Received June 20, 2018, accepted July 21, 2018, date of publication July 31, 2018, date of current version August 20, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2860519

An xAPI Application Profile to Monitor Self-Regulated Learning Strategies

MARIO MANSO-VÁZQUEZ, (Member, IEEE),
MANUEL CAEIRO-RODRÍGUEZ, (Senior Member, IEEE),
AND MARTÍN LLAMAS-NISTAL, (Senior Member, IEEE)
Department of Telematic Engineering, University of Vigo, 36310 Vigo, Spain

Corresponding author: Mario Manso-Vázquez (mario.manso@gist.uvigo.es)

This work was supported by the Galician Regional Government under Projects ED431B 2017/67 and ED431D 2017/12.

ABSTRACT Self-regulated learning (SRL) is being promoted and adopted increasingly due to the needs of current education, student centered and focused on competence development. One of the main components of SRL is learners' self-monitoring, which eventually contributes to a better performance. Monitoring is also important for teachers, as it enables them to know to what extent their learners are doing well and progressing properly. At the same time, the use of technology for learning is now common and facilitates monitoring. Nevertheless, the available software still offers poor support from the SRL point of view, especially, for SRL monitoring. This clashes with the growth of learning analytics and educational data mining. The main issue is the wide variety of SRL actions that need to be captured, commonly performed in different tools, and the need to integrate them to support the development of analytics and data mining developments, making imperative the search of interoperable solutions. This paper focuses on the standardization of SRL traces to enable data collection from multiple sources and data analysis with the goal of easing the monitoring process for teachers and learners. First, the paper analyzes current monitoring software and its limitations for SRL. Then, after a brief analysis of available standards on this area, an application profile for the eXperience API specification is proposed to enable the interoperable recording of the SRL traces. The paper describes the process followed to create the profile, from the analysis to the final implementation, including the selection of the interactions that represent relevant SRL actions, the selection of vocabularies to record them and a case study.

INDEX TERMS Educational technology, learning analytics, self-regulated learning, standardization.

I. INTRODUCTION

Self-regulated Learning (SRL) is, according to one of the most accepted definitions, "an active, constructive process whereby learners set goals for their learning and then attempt to monitor, regulate, and control their cognition, motivation, and behavior, guided and constrained by their goals and the contextual features in the environment" [1]. In Europe, SRL has become especially relevant in recent years due to the Bologna Declaration [2], in which learners are required to be more autonomous and accountable. In a broader context, self-regulation is also recognized as crucial to achieve academic success [3], empowering the learners to control their own learning and preparing them for lifelong learning.

The adoption of SRL principles involves many changes in the role of learners and educators. From a learner's perspective, SRL implies the management of a variety of strategies: planning, forethought, task analysis, self-monitoring, control, self-evaluation, awareness, reflection and reaction [1], [4]. This work is focused on self-monitoring, as it is an essential part of SRL. According to the main SRL models [1], [5]–[8], it is important for the learners to monitor their learning, in order to be aware to control and evaluate their methods, actions and outcomes. Traditionally, this process relies on students' awareness, memory and note taking. From an educator's perspective, SRL also demands the development of new activities. Although self-regulating is mainly a learner's task, increasing autonomy does not mean leaving learners alone. The role of educators is important, because SRL can be taught and supported, and to this end, it should be tracked and assessed. Therefore, it is crucial for teachers to know how their learners are self-regulating in order to guide them when necessary. Nevertheless, educators' tracking students' SRL development is usually a time-consuming task [9], relying mainly on interviews and reports.

As in other fields, the use of technology to support SRL approaches is a growing trend [10]. In particular, monitoring students' activity is an active research field as shown by the amount of work in Learning Analytics (LA) and Educational Data Mining (EDM) [11]–[14]. Both learners' self-monitoring and educators' SRL tracking and assessment could benefit from these approaches. However, despite progress, from a SRL point of view monitoring is still far from the needs of students and educators as long as it relies mainly on traditional system logs [15]. Some of the core functionalities needed for a more comprehensive SRL support are not yet implemented in existing systems or have limitations, lacking relevant traces from key activities recognized in the main SRL models [16].

The goal of this work is to improve the support for SRL monitoring by focusing on the relevant SRL traces. Rather than staying limited to the traces from current learning software, which produce partial and incomplete results, a different approach is tried, a fresh start, laying out two main questions: *(1) what information is relevant to enable a more comprehensive monitoring of the SRL processes?* and *(2) How such information can be represented?* The answer to these questions is conditioned by some distinctive features of the SRL scenario. First, SRL implies many different activities that are usually performed using different software tools, often considered as part of a Personal Learning Environment (PLE) [17]. The range of tools varies from general ones, such as task managers, to specific software, such as LMSs [10]. Therefore, this scenario requires to gather, select and analyze traces from multiple sources, even including offline activities. Second, we have to define what actions should be recorded (since many actions would be irrelevant from a SRL perspective) and how to represent them, since the same actions may have different representations in different tools. Besides, this should be done while maintaining coherence in vocabulary and structure in order to ease data processing. Therefore, a standard solution is needed defining what actions to record and a how to represent them. The paper introduces such a solution as an application profile of the well-known monitoring standard Tin Can API (also known as eXperience API or xAPI). The eventual goal is to facilitate the interoperability among learning systems for monitoring SRL.

The rest of the paper describes the process followed to produce the xAPI SRL application profile. First, self-regulated learning and the main models about this theory are introduced in Section II. In Section III, the state of the art about measuring and monitoring SRL is briefly reviewed, paying attention to SRL interactions. Section IV analyzes current LA standards for monitoring learning. Section V shows the analysis made to select the interactions representing SRL actions defined after this analysis. Section VI presents the xAPI profile implementation, along with a case study.

II. SELF-REGULATED LEARNING

There are several models about self-regulated learning, but all of them share a similar vision. As we synthesized in [18], "the self-regulation process is based upon the students' consciousness about their possibilities and limitations, their task centered goals and their use of appropriate strategies. Students can improve their performance and academic success using strategies to control and regulate aspects of their cognition, motivation and behavior, to select and build learning environments and to set goals and monitor their compliance."

A. SRL MODELS

Three main models have been proposed to arrange the different strategies and processes involved in SRL: Zimmerman's cyclical model [6], Pintrich's general model [1] and Wine and Hadwin's information processing model [7], which are being used by most experts [19]. All of them tackle the main dimensions of SRL. The information processing model is focused in cognitive and metacognitive aspects, while cyclical and general models include other aspects as motivation and behavior. For this research, Zimmerman's and Pintrich's models were taken as reference because of their wider focus.

- Zimmerman's model defines the existence of three phases [10]: (1) forethought phase, in which the students do the task analysis and planning, and establish motivational beliefs; (2) performance phase, in which self-control and self-observation take place; and (3) self-reflection, in which the learners self-judge their learning process and react to their own conclusions.
- Pintrich proposes four phases [10]: (1) forethought, planning and activation, (2) monitoring, (3) control and (4) reaction and reflection. In this model, the four phases are not cyclical but the learner jumps from one to another when needed. This general model also defines four areas: cognition, motivation/affect, behavior and context, establishing a very comprehensive model.

Here, an integrative model based on common elements involving three phases and five areas (see Fig. 1) has been used, compiling recent proposals by several authors [3], [20]. The three phases are: (1) forethought, planning and activation, (2) performance, monitoring and control, and (3) evaluation, reflection and reaction. This is because the elements of each phase are concurrent within the learning episode. The five areas are: cognition, metacognition, motivation / affect, behavior, and (social and environmental) context. Cognition and metacognition appear integrated in Pintrich's model, but are considered as different areas because the processes of each of them are very different [3]. This integrative model arranges many different strategies related to SRL in a matrix and in this way it facilitates a comprehensive visualization of the many issues involved.

