Timeline Of Learner Success

As learners engage in a blended eLearning ecosystem, they will build up a history of learning experiences. When that eLearning ecosystem adheres to a framework dedicated to supporting and understanding the learner, such as the Total Learning Architecture (TLA), it becomes possible to retell their story through data. One important aspect of that story is the learner's history of success.

1 Ideal Statements

In order to accurately portray a learner's timeline of success, there are a few requirements of the data produced by a Learning Record Provider (LRP). They are as follows:

- the learner must be uniquely and consistently identified across all LRPs
- learning activities which access a learner's understanding of material should report if the learner was successful or not
  - if the assessment is scored, the grade earned by the learner should be reported
  - if the assessment is scored, the minimum and maximum possible grade should be reported
- The learning activities must be uniquely and consistently identified across all LRPs
- The time at which a learner completed a learning activity must be recorded
  - The timestamp should contain an appropriate level of specificity.
  - ie. Year, Month, Day, Hour, Minute, Second, Timezone

1.1 statement parameters to utilize

The statement parameter locations here are written in JSONPath

- $\$.timestamp$
- $\$.result.success$
- $\$.actor$
- $\$.verb.id$

2 TLA Statement problems

The data collected at the TLA pilot run supports the following algorithm.

3 Algorithm

3.1 Summary

1. Query an LRS via a GET request to the statements endpoint using the parameters agent, since and until
2. Filter the results to the set of statements where:
   - $.*verb.id* is one of:
     - http://adlnet.gov/expapi/verbs/passed
     - https://w3id.org/xapi/dod-isd/verbs/answered
     - http://adlnet.gov/expapi/verbs/completed
   - $.*result.success* is true

3.2 Query an LRS via REST

How to query an LRS via a GET request to the Statements Resource [1]

```
Agent = "agent={"account":
                {"homePage": "https://example.homepage",
                 "name": 123456}}"

Since = "since=2018-07-20T12:08:47Z"

Until = "until=2018-07-21T12:08:47Z"

Base = "https://example.endpoint/statements?"

endpoint = Base + Agent + "&" + Since + "&" + Until

Auth = Hash generated from basic auth

S = curl -X GET -H "Authorization: Auth"
         -H "Content-Type: application/json"
         -H "X-Experience-API-Version: 1.0.3"
         Endpoint
```

3.3 Z Specifications

3.3.1 xAPI Schema

[*Statement*] [*Actor*] [*Verb*] [*Object*] [*Result*] [*Context*] [*Timestamp*]

[1] S is the set of all statements parsed from the statements array within the HTTP response to the Curl request. It may be possible that multiple Curl requests are needed to retrieve all query results. If multiple requests are necessary, S is the result of concatenating the result of each request into a single set

$$
\begin{array}{|l}
\hline
Statement \\
\hline
s : Statement \\
\hline
s = \{Actor, Verb, Object, Timestamp\} \mid \\
\quad \{Actor, Verb, Object, Timestamp, Context\} \mid \\
\quad \{Actor, Verb, Object, Timestamp, Result\} \mid \\
\quad \{Actor, Verb, Object, Timestamp, Result, Context\} \\
\hline
\end{array}
$$

- The variable s is of type $Statement$ and consists of an $Actor$, $Verb$, $Object$, $Timestamp$ and optionally $Context$ and $Result$

$$
\begin{array}{|l}
\hline
Statements \\
\hline
S : Statements \\
\hline
S = \{ s : Statement \mid S \neg \emptyset \} \\
\hline
\end{array}
$$

- The variable S is of type $Statements$ and is a set of objects s, each of type $Statement$
- The variable S is a non empty set

3.3.2 Timeline Leaner Success System State

$$
\begin{array}{|l}
\hline
TimelineLearnerSuccess \\
\hline
S_{extra}, S_{completion}, S_{success}, S_{failure} : \mathbb{P}\, S \\
\hline
S_{extra} \cup S_{completion} = S \\
S_{extra} \cap S_{completion} = \{\} \\
S_{success} \cup S_{failure} = S_{completion} \\
S_{success} \cap S_{failure} = \{\} \\
\hline
\end{array}
$$

