sensors and actuators. The verb 'moved', for example, can be used to describe commands sent to different engines, or to describe the use of a visual programming interface, such as the lab block.ino [4].

C. Objects

The object defines the element in which the action was executed, and in this model, when we consider the communication between student and laboratory in bidirectional way, we need to specify objects of two types: activity and agent.

In cases where the student is the actor and performs an action under an lab, the object of the statements will be of the activity type, using the type of online laboratory activity defined in this model.

In these statements, the object definition points to the laboratory description - name, type of activity and id, indicating which sensor or actuator of the laboratory was triggered in the event. An example of an object extracted from an activity object statement is shown in Figure V-C.

The statements in which the laboratory is the actor have an object type 'agent', in which a user is described in the object property. An example object extracted from an agent object statement is shown in Figure V-C.

[4] http://relle.ufsc.br/labs/16

```
"object": {
  "definition": {
     "name": {
        "en-US": "Plano Inclinado "
      },
      "type": "http://example.com/gtmre/activity-
          types/online-laboratory"
   },
   "objectType": "Activity",
   "id": "https://example.com/gtmre/inclinedPlane/
       actuator/servoMotor"
}
```

Fig. 7. Example of an activity object

```
"object": {
"objectType": "Agent",
"name": "Jose Simao",
"mbox": "mailto:simaoj@mail.com"
}
```

Fig. 8. Example of an agent object

D. Context

Using a URI that points to the definition of lab information in the context's 'extension' attribute, in this template represented by " https://example.com/relle/lab-information ", is sent information about the state of the experiment at the moment of the construction of the statement.

The sensors and actuators of a laboratory, especially when using the smart device specification, are described by a set of metadata that can be used in these attributes. When sending a command to the lab, the statement describes in its context which settings are defined by the user. When returning data to the user, the statement describes the values read by each sensor, such as the example described in Figure V-D.

```
"context": {
   "extensions": {
     "https://example.com/relle/lab-information": {
       "https://example.com/relle/heatPropagation/
           sensors/thermometer": {
         "1": 21.4,
         "2": 21.2
       }
     }
   }
}
```

Fig. 9. Example of context

VI. Prototype

The prototype used to verify the model was developed using the RLMS RELLE as the basis, integrating it with xAPI and using the specified vocabulary.

A. RELLE

RELLE (Remote Labs Learning Environment[5]) is a RLMS developed by the Remote Experimentation Laboratory (RExLab) of the Federal University of Santa Catarina (UFSC) in 2015 as part of the GT-MRE project. The environment can be used to access and manage remote laboratories and currently hosts 17 laboratories developed by UFSC and other 2 universties: Federal University of Uberlândia (UFU) and State University of Campinas (Unicamp).

The system was developed in PHP 5.5 using the Laravel MVC (Model-View-Controller) framework in its back-end, and HTML, the CSS Bootstrap framework, and the JavaScrip library jQuery on the front-end.

Its flexible architecture, described in Figure 10, allows the integration of different types of laboratories, and is divided into 3 modules: RLMS, LIS (Instance Scheduling Service) and Laboratory, which are described in sections below.

