DATASIM Needs Analysis Report

Data and Training Analytics Simulated Input Modeler

27 May 2021

This work was supported by the U.S. Advanced Distributed Learning (ADL) Initiative (HQ003419C0061). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ADL Initiative or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes.

Needs Analysis: DATASIM

Data and Training Analytics Simulated Input Modeler
Prepared by Shelly Blake-Plock, PI, Yet Analytics, Inc.
Submitted to ADL on June 24, 2020

Purpose of this Document

This document presents the definitions, needs, features, and considerations necessary to examine the Option Year 1 iteration of the work to continue development of the Data and Training Analytics Simulated Input Modeler (DATASIM). The needs identified in this document will contribute directly to the development of the Beta DATASIM specifications, building on top of the Alpha implementation released in March 2020.

Table of Contents

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. Through experimentation with this Alpha version since its release, and based on the findings of the Alpha Testing and TRL 4 report, the ADL and Yet Analytics have elicited areas of necessary enhancement to make DATASIM more useful to Government Stakeholders.

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.

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

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.

3 Milestones

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

- the concurrent research necessary to facilitate the improvement to the simulation, UI/UX and deployment considerations of the DATASIM Beta platform
- stakeholder feedback and analysis
- the development of testing and evaluation criteria and implementation of testing
- the development of specifications and Beta software
- completion and publication of the Beta software documentation

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
Deliverable	Technical Webinar (TBD)	
Deliverable	Interview of Project (TBD)	

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**

4 Identified DATASIM Beta Needs

For this part of the document we will detail the needs elicited from analysis of the Alpha implementation by the ADL and Yet Analytics and the requirements which meet those needs. All of these needs were identified as being necessary to move DATASIM from its current Alpha implementation to Beta TRL5 in Option Year 1. All currently identified needs fall into a category of enhancement to simulation capabilities, enhancement to UI, or enhancement to the architecture and deployment model of the platform.

4.1 Simulation Modeling Capabilities

This section explores identified needs for DATASIM development in this phase of the project related to increasing the capabilities of a researcher when using the input specifications. DATASIM has four primary input specifications: Profile(s), Actors, Alignments, and Parameters. In this iteration of the project we have identified needs related to two of those specifications, Alignments and Actors.

4.1.1 Alignment Specification Need: Zero-Statement Alignments

In the Alpha implementation of DATASIM the probability of an Actor engaging with a Profile Pattern, Concept, or Statement Template is determined by a combination of a deterministically random value and the appropriate Alignment, if one exists. An Alignment weight of -1 strongly indicates that the Actor does not engage with the component, and conversely a weight of 1 strongly indicates that the Actor engages with the component.

The deterministically random value serves to introduce the necessary entropy into Actor behavioral preferences to create a realistically diverse dataset. The effect of that, however, is that even if a researcher aligns an Actor with a -1 weight to a component, within a sufficiently large simulation the Actor will still occasionally engage. This is not always ideal for modeling a simulation. There are use cases in which it is desired for a given Actor in a simulation to never engage with a component and the expectation is an xAPI dataset which does not contain a single statement in which they do so.

The requirement to meet this need in the DATASIM Beta implementation is to modify the Alignment specification to allow for a weight input which disallows the Actor from engaging with the component, rather than just heavily suggesting it. The requirement for simulation execution is that DATASIM respects this input and does not generate any xAPI data with the specified combination.

4.1.2 Alignment Specification Need: Object Override Alignments

There exist use cases, not currently supported by the DATASIM Alpha implementation, which require the generation of xAPI Statements containing an Actor defined in the simulation as the Object of the Statement. Current DATASIM input specification allows only for the Object to be dictated by the Statement Templates and Rules in the xAPI Profile itself.

The change required to accomplish this lives most naturally in the DATASIM Alignments specification, where the relationship between an Actor and a component are already detailed. There is, however, no structure in the current specification which can accomplish the goal, as seen below:

```
{"mbox::mailto:bob@example.org":
 {"https://example.org/activity/a": 0.5,
  "https://example.org/activity/c": -0.2},
 "mbox::mailto:alice@example.org":
 {"https://example.org/activity/c": 0.7,
  "https://example.org/activity/d": -0.02}}
```

An Example implementation of the DATASIM Alpha Alignment Specification

The requirement to meet this need is to add the ability to specify an existing Actor in the simulation (via an identifier) as the desired Object when the component in question is a Statement Template. At runtime, if the other requirements are met for use of the Alignment in generation of a Statement, the Object of the output Statement should then be the specified Actor. This capability may also be expanded in the future to add additional overrides to Alignment specification for further detailed modeling.

