DATASIM Requirements Report

Data and Training Analytics Simulated Input Modeler

27 May 2021

Distribution Statement A
Approved for public release: distribution unlimited.

DATASIM Requirements Report

Project DATASIM

Data and Training Analytics Simulated Input Modeler

Prepared by Shelly Blake-Plock, PI, Yet Analytics, Inc.
Submitted to ADL on 8/14/20

Purpose of This Document

This document enumerates the technical and design requirements for the Beta development phase of DATASIM.

DATASIM is an open source application that leverages xAPI Profiles to generate realistic xAPI statement data for a cohort of simulated actors. These datasets may be used to benchmark and stress-test components of the Total Learning Architecture (TLA) and other distributed learning projects.

Additionally, DATASIM may help learning scientists, engineers, ISDs, IT staff and decision-making stakeholders to determine the effectiveness of xAPI data design and implementation across the TLA.

Yet Analytics is on schedule to deliver the Beta implementation of DATASIM as planned.

1 About DATASIM

DATASIM is an open source platform and set of open source specifications which can be used to generate simulations resulting in realistic xAPI data. The platform uses xAPI Profile Patterns, Templates, and Concepts to model behavior for a cohort of simulated actors in the form of xAPI datasets.

These datasets offer insight into the available Patterns and paths available in an xAPI Profile, and can be used to model complex learner behaviors to evaluate the effectiveness of xAPI data design. The output of these simulations can be used as a design tool to iteratively improve upon xAPI Profiles themselves. Because DATASIM is capable of producing a high throughput of records, these datasets may additionally be used to benchmark and stress-test components of the Total Learning Architecture (TLA) and distributed learning projects. DATASIM requires valid xAPI Profiles in order to generate datasets, so as part of the platform it has the capability to validate xAPI Profiles and other inputs against specification.

DATASIM is open source under the Apache 2.0 License and is funded by the Advanced Distributed Learning Initiative at the United States Department of Defense. The source code of the project[1], as well as the source code of an accompanying frontend user interface / client[2] can be found on GitHub.

The Alpha version of DATASIM, released in March 2020, has demonstrated the capability to generate conformant Profile-aligned xAPI datasets at scale. It has also demonstrated the ability to model and guide simulations based on Actor Alignments and other parameters.

---

[1] DATASIM Code Repository: https://github.com/yetanalytics/datasim
[2] DATASIM-UI Code Repository: https://github.com/yetanalytics/datasim-ui

2 Definitions

For the purposes of this document, the following are working definitions.

Researcher: This is a general term indicating the end-user of DATASIM.

Simulation: The modeled activity of agents within an environment.

Scenario: This is the narrative or conceit comprising the text or subtext of the activity within the simulation.

Agent: This is a simulated learner in a defined DATASIM simulation. Sometimes used interchangeably with Actor in an xAPI simulation.

Personae: A collection of Agents, Groups and their characteristics, to be used in a simulation

Alignment: A method of influencing the outcome of a simulation by assigning numeric weight multipliers to the combination of Agents or groups of Agents and xAPI Profile components

Parameters: Operational parameters necessary to generate a simulation

Population: This is the universe of all agents in the simulation.

Environment: This is the combination of platforms, courses, activities, and events, as defined by a taxonomy. The researcher has chosen the environment for the simulation.

xAPI Profile: An xAPI Profile is a collection of concepts, statement patterns, extensions, and statement templates used when implementing xAPI in a particular context.

IRI: Internationalized Resource Identifier. A set of characters which uniquely and unambiguously identifies a particular resource.

3 Technical and Design Requirements

DATASIM core functionality is reliant on three primary domains. Input is provided by the user, processing is performed by DATASIM, and a simulation is run returning output in the form of xAPI data. This can be done either from the command line or a purpose-built user interface.

