

# ML2++ Tutorial

July 2025

## 1 Introduction

*This section introduces the ML2++ framework, outlining its capabilities for integrating machine learning and time series forecasting into IoT system modeling.*

ML2++ extends the ThingML modeling language to seamlessly integrate time series forecasting and machine learning into IoT system design. This tutorial will show you how to use the system.

## 2 Using ML++ Graphical DSL

### 2.1 Uploading the Sirius Web Abstract Syntax

#### Goal

Download the abstract syntax for a Sirius Web grammar and upload it into Sirius Web so you can use the abstract syntax to define ML++ instance models.

#### Prerequisites

- The abstract syntax packaged as a .zip.
- Access to Sirius Web and the Home / Project overview page (localhost:8083).

#### Steps

##### 1. Download the abstract syntax

Download the Sirius Web abstract syntax package to your computer as a .zip file (this is the file you will upload).

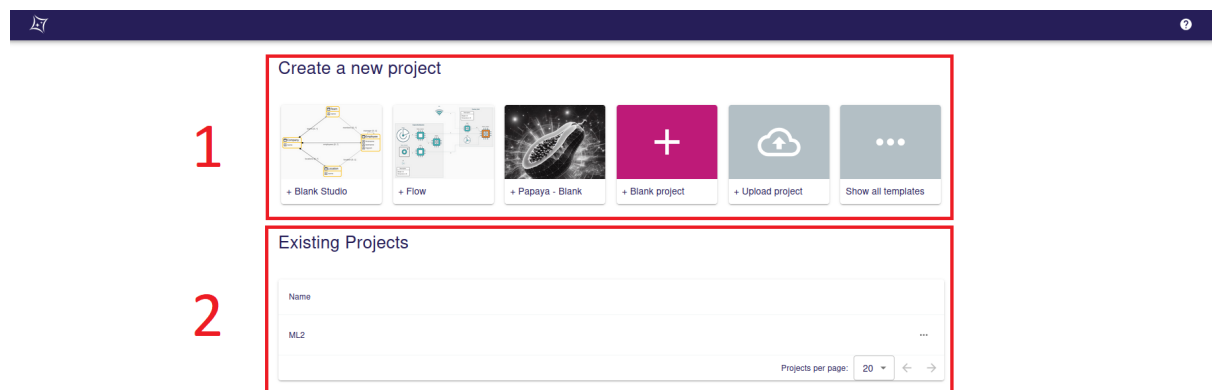


Figure 1: Sirius Web's Homepage

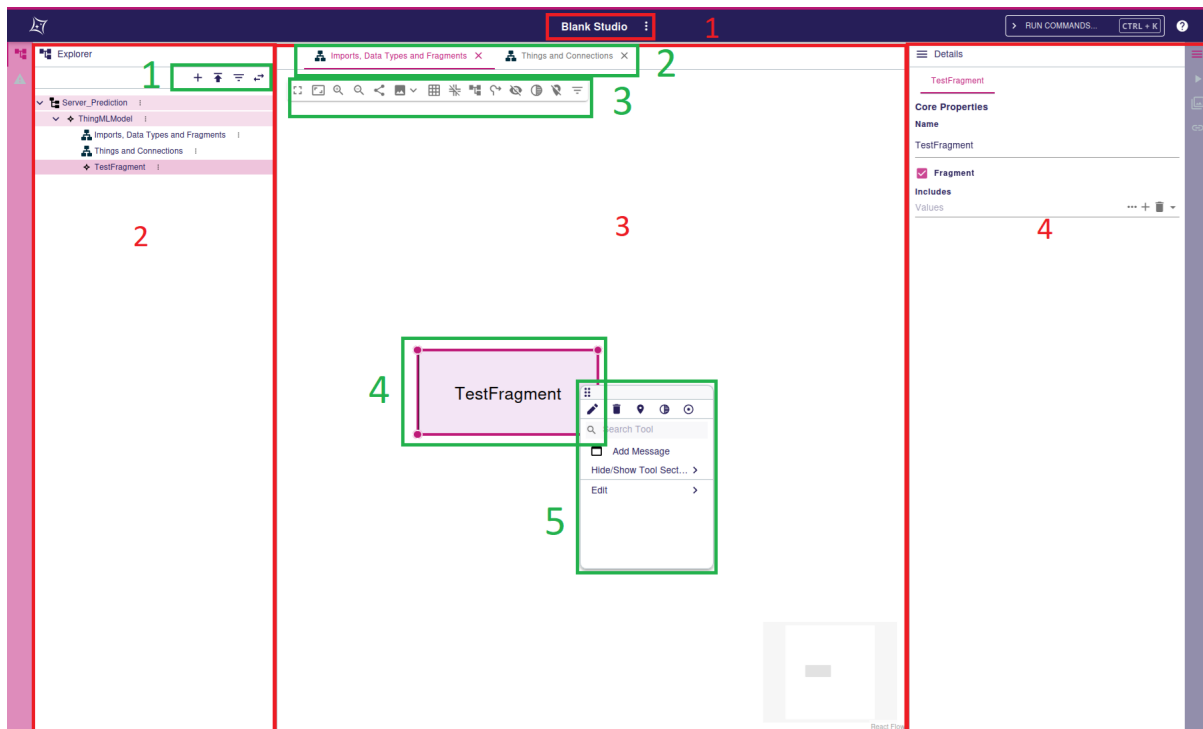


Figure 2: Sirius Web's Model Editor

## 2. Open Sirius Web and go to the Home / Project overview page

You should see the project overview page 1. Note the two important areas in that figure:

- **Area 1 (red)** — where you create new projects / upload projects.
- **Area 2 (red)** — the list/overview of projects already in the system (e.g., the ML2 grammar shown).

## 3. Start the upload wizard

In **Area 1**, click *Upload Project*. A small wizard / dialog will pop up.

## 4. Select the .zip project file

In the wizard either *drag & drop* your .zip file into the dialog or click the file selector and choose the file you downloaded.

## 5. Confirm and upload

Complete the wizard prompts (usually an *Upload* or *Confirm* button). Wait for the upload to finish.

## 6. Verify the project appears

After the upload completes, check **Area 2** on the Project overview page, the grammar should now be uploaded (like the ML2 grammar in the figure).

