DATASIM Option Year 1 Final 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.

Final Option Year 1 Report: DATASIM at TRL5

Data and Training Analytics Simulated Input Modeler

Prepared by Shelly Blake-Plock, PI, Yet Analytics, Inc.

Submitted to ADL on March 31, 2021

1. Purpose of this Document

The purpose of this document is to present a report and related documentation regarding the Technology Readiness Level 5 (TRL5) prototype of the Data and Training Analytics Simulated Input Modeler (DATASIM). The report explores the progress that has been made on the prototype reference software, new features that have been introduced, and the aspects of the application which would benefit from further development in the next iteration of the technology.

Table of Contents

2. Project Description

DATASIM is an open source software project currently under development for the purpose of providing a platform for running simulations that result in the generation of realistic xAPI data at any scale. The environment uses xAPI Profiles to model behavior 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 can help learning scientists, engineers, ISDs, IT staff and decision-making stakeholders to determine the effectiveness of xAPI data and xAPI Profile design and implementation across the TLA.

DATASIM is funded by the Advanced Distributed Learning Initiative and is released as open source under the Apache License Version 2. It is available on GitHub here:

Backend <https://github.com/yetanalytics/datasim>
Frontend <https://github.com/yetanalytics/datasim-ui>

3. Technical Summary

This section of the report explores in detail the new features introduced, how they work together, and what has been implemented to date in this stage of the development of the prototype platform.

3.1 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. Predefined xAPI Actors (upon whom the simulation will be based) are required to run a DATASIM simulation. This takes the form of a JSON array of xAPI Groups, each object containing an array of conformant Actor members.

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. An alignment represents a way to influence the simulation by explicitly weighting an Actor's relationship to a part of the xAPI Profile. Each actor can have alignments to multiple parts of the Profile, and the weight system ranges from -1 to 1 (with 1 being an extremely high propensity for interaction in the simulation and -1 indicating that zero statements should be created for that Actor and that Profile Component). During the simulation these weights factor in but do not completely predict the outcome as there is still randomness in Actor behavior. The records are an array of objects where each object is a combination of Actor (id in IFI format), type ("Agent", "Group", or "Role") and an array of IRIs to align to, and weights for each.

Parameters: Operational parameters necessary to generate a simulation. The simulation parameters input covers the details of the simulation not covered by other pieces. This includes Start Time, End Time, Timezone, Max (number of statements) and seed. When run, the simulation will create a time sequence from the Start Time to the End Time and generated xAPI statements will have corresponding dates and times. The seed is important as it controls the inputs to all random value generation and corresponds to repeatability. A simulation run with the same inputs and the same seed will deterministically create the same xAPI Statements, but changing the seed value will create an entirely different 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. A valid xAPI Profile is required for DATASIM to generate xAPI Statements. This input can either be a single Profile JSON-LD document or an array of JSON-LD format profiles. At this time all referenced concepts in a Profile must be included in the input. For instance if in "Profile A" I have a Pattern that references a Statement Template found in "Profile B", both Profiles must be included in an array as the Profile input.

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

3.2 Alignment Specifications

The Alignment record consists of an Actor, IRI (Internationalized Resource Identifier), and weight. The Alignment is meant to provide the user with the ability to noticeably tip the scales of an Actor's relationship to a part of the xAPI Profile in order to increase or decrease the likelihood of that statement generation. The Alignment weight system consists of a -1 to 1 scale.

- 1 substantially inflates the number of statements generated for that Activity
- -1 respectively decreases the amount of statements for the Activity.

The statements produced are representative of the weight adjusted within the Alignment. Using DATASIM, the user can further hone the alignment specification to accommodate the modeling of all training requirements and scenarios.

In this iteration of the DATASIM platform, the Alignment specification was modified to account for a number of new features. This was done to accommodate the expansion of the Alignment object itself to allow for new fields, such as in the Object Override feature discussed below in section 3.1.2.

A comparison of the old and new Alignment specification can be 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}}
```

Fig. 1 Original Alignment Specification format.

```
[
  {
    "id": "mbox::mailto:bob@example.org",
    "type": "Agent",
    "alignments":[
      {
        "component": "https://example.org/activity/a",
        "weight": 0.5
      },
      {
        "component": "https://example.org/activity/c",
        "weight": -0.2
      }
    ]
  }
]
```

Fig 2. The revised TRL5 Alignment Specification.

3.2.1 Zero-Statement Alignments

The refined Alignment specification allows for the capability to specify that no statements should be generated for a matched Profile IRI and Personae. Previously, due to the random entropy involved in generation, certain actions that were weighted -1 were still creating statements because weight was only a component of the related probability of their occurence. There are scenarios where certain statements should not be generated no matter the other factors of the simulation.

3.2.2 Object Override Alignments

This new feature of the Alignment specification allows the researcher to specify an Object directly in the spec, and that Object will be used in the resulting statement (where the component IRI is in a specific Statement Template). Object override allows the user to accommodate inherent inaccuracies within the data to avoid invalid and improbable scenarios. Allowing the user to customize the dataset to reflect reality will greatly increase the usefulness of the resulting data.