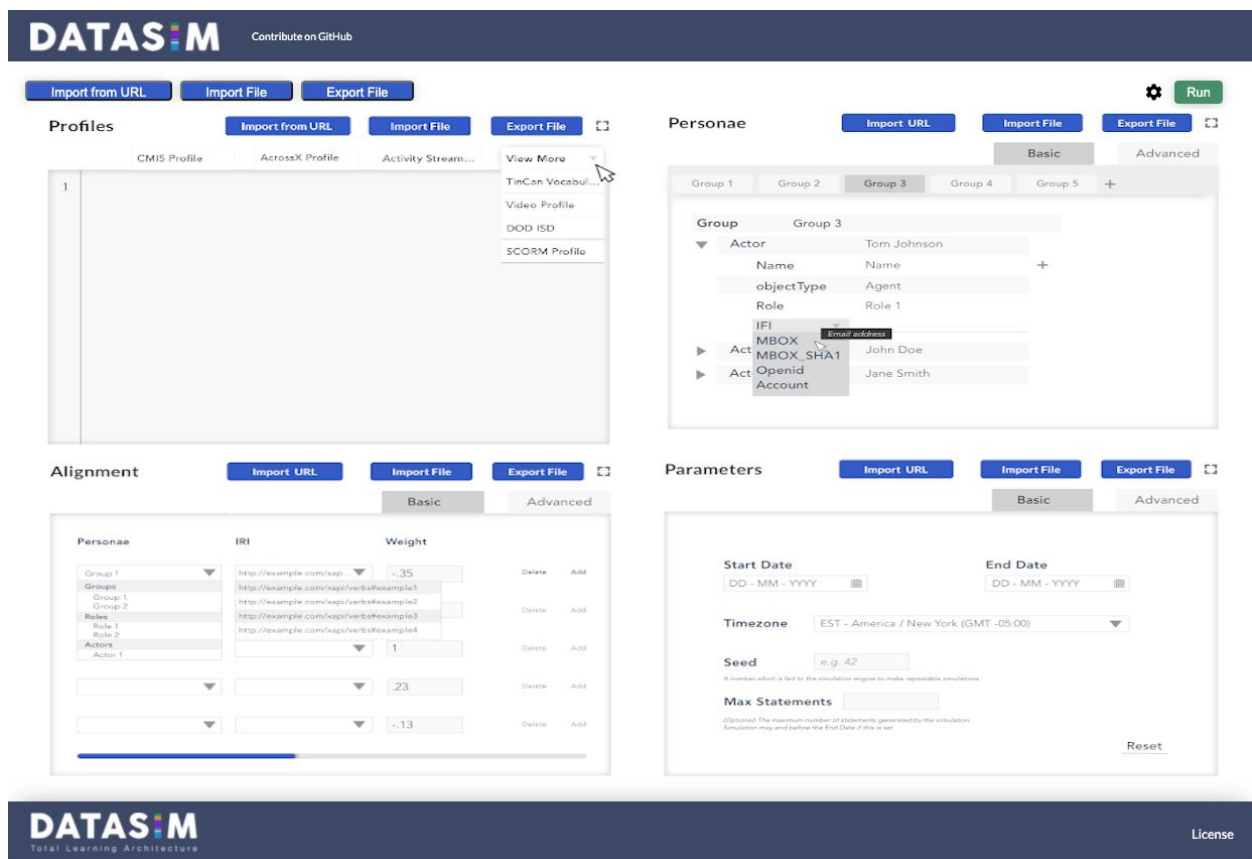The requirements for the initial development of DATASIM fall along these domains.

Fig. 1.   Illustration of DATASIM Beta Implementation's proposed interactions

1. Input Specification Requirements

1.1. Alignment Specification Requirements

1.1.1. Zero-Statement Alignments

1.1.1.1.   Add to Alignment Specification the capability to specify that no statements should be generated for a matched Profile IRI and Personae

1.1.1.2.   Backwards Compatibility with Existing Alignment Specification

1.1.2. Object Override Alignments

1.1.2.1.   Add to Alignment Specification the ability to specify an Actor in the simulation (by Inverse Functional Identifier) as the Object of a Statement for an Alignment where the IRI is a specific Statement Template

1.1.3. Group and Role Specification

1.1.3.1.   Add the capability to the Alignment specification to specify any one of the following as a Personae for a given Alignment: Group, Role, Actor.

