Recipe for Success — Lessons Learnt from Using xAPI within the Connected Learning Analytics Toolkit

Aneesha Bakharia
Information Systems School
Queensland University of Technology
Brisbane, Australia
aneesha.bakharia@qut.edu.au

Kirsty Kitto
Information Systems School
Queensland University of Technology
Brisbane, Australia
kirsty.kitto@qut.edu.au

Abelardo Pardo
School of Electrical and Information Engineering
The University of Sydney
Sydney, Australia
abelardo.pardo@sydney.edu.au

Dragan Gašević
Moray House School of Education and School of Informatics
University of Edinburgh
Edinburgh, Scotland
dgasevic@acm.org

Shane Dawson
Teaching Innovation Unit
University of South Australia
Adelaide, Australia
Shane.Dawson@unisa.edu.au

ABSTRACT

An ongoing challenge for Learning Analytics research has been the scalable derivation of user interaction data from multiple technologies. The complexities associated with this challenge are increasing as educators embrace an ever growing number of social and content-related technologies. The Experience API (xAPI) alongside the development of user specific record stores has been touted as a means to address this challenge, but a number of subtle considerations must be made when using xAPI in Learning Analytics. This paper provides a general overview to the complexities and challenges of using xAPI in a general systemic analytics solution - called the Connected Learning Analytics (CLA) toolkit. The importance of design is emphasised, as is the notion of common vocabularies and xAPI Recipes. Early decisions about vocabularies and structural relationships between statements can serve to either facilitate or handicap later analytics solutions. The CLA toolkit case study provides us with a way of examining both the strengths and the weaknesses of the current xAPI specification, and we conclude with a proposal for how xAPI might be improved by using JSON-LD to formalise Recipes in a machine readable form.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*; D.2.10 [**Software Engineering**]: Design—*Representation*; E.2 [**Data Structures**]: Data Storage Representation—*Composite structures, Linked representations, Object representation*

Keywords

xAPI, CLA toolkit, CLRecipe, Architecture, Learning Analytics, Learning Record Store

1. INTRODUCTION

Learning Analytics has evolved as a field of research that uses data driven methods to improve student learning processes and outcomes [21]. However, the learning process is complex and influenced by a wide variety of contextual and personal factors. Researchers have suggested that the true potential to offer meaningful insight comes from combining data from across different sources [17]. However, student learning data is commonly generated from numerous platforms, often with different underlying data structures. Hence, establishing a combined data set can be a challenging task. For example, a video annotation platform may be able to provide detailed accounts of its specific events (e.g., play, pause, stop, annotate, and comment), but when this data is combined with another dataset extracted from a social network platform the intersection in vocabulary may not overlap or be consistent. If these two data sets are to be used in the context of a learning experience, their terms, objects and actions need to be reconciled into a common notation, often a time consuming and difficult task.

The challenges involved in collating and analysing distributed learning events are well documented in the Learning Analytics literature, and numerous data formats have been proposed as potential solutions, including: Contextualised Attention Metadata [19], learning context ontologies (LOCO framework) [13], and ontologies for organizational learning (IntelLEO framework) [20]; and the very recently (October 2015) released IMS Caliper [3].

LAK '16, April 25-29, 2016, Edinburgh, United Kingdom

DOI: http://dx.doi.org/10.1145/2883851.2883882

1.1 Experience API

The Experience API (xAPI), provides a platform-neutral formalism to collect events occurring in any learning experience. xAPI was released in 2013 as the outcome of an ADL project that aimed to both: (i) improve interoperability between elearning systems that collect and exchange student learning data, and (ii) overcome the limitations of SCORM [2]. The xAPI specification [4] describes the format to represent discrete learning activities (as JSON statements) and the requirements for Learning Record Stores (LRS) that are able to collate and exchange learner records. The xAPI statement data format is based on WC3 Activity Streams 1.0 [1] with notable changes made to include results and context for a learning activity [8]. The design of xAPI has been influenced by the socio-cultural framework of Activity Theory [22] with the unit of analysis being the activity. As Activity Theory is closely related to constructivist learning theory, Kevan and Ryan [14] suggest that xAPI is ideally suited to tracking constructivist learning activities. However, the xAPI specification has no defined core vocabulary. In this case, the community is required to both define and share the structure of xAPI statements and the vocabulary as *Recipes* specific to a domain. These Recipes are analogous to the semantic definitions included in ontologies. Without them, xAPI only provides the syntactical structure to compose statements.