## 2.2 Creating a model instance using ML++

### Goal

Open the Sirius Web model editor, create a domain model and its root, then build and edit model instances using the tree (model explorer), graphical views and form views.

## Prerequisites

- A Sirius Web project already uploaded to the system.
- The ML-Quadrat domain available in the project.
- Access to Sirius Web and the project home.

## Interface overview (red colours)

- (1) **Project title & menu:** shows the project name; the “three dots” menu lets you download the project or edit the title.
- (2) **Model explorer (tree):** hierarchical view of the model instance. You can create and manipulate elements here; required when first creating a new model.
- (3) **Graphical canvas (visualization):** main drawing area. Each *view* provides its own context-dependent commands and appearance.
- (4) **Form view:** shows and edits attributes of the currently selected element (from the tree or canvas).

Additional (green) menus:

1. Action menu: create or upload models.
2. Views/tabs menu: open/select active graphical representations.
3. Graphical manipulation menu: zoom, arrange, show/hide elements, grid, etc.
4. Object instance (example visual element): shows how objects appear in the canvas and in the tree.
5. Context pop-up wizard: appears on click/right-click; provides context-sensitive commands (different when clicking canvas vs an object).

## Workflow: create and populate a model

1. **Create a new project workspace:** On the Sirius Web homepage 1 select *Blank Studio* to open the model editor for a new project.
2. **Open the model explorer (tree):** confirm you can see the tree (Area 2). The first model creation steps use this tree.
3. **Create a new model (Domain) (Fig. 3):**
  - (a) Click the + button inside the tree-view.
  - (b) In the pop-up wizard give the model a name.
  - (c) Choose model type **Domain** (not “View”) — *Domain* is required to create instance models.
4. **Create the root object (ThingMLModel) (Fig. 4):**
  - (a) Select the domain node in the tree.
  - (b) Click the domain node’s “three dots” menu and choose the option to create/add a root object.