B. SRL STRATEGIES

There are many definitions of what learning strategies are [21]. According to Nisbet & Shucksmith [22], learning

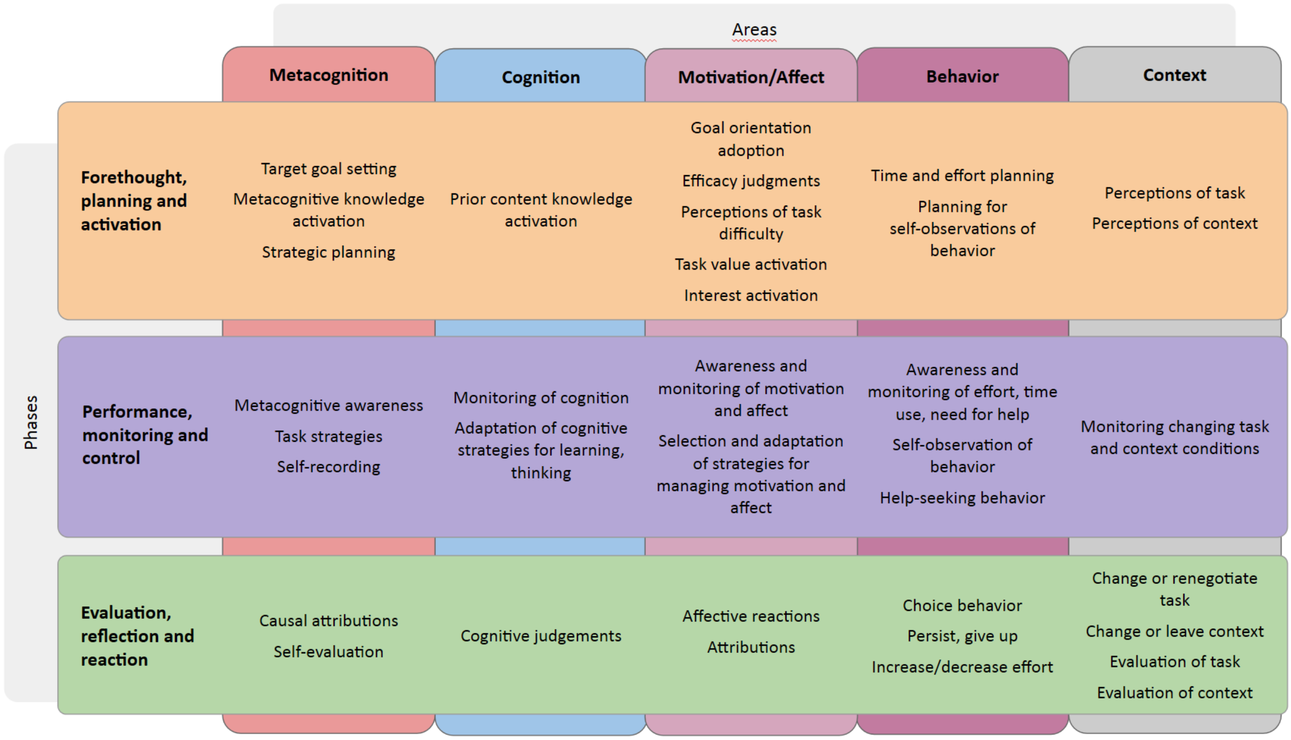


FIGURE 1. Integrative SRL model.

strategies are "integrated sequences of procedures, the appropriate selection and flexible adaptation of which is to meet the needs of a specific learning situation." There are two main types of learning strategies: self-regulation strategies and cognitive learning strategies.

Self-regulation strategies focus on metacognitive aspects, as well as behavior, motivation and context control. Self-regulation strategies can be really complex, being formed by several simpler strategies. We called these complex strategies *super-strategies*. An example of a self-regulation strategy is the *goal definition* strategy. To perform a goal definition, the learner names and describes a goal following the SMART scheme: Simple, Measurable, Attainable, Realistic and Time-bounded. Another consideration is that long-term goals should be divided into short-term ones to meet the requirements. This is a simple strategy, i.e., cannot be divided into more simple strategies. In this case it is part of a super-strategy called *goal-based learning planning*, a complex strategy formed by five simple strategies: goal definition, goal planning, plan monitoring, plan evaluation and plan correction.

Cognitive strategies provide a structure for doing complex tasks. Some of them are also known as study techniques and tackle cognitive actions as information acquisition and transformation, memorization, reading comprehension, problem solving, etc. A representative example of a cognitive strategy could be SQ3R (Survey, Question, Read, Recite, and Review), a reading comprehension strategy. It is based on five steps:

1) Survey: this first step consists in glancing through the text in order to identify headings, sub-headings and other outstanding elements.
2) Question: formulate questions about the content before reading it.
3) Read: read the text trying to find answers to the questions.
4) Recite or write: after reading the text, sum up and try to answer the questions.
5) Review: review the important contents.

From our point of view, learning strategies are the practical element that, supported by technological systems, can provide traces to track learners' SRL actions that can be used to ease monitoring, as shown in Section V.

III. MEASURING AND MONITORING SRL

To date, learners' self-monitoring and educators' SRL monitoring are done very differently. Learners rely on their own memory and notes to monitor, control and evaluate their own progress and performance. Teachers and educators have two main options that complement each other to measure and monitor SRL. On the one hand, the general assessment of students' SRL skill level is commonly measured using self-completion questionnaires [23], like LASSI [24], MSQL [25] or CEVEAPEU [26]. These provide the educator with a picture of each student's skills at a certain moment in time, but do not enable SRL performance monitoring. On the other hand, continuous tracking of students' SRL performance allows the educators to support their students in real time, addressing problems when it is needed, but it is a demanding task that relies on periodical interviews and reports [9]. In both learners' and educators' cases, monitoring SRL is commonly performed using traditional tools. However, monitoring can be supported through software, especially in e-learning or blended learning scenarios [27].

We reviewed current e-learning systems with monitoring support in order to know the state of the art [19]. One of the conclusions of this review is that SRL monitoring support is limited by the SRL related functionalities that the system has, or has not, since the main problem seems to be the lack of SRL specific functionalities [19], [27]. Typically, e-learning systems record activity traces that reflect the use of certain parts of the system, like "the user commented on thread X", "the user opened document Y" or "the user joined group Z." In general, these traces are recorded using proprietary formats and are mainly for logging purposes. Thus, most e-learning systems do not provide a specific monitoring solution. Nevertheless, we found some systems that enable monitoring some aspects of the learning process using different approaches, as shown in the next sections.

A. TRACKING SOCIAL INTERACTIONS

Systems like TrAVis [28] and StepUp! [29] use data from students' interactions among themselves and with teachers to support self-monitoring.

TrAVis (Tracking Data Analysis and Visualization Tools) [28] is "a web-based system that assists the students in visualizing their communication activities in distance learning situations." TrAVis accesses a tracking data repository, computes certain data indicators and displays them in different forms, letting students and teachers monitor individual and group activities. The indicators selected differentiate four levels of interaction: aggregation (individual), discussion, cooperation and collaboration. It uses time as a reference to allocate the interactions.

StepUp! [29], a system developed within the ROLE project,[1] records mainly social interactions, adding also time spent (through Toggl),[2] produced artifacts (content creations in blogs) and resource use. It also uses time as a reference for all the interactions. Data is shown in a table that contains a summary of all the traces and shows bar activity graphs, having also a mobile application that shows a simplified version of them.

B. TRACKING IN-COURSE PROGRESS

Systems like LearnTracker [30] or Student Activity Meter (SAM) [31] are more focused on cognitive tasks. Both provide visualizations of progress in the course for teachers and learners, based on time spent and resource use.

LearnTracker is a mobile learning system that tracks the time and location of the activities performed by the learners, in order to know their learning habits. Learners use a timer embedded in LearnTracker that shows the accumulated time for each activity or an asynchronous time recorder. It has a content creation tool based on a web service and a reporting tool to display graphs that show the time spent per activity, per user or compared to teacher time estimations. Despite being a great concept, it is not possible to export or import any data.

SAM [31] also provides different visualizations of time spent on learning activities and resource use tracked from various learning environments, including Moodle. It relies on CAM [32] data, which uses system logs. It enables the discovery of averages and trends, assisting teachers in the detection of students at risk. Resource use indicators also provide the students information about learning material used by other students, which can be especially useful in SRL [6].

C. PROCESS MINING AND SRL

A different project called SoftLearn [9], aimed at easing the SRL assessment for the teachers, uses process mining to discover and depict the learning paths followed by the students. In a blended learning scenario, using ELGG as a PLE, SoftLearn uses activity traces from blog posts, microblogging, forum messages, personal walls, favorites and content creations. One of the main achievements of this project was to obtain relatively simple, comprehensive and precise information about the learning process from a large amount of data.

Similarly, in [33] student's activities captured in event logs are analyzed by process mining techniques (Fuzzy Miner and ProM) to discover SRL processes followed by students in a specific course. Basically, the processes, patterns and frequencies of the most successful and least successful students were analyzed to identify differences among them. Activities had been classified according to the SRL phase (Metacognition, Cognition, Organization and Motivation) and strategy (e.g. Planning, Goal Setting, Repeating, Search, Elaboration). Process mining techniques were used to check the conformance of the event logs to reference process models. In addition, a fitness metric was defined to measure the similarity of a set of traces to such a reference process model. This approach provides indicators about self-regulation, but it requires a great effort because activities of interest have to be identified and reference process models defined in advance according to the course.