Example Recipes currently exist for attendance[1], video interaction[2] and open badges[3]. However, the current focus in the xAPI community upon data collection tends to mean that the analytics implications of vocabulary choice are often not considered. More complex Recipes need to be defined, and we consider it essential that the Learning Analytics (LA) community participate in this process, as poor design decisions will make it far more difficult to implement LA systems. Here, we share knowledge and lessons learned from participating in this process with the design of the Connected Learning Analytics (CLA) toolkit. We provide some of the best practice lessons that we have learned, along with guidelines for the appropriate capture and analysis of learning records.

2. EXAMPLE: THE CLA TOOLKIT

The Connected Learning Analytics (CLA) toolkit [16] is currently being created as a part of an Australian Government funded Office for Learning and Teaching project. This project aims to collate and analyse student behaviour within defined learning activities that are run in the "wild" using standard social media platforms, i.e. beyond an institution's adopted Learning Management System (LMS).

The CLA toolkit is open source (GPL3.0), and implemented in Python, using the Django web framework. It consists of two main components:

Data Collection is achieved by interfacing with standard social media APIs to retrieve specific data about student participation in a pre-defined learning activity. This data is stored in a Learning Records Store (LRS) using the xAPI format. Full functionality is currently implemented with the Facebook, Twitter, YouTube and Wordpress platforms.

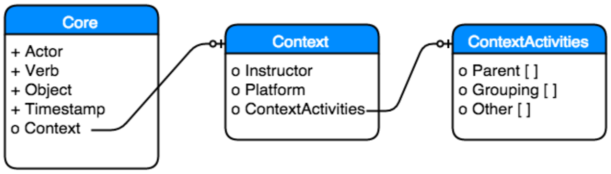

Figure 1: Simplified xAPI statement schema.

Analytics and Reporting are enabled by pulling data out of the LRS and storing it in a secondary database (presently PostgreSQL) which provides full functionality for querying the xAPI JSON document structure. Section 5 discusses the current reporting capabilities of the CLA toolkit, which has both student and instructor facing dashboards.

3. DESIGNING XAPI STATEMENTS

In this section, an overview of the structure of xAPI statements will be presented followed by a discussion on the advantages and disadvantages that xAPI syntax flexibility brings. xAPI statements contain 3 main elements, namely metadata (i.e., id, timestamp), descriptive information (i.e., actor, verb and object) and complementary data (i.e., context). xAPI statements are made up of <*subject*>, <*verb*> and <*object*> triplets. Each verb and object in an xAPI statement requires a unique identifier that resolves to a URL that contains the required metadata although no specific schema (i.e., typing) needs to be defined.

The <*subject*>, <*verb*>, <*object*> triplet representation is an oversimplification of the xAPI syntax, as well as being misleading in terms of the actual data required for analytics. Our experience in developing the CLA toolkit has shown that the correct modelling and population of contextual data within xAPI statements is critical. This is the context sub-section within an xAPI statement, which includes the instructor, team, and other important information about the learning context. In particular, the inclusion of the platform and the ability to link the activity with a course, a course section and an instructor are all equally important from an analytics perspective and should not be neglected when designing the mandatory fields that must be populated in xAPI statements designed for Learning Analytics. We consider it essential that all statements include a reference to uniquely identify the learning experience (e.g., course, event, field trip) and platform (e.g., Moodle). All statements generated by the CLA toolkit include the course code and the originating social media platform.