- (c) In the wizard choose domain = **ML-Quadrat** and object type = **ThingMLModel**.  
*Important:* **ThingMLModel** is the root of all domains; subsequent steps require this selection.
5. **Choose how you will develop (two options):**
    - **(A) Tree + Form view (structured editing):** use the tree to add objects and the form view to edit values.
    - **(B) Graphical views + Form view (visual editing):** create graphical representations and edit via canvas and forms.
  6. **Add objects via the tree (option A):**
    - (a) In the tree, click the domain or parent object, open its “three dots” menu and choose **Add Object**.
    - (b) Select the desired object type from the context-dependent drop-down list.
    - (c) The new object appears in the tree and gets default attribute values; edit them in the form view.
  7. **Add a graphical representation (option B) (Fig. 5):**
    - (a) In the views/tabs area select **New representation**.
    - (b) In the wizard select a graphical view from the list and optionally rename it. Recommended first view: **ImportsDataTypesAndFragments**.
    - (c) After creation, open that view tab to see the canvas (may be empty or pre-populated depending on the view).
  8. **Use the canvas context menu (Fig. 6):**
    - Right-click the empty canvas or an object to open the context pop-up wizard (menus and sub-menus).
    - Menus are context-dependent: commands differ when the canvas is selected vs when an object is selected.
  9. **Create graphical objects and relationships (Fig. 7):**
    - Choose a *create* action from the canvas/object context menu to place a graphical object.
    - Click a created object to get object-specific actions.
    - Creating a relationship between objects creates a visual link between them .
  10. **Edit default values via the form view (Fig. 8):**
    - Newly created objects get default values; select the object and edit attributes in the form view.
    - Example: a **Port** shows name (string) and relationships to messages it sends/receives; a **Property** shows a name string only.
  11. **When finished, download the Sirius Web instance model (Fig. 10):**
    - Click on the three dots of the project root
    - Select download.

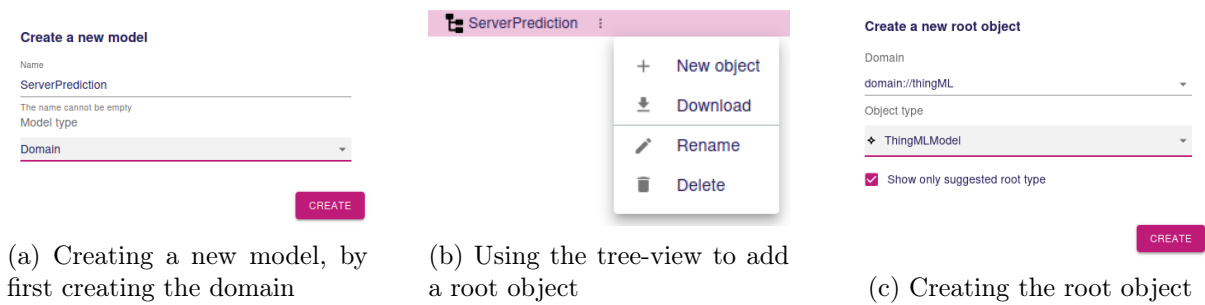


Figure 3: The first few steps after creating a new project.

## Special forms: DAML and machine-learning models

- DAML (data analytics & machine learning) elements and ML model elements provide richer, multi-page form representations. (Fig 9)
- Pages: each page contains editable values grouped by section; this keeps the form focused and readable.
- **Purple action buttons:** many DAML fields use purple buttons to *set* attribute values (pressing a button fills/replaces the attribute value). A **NOT SET** button unsets the attribute.
- **ML model selection:** the DAML form’s last page lists available ML models as purple buttons (e.g., **LINEAR REGRESSION**). Clicking a model button adds that model to the DAML–algorithm relationship. Selecting a different model replaces the previous model.
- Each ML model has its own separate form representation you can edit after adding it.

## Tips & troubleshooting

- If commands are missing, check what is selected (canvas vs object) and which *view* is active, many menus are view- and context-dependent.
- Use the project title “three dots” to download or export the project periodically.
- When a graphical view appears empty, confirm whether that view should be pre-populated (some views aggregate objects from other views).