```
[
 {
   "id": "mbox::mailto:bob@example.org",
   "type": "Agent",
   "alignments": [
     {
       "component": "http://example.org/xapi/statements/course-taken",
       "weight": 0.9,
       "objectOverride": {
                "objectType": "Activity",
                "id": "https://www.example.com/activities#course2",
                "definition": {
                    "name": {"en-US": "Course 2"},
                    "description": {"en-US": "Course Description 2"},
                    "type": "http://adlnet.gov/expapi/activities/course"
                }
            }
        }
     }
   ]
 }
]
```

Fig 3. The illustration shows how to use Object Override Alignments in the updated Alignment Specification.

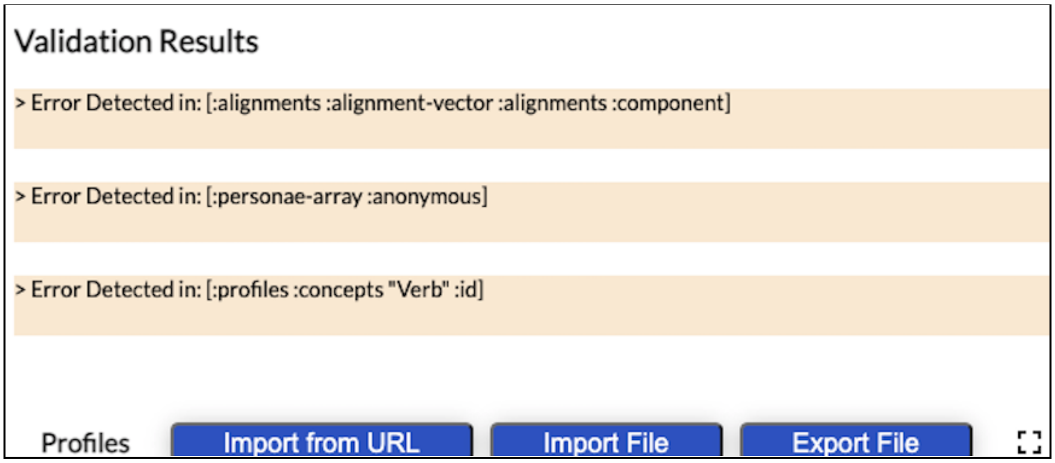
3.2.3 Group and Role Alignments

This Alignment allows the capability to specify Group, Role, or Actor as Personae for a given set of Alignments in the updated Alignment specification. In the Personae section of DATASIM, the user now has the ability to add and name multiple Groups and place each Actor within exactly one Group. An Actor must be in a Group, but if the Group does not have an Alignment it does not impact the simulation. In the Groups structure, the use can allow the assignment of a Role attribute to the Actor’s entry to describe their role within the context of the group.

3.3 Visual Validation

This update to the DATASIM platform has added detailed validation feedback to each of the parts of the Simulation Specification. The user is alerted when an error has occurred as DATASIM validates the specifications of the simulation. In order for the user to remediate each component, all of the errors appear in a text field at the top of the page. Each type of error can be rendered individually alongside other errors to ensure the correct simulation specification errors appear when a certain mistake is made. Each line is a high-level summary including the location of the error. The summary can be expanded into a detailed description of why and where the specification validation failed.

Visual validation allows the simulation configurations to be readable and allow a user to find the appropriate error in their inputs by the information provided in the validation messages.



Validation Results

> Error Detected in: [:alignments :alignment-vector :alignments :component]

> Error Detected in: [:personae-array :anonymous]

> Error Detected in: [:profiles :concepts "Verb" :id]

Fig 4. The validation section of the updated DATASIM UI.

3.3.1. Parameter

The Parameters of a simulation are now added through text fields. A clickable widget has been implemented in order to provide the user with a simple way to add the dates. However, even with the new user interface, the user still has the capability to input an erroneous value. Therefore, DATASIM's validation capability ensures that the parameter errors appear when applicable and that they include straightforward directions to fix the issue.

3.3.2 Personae

With the addition of groups and roles, there is a need to test different types of group hierarchies against multiple types of roles in order to assess the possible errors that could prevent a simulation from running. The goal of this test is to validate that the resulting statements are following the proper group and role hierarchy and that statements can be generated for groups themselves.

3.3.3 xAPI Profile

DATASIM contains the onboard ability to test and validate xAPI Profiles. The xAPI Profile validation test shows the violation type with a description and the location within one or more xAPI Profiles for a variety of errors. The validation library implemented is based directly on the xAPI Profile specification and is built to evolve with subsequent changes to that specification, including the ongoing development of IEEE P9274.2.1.

3.4 Distributed Generation

This type of scaling increases additional resources by adding more machines to the network. It enables those machines to share the workload across multiple devices and threads.

Multi-node generation is a clustered communication architecture that facilitates individual nodes working in collaboration to create a one concerted simulation. This enables a high degree of scaling and allows a user to proliferate the volume of generation depending on the project's data volume needs. Scaling horizontally allows the user to take advantage of multiple machines to scale the overall simulation throughput.

This iteration of the project focused on **low overhead linear / near-linear scaling** during development which allows the user to generate data at the scale their systems can handle. Any user can increase the size of their dataset if they have the appropriate hardware to generate at that capacity. Most practical implementations were undertaken in Amazon AWS with Scaling Groups of virtual machines.

Note that due to the potential for abuse of this capability, a disclaimer was added to the documentation accompanying the software license.

3.5 Pattern Consistency

When a Pattern sequence begins, this feature forces subsequent statements generated by the actor to be the next step in the sequence. Patterns in an xAPI Profile can be used to demonstrate complex learner behaviors and can be used to evaluate the effect of the xAPI data design. Whenever a certain action is taken, these sequences make the datasets follow the specified following group of Activities.

For example, the user can designate a Pattern for the most common paths through a training and, if the first step of that path is taken, the simulation generates statements for the next steps in that Pattern. The resulting DATASIM simulations are more consistent and representational of existing training drills as represented by xAPI Profiles.

3.6 User Interface

DATASIM has a fully functional graphical user interface (GUI) built for modern web browsers. In addition to the Alpha capabilities of being able to edit inputs separately and to manage the settings of the simulation, the DATASIM UI has been updated to include business-style editors for each input specification type.

A user can graphically build Personae lists, Parameters, and Alignments. A user can also separately manage multiple profiles. The new user-friendly experience allows for an expansion of the user base for DATASIM to include anyone in the enterprise who wants a large xAPI dataset but is not comfortable coding in JSON or dealing directly with the simulation specification.

In the new interface, the user is guided through each stage of building the simulation and is shown suggestions using the visual validation errors listed after attempting to run the simulation. This allows the user base to shape better simulations (and quickly to begin to understand common errors in xAPI data design).

The user now has access to a Basic and an Advanced Mode in the Personae, Parameters, and Alignments sections. The Basic Mode enables the user to enter simulation specifications using buttons and fields, while the Advanced Mode contains the legacy DATASIM interface using JSON. Each of these simulation specifications (except arguably Profiles) can be fully built without engaging in the writing of any code.

3.6.1 Parameter