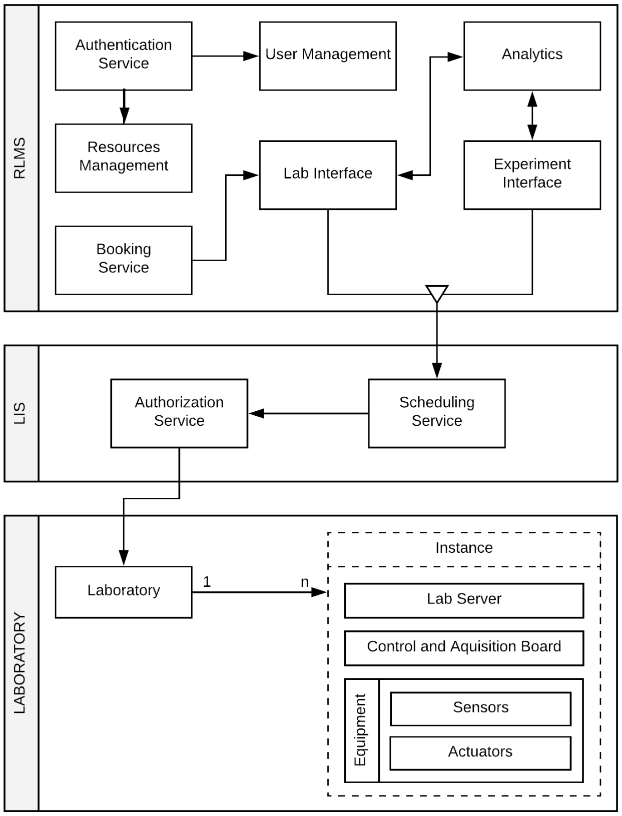


Fig. 10. Architecture of RELLE

The RLMS module is responsible for the features common to multiple laboratories, such as authentication, user and resource management, being available for access in 3 languages: Portuguese, English and Spanish.

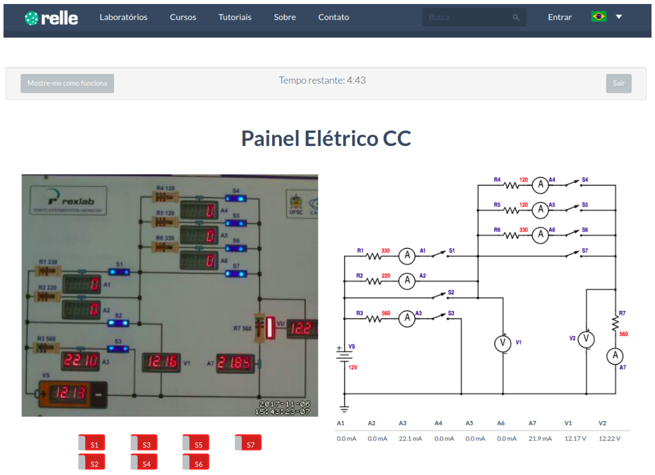
The authentication service gives access to the specific functionalities of each user role: while the average user can

[5] http://relle.ufsc.br

export reports with their identification in the labs and edit their profile, the administrator has access to the control panel, where all the administrative functions of the system are available . Administrator privileges include managing users, labs, scheduling, documents (including, editing, and deleting), access to logs, and permission to change the role of other users.

The system allows the registration of documents (such as teaching material, user manuals and technical documentation, for example) and other types of files, and their link to one or more laboratories. These are then available on the page that precedes access to the experiment, along with a video access tutorial, and laboratory metadata.

The lab client interface (Figure 11) gives the user control over the actuators and sensors available on the equipment. On this page, students have access to tutorials on experiment control, and can export the experiment data to a CSV file or a PDF report file.



Tempo restante: 4:43

Painel Elétrico CC

A1	A2	A3	A4	A5	A6	A7	V1	V2
0.0 mA	0.0 mA	22.1 mA	0.0 mA	0.0 mA	0.0 mA	21.9 mA	12.17 V	12.22 V

Fig. 11. Laboratory access interface

The lab client interface is developed by the administrator according to the specificities of the lab, using templates that are available in the public repository of the project [6].

Considering that in the interactive access model, in which the vast majority of laboratories available in the system are based, the equipment can only be accessed by one user at a time, it is necessary to have a service that deals with resource allocation.

LIS (Laboratory Instances Scheduling Service) is the service responsible for scheduling remote laboratories hosted in RELLE. The service manages the queue, and distributes access between multiple instances of a lab, in both lab and experiment interfaces.

The scheduling service used to be FCFS (First Come First Served), as described by [32], but from the implementation of scheduling functionality, the allocation of users in a lab began to consider weights, where the user that booked previously has a greater weight for that period than other users, receiving first access to the resource.