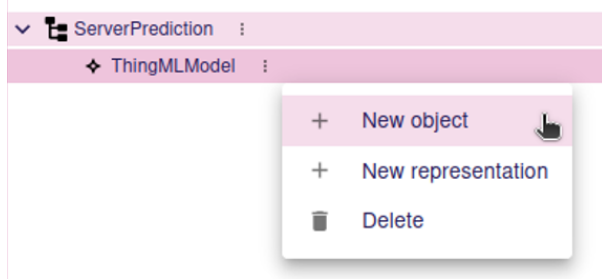
## 2.3 Using the Orchestrator

### Goal

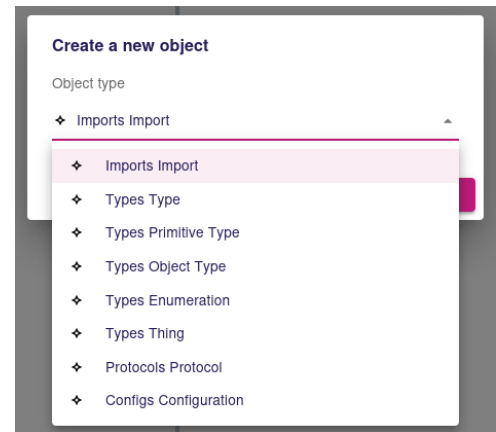
Access the Orchestrator, manage Sirius Web instance models, generate projects, and execute ML-Quadrat workflows.

### Prerequisites

- Login credentials (test,1234)
- One or more Sirius Web instance model files (.xml).
- (Optional) Dataset files for the project.

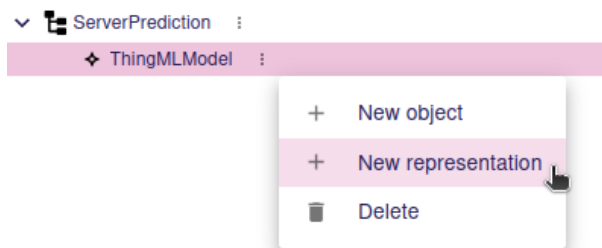


(a) Selecting "New Object" to prompt the wizard

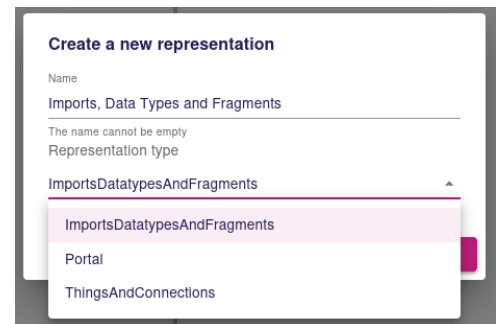


(b) Use the drop-down menu to select the appropriate object to be added

Figure 4: Adding a new object using the tree view

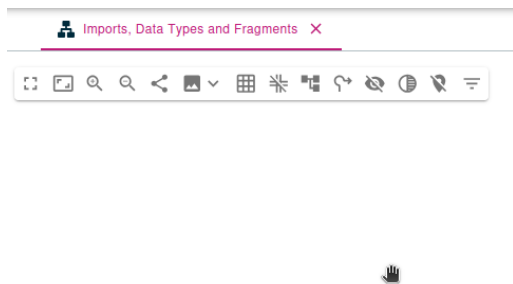


(a) Selecting "New representation" to prompt the wizard

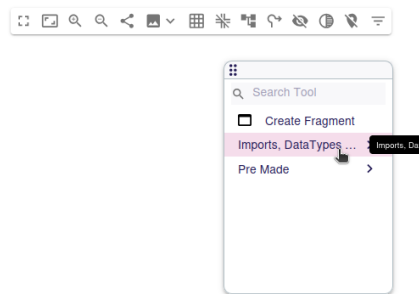


(b) Use the drop-down menu to select the appropriate view to be added and change the title of the view