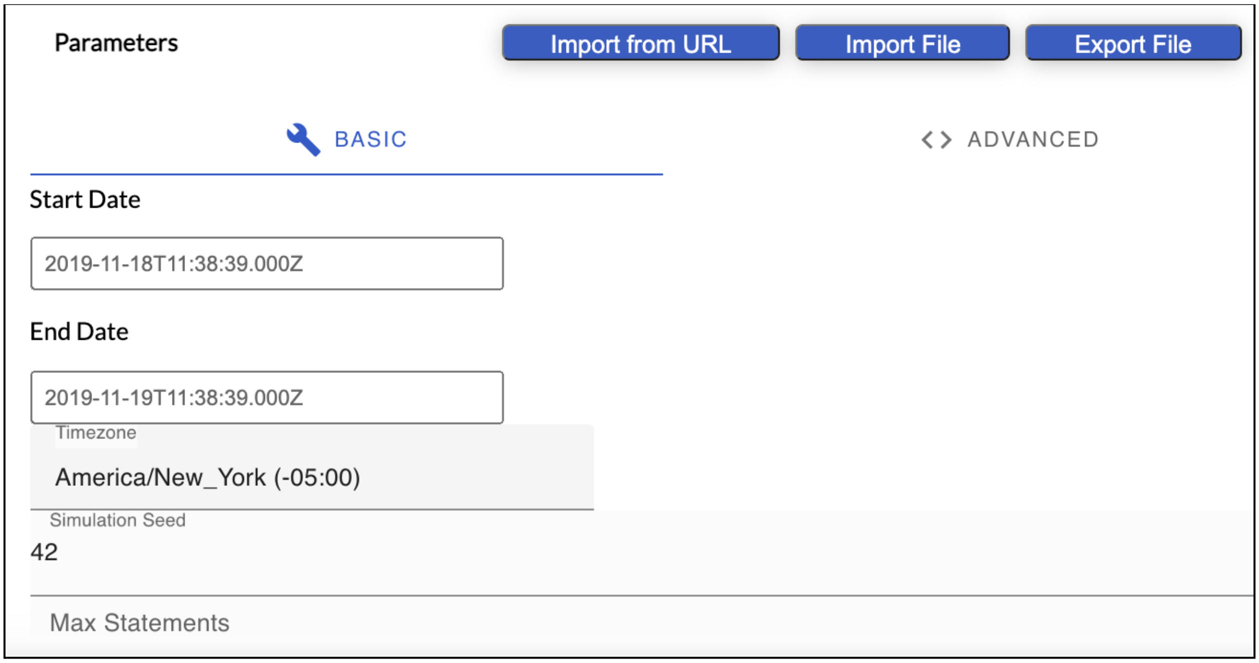
The simulation parameters input dictates aspects of how the simulation runs which are not covered by the other inputs. This includes:

- Start Time
- End Time
- Timezone
- Max Statement
- Seed

The new user interface allows the user to use a date-picker widget or text field to input the specifications of the simulation. Specifications that the user adds in the Basic Mode is reproduced automatically in the generated simulation code available in Advanced Mode.

By inputting the specifically formatted time and date, the UI substantially decreases the likelihood for error when creating simulation specifications. This empowers users to create accurate simulation specifications.

The Parameter interface allows the user to add the dates, timezone, numerical Max Statement and numerical Seed value using text fields and selectable input.



Parameters

Start Date

2019-11-18T11:38:39.000Z

End Date

2019-11-19T11:38:39.000Z

Timezone

America/New_York (-05:00)

Simulation Seed

42

Max Statements

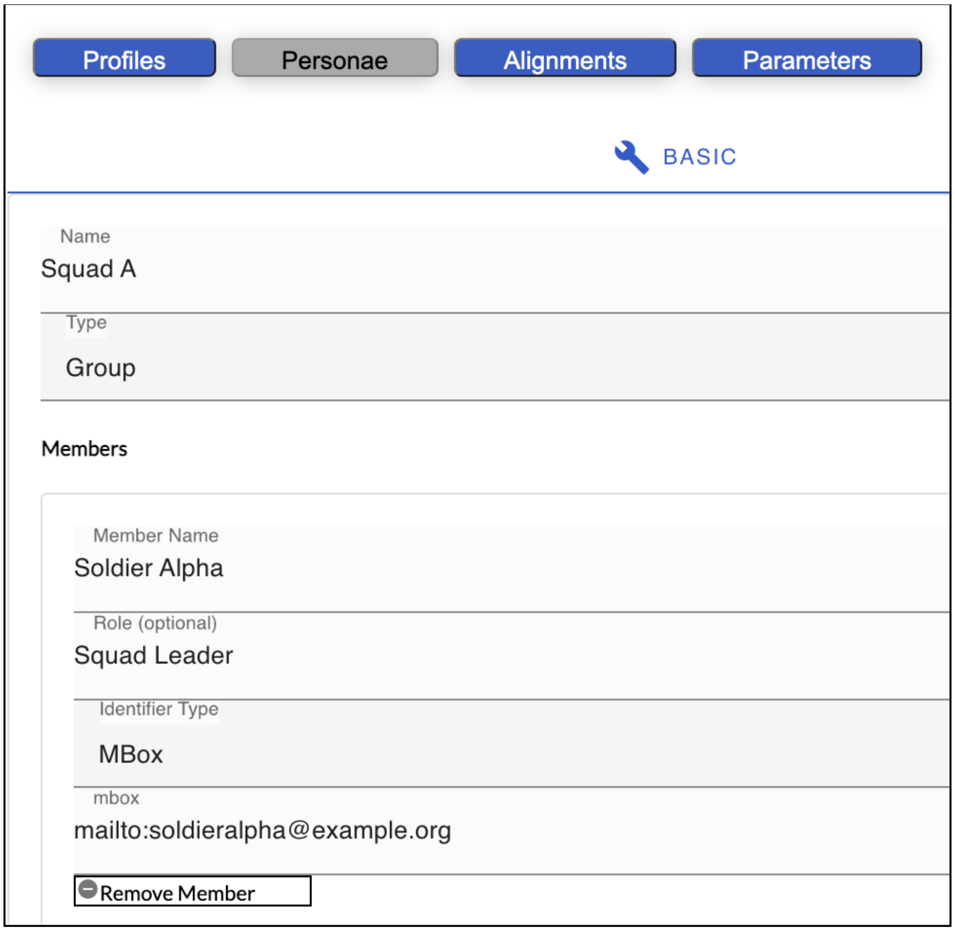
Fig 5. The Parameters interface.

3.6.2 Personae

The Personae feature enables the researcher to add unlimited Personae within a simulation. Each Persona in a simulation details the relevant attributes of a simulated character whose activity comprises the modeled learning or training event. The user is able to construct Personae even if they are unfamiliar with JSON or other data structures.

Configurable fields permit the user to add Group names, Group members, and Roles to each Persona. Any user is able to use DATASIM to create new Personae and culminating datasets much faster than was the case in the prior TRL4 JSON method. Any information added in the Basic Mode automatically updates the simulation specifications.

The new user interface also allows the user to easily add Actors into a Group and assign them to a Role. It allows for multiple Groups within a simulation. The user can create multiple team hierarchy scenarios to show the capacity to have adjustable team structures within the simulation.



Name
Squad A

Type
Group

Members

Member Name
Soldier Alpha

Role (optional)
Squad Leader

Identifier Type
MBox

mbox
mailto:soldieralpha@example.org

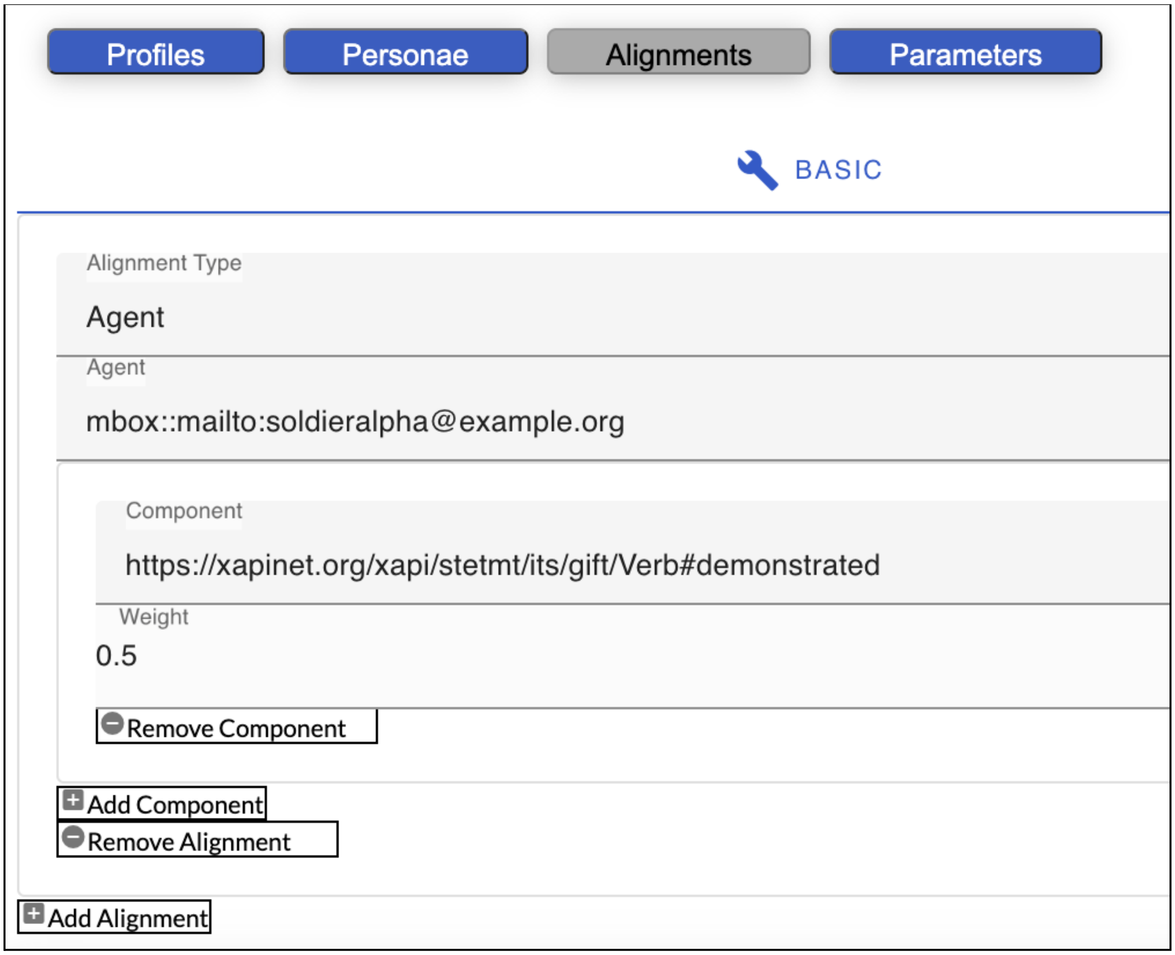
Fig 6. The Personae interface.

3.6.3 Alignment

Alignments can be entered completely through the new user interface from beginning to end. The user is able to add the Actor, IRI, and Weight for each alignment using dropdowns and text fields.

The Alignment editing interface permits the user to add and update any of the Alignments using a numeric field to change the range of each behavior associated with any environment template or pattern within the simulation. The sequential hierarchy allows the user to add each group and member using Add and Remove buttons at the bottom of each element to visualize each aspect of that Alignment and make changes according to their testing needs.

Each Alignment can now be added via a series of dropdown menus to select the Persona and Component with a field to assign the weight between -1.0 and 1.0. Each of the Alignments should accurately generate the correct amount of statements according to simulation specifications.



Alignment Type
Agent

Agent
mbox::mailto:soldieralpha@example.org

Component
https://xapinet.org/xapi/stetmt/its/gift/Verb#demonstrated

Weight
0.5

Fig 7. The Alignments interface.

4. TRL5 Testing Results

4.1 Alignment Behavior

In this series of Alignment tests, all specifications were set to the same parameters prior to testing and evaluation.

The assessment examines the output of the relative number of dataset records corresponding to an Activity and Actor over a range of Alignment weight values. The expectation is that the output should demonstrate a direct relationship between weight and volume.

4.1.1 Groups and Roles Alignments

The results of this test concluded that all three Alignments factored into the Actor's overall weight for 'assessed'. The Max Statement was set to be 10,000 to make it easier to see the comparison of weighted Components.

Actor 1 'Persona 1'	https://w3id.org/xapi/tla#assessed	.5
Role 1	https://w3id.org/xapi/tla#assessed	-.25
Group 1	https://w3id.org/xapi/tla#assessed	-.3

Test Results
This simulation was run without any weight assigned and then was run a second time to the Alignment specifications above.

Simulation	Amount of Aligned Statements
Control Simulation	27
Weighted Simulation	25

The difference between controlled and weighted simulations was -.01666666, thus demonstrating that weighted components produce different simulations.

4.1.2 Zero Statement Alignment

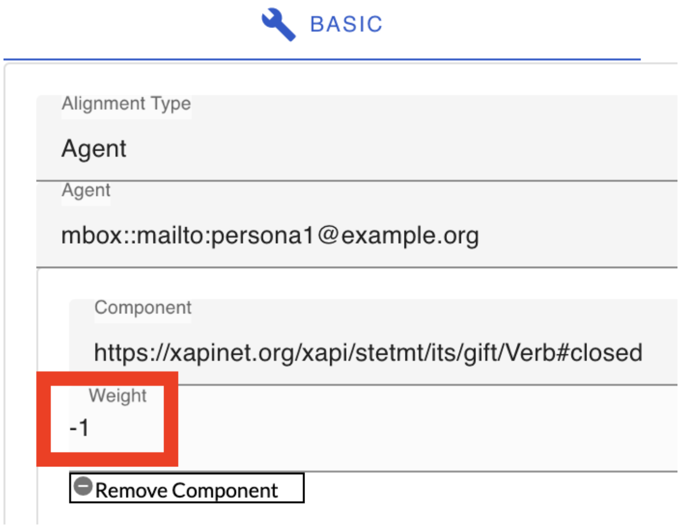
To test this type of Alignment, an Alignment was created with a weight of -1 for a particular Activity. In total, ten simulations were run: one Zero Statement Alignment simulation and then nine simulations with multiple varying Zero Statement Alignments to ensure there are no statements generated for any Alignments weighted -1.

These tests concluded that none of the Agents, Roles, or Groups had statements created for these test Alignments.

4.1.2.1 Individual Zero Statement Alignment

In this test, an Alignment was created for Actor 1 which weighted the component 'captured' to be -1.

The results concluded that by weighting 'captured' -1, resulted in no statements with this component being created.



Alignment Type
Agent

Agent
mbox::mailto:persona1@example.org

Component
https://xapinet.org/xapi/stetmt/its/gift/Verb#closed

Weight
-1

Fig 8. Zero state Alignment testing.

4.1.2.2 Multiple Zero Statement Alignments

This test added additional Alignments to the same simulation specifications with the weight of -1. Each of these Alignments were run in combination with other zero statement Alignments 9 addition times. The Personae specifications for this test are 2 Actors assigned two different roles and all assigned to Group 1. The remaining tests run this varying multiple Alignments listed below.

These tests concluded that no statements were created for the tests below in any test set.

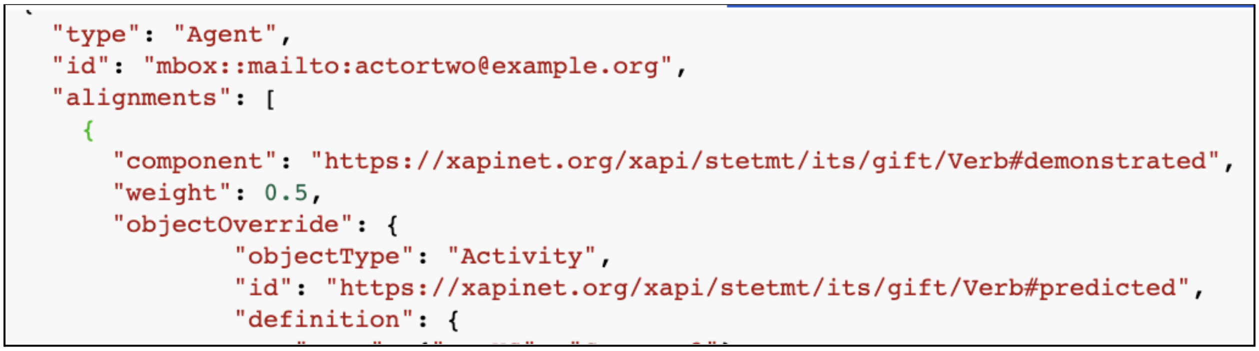
Alignment	Component	Statements Generated
Actor 1	https://w3id.org/xapi/tla/verbs/captured	0
Actor 1	https://w3id.org/xapi/tla/verbs/recruited	0
Actor 2	https://w3id.org/xapi/tla/verbs/recruited	0
Actor 2	https://w3id.org/xapi/tla/verbs/certified	0
Group 1	https://w3id.org/xapi/tla#abandoned	0
Role 1	https://w3id.org/xapi/tla/verbs/captured	0
Role 2	https://w3id.org/xapi/tla/verbs/recruited	0

4.1.3 Object Override Alignment

In this series of tests, two Object Override configurations were created. The first is overriding an activity for an Activity and the second overrides a component with a Persona. Any information that is added to the Object Override field in the Advanced Mode, if the IRI and Actor match, will have that body in the object field instead of a rule-generated Object. Additionally, that field is validated against the xAPI spec for anything that can be in an Object.

4.1.3.1 Activity Override

In the first test, one activity ‘demonstrated’ is overridden for another activity ‘predicted’. This allowed the user to always use the Verb predicted instead of demonstrated for Actor Two.