D. LACK OF KEY ACTIONS

In general, the reviewed solutions focus on information access actions, interactions between users, communications, blog posts, comments and content creations (wikis, pages or uploaded files), corresponding to the functionality provided by the systems. The main problem we have found regarding the actions to record is that current SRL support is partial and just tackles part of the SRL processes, even in systems with SRL relevant functionalities, like time use measurement (relevant for behavior control in SRL). These variables are an exception. The most common tracked activities are cognitive and communicative ones, leaving out other important activities related to metacognition or

[1] http://role-project.archiv.zsi.at
[2] https://toggl.com

self-regulation, like task planning, goal definition, time monitoring, etc. These missing interactions are key to monitor SRL.

IV. MONITORING STANDARDS FOR E-LEARNING

Selecting the right standard for a LA application is key. There are great reviews of current LA standards like the one presented in [34], which explains the main features of xAPI and IMS Caliper and their origins. Both are based on XML and enable trace interoperability. This section provides a summarized review to justify the election of xAPI for the SRL profile.

A. IMS CALIPER

Developed by the IMS Global Consortium, Caliper Analytics [35] is an open standard to capture educational traces to measure learning activities. Caliper builds an ecosystem around the *Learning Sensor*, an API to capture data from any browser. It uses IMS LTI to integrate traces from tools and other standards.

Caliper captures activity traces in the form of *Events*, whose structure comes from the social activity standard *Activity Streams*.[3] The data of an event consist on an actor, an action and an object, besides seven different elements to describe the action, the learning context and the activity context.

It is based on several metric profiles defined by IMS to describe different learning activities, like evaluation, reading, etc. A metric profile defines the activity metrics, the activity information model and the engagement and/or performance metrics. Although its profiles tackle part of the SRL actions, it would be necessary to create a SRL specific profile and extend other profiles currently available in order to define all actions derived from the selected strategies. Unfortunately, the creation of metric profiles is performed exclusively by IMS and not open to external parties.

B. EXPERIENCE API

Experience API, also known as Tin Can API or xAPI,[4] is a community-driven specification for learning technology managed by ADL, the owners of SCORM. It was born from applying the Activity Streams concept to e-learning [36]. It defines both data and communication models to track user activities within learning software applications.

In xAPI, an event is captured as a *Statement* in the form of a sentence ''I did this'' (actor, verb, object) in the software where the event took place (*Activity Provider*) and then stored in a Learning Record Store (LRS) in chronological order. A LRS is a repository that can be accessed by any authorized tool, enabling communication with different tools (LMS, games, e-portfolios, reporting tools, etc.) and with other LRS.

xAPI has four APIs: activities are identified in the *Activity Profile API*; activity states are managed by the *State API*, which enables resuming an activity between sessions; users are identified by the *Agent Profile API*, even if using different accounts; and statements are managed by the *Statement API*.

[3] https://www.w3.org/TR/activitystreams-core/
[4] https://experienceapi.com/overview/

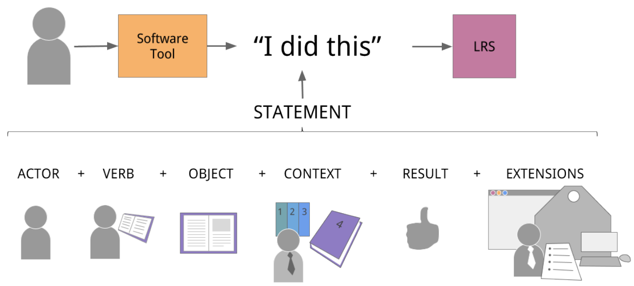


FIGURE 2. xAPI statement.

Experience API inherits from *Activity Streams* the nomenclature and the concept of the statement, but its structure is different. The main elements in a statement are the *Actor*, the *Verb* (action) and the *Object* (see Fig. 2). These can be completed with the *Context*, which contains information about the context of the action (parent activity, instructor, platform, etc.); the *Result*, which has different options to record grades and duration of the action; *Extensions*, a multipurpose object to define data that does not fit into any other field; and *Authority*, which specifies who assures the truthfulness of the statement and *Attachments*. There are also other formal attributes, like *UUID* (Universal Unique IDentifier of the statement) and *Timestamp*.

Experience API statements are built using XML language. The main elements are represented by an URI (with the exception of the actor, represented by a *mbox* or an account), but a human readable version for a specific language is also provided; e.g., the actor and the object have a name attribute and the verb has a display attribute that are human readable:

```
"verb": {
    "id": "http://id.tincanapi.com/verb/
           rated",
"display": {"en-US": "rated", "es":
             "calific\'{o}" }
}
```

The main features of xAPI, key to its wide adoption, are its openness and flexibility. The vocabulary is limited but expandable, making it easy to record new actions. Every new verb, activity, extension and profile is curated by ADL in order to avoid duplicates or possible problems with homonymous or polysemic words, and then added to the specification for community use. Unlike Caliper, xAPI enables the creation of new verbs and activity types, and even new attributes using the *Extensions* field. These features make the creation of specific domain profiles possible, making xAPI an ideal candidate for the creation of the SRL profile, presented in the next section.

V. DEFINITION OF SRL ACTIONS

As stated in the introduction, one of the main issues that makes monitoring SRL through software a complex problem is that the wide variety of SRL actions is not recognized in learning software, i.e., available e-learning software does not have enough functionality to support main SRL actions, limiting SRL monitoring. Considering this, we focused on how to represent those actions to enable SRL monitoring. Rather than working on what information we can get with the traces we have, we approached this issue wondering (1) what information is needed to enable proper and comprehensive SRL monitoring, (2) how can it be gathered and (3) how can it be processed to maximize the amount of relevant information for both learners and educators. The first question led us to study and analyze the SRL process. Nevertheless, despite SRL theory and the three main SRL models describe the process, they are too general to identify activities and actions of interest. Therefore, we changed our focus to *self-regulated learning strategies*, the practical element of SRL that, as described in Section II.B, defines the procedures that drive self-regulated learners' actions.

A. LEARNING ANALYTICS STRATEGY ANALYSIS

Learning strategies define and describe the actions that learners perform to plan, monitor, regulate and evaluate their learning, providing key information to define the traces that represent relevant SRL actions. The level of detail in which the actions are described depends on the strategy and the source. Therefore, several sources were studied [22], [37]–[39] to compile as many strategies as possible with enough detail regarding action definition, resulting in 72 strategies. This number was reduced to a selection of 55 attending to the following criteria regarding software implementation:

- The *object* of the strategy should be *measurable*, quantifiable or tangible. For instance, time management strategies are implementable as time is measurable.
- The *action* itself and the information it manages can be *recorded* using an existing language or specification.
- It requires an *input* and/or generates an *output*.
- It is procedural, i.e., it is based on steps (not required but recommended).

These criteria pretend to guarantee that the software implementation of a strategy is possible and produces traceable results. An example of a strategy that was rejected is the "avoiding inner and outer distractions" strategy, a motivational and self-control strategy focused on preparing the study environment and mindset to avoid distraction and postpone other activities. This strategy does not meet the first three criteria.

The strategies were classified following Pintrich's SRL model but dividing metacognitive and cognitive strategies [19] as explained in section II.A.

Table 1 shows a list of the selected strategies following this classification. (Note that metacognitive and behavioral categories were merged as most strategies touch both areas and the rest are strongly related; super-strategies were not included as they are formed by several strategies from different categories; and information organization and memorization strategies were grouped for the sake of clarity, resulting in 39 entries.)

The analysis of each strategy followed these steps:

1) *Dividing complex composite strategies into simple strategies*.
2) *Identifying the process described in the strategy*, i.e., the actions and steps learners should follow to use it.
3) *Identify a software implementation of the strategy*, outlining the improvements it may need to address the limitations for the development of the strategy. This is, the learner has to perform certain actions to develop the strategy in the software. For that, we identified software (generic and specific e-learning software) that provides support for the development of each strategy. In some cases, the software does not provide support for all the actions in a strategy. Therefore, the functionality to support those missing actions should be described in order to know the data and interactions involved in its development. For instance, a generic task manager fits most of the actions involved in the *goal-based learning planning* strategy, but has some limitations to its full development, like the differentiation of tasks and goals, that should be described and addressed [19].
4) *Define the traces resulting from a software implementation of the strategy*, i.e., the information and data resulting from the performance of the strategy in the software, including the actions themselves (actor, action, object of the action, results, context, time, etc.). This definition was raw as it did not include a representation using any of the available specifications, which was performed later in the process (see section VI).

This fourth step is the most relevant one for this study. Here we show an example of the trace definition for the *time estimation* strategy. This simple strategy consists on assigning a time duration for a future event, task or activity. This strategy generates this data:

- The actor
- The time estimation (time duration)
- The element for which the action estimates the duration
- Additional contextual information:
  - the time of the action (timestamp)
  - the parent element, if any (chapter, course, etc.)
  - the system

Despite the raw and rather simplistic approach, the objective is that the data defined for each trace is detailed enough to describe the action it represents, so it translates well into any specification with minor adjustments.