1.2. Actor Specification Requirements

1.2.1. Add Groups and Roles

1.2.1.1.   Add to the Personae Specification the ability to create and name multiple Groups and place each Actor within a Group

1.2.1.2.   Within the Groups structure, allow the assignment of a Role attribute to the Actor's entry to describe their role within the context of the group.

2. Processing Requirements

2.1. Validation

2.1.1. xAPI Profile Validation

2.1.1.1.   Return a json data structure containing detailed information for all xAPI Profile Specification conformance issues found during validation, including:
- Violation Type and Description
- Location in within profile(s)

2.1.2. Alignment Specification Validation

2.1.2.1.   Update existing Alignment validation to account for specification changes resulting from requirements 4.1.1.1, 4.1.2.1, and 4.1.3.1

   2.1.2.2. Validate existence of Alignment IRIs and Personae in associated inputs

  2.1.3. Actor Specification Validation

   2.1.3.1. Update existing Personae validation to account for specification changes resulting from requirements 4.2.1.1 and 4.2.1.2

3. Simulation Requirements

3.1. Pattern Sequence Consistency

  3.1.1. Adhere Simulation to Sequence property of Pattern

   3.1.1.1. If an Actor begins a Pattern sequence, force subsequent statements generated by an actor to be the next step in the sequence

   3.1.1.2. Follow appropriate sequence rules, until completion

   3.1.1.3. Maintain Context consistency

3.2. Alignment Runtime

  3.2.1. Group and Role Weights

   3.2.1.1. Group and Role-based Alignments factor into the weight of an individual Actor when making deterministic decisions about an IRI

   3.2.1.2. The average of all applicable Alignments which exist will determine the final weight to apply to the IRI at runtime

  3.2.2. Zero-Statement Alignment Runtime

   3.2.2.1. At runtime the simulation must generate zero Statements for the Actor and IRI designated with the Alignment weight value specified in 4.1.1.1

  3.2.3. Object Override Alignment Runtime

   3.2.3.1. If an Object Override is specified in the Alignment for the statement being generated, the Object Override must be used within the resulting statement

4. UI / UX Requirements

4.1. xAPI Profile UI Requirements

  4.1.1. Provide separate management of individual xAPI Profiles

   4.1.1.1. Utilize tabs for management and navigation of Profiles

   4.1.1.2. Allow for multiple xAPI Profiles to be uploaded at once

4.2. Parameter Specification UI Requirements

4.2.1. Allow for the UI-based input of the xAPI Simulation Parameters specification

4.2.1.1. Datepicker for Start Date, End Date

4.2.1.2. Drop-down to Timezone

4.2.1.3. Text box (numeric entry only) for Max Statements

4.2.1.4. Text box (numeric entry only) for Seed

4.3. Personae Specification UI Requirements

4.3.1. Allow for UI-based input of the xAPI Personae Specification

4.3.1.1. Groups are in Tab form and editable within each tab

4.3.1.2. Actors are added in a Group tab

4.3.1.3. Actors can be dragged to other Group tabs

4.3.1.4. Actors can be edited according to xAPI Actor specification

4.3.1.4.1. Text box for name

4.3.1.4.2. Text box and Dropdown (for type, e.g. mbox, account) for Inverse Functional Identifier

4.3.1.5. Actors can have a Role

4.3.1.5.1. Text box for Role name

4.4. Alignment Specification UI Requirements

4.4.1. Allow for tabular input of Alignments

4.4.1.1. On each row is a Personae Drop-down

4.4.1.1.1. Drop-down contains all Groups, Roles and Actors in the Personae Specification

4.4.1.1.2. Drop-down is updated instantly when changes occur to the Personae input

4.4.1.2. On each row is an IRI Drop-down

4.4.1.2.1. Drop-down contains all IRIs within the provided Profile(s)

4.4.1.2.2. Drop-down is updated instantly when changes occur to the Profile input.

5. Architecture and Deployment Requirements

5.1. Distributed Generation

5.1.1. Configuration

5.1.1.1. Infrastructure-level options should be provided to enable multi-server communication over an accepted protocol

5.1.1.2. Servers should be able to failover and leave the cluster without disabling overall generation (but not mid simulation)

   5.1.2. Generation

     5.1.2.1. If multiple servers communicating in a cluster, simulation initiation on one server should begin synchronized simulation across servers

   5.1.3. Consistency

     5.1.3.1. Final produced statements should be deterministic regardless of number or configuration of involved servers

     5.1.3.2. Timestamp should be identical on every run with the same input

     5.1.3.3. Order of storage may not be identical with the same inputs on every run

6. Process Requirements

6.1. Documentation

  6.1.1. General documentation for DATASIM should be available on GitHub in the README.md and associated files in the /doc path of the repo.