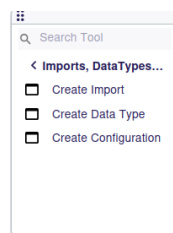
Figure 5: Adding a graphical representation using the tree view



(a) The empty canvas



(b) Menu to manipulate the model after right-clicking



(c) Sub-menu for different actions

Figure 6: Selecting elements and their menus

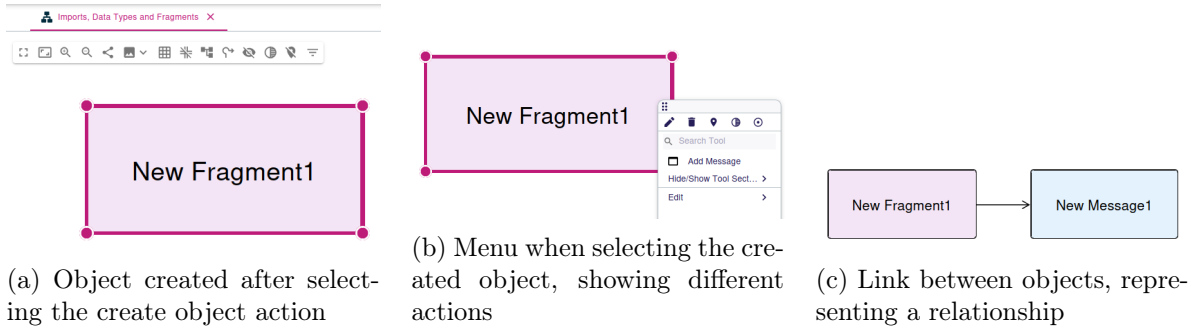


Figure 7: Adding a new object using the graphical view

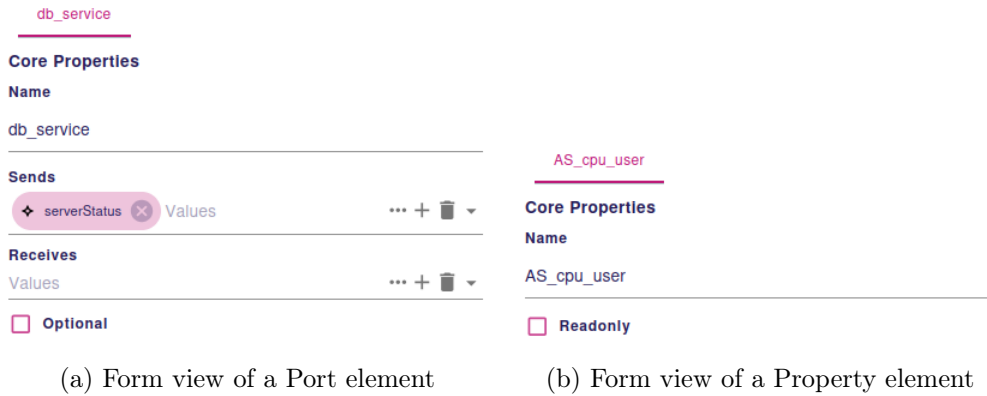


Figure 8: Overview of two elements having different editable values in the form view

## Workflow

### 1. Login to the Orchestrator

- Navigate to the Orchestrator URL. (localhost:8081)
- Enter your username and password in the login form.
- Click **Login** to proceed.

### 2. Project overview page (after login) (Fig. 11)

- At the top section (**Area 1**), you can upload new Sirius Web instance model files.
- Below this section is the list of all projects you have uploaded. For each project, the following information/actions are available:
  - File name of the uploaded instance model.
  - Upload date.
  - Inspect** button — opens the project overview page.
  - Re-upload** option — replace with a newer instance model.
  - Delete** option — remove the project.

### 3. Choose how to proceed

- Option A:** Upload a new Sirius Web instance model (Area 1) and then select **Inspect**.
- Option B:** Select **Inspect** on an already existing project from the list.