B. ACTIONS AND VARIABLES

The data needed to track SRL actions is provided by the traces for each of the selected SRL strategies (defined in the fourth step of the analysis described above) plus the

TABLE 1. Selected strategies and its classification.

Metacognitive and behavioral strategies	Planning	Goal definition
		Goal planning
		Step identification
		Task definition/identification
		Time estimation
		Time organization/schedule
		Event deadline
	Monitoring and control	Monitoring task performance
		Step recording
		Time control
		Monitoring comprehension level
	Evaluation, reflection and reaction	Experimentation (strategy use and rating)
		Self-criticism
		Evaluation of results
		Performance evaluation
		Strategy use evaluation
		Strategy adaptation
		Causal attribution
		Time use evaluation
		Reactions after evaluation of results
Motivational strategies		Expected results
		Rewards
Context strategies		Help seeking
		Collaboration
Cognitive strategies	Search, gathering and selection of information	Selection of information sources
		Information expansion
		Summary and synthesis*
	Acquisition, processing and use of information	Information acquisition
		Vocabulary control
		Annotation (note taking, highlighting)
		Reading speed*
		SQ3R*
		KWL*
		Information organization (concept maps, mind maps, tables, semantic networks and schemes)
		Memorization (grouping, acronyms, analogy, imagery, mnemonics)
		Content/information review
		Problem solving (generic and science)*
		Brainstorming
		Dissertations and monographs*

*These strategies can be provided as a guide for the students.

context of the action (time and other information like location, instructor, etc. if needed). The main variables that enable tracking these actions are already defined by LA standards, like Tin Can API (xAPI) or IMS Caliper, as shown in the next section. In general, the basic variables to represent any action are the time when the action took place, the user who was the performer, the action that was performed, the object of the action (if any) and the results produced (if any). These basic variables are shared by both standards and by most proprietary systems in the examples above. That set of variables describes an action, but does a single action describe a strategy? In a SRL monitoring scenario, single actions provide information about simple strategies, but due to the complexity of most strategies, this information will be partial and will need other traces to be comprehensive. In general, an action will be followed by other actions that would affect the same object, which eventually will not be needed again. For instance, a task can be defined, planned, performed and completed, following several states. The information about planning the action is relevant, but it is part of a bigger process. Other actions, like single events, will be isolated and self-defined. This way, two types of actions were identified: event-based and state-based. Event-based can be recorded with a single trace and state based need more than one trace to record the state evolution, which should be limited to a final state.

The different nature of each strategy makes it impossible to define a single interaction model that works for every action, so each group of actions was considered separately.

VI. THE XAPI SRL PROFILE

The xAPI SRL profile [40] is already available online at the xAPI official repository: *The Registry*. In this section we detail the process followed to transform the actions identified at each strategy into xAPI *statements*, and how such strategies were grouped into xAPI *recipes*.

TABLE 2. xAPI-SRL recipes.

Goal and task management	Defines the actions related to learning project management using tasks and goals as the core elements.
Planning	Is aimed at recording goal and task planning events, although it can be used with other elements.
Time and flow control	Focuses on recording the use of time, enabling controlling the time and activity flow across applications
View control	Defines the actions related to the act of viewing learning content, i.e., reading a document, watching a video, etc. It is complemented by the time and flow control recipe.
Resource and learning content management	Defines the actions related to the management of learning objects or resources. Its actions can be applied to any other element that can be added to a container, removed from it, or edited in it. For instance, adding a video tutorial to a task.
Annotation	Defines the main annotation actions, adding semantics to the notes (note, doubt, solution, etc.)
Curation	Defines content curation actions. It can be applied also to other elements as strategies or any kind of resources. For instance, rating a resource or selecting a strategy as favorite, etc.
Motivation	Contains the actions related to motivation strategies. It can complement any recipe related to strategies in which the learner can use rewards or expected results to control motivation. In most cases, the motivation strategies will be related to a goal or a task. For instance, a reward could be set for completing a particular book, which can be a goal or a task, depending on the user's definition.
Strategy management	Defines the actions related to the management of (SRL) strategies. These strategies can be embedded in the software or described as a set of instructions that typically would be stored in a repository or collection. For instance, enabling or disabling a strategy or adding a strategy to a repository.
Step sequencing	Defines the actions related to the management of sequences of steps. This can be used for elements that can be divided into them, like strategies or tasks.
Vocabulary control	Defines the actions related to the management of a vocabulary collection.
Self-monitoring and self-evaluation	Defines self-monitoring and self-evaluation actions.

A. PROFILE STRUCTURE: RECIPES AND RECOMMENDATIONS

An xAPI profile is formed by one or more *recipes*. In this particular case, the large number and variety of strategies makes dividing the identified actions into recipes necessary. Although the first idea was using the strategy classification presented in section III.A to sort the actions into xAPI recipes, a different criteria based on strategy-related software functionalities was preferred, seeking for practicality and coherence with the final use of the profile. Attending to this criteria six main functionalities (or groups of activities) that potentially provide support for SRL strategies are recognized. The first four can be easily found on available software:

- Project or task management and planning
- Information and resource management
- Time management
- Content delivery
- Strategy management

Self-monitoring

- and self-evaluation

Taking these 6 groups as a foundation for the recipes would make some of the actions to be in several recipes. For this reason a mixed criteria of functionality and strategy relationship was used, resulting in 11 recipes [40], shown in Table 2. This is, each recipe contains statements for strongly related strategies that are supported by the same functionality. Finally, for each of the six functionalities listed above, a list of recipes should be considered. For instance, an application that provides content delivery (like a LMS) should implement the following recipes: view control, time and flow control, curation and annotation.

B. STATEMENT DEFINITION AND VOCABULARY

The process of defining the xAPI statements was not trivial, as the wide variety of strategies selected implies a wide variety of different actions which at the same time generate different types of results. Besides this, the versatility of xAPI makes that one action can be recorded in several ways, so a unique way of recording each action should be defined. Thus, each strategy was studied separately while trying to maintain coherence in statement definitions. Similarly, ADL's

TABLE 3. New additions to the xAPI vocabulary resulting from the SRL profile definition.

xAPI attribute	New vocabulary
Verb	defined, estimated-duration, performed-offline, expected, arranged (in collection, use in combination with the extension *position*), selected (from collection), discarded, enabled, disabled and personalized.
Activity Type	project, goal, step, strategy, embedded-strategy, reward, resource, doubt, solution, to-review, vocabulary-word, simple collection and mixed collection.
Extensions	position, reflection, condition-type, condition-value, purpose and collection-type.
xAPI-SRL specific activities	vocabulary-collection and strategy-collection

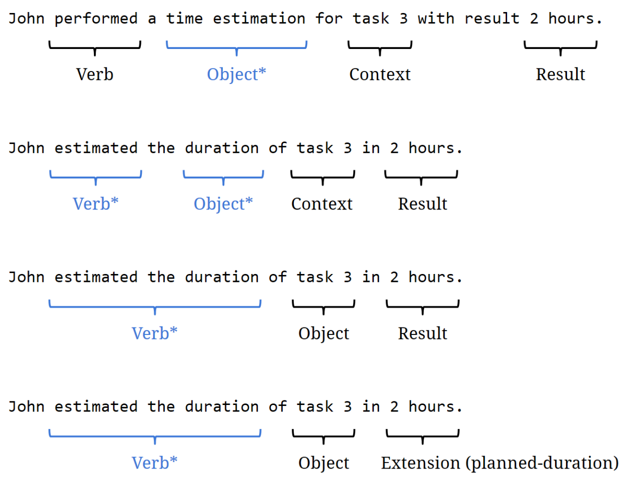


FIGURE 3. Time-estimation statement definition options (*new vocabulary).

recommendations related to the use of existing verbs, activity types and extensions available in the Registry were followed. Fig. 3 shows an example of how the time estimation statement was defined, starting from different options that use different xAPI attributes to store the information and selecting the most suitable structure, illustrating the versatility issue. The first option uses an existing verb, performed, while the activity type of the object would be a new definition (time estimation, marked with a *). The second option uses both new verb and activity type. In both cases, the verb does not define the action completely, which is strongly recommended by ADL, and the recipient of the action (task 3 and 2 hours) is stored in the context, rather than in the object, where it should be. The third and fourth options fix these problems adding a new verb estimated the-duration, but lay out a new one: where to record the time estimation itself. The duration field in the Result attribute is devoted to store the duration of the action, i.e., the duration of the time estimation. At the time when this statement was created, there was already in the Registry an extension called planned-duration that suited semantically the purpose of the time estimation, being the one selected to store it. This would allow to record the duration of the time estimation action in the future if needed.

