xAPI Deep Dive Verbs

In continuing with our “Anatomy of a xAPI Statement” series, here’s the next installment — verbs. In this post, I’ll tell you a *lot* about how verbs work with the Experience API. If you have any questions at all, please leave them in the comments below, or contact us here.

Inclusion in Statements

Verbs are a required part of statements and including them is simple enough. Set a Verb object into the “verb” property of a statement to indicate the action being taken for a given experience. A Verb object can consist only of an “id” property pointing to a URI (well, IRI). Here is an example:

```
{
    id: "http://adlnet.gov/expapi/verbs/experienced"
}
```

While that’s sufficient it seems people prefer something a little more akin to their own language, therefore Verbs should include a “display” property as well. The ‘display’ property’s value is a language map (a list of language codes with corresponding string values). Language maps are central to giving the xAPIAPI internationalized data interoperability. Here is the same verb, but with a human readable display value:

```
{
    id: "http://adlnet.gov/expapi/verbs/experienced",
    display: {
        "en-US": "experienced"
    }
}
```

Additional language values can be easily added to the language map using the RFC 5646 language tags, “en-US” above is an example of American English.

History

Early in the specification process there was a pre-defined set of verbs. In the development of the 0.95 version of the specification that list moved to the object form with full URI for “id” and was moved out of the specification proper in favor of letting new verbs be created at will. ADL still maintains a list of verbs that are specifically designed for the learning community, though there is no reason those verbs can’t be used for other purposes as well. Verbs such as “attempted”, “experienced”, “passed”, “failed”, “answered”, and “completed” (in their URI form of course) match up well with previous standards and have become some of the most common used in xAPI so far. It is expected that communities of practice will evolve to create their own set of specific verbs known and used within a particular community. And the exception to the rule, since they all have one, there is one predefined verb included in the specification which serves the special purpose of voiding a statement. To void a statement send a new statement with this special verb having id “http://adlnet.gov/expapi/verbs/voided” along with a statement ref (more about these in a future post) to any LRS that may have received the statement.

Past Tense

Verbs should be past tense. xAPI is designed to track experiences which by their nature are time based, consequently verbs are past tense because a statement has to be recorded (note past tense) for the experience. No matter how soon after a recorded experience is reported on, that portion of the experience must already be in the past. (This is also one of the reasons why a statement no longer indicates something as “in progress”.) The concept of time passing as relates to streams of activities, potentially within the same experience, provides a significant amount of the complexity required to

derive meaning from just a pile of statements, but at the same time provides the flexibility that allows content creators' imaginations to flourish.

Resolvability

Verb IDs as URIs mean that many verbs can be resolved to a location (URL), and when they do they can provide additional meta information. The meta information should contain an object with a "name" and "description" properties, at least when requested as JSON, per the specification. These properties are used to provide information specifically about the verb rather than the representation of the verb itself (what the "display" property is for). This is a good start and as verbs and xAPI evolve we'll have a way to extend the information surrounding verbs (and other items using URIs). The Internet purist in me says that because a Verb uses a URI and because I can pick a URI that can be a URL that I should, and that all verbs should resolve, but the specification (rightly so, cause I'm not always a purist) leaves it open that Verbs don't have to resolve.

To Coin, or Not to Coin

It really isn't a question! You should avoid coining new verbs except as a last resort. Okay, last resort may be a bit strong as early as xAPI adoption is, but eventually the set of verbs should move towards a fixed state. Consider the three required parts of a statement–actor, verb, object–only one of these, the verb, can be consistently matched across experiences for different people, or indicate different actions within an experience for the same actor. Those two dimensions are fundamental to reporting and don't work if every new experience comes with a whole new set of verbs. This is where the community of practice comes in, verbs will gain traction through adoption. As verbs gain traction their common use allows system implementers to rely on their semantic meaning which is the foundation of the interoperability that a specification like xAPI seeks to provide. Statement creators should look for and leverage existing verbs whenever possible.

Registry

We've taken to calling a list of verbs (and other URI based components) a "registry," and have implemented one, specifically The Registry where you can go to find Verbs that are being used in the wild. Right now it consists of the list of ADL verbs, but we are working on functionality (when not writing blog posts) to allow users to create new verbs through a curated process precisely to prevent the explosion of verb creation that could lead to less interoperability. Certainly we can't prevent people from coining and using new verbs, and new verbs will be necessary over time, but we also want to help those verbs to be ever lasting (as they need to be since statements are) and that those ever lasting verbs will continue to resolve, so we've opened up the "id.tincanapi.com" domain namespace to be used for URI based ids. Verbs (and other items) created in The Registry will have resolvable URLs that will serve the meta data associated with them as defined by the specification.

Meanings, Not Words

Verbs are tough, and English (other languages do too I'm sure) does its best to make them tougher. So far we've said to reuse verbs when possible to allow interoperability, but we've also said there will be communities of practice that will adopt their own set of verbs. There is a conflict in these two best practices that arises because a single verb may have different meanings depending on context, a synonym if you will. To say it another way to stress the impact of this, Verb objects have an identifier that maps directly to a singular meaning, not to a specific word. It is the meaning therefore that must match when determining when a Verb object with a given ID can be reused in different cases.

"Fired" is a commonly cited one, probably because it is the one used by the specification. The word "fired" has very different meanings depending on the situation in which it is used. The specification calls out this fact and suggests that verb IDs be used to separate these meanings, the problem with that is how to demarcate the line. For instance are "fired a gun" and "fired a cannon" different verbs? One could argue that

both are for the rapid expulsion of a projectile, so the same, similarly arguments can be made that the act of firing the two different instruments require significantly different skill sets, equipment, etc. which might make them different verbs. We are left with a gray (or is that grey?) area that will have to be filled in by systems consuming the statements and ultimately up to us fallible humans to step in and create meaning from difficult semantic relationships. Really only time and adoption can point us in the right direction when it comes to coining verbs and their synonyms.

To conclude, I mean to finish, I mean to sum up, I mean to wrap up… ah, the heck with it!