### 4. Inside the project overview page (Fig. 12)

da Form X

Data Preprocessing Time Series Model Evaluation Visualisation Model Algorithm

Dataset  
data/appserver01\_data\_string.csv

Features

DA\_cpu\_user DA\_cpu\_system DA\_cpu\_wait DA\_disk\_read DA\_disk\_write DA\_mem\_used  
DA\_swap\_used DA\_net\_eth0\_rx DA\_net\_eth0\_tx DA\_net\_eth1\_rx DA\_net\_eth1\_tx DA\_sch\_threads  
DA\_app\_status Values

Input Features  
Values

Output Features  
Values

Labels ON  
NOT\_SET OFF ON SEMI

Timestamps OFF  
NOT\_SET OFF ON

Common Period Threshold  
0

(a) A page of the data analytics form

da Form X

Data Preprocessing Time Series Model Evaluation Visualisation Model Algorithm REMOVE ALGORITHM

Supervised

LINEAR CLASSIFIER LOGISTIC REGRESSION  
LINEAR REGRESSION  
NAIVE BAYES GAUSSIAN  
NAIVE BAYES MULTINOMIAL  
NAIVE BAYES COMPLEMENT  
NAIVE BAYES BERNOULLI  
NAIVE BAYES CATEGORICAL  
DECISION TREE REGRESSOR  
DECISION TREE CLASSIFIER  
RANDOM FOREST REGRESSOR  
MULTILAYER PERCEPTRON

Unsupervised

K MEANS MINI BATCH K MEANS DBSCAN SPECTRAL CLUSTERING GAUSSIAN MIXTURE

Semi Supervised

SELF TRAINING CLASSIFIER LABEL PROPAGATION LABEL SPREADING

Time Series

Deep Learning

MLP GRU CNN LSTM RNN TCN TRANSFORMER

Statistical

ARIMA SARIMA HWES ETS STATE SPACE MODEL

Machine Learning

SVR RFR GBM XGBOOST

Hybrid

ARIMA GARCH PROPHET

(b) Model Algorithm Section

Figure 9: Data Analytics form representation

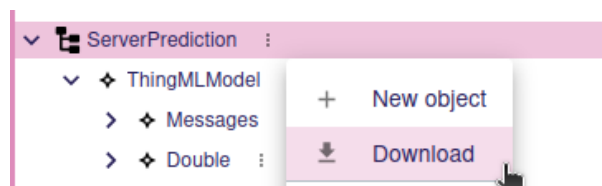


Figure 10: Downloading the instance model



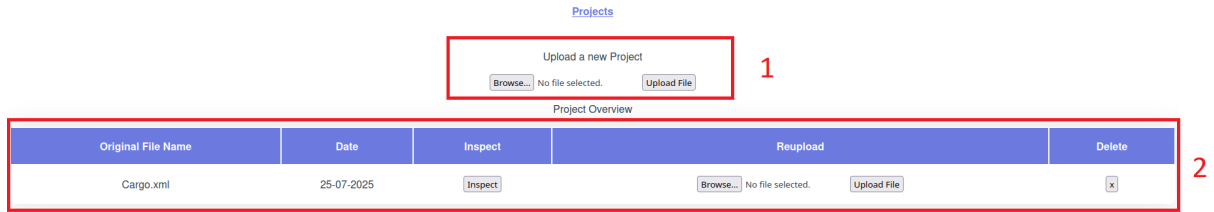


Figure 11: Caption

- (a) The page is divided into two major sections.
- (b) **Section 1 — Project commands and file management:**
- **Generate project** — compiles the converted ML2 file into a project.
  - **Delete project** — removes the project from the orchestrator.
  - **Delete dataset** — removes the dataset associated with the project.
  - **Update model instance** — upload a new Sirius Web model instance to update the project.
  - **Upload dataset** — upload a dataset to be used by the project.
  - **Generated files list:** shows file type, name, and status (successful or not).
  - Download individual files by clicking on their names.
- (c) **Section 2 — Project execution:**
- **Execute project** — runs the ML-Quadrat project (if it exists).
  - Execution output is shown in the **Generated Output Section** (collapsible).
  - Generated images are shown in the **Generated Images Section** (collapsible).
  - Generated report is shown in the **Generated Report Section** (collapsible).
  - Three download buttons are available: for images, project output and report.