Related to the second issue, the use of existing verbs, activity types and extensions, it is necessary to mention that some particular cases needed new definitions to represent the SRL actions in a consistent way. A summary of the new xAPI vocabulary resulting from the definition of this profile is shown in Table 3. Below, the adjustments and considerations made to each field of the xAPI statements are introduced:

- *Actor*. The main *actor* is the learner, and educators have a secondary role. All the actions related to managing tasks, viewing content, annotation and so on are performed only by the learner. Monitoring and evaluation actions can be performed by the learner or the educator. The actor needs a user ID from any account, like an email or a Twitter account.
- Actions are defined by the *verb*. Each verb is referenced with a unique internationalized resource identifier (IRI) which returns the verb definition in a machine-readable format. In this profile, 10 new verbs are proposed: defined, estimated-duration, performed (offline), expected, arranged (in collection, use in combination with the extension *position*), selected (from collection), discarded, enabled, disabled and personalized.
- *Object*. xAPI offers four types of *object*: Activity, Agent (or group), Sub-statement (another statement) or StatementRef (a reference to a past statement). In this profile just two types are proposed: Activity and StatementRef. For the object type Activity, the xAPI-SRL profile uses 21 Activity Types already defined by the community and 13 new Activity Types that were defined for this profile: project, goal, step, strategy, embedded-strategy, reward, resource, doubt, solution, to-review, vocabulary-word, simple collection and mixed collection.
- The *context* is used to provide information about the parent element of the object, the collection or course related to it and, if necessary, a statement to which the action is referred or related. Any activity that is set as context may be classified into one of these groups: Parent, Grouping, Category (tag) or Other. For instance, for the statement "Anna answered 3/5 to question 3 of Maths II final exam from 2nd grade course", the object is the question, the Parent context is the exam and the Grouping context is the course.

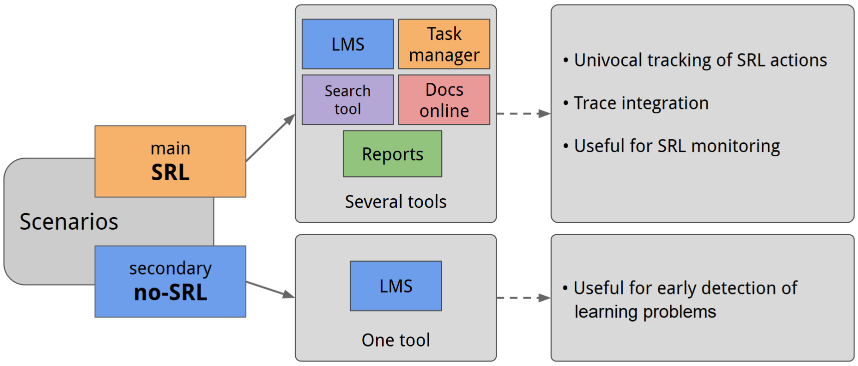


FIGURE 4. **SRL and non-SRL scenarios.**

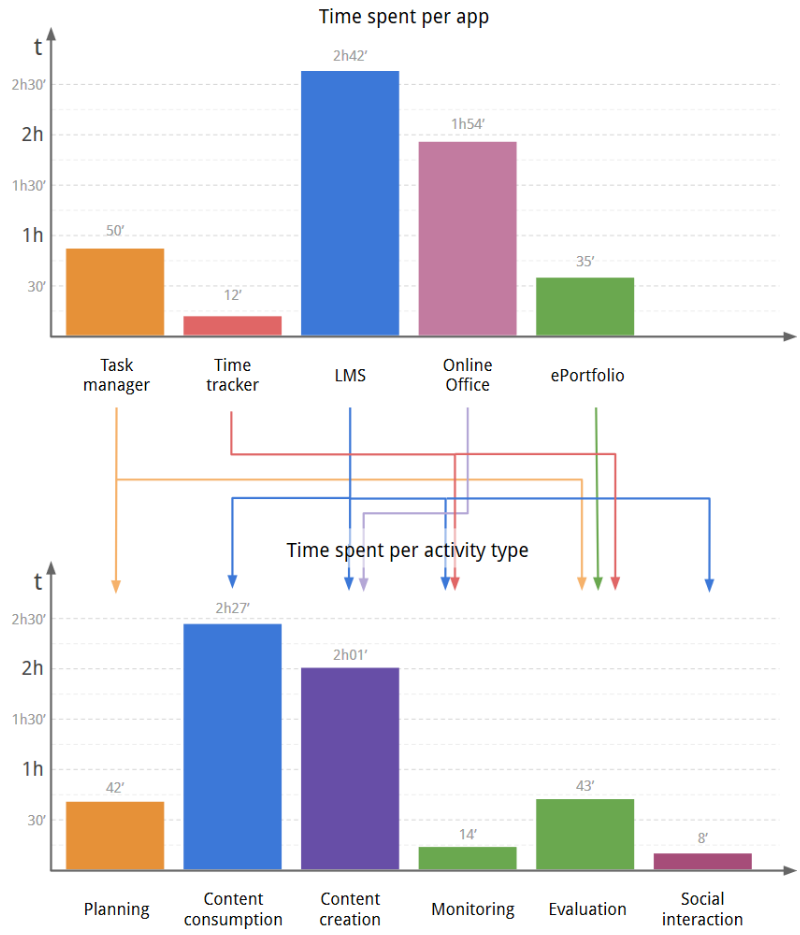


FIGURE 5. **Grouping criteria: data origin (above) vs. activity type (below) [19].**

- The *result* field was designed to store the duration of the action and a numeric and/or text value.
- Some recipes use *extensions* to record additional information into the objects, results or context. However, we tried to minimize its use following ADL's recommendations while trying to record all the information needed to describe the action. This profile added seven new extensions to xAPI: position, reflection, condition type, condition value, type, purpose and collection type.

For the sake of clarity, a final summary is presented in Table 4. It gathers all the xAPI-SRL profile recipes including the Activity Types and Verbs considered in each recipe. An example in its xAPI readable format is also provided for each different verb. Note that the "human readable" version of the xAPI statement was used to keep readability and get a manageable table size. The full version of the xAPI-SRL profile is available in [40].

TABLE 4. xAPI-SRL recipe summary.

Recipe	Activity Types	Verbs	Example statements
Resource and learning content management	article, page image, audio, video, slide, slide deck file, game, media (generic type), book module, course, device note, resource	*added* *edited* *deleted*	Estella added the article "guide to understanding aperture" to the goal "aperture". Estella edited the article "guide to understanding aperture". Estella deleted the article "guide to understanding aperture" from the goal "aperture".
View control	article, page image, audio, video, slide, slide deck file, game, media (generic type), book module, course, device note, resource	*previewed* *viewed* *read* *reviewed*	Estella previewed the article "guide to understanding aperture". Estella viewed the video "3 keys to a perfect exposure". Estella read the article "guide to understanding aperture" with reflection "review section 2". Estella reviewed the article "guide to understanding aperture".
Task and goal management	task goal project	*created* *defined* *edited* *deleted*	Estella created the task "review chapter 2" with description "focus on possible questions". Estella defined the goal "learn how to use aperture" with description "try manual mode and practice!". Estella edited the task "review chapter 2" with a new description "focus on possible questions for the exam". Estella deleted the task "review chapter 2" from "Unit 1: The Camera".
Time and flow control	task goal resource performance	*estimated-duration* *performed* *started* *suspended* (intended for ending a learning episode) *paused* (intended for interruptions, records the pause) *resumed* *completed*	Estella estimated the duration of "task 3" in 2 hours and 30 minutes. Estella started "task 3" at 2:00pm. Estella suspended "task 3" at 3:00pm. Estella performed "task 3" for 1 hour. Estella resumed "task 3" at 5:00pm. Estella paused "task 3" at 5:15pm. Estella resumed "task 3" at 5:30pm. Estella completed "task 3" at 18:00pm with a total duration of 2h 45'.
Planning	task goal project step strategy event	*scheduled* *set-deadline** *selected, discarded*	Estella scheduled "task 3" to 25/11/2015 3:00pm. Estella set a deadline for the task "review chapter 2" for the 5/1/16 at 15:00. Estella selected the strategy "going over content" for the task "review chapter 2". Estella discarded the strategy "going over content" from the task "review chapter 2".
Curation	task, goal, project step, strategy, article, page image, audio, video, slide, slide deck file, game, media (generic type), book module, course, device note, resource	*rated* *favorited, unfavorited* *selected, discarded* *shared, unshared*	Estella rated the resource "guide to understanding aperture" with 4 stars. Estella favorited the resource "guide to understanding aperture". Estella unfavorited the resource "guide to understanding aperture". Estella selected the resource "guide to understanding aperture" for task "Take test pictures". Estella discarded the resource "guide to understanding aperture" for task "Take test pictures". Estella shared the resource "guide to understanding aperture" in facebook.
Annotation	highlight underline note doubt solution to-review*	*annotated* *modified annotation*	Estella annotated "the higher the f stop number, the smaller the aperture" as a highlight in "guide to understanding aperture". Estella annotated "ask Mark about blades and bokeh" as a doubt in guide to understand aperture". Estella annotated "guide to understanding aperture" to review.
Vocabulary control	vocabulary word vocabulary collection (based on simple collection)	*added, edited, deleted*	Estella added "diaphragm" to "photography words" vocabulary collection.
Motivation	reward task, goal, project step strategy resource	*set*, earned* *expected (results)*	Estella set a reward "ice cream!" for completing the task "study chapter 2". Estella earned the reward "ice cream!" for completing the task "study chapter 2". Tom expected the results "understand centrifugal force" from the task "review chapter 2".
Strategy management	embedded strategy strategy strategy collection (based on simple collection)	*enabled, disabled (embedded strategy)* *personalized* *added, edited, deleted*	Estella disabled the "completion progress" strategy. Estella added "KWL" to "reading comprehension" strategy collection. Estella personalized the "KWL" strategy.
Step sequencing	goal task strategy	*added, edited,* *removed* *arranged*	Estella added the step "identify main ideas" to the task "review chapter 2". Estella removed the step "identify main ideas".