First, the authorization service validates the session through a token sent by the RLMS, confirming the user's authenticity. The service then checks whether the request is linked to a reservation in the system, and if so, sets a higher priority in the queue. Once authorized, the user is directly connected to the first available instance of a laboratory, and after this connection, the communication between the client and the lab server becomes direct, through the websocket protocol, and without the intermediary of the LIS.

LIS is also responsible for managing access to laboratories performed by other RLMSs in order to ensure that all requests comply with the rules of the queue. RLMS LabsLand now makes available 5 of the laboratories from RELLE [7].

Different models of laboratory access are supported by the platform, which in spite of offering this scheduling model allows the laboratory to adopt its own method, as in the case of VISIR, which uses its own service with batch access instead of the interactive model.

RELLE hosts 17 remote laboratories: (a) Electrical Panel CC, (b) AC Power Panel, (c) Conduction of Heat in Metal Bars, (d) Means of heat propagation, (e) Remote Microscope, (f) Inclined Plane, (g) Optical Bank, (h) Conversion of Light Energy in Electrical, (i) Newton's disk, (j) Development Environment in Arduino, (k) block.ino, (l) VISIR, (m) Observing the Water, (n) LTE Remote Microscope, (o) LTE Remote Microscope, (p) Online Titulator and (q) Thomson experiment.

Most laboratories (a-h) use the same modular architecture described in Figure 10: a certain equipment (a microscope, for example) has inputs and outputs automated by integrating sensors and actuators into one control and acquisition board developed by RExLab, which can be accessed via the lab server - a Node.JS service hosted on a Single-Board Computer (SBC), usually a Raspberry pi.

The VISIR laboratory uses a totally different architecture from the other experiments available on the platform. The lab consists of switching matrices, a measuring server, an equipment server, and the National Instruments PXI platform with its multimeter modules, function generator, oscilloscope, and power supply.

B. Learning Record Store

Learning Locker, a free and open source LRS developed by HT2 Labs, was installed on the RExLab servers. The application acts as a data repository designed to store instructions for learning activities generated by tools with support for xAPI, and is the most commonly used LRS [26].

In addition to the basic logging, verification and redemption functionality of statements, the main application features support multiple LRSs and institutions, multiple personas (actors of a user can be grouped under the same persona), sharing statements between LRSs and other applications, as well as visualization tools that allow the creation of graphs

[6] https://github.com/RExLab/lab_package

[7] https://labs.land/relle/?lang=en

with custom series and axes, and an API integrated with all UI functionalities, which can be used to request reports and graphs, for example.

The version used for model validation, v2.0, is developed using Node.JS (unlike the previous version that was developed in PHP), Mongo DB and Nginx. This release supports the specification of xAPI v1.0 and above.

The architecture of the application is divided into two elements: the Learning Locker itself, and the xAPI service. The first one comprises the user interface (UI), an HTTP API, and web workers, elements that enable you to perform a script operation on a thread separate from the main execution thread of a web application [33]. The second one provides services for statements, activity profiles, agent profiles, and states.

C. Integrating to Experience API

From the interface of the laboratories available on RELLE, were defined triggers for every event registered. A JavaScript client has been integrated into the front-end of the labs to send the statements to the LRS. TinCanJS [8] library, developed by Rustici Software, was used instead of the official version provided by ADL, xAPIWrapper [9], because the first continues to be updated by the developers.

Using jQuery, a cross-platform JavaScript library designed to simplify the client-side script of HTML, a set of functions has been developed to facilitate the assembly of statements and their sending by by TinCanJS. Each element of a statement is usually composed of a series of descriptors and IRIs, so these functions allow the code for the formulation of elements to be reused, not being repeated several times in each component that triggers an event.

The xapiCreateStatement function takes as its argument an object containing the labels for verb, object, and context, and calls other functions that form the JSON object sent to LRS.

The remote laboratories AC Panel, Remote Microscope, Inclined Plane and Disc of Newton were integrated to the xAPI using the proposed model. Trigers have been defined for interaction between student and laboratory, and also for complementary resources, such as tutorials.