When comparing xAPI statements to traditional tabular log formats (e.g., Apache Server access logs or the Accumulator table in the Blackboard LMS), an additional advantage of xAPI emerges; the xAPI statement is in a JSON document format and is therefore able to encode multiple relationships. These can be included in the *grouping*, *parent* and *other* sections of *contextactivities*. The CLA toolkit takes advantage of this functionality and is able to include multiple @mentions, hashtags and tags with an activity occurring on a social media platform as *contextactivities.other*. While the ability to include multiple objects as items in the *group-*

---

[1] http://xapi.trainingevidencesystems.com/recipes/attendance/0_0_1/

[2] https://registry.tincanapi.com/#profile/44

[3] https://tincanapi.com/recipes-designers/

ing and *parent* sections of *contextactivities* provides much more flexibility than tabular log formats, there is ambiguity as to how these relationships should be used in analytics, because multiple objects of different types can be included.

The extensibility of xAPI means that new formats can be defined as JSON sub-documents and incorporated into the structure of an xAPI statement. For example, the CLA toolkit uses the rating extension to include numeric ratings of social media content[4]. However, the ability to define new JSON constructs as extensions without the use of a mandated JSON schema is problematic from an analytics perspective. The provision of a JSON schema for extensions would allow the LRS and subsequent analytically processing code to use the information contained within the extension in an automated way. There is ongoing discussion on whether JSON for Linked Data (JSON-LD) which incorporates object and value typing should become part of the xAPI specification. We shall not discuss this point here, but will return to this question in Section 6.

4. THE IMPORTANCE OF XAPI RECIPE DESIGN

The xAPI specification initially included a core vocabulary but this was removed from version 0.95 onwards with ADL favouring a community driven approach to defining verbs and activities [14]. Rustici Software currently hosts a repository[5] with community submitted verbs, objects and Recipes. From an analytics perspective, using a common vocabulary to represent similar activity is not just desirable; it is a necessity given that LRSs are designed to collate xAPI statements originating from disparate systems. xAPI Recipes have been proposed to address this need [7].

As no Recipe unifying the description of learning events in social media was available, the CLA toolkit project has designed an open Connected Learning (CL) Recipe. CLRecipe has played a crucial role in creating a consistent data model for social media activity, and its consistency has been tested through the efficient creation of analytics and visualisations showing temporal activities, content evolution, and social network analysis. CLRecipe describes a variety of different learning scenarios using a unified vocabulary:

Microblogging on platforms such as Twitter and Facebook, where users only post short notes.

Content Authoring of any long text that is written by a single user (e.g. a blog post made on Wordpress).

Content Curation of a collection of artefacts (e.g., documents, audio, video, images, etc.).

Table 1 contains the current mapping, which has simplified aggregate analytics across social media platforms. Enforcing the mapping in the Recipe played a key role in simplifying the processing required to obtain social media activity by platform and verb at a course and individual student level. The verb and object vocabulary used in CLRecipe were all selected from the core W3C Activity Streams 1.0 vocabulary, which was designed to provide streams of social media activity [1]. A description of each verb is available in the CLRecipe `readme.MD` file [15].

While xAPI statements represent discrete social media activities, these do not occur in isolation. This is because on social media, students might interact with content created by other students (e.g., they might like and share content), or directly comment on or reply to posts to create threaded discussions. Shares, likes and replies must include a reference back to the object being mentioned using *contextActivities.Parent*, which was chosen over using *contextActivities.Grouping* because the xAPI specification says that *contextActivities.Grouping* indicates an indirect relation while *contextActivities.Parent* represents a direct parent-child relation. The inclusion of the *parent id* creates a reference to the statement containing the post being commented on, replied to, liked or shared. Including the *parent id* in the xAPI statement allows for the construction of hierarchical relationships and is used to model threaded discussions. The use of a *parent id* to model a tree structure in a relational database is known as the adjacency list model [9].