* in curation by ADL

In order to provide a detailed example, an XML excerpt of a statement from the *time and flow control* recipe is included. In this example, the learner pauses a task after one hour of work, resulting in a new state for the object. The xAPI readable version of the statement is 2016-05-18T07:55:32.654Z: "David Winters paused 'Summarize lesson 2'."

```
{
 "actor": {
   "objectType": "Agent",
   "name": "David Winters",
   "mbox": "mailto:dwinters@terra.es"
   },
 "verb": {
   "id": "http://id.tincanapi.com/verb/
          paused",
   "display": {"en-US": "paused"}
},
"object": {
  "id": "http://217.116.0.237/taskmgr/
        usr003/26",
  "definition": {
    "name": {
      "en-US": "Summarize chapter 6"
    },
    "description": {
      "en-US": "Review the chapter and
       look for practical examples"},
    "type":"http://adlnet.gov/expapi/
     activities/task"
   },
   "extensions":{
    "http://id.tincanapi.com/extension/
     attempt-id": "1"
   },
   "objectType": "Activity"
  },
  "result": {
    "duration": "PT57M35S"
  },
  "context": {
    "contextActivities": {
  "parent": {"id":"http://217.116.0.237/
                  taskmgr/usr003/22"}
  "category": [
    {
      "id": "http://id.tincanapi.com/
    recipe/srl/time-flow-control/0.1",
      "definition": {
        "type":"http://id.tincanapi.com/
                 activitytype/recipe"
          },
       "objectType": "Activity"
         }
    ]
  }
},
"timestamp": "2018-05-27T18:13:01.480Z"
"stored": "2018-05-27T18:13:02.040Z",
}
```


In this example, the actor has a mbox identifier, the verb has an id and an English readable version, the object is a task (type) called "Summarize chapter 6" which has an extension that contains the identifier of the attempt, the result includes the duration of the attempt (57'35") and the context includes the parent element of the object and the category attribute, which includes the information about the recipe to which this statement belongs, in this case, the xAPI-SRL recipe. The statement ends with the timestamp and the stored attributes, representing the time when the statement was created and the time when the statement was stored.

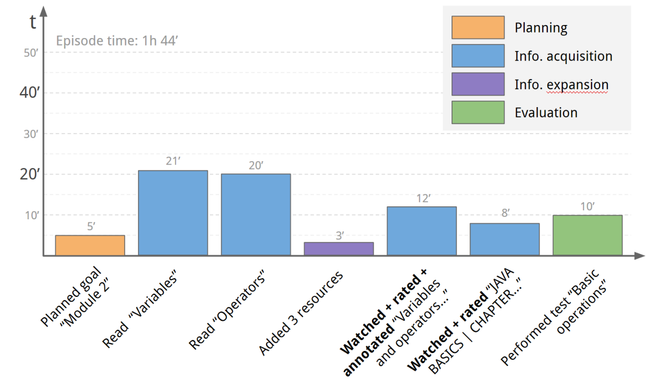


FIGURE 6. Time per activity for episode 1 (SRL scenario).

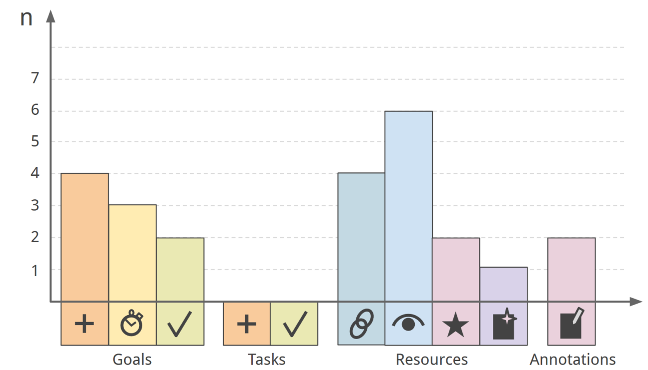


FIGURE 7. Number of time each strategy (activity) is performed.

C. CASE STUDY: SRL VS. NON-SRL

In order to demonstrate the usefulness of this profile, we used traces resulting from two different scenarios that represent the activity performed by a self-regulated learner and a traditional learner during a week of a course (Fig. 4). In the SRL scenario, the learner used different tools to perform several activities, including the use of strategies through the tools. In the non-SRL scenario, the learner used the LMS and a traditional learning approach.

The traces from both scenarios were selected and processed following the procedure presented in [41]. First, the traces were filtered and sorted for each graph, following the next steps:

1. Filtering by author.
2. Filtering by profile, in this case, xAPI-SRL.
3. Filtering by time window and sorting the records time wise.
4. Filtering or organizing by object, context, etc. if needed, depending on the purpose of the query.

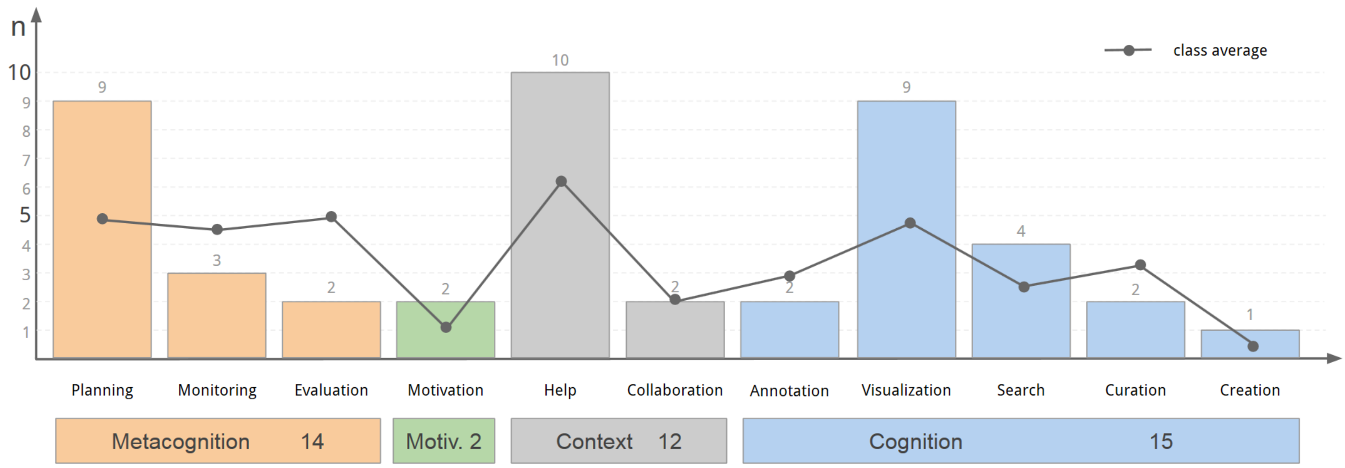


FIGURE 8. Number of actions per activity type.

The last step is grouping the traces and processing them to create the graphs. A processing system based on the combination of variables was used. This is based on the idea of combining two related variables whose comparative to a reference element would provide new information. For instance, selecting the estimated time and the spent time for a particular task (reference element) generates a differential value that represents the deviation of time use from the estimated time. This deviation provides new information that was not present in the original two variables. The key is selecting two variables that will create a meaningful comparison. This may need some processing in some cases. Following the previous example, the time estimation for one task can be obtained from one statement, whether the time spent on that task may require some processing, since the final value must be calculated from the information stored in several records.