In addition, all other labs available in RELLE have at least the start and end of session events being tracked, so that data on session number and duration can be retrieved.

In the CA Panel lab, the verb 'set' was used to record interactions with keys. This same verb was used in the Inclined Plane to indicate the user-defined angle for slope of the base, and Newton's Disc to indicate when the user turned the laboratory on or off.

In the Remote Microscope, the verb 'moved' indicated which side and which sample the student asked the laboratory. And when the student consulted the tutorial, a statement with the verb 'viewed' was sent to the LRS.

Since the prototype was deployed on November 2017, approximately 60,000 statements have been stored. Of these, a total of 215 authors recorded events described by 13 of the verbs from the model.

VII. Final Remarks

The popularization of ICTs has promoted profound and perhaps irreversible changes in the dynamics of the relationship between people and their daily activities. And from the reflection of these changes in education and the increasing development and use of digital educational tools, a great amount of valuable data about teaching and learning process has been created.

The purpose of this research was to propose a model for the formalization and storage of data on learning experiences in scenarios where experimentation practices are supported by remote laboratories. The model presented in this paper allows the registration of data in different application scenarios.

The registration of the interaction between student and remote laboratory was made through the xAPI specification, which was used from a study of the different technical specifications and architectures of remote laboratories and models for data interoperability in educational applications.

The xAPI specification proved to be the most adequate for the proposal, given its flexibility and richness in the description of the events recorded. There was no profile or model to record data using this specification specifically focused on remote laboratories.

When developing the vocabulary proposed in this model, the definition of verbs was facilitated by the fact that xAPI allowed the community to define its own verbs and types of activity. However, the implemented prototype, using the model to record activities developed by the students in 4 remote laboratories hosted in RELLE, allowed to verify that due precisely to the flexibility of the vocabulary and the need to assign detailed descriptions, the more general a model using the xAPI specification is, more complex it becomes.

In order to expand the model for more application scenarios and deploy it in different laboratory architectures, it would be important to collaborate with other institutions and researchers in the area to share the experiences related to the registration and analysis process conducted by each research group.

The research developed is not complete with the conclusion of this work. In addition to adopting the model to record events in all RExLab remote laboratories, a future study would be interesting about the mining of the learning experience data recorded through this model, and possible applications for student evaluation, verification of effectiveness of pedagogical practices and management of technological resources.

Acknowledgment

The authors would like to thank the Foundation for Research and Innovation Support of the State of Santa Catarina (FAPESC) and the Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES) for the master's scholarships, and Global Affairs Canada for the mobility fellowship granted through the Emerging Leaders in the Americas Program (ELAP).

[8] https://github.com/RusticiSoftware/TinCanJS

[9] https://github.com/adlnet/xAPIWrapper

References

[1] J. R. Brinson, "Learning outcome achievement in non-traditional (virtual and remote) versus traditional (hands-on) laboratories: A review of the empirical research," *Computers & Education*, vol. 87, pp. 218 – 237, 2015.

[2] M. Auer, A. Pester, D. Ursutiu, and C. Samoila, "Distributed virtual and remote labs in engineering," in *IEEE International Conference on Industrial Technology, 2003*, vol. 2. Maribor, Slovenia: IEEE, 2003, Conference Proceedings, pp. 1208–1213.

[3] J. Machotka, A. Nafalski, and Z. Nedić, "The history of developments of remote experiments," in *Procedings of the 2nd World Conference on Technology and Engineering Education*. Ljubljana, Slovenia: University of Ljubljana, September 2011, pp. 96–100.

[4] C. Bohus, B. Aktan, M. H. Shor, and L. A. Crowl, "Running control engineering experiments over the internet," in *13th World Congress of IFAC*. San Francisco: Elsevier, 1996.

[5] J. Henry, "Running laboratory experiments via the world-wide web," in *ASEE Annual Conference Proceedings*. Seattle: American Society of Engineering Education, 1998.

[6] M. Predko, *Programming and customizing the 8051 microcontroller*. New York: McGraw-Hill, 1999.