```
  "type": "Agent",
  "id": "mbox::mailto:actortwo@example.org",
  "alignments": [
    {
      "component": "https://xapinet.org/xapi/stetmt/its/gift/Verb#demonstrated",
      "weight": 0.5,
      "objectOverride": {
              "objectType": "Activity",
              "id": "https://xapinet.org/xapi/stetmt/its/gift/Verb#predicted",
              "definition": {
```

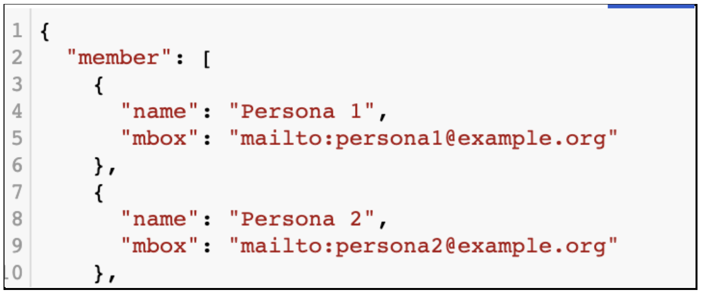
Fig 9. Activity overrides another Activity.

4.1.3.2 Personae Override

The second test was done using two Actors created in the Personae specification and the Advanced Mode with the Alignment section.

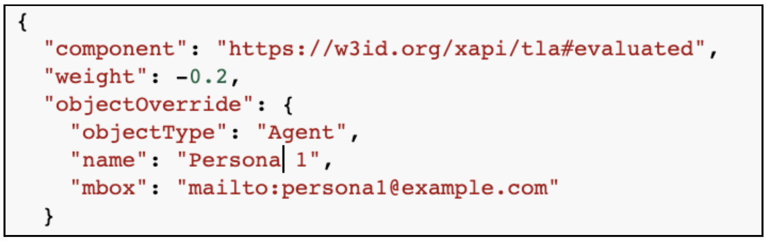
In Fig 10, Persona 2 will always evaluate Persona 2.

First, this test concludes an Object can be overwritten by a Persona.



```
{
  "member": [
    {
      "name": "Persona 1",
      "mbox": "mailto:persona1@example.org"
    },
    {
      "name": "Persona 2",
      "mbox": "mailto:persona2@example.org"
    },
```

Fig 10. Multiple personae represented in the specification.



```
{
  "component": "https://w3id.org/xapi/tla#evaluated",
  "weight": -0.2,
  "objectOverride": {
    "objectType": "Agent",
    "name": "Persona 1",
    "mbox": "mailto:persona1@example.com"
  }
```

Fig 11. Resulting Statements: 'Persona 2' Evaluated 'Persona 1'.

Secondly, the conclusion to this test is that the Object Override Alignment is fully functioning and can be configured using an Actor as an Object.

4.2 Simulation Repeatability

In this test, each simulated dataset from one-hundred simulations (fifty each on two different hardware deployments) were stored. The seed was changed for a subset of every set of simulations run.

This test ran using the Master Object Model xAPI Profile, one Group, and three Actors assigned to that group — with no Alignments. The resulting data was run through a cryptographic hash algorithm (SHA) to show if it were identical or not. All one-hundred simulations where the specification were the same created an identical dataset to the previous set of data.

Run	Seed	Result SHA
1 (Machine 1)	42	...3f73c35a5e8367c
2 (Machine 1)	42	...3f73c35a5e8367c
...		
25 (Machine 1)	42	...3f73c35a5e8367c
26 (Machine 1)	43	...91e8c98f03a48c6
27 (Machine 1)	43	...3f73c35a5e8367c
...		
50 (Machine 1)	43	...3f73c35a5e8367c
1 (Machine 2)	42	...3f73c35a5e8367c

2 (Machine 2)	42	...3f73c35a5e8367c
...		
25 (Machine 2)	42	...3f73c35a5e8367c
26 (Machine 2)	43	...91e8c98f03a48c6
27 (Machine 2)	43	...91e8c98f03a48c6
...		
50 (Machine 2)	43	...91e8c98f03a48c6

The results above show that the simulation can be repeated many times with the same input and seed on different machines and will result in the same outcome. A change to seed will completely change that output, but a change back will reproduce the original dataset.

4.3 Profile Compatibility

In this test, all of the Profiles used the same simulation specifications. The simulations were able to generate a deterministically simulated dataset — meaning that the simulations generated featured no randomness between the starting condition and the future state of the activity, but rather were governed by the xAPI Profiles and the simulation specification.

Profile	**Original**	**Identified Problem?**	**Fixed**
cmi5	Pass	N/A	N/A
Yet Calibration Profile	Pass	N/A	N/A
TCCC	Pass	N/A	N/A
DoD MOM	Pass	N/A	N/A
STEEL-R GIFT	Pass	N/A	N/A

4.4 Micro-Scale Fidelity

To test the fidelity of a small batch of Actors in a short time span, the goal was to simulate a 90 minute activity in a group of thirty-five Personae. This simulation was run 100 times with varying Alignments. In this scenario, it was expected that DATASIM would produce:

- a limited number of statements due to the time limit
- statements generated for each of the 35 Personae
- and for the statements to follow the given Patterns within the simulation.

The outcome generated on average 149 statements:

# of Simulations	Average # of Statements Generated	Average Statement per Actor	Average # of Statements per Template
100	149	4	59

4.5 LRS Cross-compatibility

In this test, generated data was sent to each of the following xAPI Learning Record Stores (LRSs) for the purpose of demonstrating the validity of the data and the consistency of validation across any conformant LRS.

LRS	Stored?
Yet Cloud LRS (Enterprise V3)	Yes
Yet SaaS Pro LRS (xAPI Sandbox)	Yes
Rustici SCORM Cloud LRS	Yes
Watershed LRS	Yes

The conclusion is that this iteration of DATASIM generates conformant xAPI data.

4.6 Simulation Performance

4.6.1 Clustered Generation and Scaling Rate

In order to test the increased statement generation capabilities enabled by the addition of distributed generation and computing clusters, numerous tests were run at varying cluster sizes. The Master Object Model xAPI Profile was used for these tests.

For distributed cloud instances, AWS EC2 m4.large instances were used.

The concurrency rate is scaled with the machine count at a rate of 1-2, so a concurrency of 4 took 8 VMs to run for example. The results below are a sample of the configurations that were deployed and tested.

Hardware	Concurrency	Statements	Time
1x local 2.6GHz 16GB	1	1,000,000	00:03:27
64x m4.large	32	100,000,000	00:44:00
128x m4.large	64	446,452,000	02:05:00

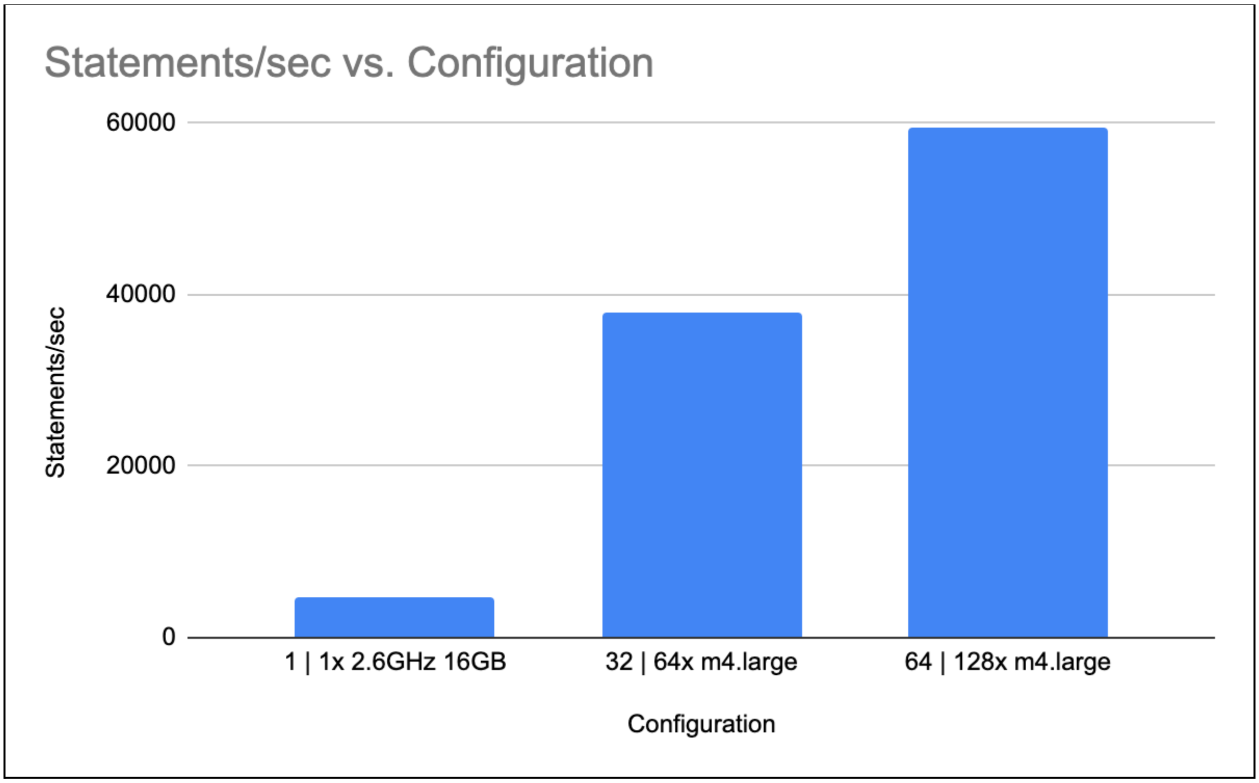


Fig 12. Comparison of DATASIM distributed generation at various configurations and a single threaded local simulation. At a concurrency of 64, DATASIM produced nearly 60,000 MOM-conformant xAPI statements per second.

This test demonstrated that the distributed generation capabilities added to DATASIM allow for a significant increase in throughput.

This distributed model can be extended to a higher concurrency than what was performed in this test — up to 128 machines.

No maximum concurrency or point of diminishing returns in our tests was observed.

Note that one of the challenges of performing these tests was constructing an adequate infrastructure to receive this data.

While AWS has very scalable storage options, bottlenecks were often hit in receiving statements both in the S3 Service and in the Kinesis Firehose service — both of which are configured already to receive large amounts of data.

4.7 Validation Testing

4.7.1 Parameters

This test demonstrates the ability to create accurate errors when the user adds specification to the Basic Mode in the Parameters section of DATASIM.

Each aspect of the Parameter specification was tested individually with information that is known to cause an error.

What follows is the record of what occurred during each test, and where applicable suggestions as to further development or mitigations against the error.

Occasionally this is a UI/UX fix which would use design to direct the user away from behavior that would produce the error. Other recommendations may suggest re-writing code for the purpose of changing the behavior of the platform.

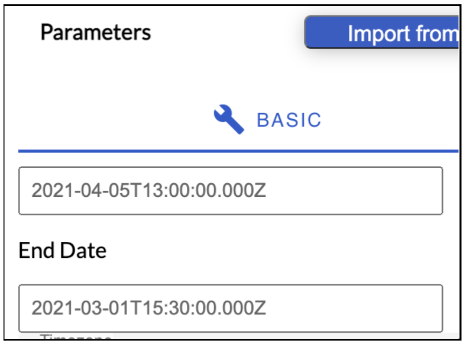
4.7.1.1Start/End Date

The Start and End Date/Time were set to where the End Date is before the Start Date.

This test did not result in an error, but it also did not create a dataset.

A suggested improvement would be to add a visual validation error notifying the user why the simulation is not executing, to better allow them to navigate to the underlying issue within the Parameter specification.

End Date before Start Date



2021-04-05T13:00:00.000Z

End Date

2021-03-01T15:30:00.000Z

Fig 13. The End Date is set to a date before the Start Date. This will cause an error, but not a visual validation.

4.7.1.2 Date Formatting

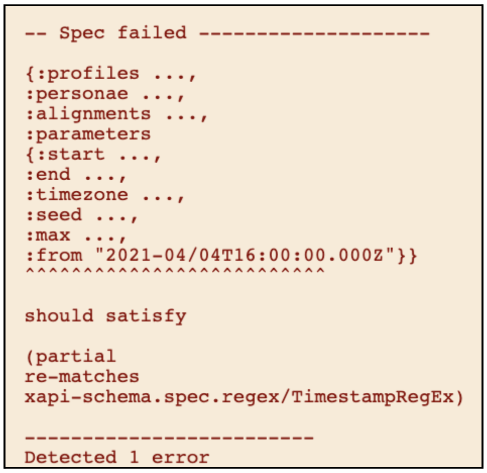
For this validation test, formatting errors were created within the date field by typing in the date or time outside of the specification.

This test can only be done in the Advanced Mode because the Basic Mode constrains user error with regards to date and time formatting.

This error was caused by the inclusion of a ‘/’ instead of a ‘-’ in the From spec.

The ‘should satisfy’ describes the Timestamp to be the issue and the carrot symbols above notate the specific line that needs attention.

Validation Error



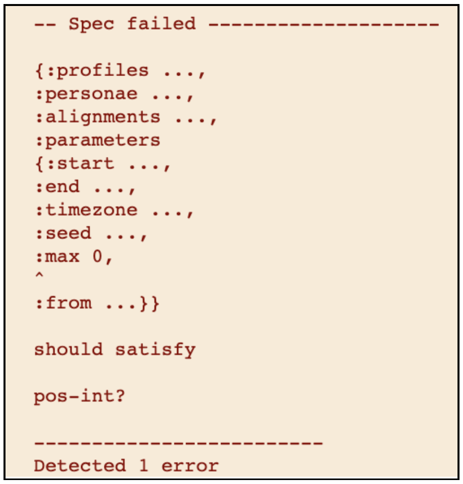
```
-- Spec failed --------------------