  6.1.2. DATASIM's public API must be fully documented.

6.2. Source Control

  6.2.1. The source code for DATASIM will be hosted on GitHub

  6.2.2. All changes to the codebase will undergo code review as pull requests

  6.2.3. Version/release tags will be used

6.3. Testing

  6.3.1. DATASIM will provide comprehensive unit tests

  6.3.2. Testing should be automated via a Continuous Integration (CI) system

  6.3.3. A test badge will be added to the DATASIM github page

6.4. Versioning + Changes

  6.4.1. DATASIM should be versioned according to semver 2.0
      https://semver.org/

  6.4.2. A changelog in the style described at
      https://keepachangelog.com/en/1.0.0/ will be maintained

4 Milestones

The following are the Option Year 1 milestones as outlined in the DATASIM Integrated Management Plan.

Action Type	Task Description	Due Date
Research	Preparation of materials for PI Meeting	4/1/19
Meeting	PI Meeting	5/12/20
Research	Preparation of materials for Kick-off Meeting	4/15/20
Deliverable	Project Profile	6/3/20
Action Type	Task Description	Due Date
Deliverable	Integrated Management Schedule / Plan	4/5/20
Meeting	Project Kick-off Meeting	4/7/20
Research	Research Needs / Requirements & Develop Report	5/30/20
Deliverable	Monthly Progress Report	5/7/20
Design	Research and Collection of Mockups, Prototypes, and Examples of Exemplar Data Vis	5/31/20
Deliverable	**Needs Analysis Report**	**6/6/20**
Deliverable	Monthly Progress Report	**6/7/20**
Research	Needs Analysis Design Processing	7/5/20
Deliverable	**Conceptual UI/UX Designs**	**7/6/20**
Deliverable	Monthly Progress Report	7/7/20
Design	Feedback and Iteration on UI/UX Designs	7/19/20
Research	Define requirements and develop report	8/5/20
Deliverable	**Requirements Report**	**8/5/20**
Deliverable	Monthly Progress Report	8/7/20
Design	Design Stakeholder Feedback Process	8/9/20
Research	Gather Feedback from Stakeholders	9/20/20
Research	Analyze Feedback & Develop Report	11/3/20
Deliverable	Monthly Progress Report	9/7/20
Deliverable	Monthly Progress Report	10/7/20
Deliverable	**Stakeholder Analysis Report**	**11/3/20**
Deliverable	Monthly Progress Report	11/7/20
Research	Research testing and evaluation criteria	1/1/21
Research	Research and design evaluation plan	1/1/21
Deliverable	Monthly Progress Report	12/7/20
Deliverable	**Testing and Evaluation Plan**	**1/2/21**
Deliverable	**Usability TRL5 Report**	**1/2/21**
Development	Develop and test prototype software	3/2/21

Deliverable	Monthly Progress Report	1/7/21
Deliverable	Monthly Progress Report	2/7/21
Deliverable	**Reference Simulated Data Set**	**3/3/21**
Deliverable	Monthly Progress Report	3/7/21
Development	Develop and test alpha software	4/3/21
Research	Prepare Final Report	4/3/21
Deliverable	**Prototype Software and Documentation**	**4/4/21**
Deliverable	**Base Period Final Report**	**4/4/21**

The Yet Team expects to have fully functional advanced Profile validation feedback (2.1.1.1) and Group and Role capabilities (1.2.1) by November 2020 prior to I/ITSEC.

We expect to be providing the full Beta implementation by the scheduled end of the Year 1 of the DATASIM project in April of 2021.