- The sets $S_{extra}, S_{completion}, S_{success}, S_{failure}$ are the powerset of S
- The union of sets S_{extra} and $S_{completion}$ is equal to the complete set of statements S
- No values are shared between the sets S_{extra} and $S_{completion}$
- The union of sets $S_{success}$ and $S_{failure}$ is equal to the set $S_{completion}$
- No values are shared between the sets $S_{success}$ and $S_{failure}$

3.3.3 Initial State of Timeline Learner Success System

$$
\begin{array}{|l}
\hline
InitTimelineLearnerSuccess \\
\hline
TimelineLearnerSuccess \\
\hline
S_{extra} = \{\} \\
S_{completion} = \{\} \\
S_{success} = \{\} \\
S_{failure} = \{\} \\
\hline
\end{array}
$$

- The sets $S_{extra}, S_{completion}, S_{success}, S_{failure}$ are all initially empty

3.3.4 Filter for Completion

$$
\begin{array}{|l}
\hline
VerbIdCompletion \\
\hline
V_{completion} : Verb \\
\hline
V_{completion} = http://adlnet.gov/expapi/verbs/passed \mid \\
\qquad\qquad\quad https://w3id.org/xapi/dod-isd/verbs/answered \mid \\
\qquad\qquad\quad http://adlnet.gov/expapi/verbs/completed \\
\hline
\end{array}
$$

- The var $V_{completion}$ has a value of one of the above IRIs and is of type $Verb$

$$
\begin{array}{|l}
\hline
FilterForCompletion \\
\hline
\Delta TimelineLearnerSuccess \\
\hline
S'_{completion} = \{ s : Statement \mid V_{completion} \in s \land s \in S\} \\
S'_{extra} = \{ s : Statement \mid V_{completion} \notin s \land s \in S\} \\
\hline
\end{array}
$$

- The updated set $S'_{completion}$ is the set of all statements s where $V_{completion}$ is in s and s is in S
- the updated set S'_{extra} is the set of all statements s where $V_{completion}$ is not in s and s is in s

3.3.5 Filter for Success

$$
\begin{array}{|l}
\hline
ResultSuccessTrue \\
\hline
R_{successful} : Result \\
\hline
R_{successful} = true \\
R_{successful} \neq false \\
\hline
\end{array}
$$

- The var $R_{successful}$ has a value of $true$ but not $false$ and is of the type $Result$

FilterForSuccess

$\Delta TimelineLearnerSuccess$

$s_{completion} : Statement$

$s_{completion} \in S_{completion}$

$S'_{success} = \{ s_{completion} : Statement \mid R_{successful} \in s_{completion} \}$

$S'_{failure} = \{ s_{completion} : Statement \mid R \notin s_{completion} \}$

- The set $s_{completion}$ is of type $Statement$ and is in the set $S_{completion}$
- The updated set $S'_{success}$ is the set of all statements $s_{completion}$ where $R_{successful}$ is in $s_{completion}$
- The updated set $S'_{failure}$ is the set of all statements $s_{completion}$ where $R_{successful}$ is not in $s_{completion}$

3.3.6 Return

Return

$\Xi TimelineLearnerSuccess$

$S_{success}! : Statements$

$S_{success}! = S_{success}$

- The returned variable $S_{success}!$ is equal to the current state of variable $S_{success}$

3.4 Pseudocode

Algorithm 1: Timeline of Learner Success

Input: S
Result: $S_{success}$
while S *is not empty* **do**
  for each Statement s in S
  if $s.verb.id = V_{completion}$ **then**
    add s to $S_{completion}$
  else
    add s to S_{extra}
  end
end
while $S_{completion}$ *is not empty* **do**
  for each Statement $s_{completion}$ in $S_{completion}$
  if $s_{completion}.result.success = R_{success}$ **then**
    add $s_{completion}$ to $S_{success}$
  else
    add $s_{completion}$ to $S_{failure}$
  end
end

3.5 Result JSON Schema

JSON schema describing the returned data structure

3.6 Visualization Description

description of the associated visualization in english

3.7 Visualization prototype

This section will be updated to a prototype viz