{:profiles ...,
:personae ...,
:alignments ...,
:parameters
{:start ...,
:end ...,
:timezone ...,
:seed ...,
:max ...,
:from "2021-04/04T16:00:00.000Z"}}
^^^^^^^^^^^^^^^^^^^^^^^^^^

should satisfy

(partial
re-matches
xapi-schema.spec.regex/TimestampRegEx)

-------------------------
Detected 1 error
```

Fig 14. Error caused by an incompatible ‘from’ specification. This is not required in the Parameter.

4.7.1.3 Max Statement - Null

In this test, the Max Statement field was set to zero in order to be sure that the error occurs. In the validation error, the user can find the section that causes the error with a ^ symbol below the specification. The “should satisfy” section outlines the need for this integer to be positive, ‘pos-int?’. The carrot symbol shows the ‘max’ specification needs to be changed.

Validation Error



```
-- Spec failed --------------------

{:profiles ...,
:personae ...,
:alignments ...,
:parameters
{:start ...,
:end ...,
:timezone ...,
:seed ...,
:max 0,
^
:from ...}}

should satisfy

pos-int?

-------------------------
Detected 1 error
```

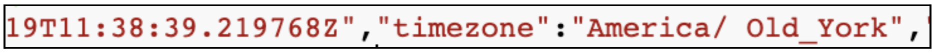
Fig 15. Error caused by zero or null Mx Statement.

4.7.1.4 Invalid Timezone

The last Parameter test was for an invalid Time Zone. This error is rare and would not occur unless the user chooses to use the Advanced Mode. In this test, the timezone code was changed from America/New_York to America/ Old_York. The results of this test did not result in an error, but also did not create a dataset.

A suggested improvement would be to create error messages for each scenario that can cause a simulation to fail.

Error



```
19T11:38:39.219768Z","timezone":"America/ Old_York",
```

Fig 16.Caused by invalid timezone code. This can only occur when using the Advanced Mode.

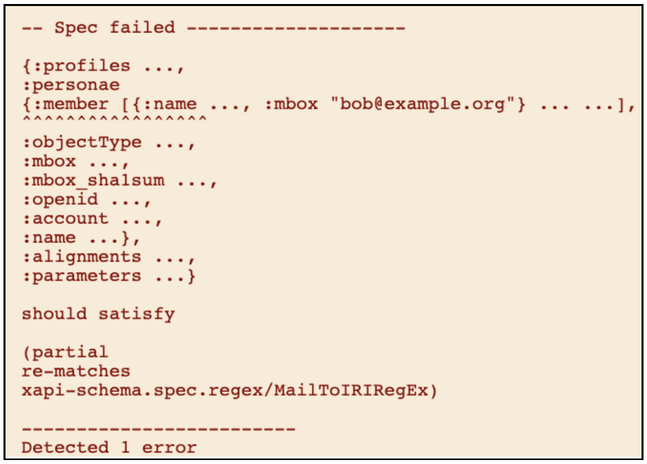
4.7.2 Personae

In the new Basic Mode interface, Personae can be added using simple fields and dropdowns. To test the Personae validation, both formatting and input errors were created.

4.7.2.1 Invalid Mbox Syntax

In the Basic Mode of the Personae section, the user can add new Actors and their email addresses identifier without coding knowledge.

The identifier element is one of the few user errors that can be created using the user interface rather than the Advanced Mode (JSON). In this test, the text ‘mailto:’ was removed from the Basic Mode and ran the simulation.

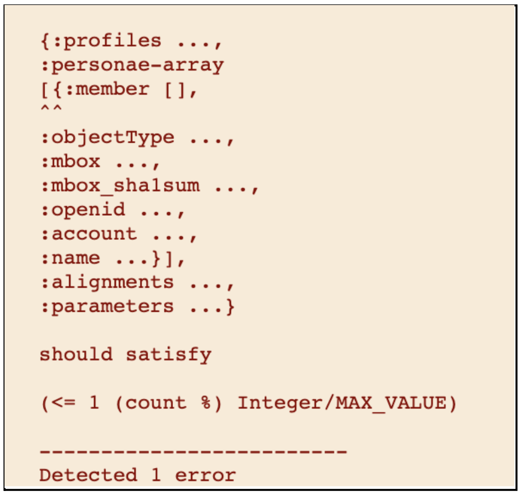


```
-- Spec failed --------------------

{:profiles ...,
:personae
{:member [{:name ..., :mbox "bob@example.org"} ... ...],
^^^^^^^^^^^^^^^^^
:objectType ...,
:mbox ...,
:mbox_sha1sum ...,
:openid ...,
:account ...,
:name ...},
:alignments ...,
:parameters ...}

should satisfy

(partial
re-matches
xapi-schema.spec.regex/MailToIRIRegEx)

-------------------------
Detected 1 error
```

Fig 17. Error caused by invalid syntax in the ‘mbox’ configuration.

4.7.2.2 Group Member Null

Another error that can be caused in the Basic Mode is failing to add members to a Group. In the error, it points out the empty member spec and shows the count needs to be more than one.



```
{:profiles ...,
:personae-array
[{:member [],
^^
:objectType ...,
:mbox ...,
:mbox_sha1sum ...,
:openid ...,
:account ...,
:name ...}],
:alignments ...,
:parameters ...}

should satisfy

(<= 1 (count %) Integer/MAX_VALUE)

-------------------------
Detected 1 error
```

Fig 18. Error caused by lack of members in a group.

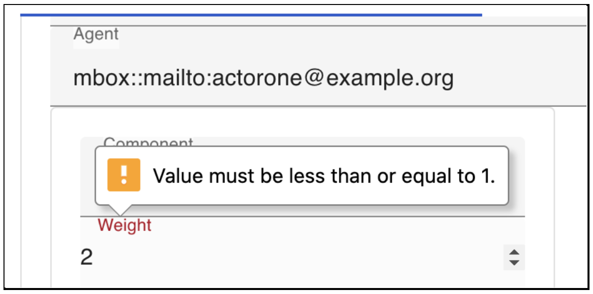
4.7.3 Alignments

This series of tests takes a set of identical simulation parameters and changes certain aspects to create seven validation errors within the Alignment configuration. The Actors, Roles, and Groups used in this test were the same as in the Personae validation testing.

Actor 1	https://w3id.org/xapi/tla#assessed	.5
Actor 2	https://w3id.org/xapi/tla#assessed	-.5
Actor 3	https://w3id.org/xapi/tla#assessed	.5
Actor 4	https://w3id.org/xapi/tla#assessed	.75
Role 1	https://w3id.org/xapi/tla#assessed	-.25
Role 2	https://w3id.org/xapi/tla#assessed	.25
Group 1	https://w3id.org/xapi/tla#assessed	-.3

4.7.3.1 Validation Error: Invalid Integer

In the first test, an invalid integer was added to the weight field. This is one of the few pop up errors that appears in red before the simulation is run. DATASIM then shows “Value must be less than or equal to 1”.



Agent
mbox::mailto:actorone@example.org

Component

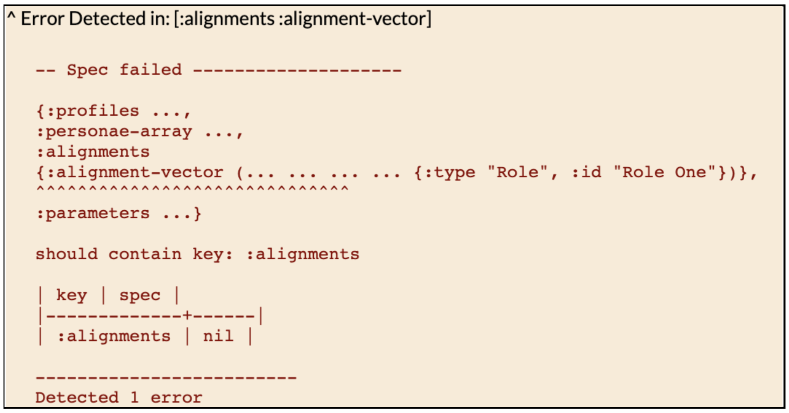
Value must be less than or equal to 1.

Weight
2

Fig 19. Error caused by invalid integer. Value must be between -1 and 1.

4.7.3.2 Alignment Weight Null

In this test, an Alignment weight was not added to one of the parameters. This was done for a Role. The error shows which Alignment contained an error and what was missing from that Alignment.