Recipes are very loosely defined by a textual description of the verbs, objects, extensions used. No formal schema is enforced and relationships between statements need to be manually inferred before automated analysis can be performed. Invariably design decisions need to be made about which elements of an xAPI statement are used and these decisions need to be known by the system performing the analysis of xAPI statements (i.e. the design decisions and rules that a Recipe serves to enforce are not described in a machine readable manner).

5. PERFORMING ANALYTICS WITH XAPI STATEMENTS

xAPI statements are stored in a Learning Record Store (LRS). The CLA toolkit uses Learning Locker, which is an open source LRS built on MongoDB (a NoSQL database). A frequent complaint about the xAPI standard concerns the limited reporting functionality of LRSs [18]. The xAPI specification does not provide a RESTful interface to perform aggregate queries (e.g., counts of verbs and object) against the statements in a LRS. Only simple queries are allowed and all matching statements are returned in full. The inability to directly perform aggregate queries using the xAPI LRS document interface was a stumbling block for the CLA toolkit project. xAPI statements are now stored as JSON documents in a PostgreSQL database where aggregate queries can easily be performed using SQL syntax. PostgreSQL has been chosen over MongoDB because of its ability to store relational data (required by the CLA toolkit web application) and JSON documents.

5.1 Temporal Analysis

Within the CLA toolkit, temporal analysis involves aggregating social media activity over time. The CLA toolkit creates graphs showing verb use (i.e., like, share, post and comment) by platform over a specified time period. In terms of processing, the star schema commonly used in traditional business intelligence (BI) applications to create high dimensional cubes, was used to perform aggregate counts by date. For example, a table containing dates (i.e., the date dimenstion) is joined to a table with extracted core fields from an xAPI statement (i.e., the fact table) which is then joined to tables containing xAPI context information such as instructor, parent and grouping (i.e., the other dimensions for

---

[4] http://id.tincanapi.com/extension/quality-rating

[5] https://registry.tincanapi.com/

Table 1: xAPI Verb Mapping in CLRecipe.

	Create	Like	Share	Tag	Rate	Comment	Add
Facebook	Post	Like	Share	Tag	-	Reply	-
Google+	Post	Like	Share	Tag	-	Reply	-
Twitter	Tweet	Favorite	Retweet	Hashtag	-	-	
Blog	Post	-	-	Tag	Rate	Comment	-
Pinterest	Board	Like	Share	-	-	-	Pin
YouTube	Video	Like	Share	-	-	Comment	-

analysis). This relational design is able to facilitate dimensional cube creation and provide aggregates by time of day, day in week, month, and year. The inclusion of other information, such as the social media platform, a related course, and contextual information (such as a tag or @mention) provides additional dimensions for analysis. While these additional fields are often seen as optional in an xAPI statement, CLRecipe mandates their inclusion for the purpose of adding further meaning and insight into the analysis.

5.2 Content Analysis

CLA toolkit includes algorithms for content analysis. At present we have implemented: Topic Modelling using the Latent Dirichlet Allocation algorithm; sentiment analysis via the Valence Aware Dictionary and sEntiment Reasoner (VADER) algorithm [12]; and a Cognitive Presence classification from the Community of Inquiry model [11]. None of these analyses can be performed without access to the content associated with an xAPI entry. The xAPI specification only mandates that a unique URL for each object involved in the activity is provided, and not that it is accessible by the LRS or the system processing the xAPI statements. As such, if the LRS does not store the content from xAPI statements that originate in a firewalled system, then this functionality will not be available in the CLA toolkit. For this reason, storing the content of a social media items is recommended by the CLRecipe, although care should be taken to meet the legal terms and conditions of different social media.