[7] M. A. Wisintainer, "Rexlab: Laboratório de experimentação remota com o microcontrolador 8051," Mestrado em Ciência da Computação, Universidade Federal de Santa Catarina, Florianopolis, 1999.

[8] C. P. Antonio, J. P. C. Lima, J. B. M. Alves, J. B. Silva, and J. P. S. Simão, "Merging a remote microscope and virtual worlds: Teaching kingdom plantae on basic education," *International Journal of Online Engineering (iJOE)*, vol. 12, no. 4, p. 3, 2016.

[9] J. P. S. Simão, J. P. C. Lima, C. Heck, K. Coelho, L. M. Carlos, S. M. S. Bilessimo, and J. B. Silva, "A remote lab for teaching mechanics," in *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*. Madrid, Espanha: IEEE, 2016, Conference Proceedings, pp. 176–182.

[10] J. G. Zubía and G. R. Alves, Eds., *Using Remote Labs in Education: Two Little Ducks in Remote Experimentation*, 1st ed. Bilbao: University of Deusto, 2011.

[11] M. A. Marques, M. C. Viegas, M. C. Costa-Lobo, A. V. Fidalgo, G. R. Alves, J. S. Rocha, and I. Gustavsson, "How remote labs impact on course outcomes: Various practices using visir," *IEEE Transactions on Education*, vol. 57, no. 3, pp. 151–159, 2014.

[12] E. Tsourlidaki and P. Zervas, "The go-lab inventory and integration of online labs – labs offered by universities," Disponível em: ¡https://telearn.archives-ouvertes.fr/hal-01201932/document¿, p. 247, 2014, acesso em: 22 set. 2017.

[13] V. J. Harward, J. A. del Alamo, S. R. Lerman, P. H. Bailey, J. Carpenter, K. DeLong, C. Felknor, J. Hardison, B. Harrison, I. Jabbour, P. D. Long, T. Mao, L. Naamani, J. Northridge, M. Schulz, D. Talavera, C. Varadharajan, S. Wang, K. Yehia, R. Zbib, and D. Zych, "The ilab shared architecture: A web services infrastructure to build communities of internet accessible laboratories," *Proceedings of the IEEE*, vol. 96, no. 6, pp. 931–950, 2008.

[14] D. LOWE, T. MACHET, and T. KOSTULSKI, "Uts remote labs, labshare, and the sahara architecture," in *Using Remote Labs in Education: Two Little Ducks in Remote Experimentation*, 1st ed., J. G. Zubía and G. R. Alves, Eds. Bilbao: Universidad de Deusto, 2012, ch. 20, p. 403.

[15] I. Gustavsson, J. Zackrisson, L. Håkansson, I. Claesson, and T. L. Lagö, "The VISIR project – an open source software initiative for distributed online laboratories," in *Proceedings of the International Conference on Remote Engineering and Virtual Instrumentation (REV)*. Porto, Portugal: IEEE, 2007.

[16] P. Orduña, J. Irurzun, L. Rodriguez-Gil, J. Garcia-Zubia, F. Gazzola, and D. L. de Ipiña, "Adding new features to new and existing remote experiments through their integration in weblab-deusto," *International Journal of Online Engineering (iJOE)*, 2011.

[17] P. Orduña, L. Rodriguez-Gil, J. Garcia-Zubia, I. Angulo, U. Hernandez, and E. Azcuenaga, "Labsland: A sharing economy platform to promote educational remote laboratories maintainability, sustainability and adoption," in *2016 IEEE Frontiers in Education Conference (FIE)*. Erie, EUA: IEEE, 2016, pp. 1–6.

[18] I. Titov, A. Glotov, I. Vlasov, and J. Mikolnikov, "Labicom labs: Rll, gnss, oscilloscope and generator remote and virtual labs: Interactive demonstration of labicom labs in spring 2015," in *2015 3rd Experiment International Conference (exp.at'15)*. Ponta Delgada, Portugal: IEEE, 2015, pp. 145–146.