4.1.3 Actor Specification Need: Grouping, Roles, and Attributes

The Alpha DATASIM Actor specification is structured as a single JSON object with properties “name”, “objectType”, and “member” which is an array of xAPI Actors. An example of the original specification can be seen below:

```
{"name": "trainees",
 "objectType": "Group",
 "member": [{"name": "Bob Fakename",
             "mbox": "mailto:bob@example.org"},
            {"name": "Alice Faux",
             "mbox": "mailto:alice@example.org"},
            {"name": "Fred Ersatz",
             "mbox": "mailto:fred@example.org"}]}
```

An Example implementation of the DATASIM Alpha Actor Specification

While this specification allows for the naming of a single group as part of the input, it does not expand on that capability and the simulation itself does not currently take into account that grouping. Furthermore, while allowable attributes in the xAPI specification may be added to Actors, there is no way to define the Actor's relationship to this simulation Group in the xAPI Actor specification. In order to facilitate the Group and Role Alignment functionality described later in Section 4.1.4 it is necessary to establish that hierarchy in the input specification for Actors.

The requirement to satisfy this need is to redesign the Actor input to allow for multiple Groups, Actors within those Groups, and a Role attribute[3] to represent the Actor's role within that group. Additionally, the simulation execution must be modified to respect this structure when generating xAPI Statements.

4.1.4 Alignment Specification Need: Group and Role Alignments

In Section 4.1.3 we described the requirement for a new Actor input specification which allowed for Group and Role structures to be input into the simulation. The corresponding need in the Alignment specification is the capability to reference a Group or Role when specifying an Alignment. The current Alignment specification accepts only an Inverse Functional Identifier[4] as a key, and a map of Profile IRIs to weights as a value.

The requirement to meet this need is to modify the Alignments specification to allow for any of a Group identifier, Role identifier, or individual Actor identifier to serve as keys of an Alignment which contains components and weights. The requirements for changes to the execution of the simulation are that the weights of an Actor's Group, Role, and the Actor themselves are all considered in combination as factors in the probability of engagement with the specified Profile component, and the generation of a corresponding xAPI Statement. It will also be required to document the algorithm for determining weight from multiple alignments, and the effect this is expected to have on execution of the simulation to aid researchers in modeling.

4.1.5 Pattern Sequence Handling

In the Alpha implementation of DATASIM when the simulation engine encounters a Pattern which contains a sequence (an array of Pattern or Statement Template identifiers[5]) it follows the sequence and generates xAPI Statements from the members. What it does not do at this time, however, is pay attention to all of the attributes of the Statements, as well as timing in order to make a consistent coherent sequence execution. It generates realistically-shaped data but does not necessarily tell the story that the sequence intends.

[3] At this time suggested to be a String datatype, definable by the researcher
[4] https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-Data.md#inversefunctional
[5] https://github.com/adlnet/xapi-profiles/blob/master/xapi-profiles-structure.md#90-patterns

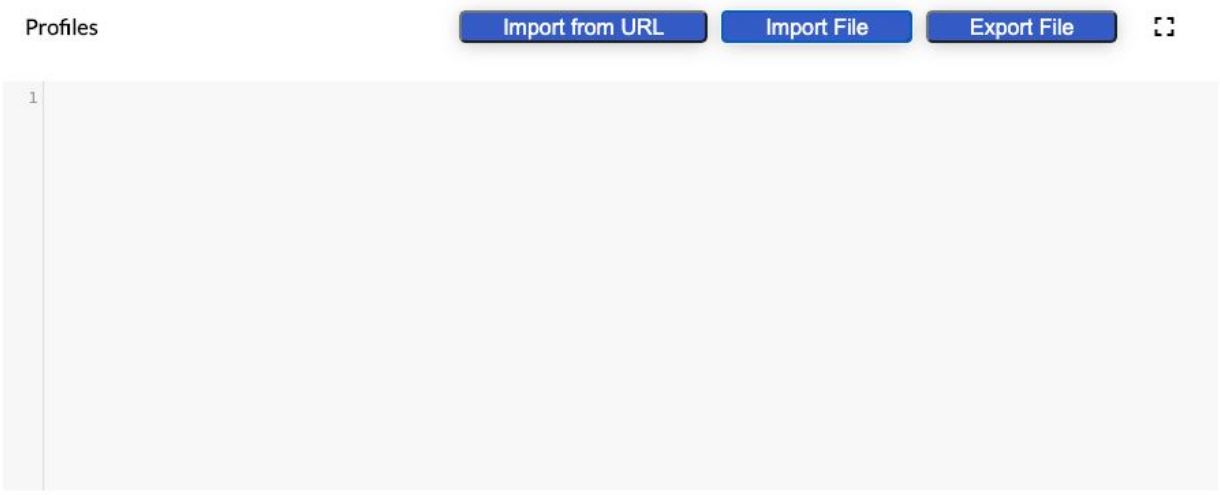
The requirement for this need is to improve the sequence handling in DATASIM such that it will produce a realistic sequence execution in the right order, back to back, while maintaining consistent Objects. After the necessary research and development to determine the best way to accomplish this goal, it may also become necessary to modify some of the input specifications in order to accomplish this goal.