```
^ Error Detected in: [:alignments :alignment-vector]

  -- Spec failed --------------------

  {:profiles ...,
  :personae-array ...,
  :alignments
  {:alignment-vector (... ... ... ... {:type "Role", :id "Role One"})},
  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  :parameters ...}

  should contain key: :alignments

  | key | spec |
  |-------------+------|
  | :alignments | nil |

  -------------------------
  Detected 1 error
```

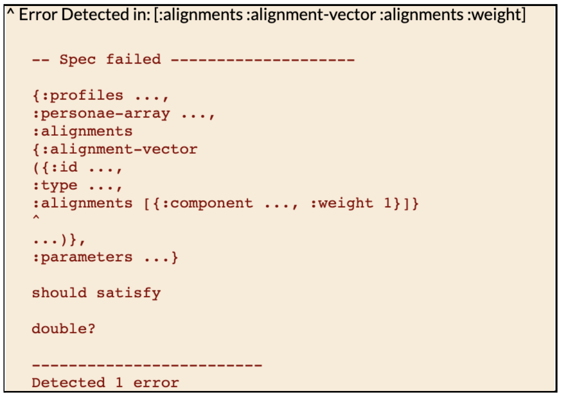
Fig 20. Error caused by invalid non-array Personae.

4.7.3.3 Invalid Integer Syntax

In this test, the weight was added in the Basic Mode using 1 and -1. It was done both by typing the number into the field and by using the up and down arrows to the right of the field.

This error shows that, even though the number is within the spec, it is not in correct JSON syntax because it lacks the decimal integer (1.0).

This could be an area of improvement by forcing the Basic Mode user interface to add the decimal integer allowing the simulation to run.

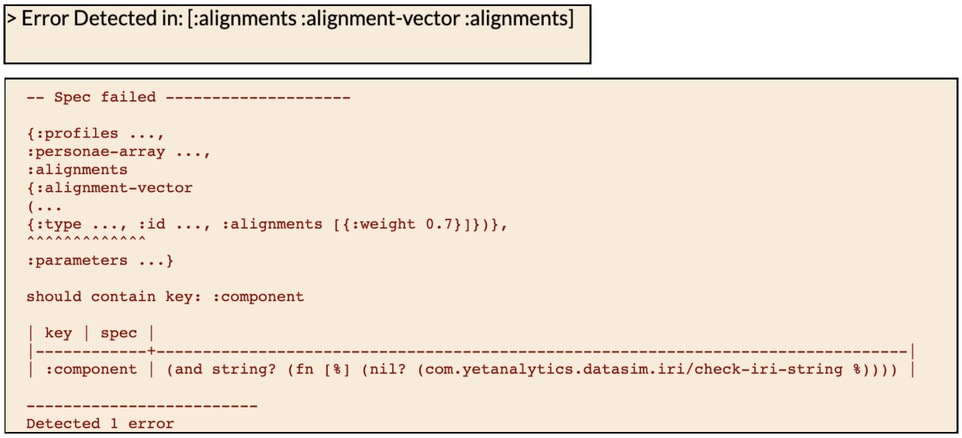


```
^ Error Detected in: [:alignments :alignment-vector :alignments :weight]

  -- Spec failed --------------------

  {:profiles ...,
  :personae-array ...,
  :alignments
  {:alignment-vector
  ({:id ...,
  :type ...,
  :alignments [{:component ..., :weight 1}]}
  ^
  ...)},
  :parameters ...}

  should satisfy

  double?

  -------------------------
  Detected 1 error
```

Fig 21. This is caused by not adding the '.0' to -1 or 1 in the weight configuration.

4.7.3.4 Invalid Advanced Mode Personae Syntax

If the user is attempting to use the Advanced Mode to create the Alignment specifications, there is far greater risk of making mistakes. In this validation test, the "mailto:" was removed from an Alignment in the Advanced Mode. This created the error below displaying



```
> Error Detected in: [:alignments :alignment-vector :alignments]
```

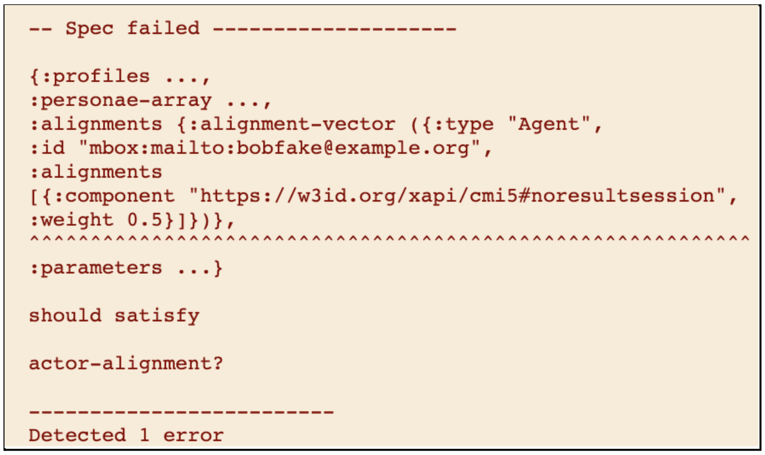
```
 -- Spec failed --------------------

 {:profiles ...,
 :personae-array ...,
 :alignments
 {:alignment-vector
 (...
 {:type ..., :id ..., :alignments [{:weight 0.7}]})},
 ^^^^^^^^^^^^^
 :parameters ...}

 should contain key: :component

 | key | spec |
 |------------+---------------------------------------------------------------------------------|
 | :component | (and string? (fn [%] (nil? (com.yetanalytics.datasim.iri/check-iri-string %)))) |

 -------------------------
 Detected 1 error
```

Fig 22. Error caused by invalid component. The Personae that was selected, was later removed.

4.7.3.5 Component Formatting

In this formatting test, the user removed just one semicolon from mbox::mailto: to see if the error message would sufficiently point the user to the correct solution.

This error shows that the error is in a specific Alignment and what type of Alignment it was. The carrot symbols show the area that needs attention, but not the exact syntax that needs to be changed. This information narrowed the area of review enough to find the missing syntax.

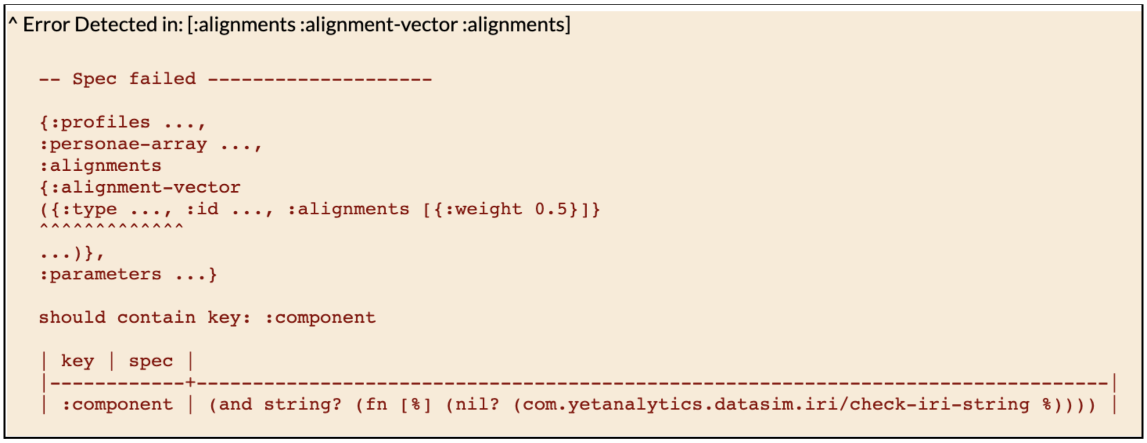


```
-- Spec failed --------------------

{:profiles ...,
:personae-array ...,
:alignments {:alignment-vector ({:type "Agent",
:id "mbox:mailto:bobfake@example.org",
:alignments
[{:component "https://w3id.org/xapi/cmi5#noresultsession",
:weight 0.5}]})},
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
:parameters ...}

should satisfy

actor-alignment?

-------------------------
Detected 1 error
```

Fig 23. Error caused by syntax error.

4.7.3.6 Alignment Component Null

For this test, the component was left blank in the Basic Mode. The error shows the parameters should contain a component and outlines which component needs to be updated.



^ Error Detected in: [:alignments :alignment-vector :alignments]

```
-- Spec failed --------------------

{:profiles ...,
:personae-array ...,
:alignments
{:alignment-vector
({:type ..., :id ..., :alignments [{:weight 0.5}]}
^^^^^^^^^^^^^
...)},
:parameters ...}

should contain key: :component

| key | spec |
|------------+---------------------------------------------------------------------------------|
| :component | (and string? (fn [%] (nil? (com.yetanalytics.datasim.iri/check-iri-string %)))) |
```

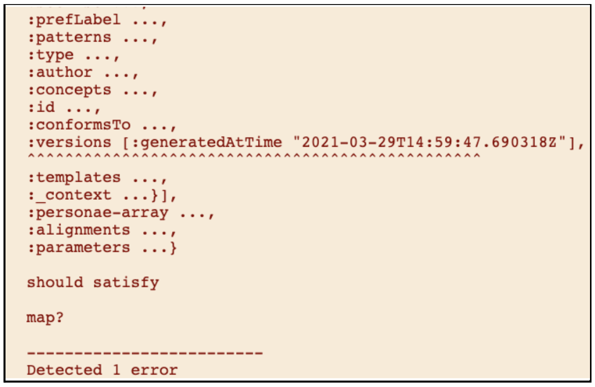
Fig 24. Error caused by having a null component.

4.7.4 Profile Validation

There are many error configurations for Profile validation and testing all of the scenarios that could occur while building xAPI Profiles in DATASIM would be difficult. Therefore, the following tests were done with the most common errors made when creating an xAPI Profile. The purpose of these tests was to ensure that the location within the xAPI Profile and error message are correct.

4.7.4.1 Map? Validation Error

This error shows that the ‘generated at time’ is not valid. This error was accompanied by another ‘map?’ error for the versions component.

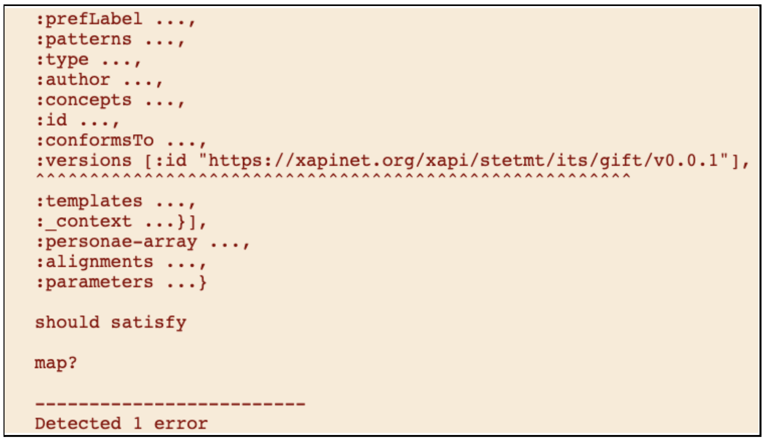


```
:prefLabel ...,
:patterns ...,
:type ...,
:author ...,
:concepts ...,
:id ...,
:conformsTo ...,
:versions [:generatedAtTime "2021-03-29T14:59:47.690318Z"],
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

:templates ...,
:_context ...}],
:personae-array ...,
:alignments ...,
:parameters ...}

should satisfy

map?

-------------------------
Detected 1 error
```

Fig 25. Error caused by invalid ‘generatedattime’.

4.7.4.2 Map? Profile Version Validation Error

This error was rendered at the same time as the error above. It is notifying the user that the ‘versions’ configuration is not valid.

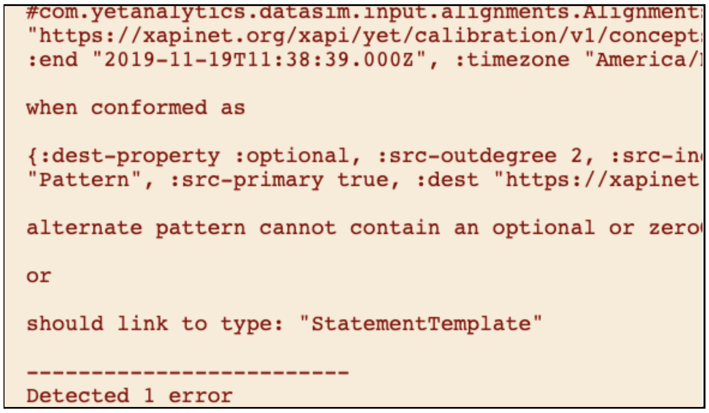


```
:prefLabel ...,
:patterns ...,
:type ...,
:author ...,
:concepts ...,
:id ...,
:conformsTo ...,
:versions [:id "https://xapinet.org/xapi/stetmt/its/gift/v0.0.1"],
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
:templates ...,
:_context ...}],
:personae-array ...,
:alignments ...,
:parameters ...}

should satisfy

map?