5.3 Social Network Analysis

The CLA toolkit can perform SNA and displays sociograms that are filterable by platform and date at both a course and individual level. Forum replies, blog comment threads, @mentions, likes and shares are all stored as social relationships between the users performing the activities. In the CLRecipe, we refer to the post being shared, liked or commented on, using a *parent id*. This addition is essential as it creates a relationship with the post's statement and allows data such as the creators details to accessed and analysed. A social relationship table is built in the CLA toolkit which contains the post creator, creation date, platform and verb. This allows for a social network to be filtered by platform, date, and user, as well as to include different edge relationships (i.e., like, share, comment and mention). In Section 4, the use of an adjacency list model to represent the parent-child relationship between statements using a parent id was discussed. The adjacency list model however requires recursive queries in order to rebuild the hierarchical tree from individual statements. A few other more efficient techniques have been proposed, such as the nested set model, which will be reviewed for representation within xAPI syntax as the development of the CLA toolkit progresses. This model would facilitate far more efficient network reconstruction, as the ability to process threaded discussions is essential.

6. IMPROVING XAPI

While the xAPI statement specification is both flexible and extensible, within this paper, we have illustrated several shortcomings in relation to the way Recipes are currently described. These include a lack of strict typing (see Section 3) in extensions and Recipes as well as the lack of a machine readable way to communicate the relationship between statements (see Section 4). As the xAPI Data Interoperability Standards Consortium (DISC) forms [23], we see a chance emerging to start thinking about how xAPI might be improved, and propose that this could be done with an extension of the notion of Recipes. We propose the adoption of JSON-LD [6] as a solution to these issues, and here we will discuss the manner in which JSON-LD introduces stricter typing and how JSON-LD framing [5] can make the relationship between xAPI statements in a Recipe explicit and therefore machine readable.

In xAPI statements, each verb and object must include the identifier for the metadata describing the main properties. However, the metadata that is required by xAPI does not include data type information and this becomes problematic for non-trivial statements that use extensions and/or are part of larger Recipes. JSON-LD uses the "@context" property (not to be confused with the context in xAPI statements) to specify a URI with details on each property and its associated data type. JSON-LD also includes specific object types (e.g., Person, Place and Event) and data types (e.g., date, temperature, coordinates and floating point numbers) which would be beneficial from a xAPI statement processing point of view. The schema that JSON-LD provides will have similar advantages to XML schema in terms of validation and compliance.

```
{
  "@context": {
    "as": "http://www.w3.org/ns/
        activitystreams",
    "ex": "http://example.org/vocab#"
  },
  "@type": "as:Blog",
  "ex:contains": {
    "@type": "as:Like"
  }
}
```

Figure 2: A Frame for a Activity Stream.

The example JSON-LD frame in Figure 2 can be programmatically applied to give structure to a collection of statements that adhere to a Recipe. The frame ensures that the structure is predictable and that there is only one way for the programming code to be implemented even though the relationship between xAPI statements can take multiple forms. Combining the stricter typing of the JSON-LD with frames for removing the ambiguity in the encoding of statement relationships, we gain a useful way of sharing machine readable xAPI Recipes. JSON-LD can also easily be translated into RDF which opens up opportunities for linked semantic student knowledge graph processing [10].

7. CONCLUSION

The modelling of xAPI statements explored in this paper is based on our experiences in building learning analytics and visualisations for the CLA toolkit. Key to the current success in our project has been the careful attention paid to creating a Connected Learning Recipe. This consideration has facilitated the easy creation of a variety of reports common to standard Learning Analytics solutions. The key take away is that while xAPI is flexible and extensible, it is essential that analytics be considered when modelling xAPI statements using Recipes. In particular, we have found adding contextual information (which is usually seen as optional extra) is key to the provision of additional dimensions for temporal analysis. In terms of social network analysis and discourse analysis, attention needs to be given to the way relationships between statements are modelled, particularly for threaded discussions. While we have emphasised the importance of considering analytics in the creation of Recipes, the lack of machine readable Recipes is a core weakness inherent in the current specification of xAPI. For this reason we have proposed that xAPI be extended with the JSON-LD framework that has already been adopted by the Activity Streams 2.0 specification.