The way in which the traces are grouped to create the graphs is crucial and will decide the usefulness of the resultant information. The criteria used to combine the traces should be meaningful for the purpose of the graph or report. To illustrate this concept, Fig. 5 [19] shows an example of two graphs created from the same group of traces from multiple sources but combining and analyzing them using different criteria. The graph at the top shows the time a user spent at each application, thus the criterion is the origin of the data. This criterion does not consider the purpose of the actions performed in each application and does not provide information about the SRL process. Meanwhile the graph at the bottom shows the time the same user spent on each type of activity. For this graph the traces were grouped by the type of activity regarding the strategies. This second criterion provides significant information about the SRL process. The graph at the bottom is an example of how the xAPI-SRL profile can help providing useful SRL information to improve SRL monitoring. Grouping the traces according to SRL criteria implies knowing the relationships between the statements and the strategies, which is inherent to how the profile was structured, as noted in section VI.A.

Figures 6 to 8 show graphs that represent the information obtained applying this processing idea to the traces. Fig. 6 shows the time spent on each activity for the self-regulated learner in the first learning episode for the second module of the course. It shows different types of activities in different colors: planning related in orange, acquisition, processing and use of information related in blue, search and selection of information related in purple and evaluation, reflection and reaction related in green. Fig. 7 shows the number of times each strategy was used by the same learner through that module. It shows the goals and tasks that were planned and completed, the resources that were linked, viewed, rated and created and the number of annotations. Fig. 8 is a teacher specific graph that shows the number of actions of the learner grouped by activity type for a period of time (a week in this case), compared to the average of the class, grouped at the same time by SRL area and strategy group. Note that, especially in figures 6 and 7, the information comes from multiple sources.

VII. DISCUSSION, CONCLUSIONS AND FUTURE WORK

This paper introduces a SRL application profile for the xAPI specification that enables SRL tracking for any xAPI compatible software. Taking into account the different areas and phases involved in SRL and focusing on learning strategies, we identified and defined a set of relevant traces for monitoring SRL. These traces were defined as xAPI statements and were classified in recipes according to the strategy of origin and its related software functionality. The main objective of this profile is improving the support for SRL monitoring defining what to record and enabling trace interoperability, thus enabling the use of traces from multiple sources for analysis and processing in learning analytics systems.

In the introduction we laid out two questions that we considered relevant to achieve our goal, which was the creation of a standard or specification to represent the main SRL actions, thus enabling future monitoring SRL possibilities.

Question 1 referred to what information is relevant to enable a comprehensive SRL monitoring. The conclusion of this work is that the information resulting from the implementation of SRL strategies, i.e., the traces generated by their software implementation, is suitable to represent the main SRL actions. This is, the strategies are an adequate source of information to track SRL activity. Note that most of these strategies are already implemented in available software, although they may offer partial functionalities. For instance, a generic task manager offers a good starting point for the main planning strategies. This is linked to question 2, referred to how we gather and represent relevant SRL information. In order to define exactly what traces would each strategy generate, we analyzed the inherent processes to identify each trace and the data that fully represent each action. Then, we selected xAPI to encode the data of each identified trace and explained the process followed for defining the statements and structuring them into recipes to create the xAPI-SRL profile. Then we used traces from two scenarios from which we were able to create different graphs for monitoring SRL.

After using xAPI to create the profile we can conclude that xAPI is suitable to represent the complexity of SRL actions. Its versatility and openness allowed us to define a wide and comprehensive selection of SRL actions, directly related to the selection of the most relevant SRL strategies. However, the translation from the strategies to the profile was not trivial. Every action had to be tackled separately while trying to maintain coherence throughout the profile. Besides, the large amount of strategies and actions and the repetition of certain actions across several strategies led to a semi-complex solution to classify the statements in recipes. However, we provide a recommendation for several types of software functionalities to ease its use in the profile.

This profile could be the first link of a chain of solutions towards the realization of a specific SRL learning analytics system for both learners and educators. It can provide several *benefits* in this direction:

- First, it can act as a *guide for developers*. Its public and open access makes it available to everyone and the classification in recipes and the recommendation for software functionalities makes it easy to know which recipe should be implemented without being a SRL expert.
- At the same time, it can be the *foundation to create reporting software for SRL*. The most common case, as stated in the introduction, is that learners use different software tools for different learning purposes. The xAPI specification grants *interoperability*, so traces from different tools can be combined to generate valuable information to monitor the learning process and learners' SRL skills and strategy use in a comprehensive manner. The use of reporting tools with this information may boost monitoring capabilities for both learners and educators. However, it is necessary a wide adoption to make this possible, and this may encounter some barriers, being the main one the need of compromise from developers and brands to adopt xAPI and this profile, even though they may not be interested in SRL. Promoting this profile and showing its benefits may be a relevant task for immediate future.

Another relevant task is publishing the self-monitoring and self-evaluation recipe. It is still under development due to the complexity caused by its inherent recursion. The self-monitoring process itself is relevant and may seem that recording it leads to a loop. It is part of the SRL core and information about which graphs were displayed, what information was visualized, what reflections and conclusions did the learners' produce, etc. is really important for teachers to see how their learners are self-regulating, and for learners to know if they are monitoring themselves properly, with the right frequency, etc., whether comparing to other learners or using automatic alerts in case of decreasing monitoring activity or abandoning it.

REFERENCES

[1] P. R. Pintrich, "A conceptual framework for assessing motivation and self-regulated learning in college students," *Educ. Psychol. Rev.*, vol. 16, no. 4, pp. 385–407, 2004.

[2] J. C. Núñez *et al.*, "Implementation of training programs in self-regulated learning strategies in Moodle format: Results of a experience in higher education," *Psicothema*, vol. 23, no. 2, pp. 274–281, 2011.

[3] P. Solano, J. C. Núñez, J. A. G. Pineda, and P. Rosário, "El aprendizaje autorregulado como medio y meta de la educación," *Papeles Psicologo*, vol. 27, no. 3, pp. 139–146, 2006.

[4] B. J. Zimmerman, "Investigating self-regulation and motivation: Historical background, methodological developments, and future prospects," *Amer. Educ. Res. J.*, vol. 45, no. 1, pp. 166–183, Mar. 2008.

[5] B. J. Zimmerman, "Theories of self-regulated learning and academic achievement: An overview and analysis," in *Self-Regulated Learning and Academic Achievement*, B. J. Zimmerman and D. H. Schunk, Eds. Mahwah, NJ, USA: Erlbaum, 2001, pp. 1–37.

[6] B. J. Zimmerman, "Becoming a self-regulated learner: An overview," *Theory Pract.*, vol. 41, no. 2, pp. 64–70, 2002.

[7] P. H. Winne and A. F. Hadwin, "Studying as self-regulated learning," in *Metacognition in Educational Theory and Practice*, D. J. Hacker and A. C. Graesser, Eds. Mahwah, NJ, USA: Lawrence Erlbaum, 1998, pp. 204–277.

[8] A. Brown, "Metacognition, executive control, self-regulation, and other more mysterious mechanisms," in *Metacognition, Motivation, and Understanding*, F. E. Weinert and R. H. Kluwe, Eds. Mahwah, NJ, USA: Lawrence Erlbaum, 1987, pp. 65–116.

[9] A. R. Groba, B. V. Barreiros, M. Lama, A. Gewerc, and M. Mucientes, "Using a learning analytics tool for evaluation in self-regulated learning," in *Proc. IEEE Frontiers Educ. (FIE)*, Madrid, Spain, Oct. 2014, pp. 1–8.

[10] M. Manso, M. Caeiro and M. Llamas, "Are learning software systems well-prepared to support self-regulated learning strategies?" *Int. J. Eng. Educ.*, vol. 32, no. 2, pp. 1015–1023, 2016.

[11] Z. Papamitsiou and A. A. Economides, "Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence," *J. Educ. Technol. Soc.*, vol. 17, no. 4, pp. 49–64, 2014.

[12] S. Katrina and M. Loganathan, "Application of big data in education data mining and learning analytics—A literature review," *J. Soft Comput.*, vol. 5, no. 4, pp. 1035–1049, 2015.

[13] P. Rojas-Castro, "Learning analytics: A literature review," *Educación Educadores*, vol. 20, no. 1, pp. 106–128, 2017.

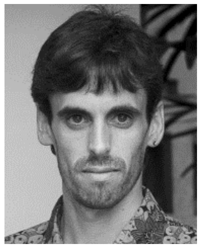
[14] A. Dutt, M. A. Ismail, and T. Herawan, "A systematic review on educational data mining," *IEEE Access*, vol. 5, pp. 15991–16005, 2017.

[15] I. Roll and P. H. Winne, "Understanding, evaluating, and supporting self-regulated learning using learning analytics," *J. Learn. Anal.*, vol. 2, no. 1, pp. 7–12, 2015.

[16] K. Verbert *et al.*, "Learning dashboards: An overview and future research opportunities," *Pers. Ubiquitous Comput.*, vol. 18, no. 6, pp. 1499–1514, 2014.

[17] N. Dabbagh and A. Kitsantas, "Personal learning environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning," *Internet Higher Educ.*, vol. 15, no. 1, pp. 3–8, 2012.