-------------------------
Detected 1 error
```

Fig 26. Error caused by invalid 'version'.

4.7.4.3 Pattern not linked to Statement Template

In this error the Researcher did not tie the Pattern to the Statement Template. The user can find the Pattern that needs to be linked to a Statement Template. The user will still need basic knowledge of how xAPI Profiles are structured.

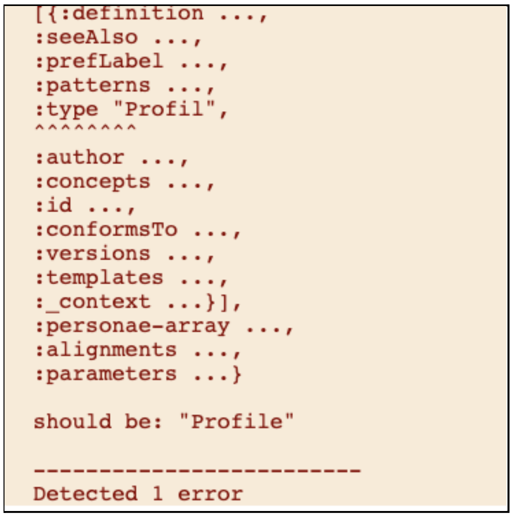


```
#com.yetanalytics.datasim.input.alignments.Alignment
"https://xapinet.org/xapi/yet/calibration/v1/concept
:end "2019-11-19T11:38:39.000Z", :timezone "America/

when conformed as

{:dest-property :optional, :src-outdegree 2, :src-in
"Pattern", :src-primary true, :dest "https://xapinet

alternate pattern cannot contain an optional or zero

or

should link to type: "StatementTemplate"

-------------------------
Detected 1 error
```

Fig 27. Error caused by the Pattern not being linked to a Statement Template.

4.7.4.4 Invalid Profile Type

In this validation error, the user incorrectly added the type of Profile. The error notifies the user of which section needs to be edited.

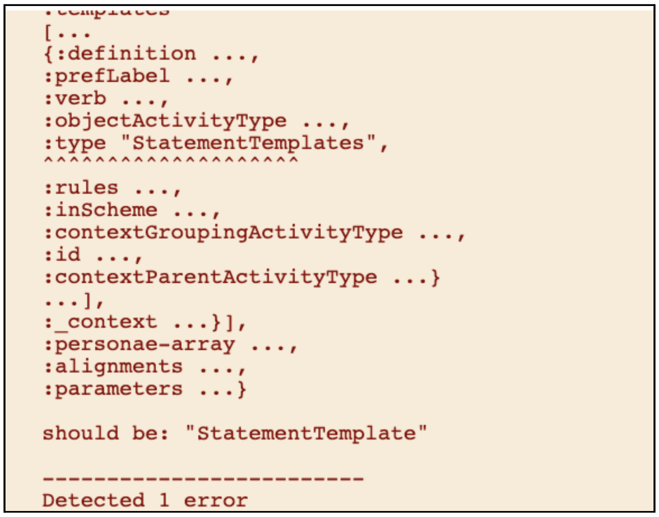


```
[{:definition ...,
:seeAlso ...,
:prefLabel ...,
:patterns ...,
:type "Profil",
^^^^^^^^

:author ...,
:concepts ...,
:id ...,
:conformsTo ...,
:versions ...,
:templates ...,
:_context ...}],
:personae-array ...,
:alignments ...,
:parameters ...}

should be: "Profile"

-------------------------
Detected 1 error
```

Fig 28. Error caused by typo in the 'type' configuration.

4.7.4.5 Invalid Statement Template Type

This error was caused by adding a typo or invalid type to the Profile specification. The particular type that is causing the error is highlighted with the carrot symbol and the correction is listed at the bottom with the correct suggested type.



```
:templates
[...
{:definition ...,
:prefLabel ...,
:verb ...,
:objectActivityType ...,
:type "StatementTemplates",
^^^^^^^^^^^^^^^^^^^^

:rules ...,
:inScheme ...,
:contextGroupingActivityType ...,
:id ...,
:contextParentActivityType ...}
...],
:_context ...}],
:personae-array ...,
:alignments ...,
:parameters ...}

should be: "StatementTemplate"

-------------------------
Detected 1 error
```

Fig 29. Error caused by invalid 'type' configuration.

4.8 User Interface

As DATASIM has been used by real world users, much has been learned which has informed the adaptation and evolution of the platform's UI to meet the stakeholder demands and expectations.

Since last year's TRL4 testing, significant changes have been made in the overall look and feel of the platform. In this section, the report indicates the efficacy of the UI changes that were made during this iteration of the project.

All UI/UX considerations have taken into account user feedback as well as best practices in design, especially as regards standards-based compliance for universal design.

The main goal of the UI is to allow non-technical business professionals to be able to navigate the product with ease.

4.8.1 Parameters Page

The results of the Parameter user interface test established that when a user adds each Parameter specification, it is updated in the Advanced Mode (JSON).

When adding the start date/time and end date/time, a drop down will appear with a calendar at the top and time at the bottom of the drop down.

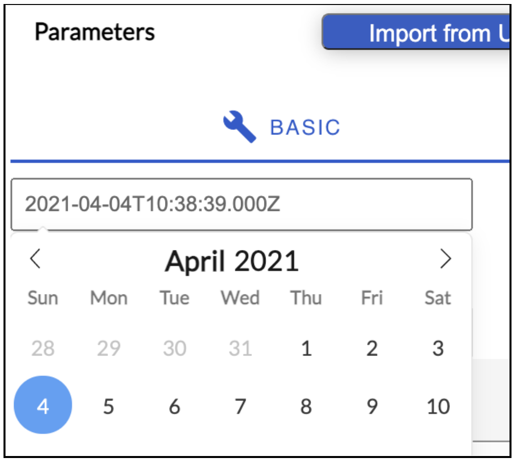
Specification Test	Successful
Start Date/Time	Yes
End Date/Time	Yes
Seed Value	Yes
Max Statement	Yes

4.8.1.1 Start/End Date and Time

In the Basic Mode, this test used the calendar widget to update the date and time to April 4, 2021 10:38:39.000.

When viewed in Advanced Mode the date has been updated to the specifications set in the Basic Mode.

Basic Mode



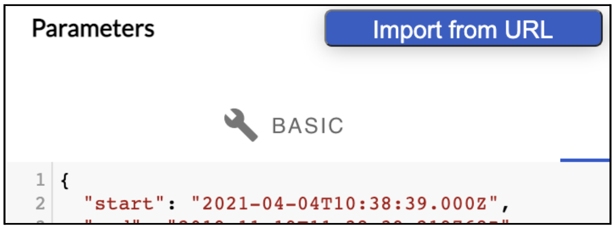
2021-04-04T10:38:39.000Z

April 2021

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Fig 30. Adding date through Basic Mode

Advanced Mode



```
{
  "start": "2021-04-04T10:38:39.000Z",
```

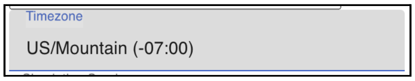
Fig 31. Showing the JSON in the Advanced Mode

4.8.1.2 Timezone

The remaining tests follow the same format for each aspect of the Parameter specification.

The Timezone was changed to US/Mountain (-7:00) in the Basic Mode and then viewed in the Advanced Mode to assure the code has been updated.

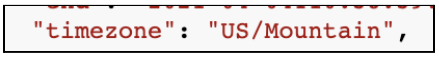
Basic Mode



Timezone
US/Mountain (-07:00)

Fig 32. Adding Timezone using the Basin Mode.

Advanced Mode



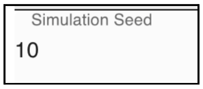
```
"timezone": "US/Mountain",
```

Fig 33. Showing the timezone in Advanced Mode

4.8.1.3 Seed Value

The Seed Value was changed to 10 in the Basic Mode and then viewed in the Advanced Mode to verify that the seed has updated to the correct integer. Any number can be added to this field and the syntax will correctly configure the specifications in JSON.

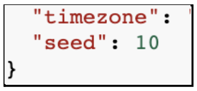
Basic Mode



Simulation Seed
10

Fig 34. Showing the seed value in Basic Mode.

Advanced Mode



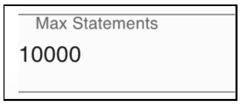
```
  "timezone":
  "seed": 10
}
```

Fig 35. Showing the seed in Advanced Mode.

4.8.1.4 Max Statement

The user interface for the Max Statement uses an integer field to allow the user to type in the desired amount of statements. In this test, the Max Statement was changed to 10,000 and then viewed in the Advanced Mode.

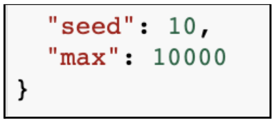
Basic Mode



Max Statements
10000

Fig 36. Showing the Max Statement in Basic Mode.

Advanced Mode



```
  "seed": 10,
  "max": 10000
}
```

Fig 37. Showing the Max Statement in Advanced Mode.

4.8.2 Personae Page

In this test simulation, Nine Actors were created in the DATASIM Personae Section and assigned to a Group (change name to Squad A).

Actor 1 was renamed to Squad Leader, assigned to Role 1 and was associated with Group 1.

Once all of the Groups and Actors were created in Basic Mode, they were viewed in Advanced Mode and a simulation was executed to ensure there were no validation errors.

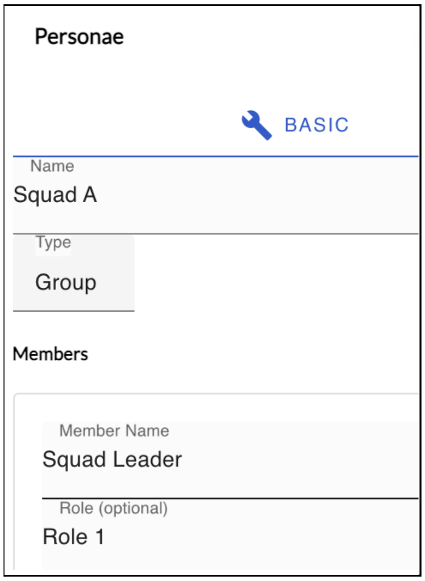
4.8.2.1 Adding Groups and Members

The user types in the Group Name at the top.

The Group is already selected as the Type and Members can be added using the button at the bottom of the section.

When the user switches to Advanced Mode, all of the members have been added via Basic Mode.

Basic Mode



Personae

Name
Squad A

Type
Group

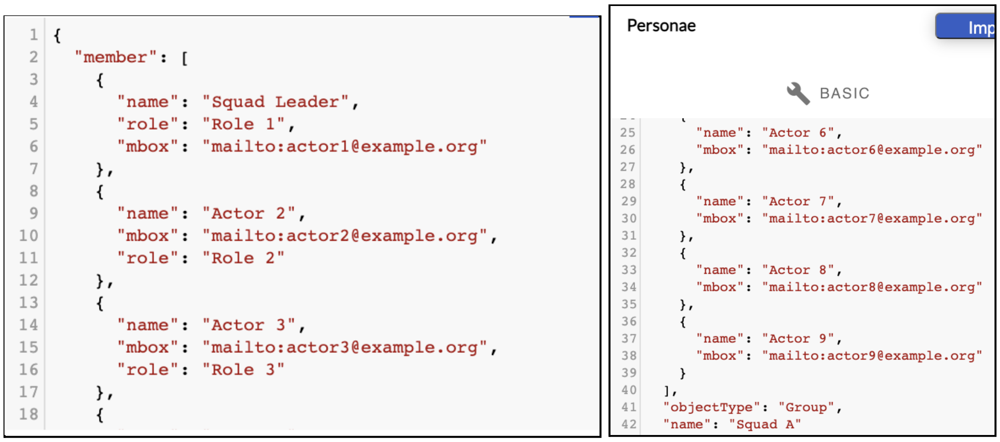
Members

Member Name
Squad Leader

Role (optional)
Role 1

Fig 38. Adding a Group, naming it Squad A and adding a member with the role of ‘Role 1’.

Advanced Mode



```
{
  "member": [
    {
      "name": "Squad Leader",
      "role": "Role 1",
      "mbox": "mailto:actor1@example.org"
    },
    {
      "name": "Actor 2",
      "mbox": "mailto:actor2@example.org",
      "role": "Role 2"
    },
    {
      "name": "Actor 3",
      "mbox": "mailto:actor3@example.org",
      "role": "Role 3"
    },
    {
```

```
      "name": "Actor 6",
      "mbox": "mailto:actor6@example.org"
    },
    {
      "name": "Actor 7",
      "mbox": "mailto:actor7@example.org"
    },
    {
      "name": "Actor 8",
      "mbox": "mailto:actor8@example.org"
    },
    {
      "name": "Actor 9",
      "mbox": "mailto:actor9@example.org"
    }
  ],
  "objectType": "Group",
  "name": "Squad A"
```

Fig 39. Showing all 9 Actors that were created using the UI in the Advanced Mode.

4.8.3 Alignment Page

To test the user interface of the Alignment specification, 4 Actors were created with varying Roles and were allotted varying weights in the Alignment. Once the Personae structure was created, the Alignments were all added to the same simulation specification using the Master Object Model xAPI Profile.

Personae Specifications

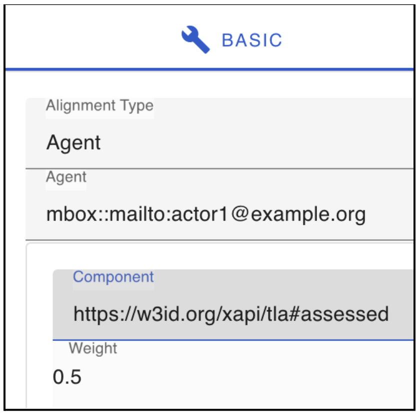
Actor 1 was assigned to Role 1 in Group 1
Actor 2 was assigned to Role 2 in Group 1
Actor 3 was assigned to Role 1 in Group 1
Actor 4 was assigned to Role 2 in Group 1

Alignment Specifications

Alignment Type	Component	Weight	Successful
Actor 1	https://w3id.org/xapi/tla#assessed	.5	Yes
Actor 2	https://w3id.org/xapi/tla/verbs/evaluated	.25	Yes
Actor 3	https://w3id.org/xapi/tla#launched	.5	Yes

Actor 4	https://w3id.org/xapi/tla/verbs/selected	-.5	Yes
Role 1	https://w3id.org/xapi/tla/verbs/evaluated	.75	Yes
Role 1	https://w3id.org/xapi/tla#assessed	.25	Yes
Role 2	https://w3id.org/xapi/tla#launched	-.75	Yes
Role 2	https://w3id.org/xapi/tla/verbs/selected	1	Yes
Group 1	https://w3id.org/xapi/tla#assessed	-.5	Yes
Group 2	https://w3id.org/xapi/tla/verbs/evaluated	.25	Yes
Group 2	https://w3id.org/xapi/tla/verbs/selected	-1	Yes

4.8.3.1 Alignment Basic Mode



Alignment Type
Agent

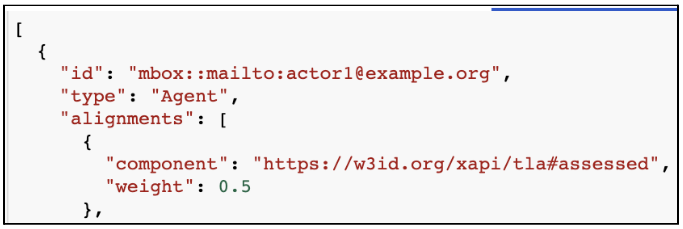
Agent
mbox::mailto:actor1@example.org

Component
https://w3id.org/xapi/tla#assessed

Weight
0.5

Fig 40. Adding an Alignment in Basic Mode.

4.8.3.2 Alignment Advanced Mode