[19] D. G. Zutin, M. E. Auer, C. Maier, and M. Niederstätter, "Lab2go: A repository to locate educational online laboratories," in *IEEE EDUCON 2010 Conference*. Madrid, Espanha: IEEE, 2010, Conference Proceedings, pp. 1741–1746.

[20] T. Richter, Y. Tetour, and D. Boehringer, "Library of labs - a european project on the dissemination of remote experiments and virtual laboratories," in *2011 IEEE International Symposium on Multimedia*, 2011, pp. 543–548.

[21] J. E. Froyd, P. C. Wankat, and K. A. Smith, "Five major shifts in 100 years of engineering education," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1344–1360, 2012.

[22] F. J. García Clemente, L. de la Torre, S. Dormido, C. Salzmann, and D. Gillet, "Collecting experience data from remotely hosted learning applications," in *Online Engineering & Internet of Things: Proceedings of the 14th International Conference on Remote Engineering and Virtual Instrumentation REV 2017*, M. E. Auer and D. G. Zutin, Eds. Cham: Springer International Publishing, 2018, pp. 170–181.

[23] W. M. A. F. W. Hamzah, N. H. Ali, M. Y. M. Saman, M. H. Yusoff, and A. Yacob, "The use of tin can api for web usage mining in e-learning applications on the social network," in *2015 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*. Langkawi, Malásia: IEEE, 2015, pp. 113–118.

[24] J. M. Kevan and P. R. Ryan, "Experience api: Flexible, decentralized and activity-centric data collection," *Technology, Knowledge and Learning*, vol. 21, no. 1, pp. 143–149, Apr 2016.

[25] A. Bakharia, K. Kitto, A. Pardo, D. Gašević, and S. Dawson, "Recipe for success: Lessons learnt from using xapi within the connected learning analytics toolkit," in *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, ser. LAK '16. New York, NY, USA: ACM, 2016, pp. 378–382. [Online]. Available: http://doi.acm.org/10.1145/2883851.2883882

[26] B. Betts and R. Smith, "The learning technology manager's guide to xapi," Disponível em: ¡https://www.ht2labs.com/resources/the-learning-technology-managers-guide-to-the-xapi/¿, 2016, acesso em: 21 dez. 2017.

[27] ADVANCED DISTRIBUTED LEARNING INITIATIVE, "Experience api: Advanced distributed learning (adl) co-laboratories," Disponível em: ¡https://github.com/adlnet/xAPI-Spec/¿, 2017, acesso em: 20 dez. 2017.

[28] J. C. Vidal, T. Rabelo, and M. Lama, "Semantic description of the experience api specification," in *2015 IEEE 15th International Conference on Advanced Learning Technologies*. Hualien, Taiwan: IEEE, July 2015, pp. 268–269.

[29] D. Brickley and L. Miller, "Foaf vocabulary specification 0.91," Disponível em: ¡https://s2.smu.edu/ coyle/cse7347.prev/handouts/s14.FOAF%20Vocabulary%20Specific 2007, acesso em: 11 nov. 2017.

[30] ADVANCED DISTRIBUTED LEARNING INITIATIVE, "Relationship between vocabularies, profiles, and recipes," Disponível em: ¡https://adl.gitbooks.io/companion-specification-for-xapi-vocabularies/content/relationship_between_vocabularies,_profiles,_and_r.html¿, 2016, acesso em: 16 dez. 2017.

[31] C. Salzmann, S. Govaerts, W. Halimi, and D. Gillet, "The smart device specification for remote labs," *International Journal of Online Engineering (iJOE)*, vol. 11, no. 4, pp. 9–29, 2015.

[32] J. P. C. Lima, "Desenvolvimento de servidores para laboratórios remotos baseado no paradigma de dispositivos inteligentes," 2016, trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) - Universidade Federal de Santa Catarina, Campus Ararangua.

[33] Mozilla, "Web workers api," Disponível em: ¡https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API¿, 2017, acesso em Nov. 15 2017.