[18] M. Manso, M. Caeiro, and M. Llamas, "Analysis of self-regulated learning strategies oriented to the design of software support," in *Proc. Frontiers Educ. Conf. (FIE)*, Madrid, Spain, Oct. 2014, pp. 1–9.
[19] M. Manso, "Contributions to a better support of self-regulated learning in virtual learning environments," Ph.D. dissertation, Dept. Telematic Eng., Universidade Vigo, Vigo, Spain, 2017.
[20] P. Solano, "Elaboración y evaluación de un programa de mejora de la competencia en estrategias de autorregulación," Ph.D. dissertation, Dept. Psychol., Universidad Oviedo, Oviedo, Spain, 2006.
[21] Y. Gu, "Learning strategies: Prototypical core and dimensions of variation," *Stud. Self-Access Learn. J.*, vol. 3, no. 4, pp. 330–356, 2012.
[22] J. Nisbet and J. Shucksmith, *Learning Strategies*. Florence, Italy: Taylor & Francis, 1986.
[23] J. C. Núñez, P. Solano, J. A. González-Pienda, and P. Rosário, "Evaluación de los procesos de autorregulación mediante autoinforme," *Psicothema*, vol. 18, no. 3, pp. 353–358, 2006.
[24] C. E. Weinstein, A. C. Schulte, and D. Palmer, *Learning and Study Strategies Inventory (LASSI)*. Clearwater, FL, USA: H&H Publishing, 1987.
[25] P. R. Pintrich and E. V. De Groot, "Motivational and self-regulated learning components of classroom academic performance," *J. Educ. Psychol.*, no. 82, no. 1, pp. 33–40, 1990.
[26] B. Gargallo, J. M. Suárez-Rodriguez, and C. Pérez-Perez "El cuestionario ceveapeu. Un instrumento pa-ra la evaluación de las estrategias de aprendizaje de los estudiantes universitarios," *Revista ELectrónica Investigación EValuación Educativa*, vol. 15, no. 2, pp. 1–31, 2009. [Online]. Available: http://www.uv.es/RELIEVE/v15n2/RELIEVEv15n2_5.htm
[27] A. Bartolomé and K. Steffens, "Technologies for self-regulated learning," in *Self-Regulated Learning in Technology Enhanced Learning Environments*, R. Carneiro, P. Lefrere, K. Steffens, and J. Underwood, Eds. Rotterdam, The Netherlands: Sense Publishers, 2011, pp. 21–31.
[28] M. May, S. George, and P. Prévôt, "TrAVis to enhance students'self-monitoring in online learning supported by computer-mediated communication tools," *Int. J. Comput. Inf. Syst. Ind. Manage. Appl.*, vol. 3, pp. 623–634, 2011.
[29] J. L. Santos, K. Verbert, and E. Duval, "Empowering students to reflect on their activity with StepUp!: Two case studies with engineering students," in *Proc. ARTELL 2nd Workshop Awareness Reflection, CEUR*, 2012, pp. 73–86.
[30] B. Tabuenca, M. Kalz, H. Drachsler, and M. Specht, "Time will tell: The role of mobile learning analytics in self-regulated learning," *Comput. Educ.*, vol. 89, pp. 53–74, Nov. 2015.
[31] S. Govaerts, K. Verbert, E. Duval, and A. Pardo, "The student activity meter for awareness and self-reflection," *Hum. Factors Comput. Syst.*, pp. 869–884, May 2012.
[32] M. Wolpers, J. Najjar, K. Verbert, and E. Duval, "Tracking actual usage: The attention metadata approach," *J. Educ. Technol. Soc.*, vol. 10, no. 3, pp. 106–121, 2007.
[33] M. Bannert, P. Reimann, and C. Sonnenberg, "Process mining techniques for analysing patterns and strategies in students' self-regulated learning," *Metacognit. Learn.*, vol. 9, no. 2, pp. 161–185, 2014.
[34] A. Serrano-Laguna, I. Martínez-Ortiz, J. Haag, D. Regan, A. Johnson, and B. Fernández-Manjón, "Applying standards to systematize learning analytics in serious games," *Comput. Standards Interfaces*, vol. 50, pp. 116–123, Feb. 2017.
[35] IMS Global Consortium. (Dec. 2017). *Caliper Analytics IMSglobal Website. Internet*. [Online]. Available: http://www.imsglobal.org/activity/caliper
[36] A. Cooper, "Learning analytics interoperability-the big picture in brief," *Learn. Anal. Community Exchange*, pp. 1–7, 2014.
[37] P. Rosário, J. C. Núñez, and J. González-Pienda, *Cartas do Gervásio ao seu Umbigo. Comprometer-se com o Estudar na Universidade*, Coimbra, Portugal: Edições Almedina, SA, 2006, pp. 119–130.
[38] J. B. Carrasco, *Estrategias de Aprendizaje. Para Aprender Más y Mejor*. Madrid, Spain: RIALP S.A., 2007.
[39] *Helping You Develop Life Skills. Skills You Need. Internet*. Accessed: Aug. 2017. [Online]. Available: https://www.skillsyouneed.com/learning-skills.html
[40] M. Manso, M. Caeiro, and M. Llamas. ADL. (2016). *Profile: xAPI-SRL, the Registry*. Accessed: Aug. 2017. [Online]. Available: https://registry.tincanapi.com/#profile/60
[41] M. Manso, M. Caeiro and M. Llamas, "Development of a xAPI application profile for self-regulated learning requirements for capturing SRL related data," in *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, Tallin, Estonia, Mar. 2015, pp. 358–365.



MARIO MANSO-VÁZQUEZ (M'12) received the M.Sc. degree in telecommunication engineering and the Ph.D. degree in telematics from the University of Vigo, Spain, in 2007 and 2017, respectively. He was a software engineer and a graphic designer in several companies, moving his focus to UX. He is currently a Researcher with the University of Vigo and he is interested in PLEs, e-learning, and UX design. He has participated in GameůTel e-learning project as a Designer, a Software Developer and the Interoperability Manager. He has also participated in ITEC-SDE and edu-AREA projects as a Leading Designer. In addition to his academic and professional interests, he is a skilled musician and plays with local and national artists.

MANUEL CAEIRO-RODRÍGUEZ (M'08–SM'10) received the M.Sc. degree in telecommunication engineering and the Ph.D. degree in telematics (European mention) from the University of Vigo, Vigo, Spain, in 1999 and 2007, respectively. He has been a Visiting Researcher with Coimbra (Portugal), Rennes (France), Budapest (Hungary), and Kumamoto (Japan). He is currently an Associate Professor with the Department of Telematics Engineering, University of Vigo. He has participated in several research and development projects in the areas of e-learning, middleware, and Web engineering, and has published over 100 papers in international refereed journals and conference proceedings. He was a recipient of several awards from the IEEE and the WWW.

MARTÍN LLAMAS-NISTAL (M'92–SM'06) received the Engineering and Ph.D. degrees in telecommunication from the Polytechnic University of Madrid, Madrid, Spain, in 1986 and 1994, respectively. From 1994 to 1997, he was the Vice Dean of the Higher Technical School of Telecommunication Engineers, University of Vigo, Spain, where he was also the Head of the ICT Area from 1999 to 2003. He has authored or co-authored over 200 papers in peer-reviewed international refereed journals and conference proceedings. He has directed several national and international research projects in telematics and technology-enhanced learning fields. He is a member of the ACM. He is a member of the program committees of different international conferences and workshops. He has been an elected member of the Board of Governors since 2008 and also has been a member of the Strategic Planning Committee since 2009. He is deeply involved in activities organized by the IEEE Education Society (ES). Since it was founded in 2004, he has been a member of the IEEE ES Spanish Chapter Board, serving in different positions, including the Chairman of the IEEE ES Spanish Chapter from 2008 to 2010. He was a member of the Steering Committee of the IEEE Transactions on Learning Technologies (TLT) from 2008 to 2013. He is a member of the program committees of different international conferences and workshops. He has received several awards from the W3C (Highlight Paper in the WWW 2001, and Education Track Best Paper and Conference Best Paper Finalist in the WWW 2002) and from the IEEE (the 2007 Chapter Achievement Award for the Spanish Chapter as an outstanding model of technical activities, membership services, and professional development in Spain and Latin America, the 2010 Distinguished Chapter Leadership Award, the 2011 IEEE Education Society Chapter Achievement Award, and the IEEE EDUCON 2015 Meritorious Service Award). He was the General Co-Chair of the IEEE-EDUCON 2012, 2013, 2014, and 2018. Since 2014, he has been an Associate Editor of the IEEE-TLT. He co-founded the IEEE *Latin-American Learning Technologies Journal* in 2006, and serves since then as the Editor-in-Chief. He has been the Vice-President of the IEEE ES for Publications since 2011.