4.2 UI/UX Needs

In this section we will explore the needs pertaining to improvements to the UI/UX components of the DATASIM platform. DATASIM-UI is a standalone project which runs independently as a client to core DATASIM, which in this model acts as a separate server. Because of this, any of the needs in this section do not describe behavioral changes in DATASIM execution, structure, specification or capability. Instead, these needs represent convenience features designed to increase accessibility and productivity for Government Stakeholders.

4.2.1 Multi-Profile Upload

In the Alpha DATASIM UI implementation there are three input mechanisms for any xAPI Profile(s). The first is a directly editable text box in which to paste or write a Profile. The second is a file upload mechanism designed to allow a local file to be uploaded. The third is a URL entry method through which a publicly accessible hosted Profile will be scraped and loaded. In the latter two methods, upon entry, the Profile is then placed in the editable text box from the first method.



Pictured above is the Alpha DATASIM-UI interface for loading or creating xAPI Profiles

If a researcher desires to use multiple xAPI Profiles in a simulation, this is possible by way of a JSON array. The format of the Profile input becomes an array wherein each element is itself an xAPI Profile. This functionality is directly correlated to the design of the core DATASIM service which expects this specification as an input.

```
[
    {
        //Profile 1
    },
    {
        //Profile 2
    }
]
```

The DATASIM input format for multiple xAPI Profiles

This can be cumbersome for a researcher, especially with numerous Profiles which they may swap in and out of simulation runs, or which they are concurrently editing between simulations. To solve this need in the UI the ideal solution would be to upload multiple Profiles at once. The requirement here is to allow for multiple xAPI Profiles to be uploaded simultaneously through the interface, and subsequently for the system to perform the work of reformatting them in the desired array structure so that they are still interoperable with core DATASIM services.

4.2.2 UI-based Parameter Specification Editing

The DATAIM Parameter specification is a JSON object containing a number of properties required to run the simulation. This specification was meant as the more operational input to the system and is concerned with the size of the dataset generated, time period for generation, and various additional inputs which allow the simulation to run. In DATASIM-UI these fields are currently entered via the editable text editor in the desired JSON format.

An example of a Parameter input to a DATASIM simulation in the UI Client

In order to facilitate greater ease-of-use, the need in this area is to be able to edit these inputs in a purpose-built user interface containing the available options and a simple input mechanism to set each. For example, in the case of the startDate and endDate inputs, they would most intuitively be a date-picker widget. The requirement to meet this need is to allow for intuitive easy to use inputs for all data types represented in this specification.

Additionally we do not want to completely remove the flexibility of the JSON input method. The reasons for this are portability of specification for import/export, and that we would like to maintain the free-form property entry in the case that core DATASIM has evolved to add

additional capabilities but the UI client has not been updated accordingly. In order to accomplish this goal, the appropriate solution is a default UI entry interface with an “Advanced” mode option that allows for the existing direct specification JSON entry method.

4.2.3 UI-based Actor Specification Editing

Similar to the need described in Section 4.2.2 it is desirable that the Actor input be editable without direct knowledge of the specification or JSON data structures in general in order to increase productivity and ease of use for all researchers and analysts, regardless of their background in data structures and JSON.

While the base need and requirement is the same as described for the Parameter specification, Actors offer an additional challenge in that there may be *n* Groups and Actors, all with variable nested properties. This requirement dictates a need for flexible, dynamically-sized, interaction UI.