```
[
  {
    "id": "mbox::mailto:actor1@example.org",
    "type": "Agent",
    "alignments": [
      {
        "component": "https://w3id.org/xapi/tla#assessed",
        "weight": 0.5
      },
```

Fig 41. Showing the Alignment in Advanced Mode.

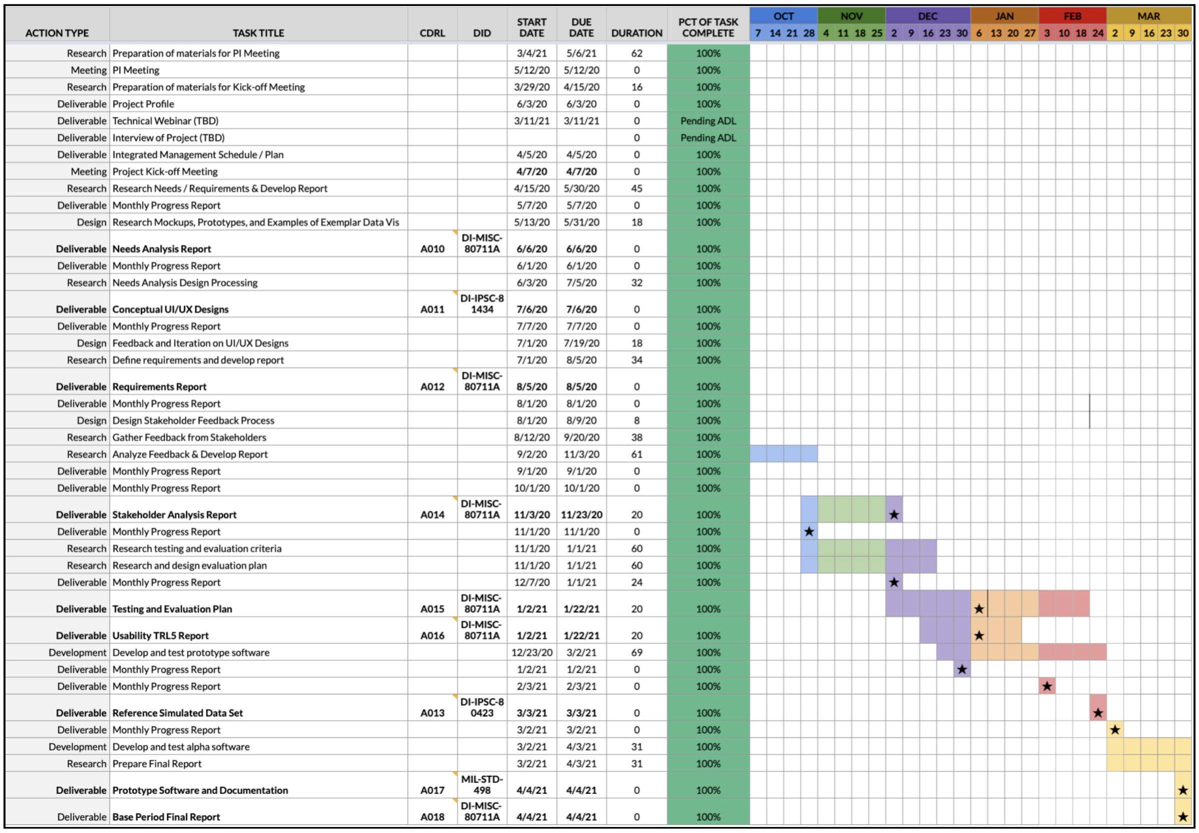
5. Option Y1 Deliverables

Scheduled Option Year One Tasks (referenced in the DATASIM SOW and Requirements Report) have been completed in their entirety. The tasks contained:

- Input Specification Requirements
  - Alignment Specification Updates including Zero-statement, Object Override, and Group and Role Alignment
  - Personae Specification Requirements including multiple Groups and Actor Roles
- Validation Changes to adhere to specification updates
  - xAPI Profile Validation
  - Alignment Specification Validation
  - Personae Specification Validation
- Simulation Requirements
  - Pattern Sequence Consistency
  - Alignment Runtime Changes
- UI / UX Updates
  - xAPI Profile Enhanced Editor UI
  - Parameter Specification input UI
  - Personae Specification input UI
  - Alignment Specification input UI
- Enabled Large-scale Distributed Generation
- Documentation
- Source Control
- Versioning + Changes
- Testing to document and demonstrate functionality
- Create a reference xAPI simulation dataset that is reproducible with a seed value
- Stakeholder Analysis Report

- UI/UX Usability Report
- Evaluation Plan
- Final Report

Find below the completed master plan for Option Year I including all deliverables.



ACTION TYPE	TASK TITLE	CDRL	DID	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
Research	Preparation of materials for PI Meeting			3/4/21	5/6/21	62	100%
Meeting	PI Meeting			5/12/20	5/12/20	0	100%
Research	Preparation of materials for Kick-off Meeting			3/29/20	4/15/20	16	100%
Deliverable	Project Profile			6/3/20	6/3/20	0	100%
Deliverable	Technical Webinar (TBD)			3/11/21	3/11/21	0	Pending ADL
Deliverable	Interview of Project (TBD)					0	Pending ADL
Deliverable	Integrated Management Schedule / Plan			4/5/20	4/5/20	0	100%
Meeting	Project Kick-off Meeting			**4/7/20**	**4/7/20**	0	100%
Research	Research Needs / Requirements & Develop Report			4/15/20	5/30/20	45	100%
Deliverable	Monthly Progress Report			5/7/20	5/7/20	0	100%
Design	Research Mockups, Prototypes, and Examples of Exemplar Data Vis			5/13/20	5/31/20	18	100%
Deliverable	**Needs Analysis Report**	**A010**	**DI-MISC-80711A**	**6/6/20**	**6/6/20**	0	100%
Deliverable	Monthly Progress Report			6/1/20	6/1/20	0	100%
Research	Needs Analysis Design Processing			6/3/20	7/5/20	32	100%
Deliverable	**Conceptual UI/UX Designs**	**A011**	**DI-IPSC-81434**	**7/6/20**	**7/6/20**	0	100%
Deliverable	Monthly Progress Report			7/7/20	7/7/20	0	100%
Design	Feedback and Iteration on UI/UX Designs			7/1/20	7/19/20	18	100%
Research	Define requirements and develop report			7/1/20	8/5/20	34	100%
Deliverable	**Requirements Report**	**A012**	**DI-MISC-80711A**	**8/5/20**	**8/5/20**	0	100%
Deliverable	Monthly Progress Report			8/1/20	8/1/20	0	100%
Design	Design Stakeholder Feedback Process			8/1/20	8/9/20	8	100%
Research	Gather Feedback from Stakeholders			8/12/20	9/20/20	38	100%
Research	Analyze Feedback & Develop Report			9/2/20	11/3/20	61	100%
Deliverable	Monthly Progress Report			9/1/20	9/1/20	0	100%
Deliverable	Monthly Progress Report			10/1/20	10/1/20	0	100%
Deliverable	**Stakeholder Analysis Report**	**A014**	**DI-MISC-80711A**	**11/3/20**	**11/23/20**	20	100%
Deliverable	Monthly Progress Report			11/1/20	11/1/20	0	100%
Research	Research testing and evaluation criteria			11/1/20	1/1/21	60	100%
Research	Research and design evaluation plan			11/1/20	1/1/21	60	100%
Deliverable	Monthly Progress Report			12/7/20	1/1/21	24	100%
Deliverable	**Testing and Evaluation Plan**	**A015**	**DI-MISC-80711A**	**1/2/21**	**1/22/21**	20	100%
Deliverable	**Usability TRL5 Report**	**A016**	**DI-MISC-80711A**	**1/2/21**	**1/22/21**	20	100%
Development	Develop and test prototype software			12/23/20	3/2/21	69	100%
Deliverable	Monthly Progress Report			1/2/21	1/2/21	0	100%
Deliverable	Monthly Progress Report			2/3/21	2/3/21	0	100%
Deliverable	**Reference Simulated Data Set**	**A013**	**DI-IPSC-80423**	**3/3/21**	**3/3/21**	0	100%
Deliverable	Monthly Progress Report			3/2/21	3/2/21	0	100%
Development	Develop and test alpha software			3/2/21	4/3/21	31	100%
Research	Prepare Final Report			3/2/21	4/3/21	31	100%
Deliverable	**Prototype Software and Documentation**	**A017**	**MIL-STD-498**	**4/4/21**	**4/4/21**	0	100%
Deliverable	**Base Period Final Report**	**A018**	**DI-MISC-80711A**	**4/4/21**	**4/4/21**	0	100%

Fig 42. Option Year 1 master plan.

6. Conclusion

This report discusses the work that was accomplished in the recent iteration of the DATASIM platform. Based on the results of testing as described in this report, DATASIM is now in Technology Readiness Level 5.

The use of enhanced DATASIM-UI capabilities has significantly broadened the potential audience of the platform to include users who may be unfamiliar with JSON. The UI also has enabled the creation of a simulated dataset through the use of only an interactive interface (without the need to write code).

Three key conclusions can be made at this stage in the project's development:

- Through the numerous specification changes and simulation enhancements, the platform has expanded the potential simulation scenarios for which it can be used.
- Through the implementation of distributed, clustered generation the platform is now capable of throughputs that should satisfy extremely high volume scenarios.
- Through real-world usage with DoD stakeholders, the applicability of and need for the platform is being clearly defined and recognized. For instance, DATASIM is currently being used to assist in the design and testing of Army STTC's STEEL-R project.

Now that the product is functioning to the specified simulation capabilities and meets TRL5, the focus of the next section is to explain what other areas for improvement DATASIM must undergo in order to enable it to directly benefit DoD and Federal Stakeholders as an operational software platform.

There are some key areas that must be developed before DATASIM is ready for transition to operational usage. **Three of these developments are considered crucial and required for DATASIM to transition into TRL6**:

- DATASIM should be integrated with the ADL Profile Server so that DATASIM's xAPI Profile validation facilities may be integrated into the conformance validation of all of ADL's published xAPI Profiles.
- DATASIM's temporal modeling facility should be built out so that users can control and model realistic scenarios cognizant of time zones, sleep schedules, and limited-duration activities such as a semester-long course as opposed to flexible-duration activities such as informal mobile learning (whereas the current implementation only provides a naive temporal implementation of simulated activity over time which results in unrealistic clustered time cycles).
- DATASIM should undergo testing and hardening necessary to complete all required cybersecurity alignments so that DATASIM may be granted Authority to Operate (ATO) in operational environments.

7. Next Steps

The Objective of this Section

This section of the report provides a cost-effective strategy for getting DATASIM into the hands of DoD and Federal Government stakeholders.

The foremost technical goal described here is to make an operational hosted prototype available in 90 days.

The Purpose of DATASIM

DATASIM provides essential modeling and data generation capabilities necessary to test and evaluate data design and technology implementation within DoD and Federal implementations of the Total Learning Architecture (TLA). Usage of DATASIM promotes every aspect of the DoD Data Strategy.

The Value of DATASIM to the Modernization of DoD and Federal Learning and Training

The sooner DATASIM's capabilities are made accessible across Joint All Domain Operations and All-of-Government learning and training technology implementations, the faster these DoD and Federal stakeholders will be able to leverage the investment made in and the capability provided by xAPI, xAPI Profiles, and the TLA.

How the strategy fits within the DoD data strategy

DATASIM provides the core modeling and data generation capabilities necessary to ensure conformity of learning and training data to DoD and relevant technical standards, to facilitate interoperability across DoD and other government systems, and to increase readiness by providing an agile and in-the-workflow testing and evaluation capability necessary to ensure the ongoing quality of DoD instructional design, data design, and IT infrastructure at the level of the TLA.

DATASIM and the DoD Data Strategy Focus Areas

This technology can provide demonstrable alignment of all aspects of the xAPI and Total Learning Architecture with the Strategy Focus Areas, namely:

- A hosted and secure DATASIM will

    - bring the power of xAPI modeling, xAPI Profile Validation, and Testing and Evaluation of TLA enabled instructional strategies and learning technologies to Joint All Domain Operations resulting in increased speed-to-the-point-of-delivery of learning and training capabilities required by stakeholders.
    - assure that the systems being used to support data-driven decisions for senior leaders across DoD management are themselves producing valid, relevant, and quality data.
    - ensure that the data used to drive informed decisions at all echelons of business analytics for TLA stakeholders is accurate, auditable, and accessible.

DATASIM and the DoD Data Strategy Guiding Principles

As a platform for continually testing and evaluating the DoD's learning and training technology and data ecosystem, DATASIM ensures that the very data that comprises the core asset of this data ecosystem is valid, accessible, rich, and meaningful to the DoD strategy and mission.

- Data is a Strategic Asset – DATASIM assures that xAPI data will be designed for lasting military advantage. Without DATASIM's capabilities to help instructional designers, analysts, and IT stewards safely test their assumptions at scale, production TLA implementations would be created in the dark.
- Collective Data Stewardship – An authenticated DATASIM supports a chain of command in data stewardship. Data sets generated by DATASIM are portable and interoperable. They can be stored in shared libraries and made accessible for testing and evaluation purposes throughout the system supporting the data lifecycle.
- Data Ethics – DATASIM provides a means for instructional designers and analysts to test data design to ensure that learning and training content, and the measurable outcomes of engagement with that content, contain no hidden bias.
- Data Collection – Strictly adhering to the xAPI Profiles specification (soon to be IEEE P9274.2.1), DATASIM ensures the validity of xAPI data and enables an auditable trail of data generated for the testing and evaluation of TLA implementations.
- Enterprise-Wide Data Access and Availability – DATASIM is designed to provide a means of testing xAPI Profile aligned activity data as it will flow across the entirety of the enterprise TLA implementation from initial data source to authoritative learner record.
- Data for Artificial Intelligence Training – During the alpha development phase of DATASIM, the application was used to generate over 100,000,000 valid and xAPI Profile conformant xAPI statements in several hours on a laptop. Such an ability to create large machine readable data sets either on commodity hardware or as processed at even larger scale through distributed computing clusters is vital in the effort to generate data

for the purpose of acquiring the necessary and relevant data sets needed for machine learning and the training of AI in the learning and training domain.
- Data Fit for Purpose – Because DATASIM generates simulated learner data, projects may be undertaken without the need for human subjects, thus reducing the potential of exposure of sensitive learner information. Further, the data itself may be modeled for the purpose of testing the data design for any sign of implicit bias.
- Design for Compliance – DATASIM adheres to the strict protocols of the xAPI specification (soon to be IEEE 9274.1.1) and the xAPI Profiles specification (soon to be P9274.2.1). It is the first full compliant and conformant Big Data engine built expressly for the testing and evaluation of TLA implementations. It follows protocol and best practices with regard to the information management lifecycle and as an authenticated and secure system will support DoD's requirements for secure data and data management.

DATASIM and the 4 Essential Capabilities necessary to enable all DoD strategic data goals

- Architecture – DATASIM ensures the integrity and value of data that will flow through and support the Total Learning Architecture. Because DATASIM can produce massive ad hoc testing sets of meaningful xAPI data, it can allow DoD instructional designers, analysts, and IT support to move faster than ever before on the testing and evaluation of data design and architectural support for technology in the enterprise learning and training domain.
- Standards – DATASIM both strictly adheres to DoD specifications and DoD-supported standards and upholds, validates, and provides technical advocacy for their proper usage in enterprise environments. By automating the process of validating that the design of xAPI data and xAPI Profiles are correct, DATASIM guides users into properly implementing DoD specifications and standards.
- Governance – As a testing and evaluation tool, DATASIM can be used to ensure that DoD data design follows DoD data governance regarding principles, policies, processes, frameworks, tools, metrics, and oversight required to effectively manage data at all levels and anywhere within the learning and training enterprise.
- Talent and Culture – DATASIM can act as a guide to instructional designers and analysts. The application will help them to better understand xAPI data, will support them in the design of more relevant and value-add metrics, and will increase the assuredness they have in working with Big Data in the DoD learning and training enterprise.

DATASIM and the 7 Goals that must be achieved to become a data-centric DoD:

Filling the role of data and design validator and evaluator, DATASIM ensures that the DoD learning and training enterprise is not only data-centric but data-positive. Meaning: because DATASIM helps to ensure the creation of clean and usable data within the ecosystem, users of that ecosystem will have a more positive experience using and leveraging data. Thus the cultural effect will be such as to encourage the development of a data-centric DoD because at the practical and day-today level, people will understand and will depend on data and the systems that are hydrated by data.

- Make Data Visible – DATASIM makes all aspects of xAPI data and xAPI Profiles visible to the end user.
- Make Data Accessible – DATASIM makes all aspects of xAPI data and xAPI Profiles accessible to end users and machine consumers including artificial intelligence.
- Make Data Understandable – DATASIM clearly guides human users in the context, meaning, and proper usage of all xAPI data attributes as well as the machine readable pattern, sequence, and concept capabilities of xAPI Profiles that will drive the ability to leverage machine learning and AI in the next generation of learning and training technologies.
- Make Data Linked – By relying on xAPI Profiles and the Linked Data underpinnings of this DoD specification, DATASIM encourages a view of data as representative of a linked world of information cutting across geography, domain, and business or mission purpose.
- Make Data Trustworthy – DATASIM was built for the express purpose of generating standards-aligned data sets so finely tuned that they could be used to test and validate the properties of the entire learning and training data lifecycle.
- Make Data Interoperable – DATASIM both ensures data interoperability at the machine-to-machine level, but also compels human users to work with data in ways that expressly foster and promote interoperability. Data designed via DATASIM is data that is inherently ensured to be interoperable with xAPI and related technologies implemented across the TLA.
- Make Data Secure – DATASIM produces auditable and immutable xAPI data.

7.1 Near Term Tasks and Objectives within Scheduled OY2

POP (Months 1 - 6)

Tasks

It is recommended that the following tasks are accomplished under the currently planned DATASIM Option Year 2. By front loading the development and implementation of these key features into the current schedule, an open source hosted version of DATASIM could get into the hands of DoD and Federal Government stakeholders by July 2021. Usability, testing, and related reports will be accomplished as scheduled.

Key Features to be developed:

Months 1 - 3

- Integration with the ADL xAPI Profile Server
- User Accounts, Authentication, Logging and Accountability
- Hosted deployment

Months 3-6

- Phase I Development of DATASIM's Advanced Temporal Modeling functionality (ATMOD)
    - This feature is the last core code element in the baseline DATASIM product. The feature will allow the user to model how activities occur over time. Phase I deliverable is to provide the capability that:
        - A user will be able to accurately model a global workforce with training operations across time zones.

Outcomes Result

TRL 6 finished **24/7 hosted DATASIM prototype solution** that links to the xAPI Profile Server to allow DoD stakeholders

- to test and evaluate different xAPI Profiles
    - including automated xAPI Profile validation
    - including automated validation of xAPI statements to xAPI Profiles
    - including granular error messaging to advance quick debugging
- to generate their own xAPI data sets based on those xAPI Profiles and store them in their own LRS

Key Performance Indicators & Measures

- The hosted solution will be accessible by DoD and Federal stakeholders

- DATASIM will be able to provide validation and error messaging regarding any xAPI Profile and will be put to the test reviewing all xAPI Profiles available through ADL's Github repo
- The solution will produce a large scale dataset of at least 100,000,000 xAPI statements for ADL test usage
- The solution will generate relevant data for testing of the ADL ELRR based on the Master Object Model (MOM) xAPI Profile

Value Proposition

DoD stakeholders will be able to test and evaluate xAPI design and xAPI Profiles quickly within a browser.

7.2 Mid Term Tasks and Objectives within Scheduled OY2

POP (Months 6-12)

Tasks

It is recommended that the following tasks are accomplished under the currently planned DATASIM Option Year 2. Usability, testing, and related reports will be accomplished as scheduled.

Key Features to be developed

- Improve UX/UI Workflow based on Stakeholder Feedback
    - Improvements will be made to the authoring workflow in response to stakeholder usage in the STEEL-R project (as appropriate), ELRR development, and in ADL usage (especially as regards xAPI Profile validation via integration with the ADL Profile Server)
        - The goal of the workflow improvements is to advance DATASIM to being an application that can be rapidly integrated into an authoring and testing workflow that will extend the functionality and capabilities of the ADL Profile Server without causing bottlenecks caused by current-state limitations to xAPI Profile validation and debugging
- Cybersecurity (time frame depends on the requirements and sponsorship / support of the government, therefore it may be necessary to begin the process earlier in the year though it is expected to take several months to complete.)
    - STIG / RMF review

    - Requirements fulfilled to attain ATO
    - Scanning and acceptance of software as fulfilling security requirements
- Phase II development of ATMOD
    - Note that ATMOD functionality
        - ensures the testability of globally distributed TLA implementations
        - increases the fidelity of simulations by including time-specific behaviors to scenarios
    - Phase II deliverable is to provide the capability that:
        - A user will be able to more accurately model informal learning and asynchronous learning workflows which may occur in a real-world context.
        - A user will be able to set time limits on certain events to reflect, for example, the length of courses in a curriculum

Outcomes Result

DATASIM will have completed key functionality requirements and have passed the necessary security requirements and **will be able to be transitioned** and implemented across the DoD.

Key Performance Indicators & Measures

- Successfully identify and iterate to meet the design needs of stakeholder workflow
- Successfully complete security requirements (as possible by attaining an ATO)
- Successfully produce time-responsive simulations at a scale of at least 100,000,000 xAPI statements

Value Proposition

This will make context-faithful xAPI generation and the testing, evaluation, and implementation of key TLA data resources accessible for the whole of government.

7.3 Long Term Tasks and Objectives beyond Scheduled OY2

POP (depends on objectives… a baseline operational capability would be deployable immediately.)

Task

The purpose of this cycle is to provide the Initial Operational Capability to the DoD.

- Operational deployment of DATASIM for use by enterprise DoD stakeholders.

Outcomes Result

24/7 **hosted DATASIM operational solution** that links to the xAPI Profile Server to allow DoD stakeholders

Key Performance Indicators & Measures

- Evaluation of stakeholder response and usage of the platform
- Operational implementation of DATASIM within the DoD enterprise environment

Value Proposition

As stated, DATASIM provides the core modeling and data generation capabilities necessary

- to ensure conformity of learning and training data to DoD and relevant technical standards
- to facilitate interoperability across DoD and other government systems
- to increase readiness by providing an agile and in-the-workflow testing and evaluation capability necessary to ensure the ongoing quality of DoD instructional design, data design, and IT infrastructure at the level of the TLA.

DoD stakeholders will have the ability launch or access an operational secure enterprise DATASIM to test and evaluate

- xAPI data design and instructional design implementing xAPI
- TLA interoperability (potentially in line with IEEE ICAP conformance assessment standards for the 9274 suite of xAPI technologies)
- CaSS and competency assertion mechanism accuracy when basing decisions on xAPI data that has passed through the TLA and throughout control loops of training, readiness, and workforce development
- Master Object Model and xAPI Profile design and implementation
- conformance of services with the ELRR

Note that a longer term integration of DATASIM and the xAPI LXP currently under development and sponsored by the Air Force Research Laboratory is highly recommended. This would provide users the ability to test real-world ad hoc xAPI-enabled learning engagement within a Master Object Model controlled xAPI Learning Record Provider. **The linkage between DATASIM and Learning Record Providers will be the foundation of a conformance test**

suite for xAPI data producers. The LXP integration is a logical first step as it is the first dual-use commercial/defense technology built on ADL's Master Object Model.

There are a number of other features that have been recommended including automated xAPI Profile generation and visual improvements which would go beyond the core functionality described here. The purpose of this document has been to describe only what needs to be done to get a hosted DATASIM solution deployed and implemented for DoD stakeholders to use effectively — and to do so in as rapid a fashion as possible.

To sum it up, the prototype can be hosted and available within 90 days. The security and functional requirements would take another 9 months. A fully operational platform modeled on commercial dual-use could be deployed immediately thereafter and further features could be developed on an Agile iteration schedule to meet ongoing need and demand.

8. Appendix

8.1 Yet Calibration Profile

The Yet Calibration Profile, used for simplified simulation behavior analysis, can be found here: https://github.com/yetanalytics/datasim/blob/master/dev-resources/bench/calibration.jsonld

8.2 Master Object Model Profile

The conformant version of the Master Object Model (MOM) Profile used for generation in this effort can be found here:
https://adlnet.gov/resources/publications/2020/04/2019-Total-Learning-Architecture-Report/

8.3 DATASIM Code Repositories and Documentation

Backend <https://github.com/yetanalytics/datasim>
Frontend <https://github.com/yetanalytics/datasim-ui>