## 5. Further development options

- **Offline development:** Download converted files from the generated files list and continue work locally.
- **Iterative cycle:** Edit the Sirius Web instance model, then update the model in the Orchestrator via re-upload.

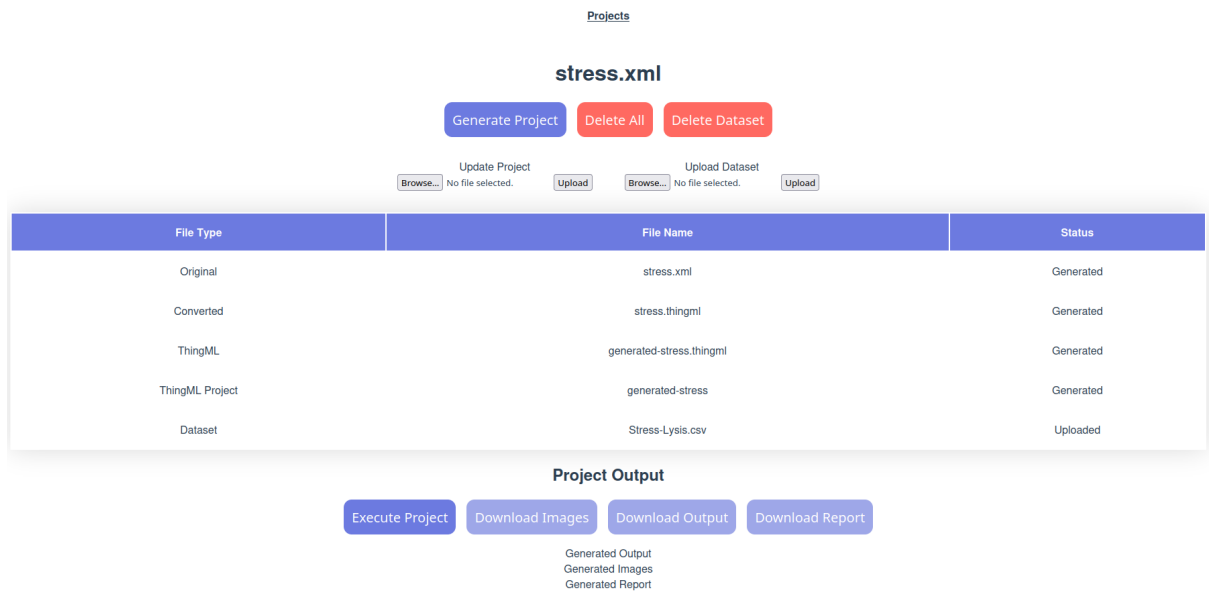


Figure 12: Caption

### 3 Quick Overview of All Available Views in Sirius Web

View Name	Type	Element	Description
ImportsDatatypesAndFragments	Graphical	Project Root	Define datatypes, objects and fragments
ThingsAndConnections	Graphical	Project Root	Define Things including their properties, ports, behaviour and DAML
Behaviour	Graphical	Thing Behaviour	Statechart flow diagram of a Thing's behaviour
Configuration	Graphical	Configuration	Define object instances of Things and connect them
DA Form View	Textual	DAML	Configure your DAML
Model Form View	Textual	Any ML model	Configure your ML

Table 1: Example of a four-column table

## 4 Common Actions for Behaviour Programming

Within action blocks, users can define control flows, variable assignments and call data analytics functions, here's a list of frequent actions: *Variable Assignment*, note that the value needs to be the same type as the variable

```
1 variable = value
```

### *If else*

```
1  \\V1 for multiple actions (preferred method)
2  if(condition) do
3      ...
4  end
5  else do
6      ...
7  end
8  \\v2 for singular actions
9  if(condition)
10     ...
11 else do
12     ...
13 end
14
15 Condition: simple comparison operations e.g. >, <, ==
```

### *Sending messages*

```
1 port_name!message_name(parameters)
```

### *DA Functions*

```
1 da_save DAML_name
2 da_preprocess DAML_name
3 da_train DAML_name
4 da_predict DAML_name
```