4.2.4 UI-based Alignment Specification Editing

The need for more accessible interaction with Alignments duplicates the exact requirements outlined in Section 4.2.2 and Section 4.2.3. In addition to what has already been described, in order to be more effective the Alignments UI must also be capable of referencing the Actors, Groups and Roles made available by the Actor input as well as the IRIs available in the Profile input[6]. This added functionality involves parsing the adjacent inputs continuously and generating lists of available components, akin to an inventory, from which to derive complete Alignments. This effort aims to achieve a highly usable guided interface which prevents erroneous input and broadens the backgrounds and skill sets necessary to efficiently utilize the Alignments specification for analysis.

4.3 Architecture and Deployment Needs

In this section we will explore needs for improvement related to the Software Architecture of DATASIM and related methods of deployment and infrastructure appropriate for a Beta implementation.

4.3.1 Distributed Generation

The Alpha implementation of DATASIM can currently be scaled by a number of methods. The host hardware of a simulation can be scaled vertically, via hardware upgrades or additional cloud computing resources. DATASIM itself leverages a streaming architecture for the purpose of removing an upper limit on the total volume of statements generated. Testing and evaluation of the Alpha platform have revealed that it will simply keep running until it is finished, as defined by the input parameters.

[6] Presumably implemented visually as drop-downs or other discrete interaction methods

The DATASIM Alpha implementation can also be duplicated, in an unsophisticated way, horizontally to deliver a linear increase in total volume of Statements generated over time. This capability has value for at least one DATASIM use case; stress testing learning ecosystem infrastructure. In this case any number of independent simulations can be pointed at the same endpoint in a learning ecosystem and their individual throughputs combine to become the total incoming Statements at that endpoint. This scaling method's application to data and Profile analysis, however, is limited. The concurrent simulations are ignorant of each other and do not operate on the same time series. The simulations may or may not have the same characteristics and because the entropy of the simulations are not synchronized from the same source they may not be deterministically repeatable when observed as a whole.

The ideal scenario for scalability with these latter use cases is that a researcher is able coordinate a singly-defined simulation distributed across *n* independent-but-collaborating nodes to reliably produce a deterministic and coordinated output dataset. In future use cases this will be instrumental to coordinated load testing, to the fast reproduction of high throughput simulated scenarios[7], and to simulations of very large Actor populations. From a requirements standpoint this improvement would dictate adding a clustered architecture to the DATASIM backend enabling accurate and reliable orchestrated xAPI Statement generation. Throughput scaling in this clustered model would by definition not be as precisely linear as the isolated simulation scaling described above due to coordination overhead, but should be able to scale to a degree that it is capable of any conceivable learning activity need. As part of the requirement, consideration should also be given to the ease of scaling such a cluster from the perspective of infrastructure administrators or researchers themselves.

7 Next Steps

In this report we defined the needs for a Beta implementation of DATASIM as identified by analysis of the current Alpha platform. Based on Section 4.2 the UI/UX design work is currently underway, and engineering analysis has begun on the simulation modeling and architectural requirements. The needs identified were elicited primarily through interaction with ADL stakeholders and engineers. In addition to the needs described here we believe it would be valuable to continue to develop the platform throughout this Option Year based on additional sources, including:

Additional Government Stakeholder(s)

Additional Government Stakeholders with defined use cases, ready to leverage DATASIM for xAPI Profile analysis or stress-testing scenarios, will be a valuable resource in helping to prove assumptions and strengthen the platform as a whole. Use of the Tactical Combat Casualty Care domain was instrumental in helping shape design choices in the Base Year. That work helped

[7] e.g. high resolution instrumented simulators, telemetry data, and instrumented real-world exercises

prove and articulate some additional use cases for the platform as a whole, such as the simulation-aided iterative xAPI Profile design process. In collaboration with the ADL we are speaking to a number of candidate Stakeholders currently.

Additional DoD xAPI Profiles and/or Use-Cases

Having access to more xAPI Profiles and related simulation use cases during the development of the Beta implementation will broaden the support of the platform and help shed light on scenarios which we have not elicited through our own observation.

Yet Analytics has supported some recent work being done by the ADL with the DoD MOM Profile, for instance. This work was directly helpful in finding edge cases and shaping some of the newer requirements found in this report.

Evaluation Results and Volume Reporting

We have asked that the ADL keep us informed of results from stress tests against TLA sandbox/reference infrastructure, or any available benchmarking which takes place with the use of DATASIM. Primarily what we are interested in is the volume of data over time that needs to be generated in these use cases. This information is valuable to aid in shaping throughput KPIs for this iteration of the project.