8. ACKNOWLEDGEMENTS

Support for this project has been provided by the Australian Government Office for Learning and Teaching. The views in this project do not necessarily reflect the views of the Australian Government Office for Learning and Teaching.

9. REFERENCES

[1] Activity streams. http://activitystrea.ms/. Accessed: 2015-10-12.

[2] ADL Initiative, Project TinCan. http://www.adlnet.gov/tla/tin-can. Accessed: 2015-10-12.

[3] Caliper Analytics. http://www.imsglobal.org/activity/caliperram. Accessed: 2015-10-29.

[4] Experience API. https://github.com/adlnet/-xAPI-Spec/blob/master/xAPI.md. Accessed: 2015-10-12.

[5] Json-ld framing specification. http://json-ld.org/spec/latest/json-ld-framing/. Accessed: 2015-10-23.

[6] Json-ld specification. http://json-ld.org/spec/. Accessed: 2015-10-23.

[7] Recipes. https://tincanapi.com/recipes-designers/. Accessed: 2015-10-23.

[8] M. Bowe. Tin Can vs. Activity Streams. http://tincanapi.com/tin-can-vs-activity-streams/, 2013. Accessed: 2015-10-24.

[9] J. Celko. Some answers to some common questions about SQL trees and hierarchies. http://www.ibase.ru/devinfo/DBMSTrees/sqltrees.html. Accessed: 2015-10-12.

[10] S. Dietze, S. Sanchez-Alonso, H. Ebner, H. Qing Yu, D. Giordano, I. Marenzi, and B. Pereira Nunes. Interlinking educational resources and the web of data: A survey of challenges and approaches. *Program*, 47(1):60–91, 2013.

[11] D. R. Garrison, T. Anderson, and W. Archer. Critical thinking, cognitive presence, and computer conferencing in distance education. *American Journal of distance education*, 15(1):7–23, 2001.

[12] C. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[13] J. Jovanović, D. Gašević, C. Knight, and G. Richards. Ontologies for effective use of context in e-learning settings. *Journal of Educational Technology & Society*, 10(3):47–59, 2007.

[14] J. M. Kevan and P. R. Ryan. Experience API: Flexible, decentralized and activity-centric data collection. *Technology, Knowledge and Learning*, pages 1–7.

[15] K. Kitto and A. Bakharia. CLRecipe. https://github.com/kirstykitto/CLRecipe, 2015.

[16] K. Kitto, S. Cross, Z. Waters, and M. Lupton. Learning analytics beyond the LMS: the connected learning analytics toolkit. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 11–15. ACM, 2015.

[17] S. Knight, S. B. Shum, and K. Littleton. Epistemology, assessment, pedagogy: where learning meets analytics in the middle space. *Journal of Learning Analytics*, 1(1):23–47, 2014.

[18] SaLTBOX. Why reporting in the LRS? http://blog.saltbox.com/blog/2015/09/23/why-reporting-in-the-lrs/, 2015.

[19] H.-C. Schmitz, M. Wolpers, U. Kirschenmann, and K. Niemann. Contextualized attention metadata. *Human attention in digital environments*, pages 186–209, 2011.

[20] M. Siadaty, D. Gašević, J. Jovanović, K. Pata, N. Milikić, T. Holocher-Ertl, Z. Jeremić, L. Ali, A. Giljanović, and M. Hatala. Self-regulated workplace learning: A pedagogical framework and semantic web-based environment. *Journal of Educational Technology & Society*, 15(4):75–88, 2012.

[21] G. Siemens. Learning Analytics: The Emergence of a Discipline. *American Behavioral Scientist*, 57(10):1380–1400, Aug. 2013.

[22] A. Silvers. Answers: How do i get started with xAPI? http://makingbetter.us/2014/11/answers-how-do-i-get-started-with-xapi/, 2014.

[23] A. Silvers. The way of xapi's consortium. *xAPI Quarterly*, 2015.