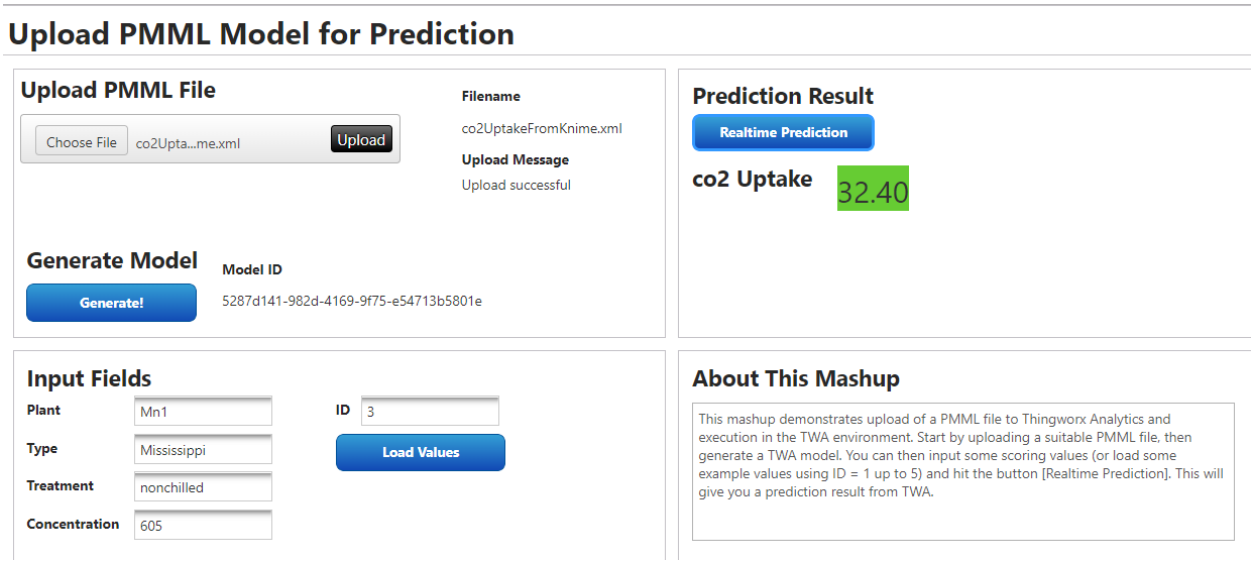
Upload PMML to Thingworx Analytics

Version 1.1, 11th February 2019

Quick Summary

This tutorial demonstrates how to upload a PMML prediction model from an external system into Thingworx Analytics, and then score data using that model. A TWX mashup is used to guide you through the process, see below:



Prerequisites

In order to use this tutorial you will need the following skills:

- Familiarity with Thingworx development, including some scripting
- Some familiarity with Thingworx Analytics, e.g. building a prediction model

You need an environment with Thingworx Foundation + Thingworx Analytics server installed, version 8.3 or later. You do not need Descriptive Analytics / Property Transforms to be installed.

Furthermore you will require the following files:

Upload-PMML.zip

About the Model

In this tutorial we are using a prediction model generated in Knime, that is, completely externally to Thingworx Analytics. The prediction model takes the following fields as input:

Plant (String, Categorical)

Type (String, Categorical)

Treatment (String, Categorical)

conc (Double, Continuous)

It then predicts the following output:

uptake (Double, Continuous)

The inputs represent different types of plants with different concentrations of CO2 (carbon dioxide) in the air. The output represents the predicted uptake of CO2 by the plant. The dataset used as a learnset for this model was co2.csv

Entity Import

To start you need to upload entities from co2Uptake.twx (or co2Uptake.xml) into Thingworx. You can do this from the menu options at the top of Composer, selecting Import with the following options:

Import Option = From File

Import Type = Entity

Use Default Persistence Provider = false

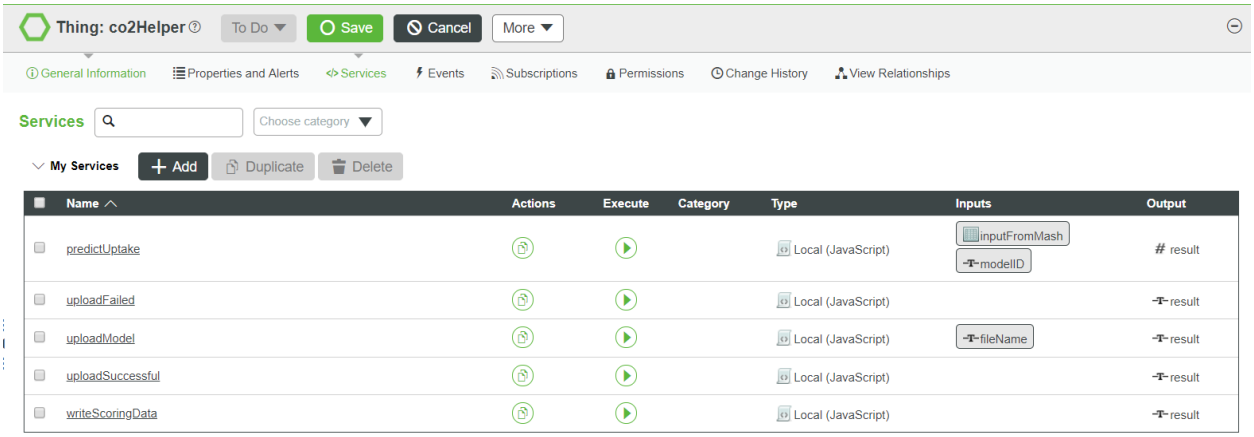
Include Subsystems = false

Import Source = Single File

File Name > Browse to your file co2Uptake.twx (or co2Uptake.xml)

Generate Data Table Entries

In order to generate some records which can be used for scoring from the mashup, open Thing co2Helper and execute service writeScoringData as shown below:



You do not need to specify any inputs. A number (e.g. 33245) is returned as an output, confirming that the service executed successfully.

Load the Mashup

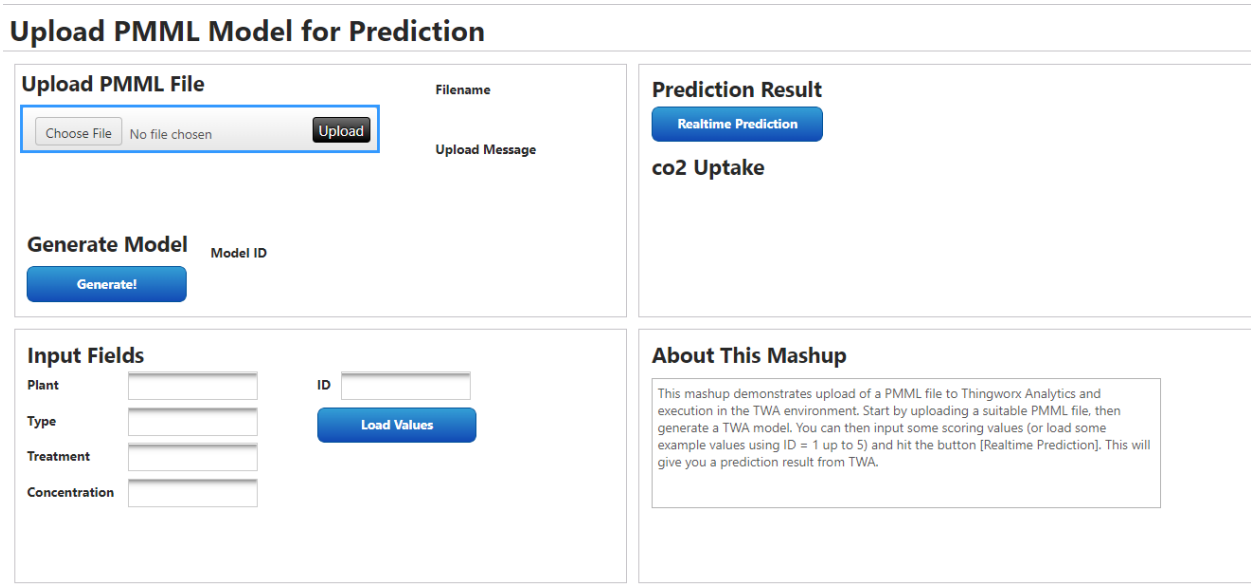
To use the mashup, navigate to the following relative path:

/Thingworx/Mashups/co2UptakePrediction

For example in my system the path is similar to this:

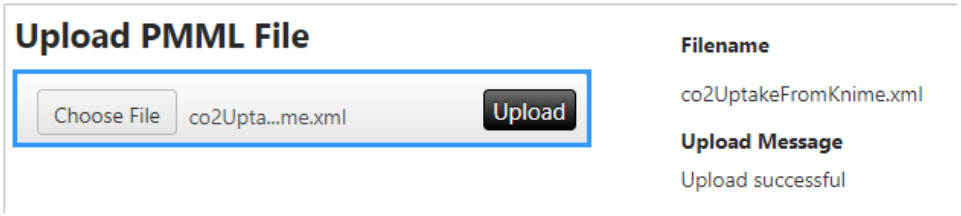
http://SERVERNAME/Thingworx/Mashups/co2UptakePrediction

You should now see the initial state of the mashup as shown below:

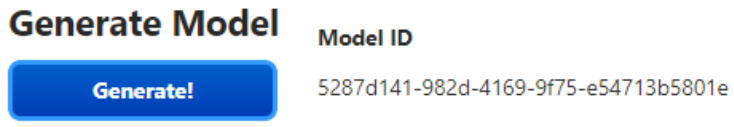


Upload PMML and Generate Model

Now you can select the PMML file (co2UptakeFromKnime.xml) and click Upload. You should see the Filename displayed on the mashup, and also the Upload Message should state “Upload successful”.

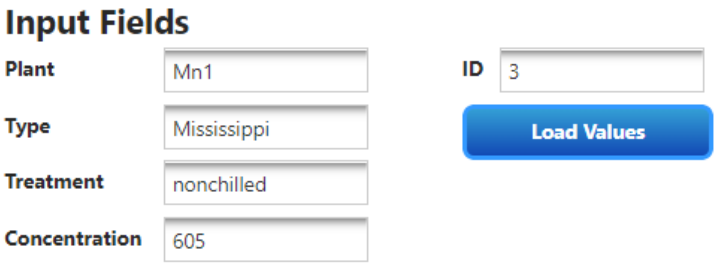


Under “Generate Model” you can now click “Generate!” and you should see a Model ID displayed on the screen.

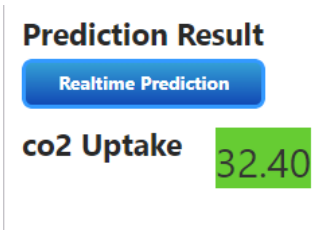


Set Inputs and Get a Prediction Result

To trigger a prediction you now need to input some values into fields Plant, Type, Treatment and Concentration. An easy way to do this is to input an integer between 1 and 5 into the ID field and click Load Values. This loads a saved scoring record from a data table. You are then free to manually change those inputs in the mashup if you wish.



Now click on “Realtime Prediction” under Prediction Result and you should see a value display on the screen. This is the predicted carbon dioxide uptake of the plants given the input conditions you specified.



Conclusion

This tutorial has demonstrated the ability to upload a PMML model generated in an external system, and execute it in Thingworx Analytics.

The Service Scripts

The following service is of particular interest to understand the functionality in TWA and perhaps adapt it to your own needs:

Things[“co2Helper”].uploadModel

In the service, the call Things[ms.name].UploadModel is used to generate the prediction model in TWA from a PMML file which can then be used for scoring.

Other services on Things[“co2Helper”] might also be of interest as they provide functionality to the mashup.

About Knime

According to the Knime website , *“Our KNIME Analytics Platform is the leading open solution for data-driven innovation, designed for discovering the potential hidden in data, mining for fresh insights, or predicting new futures.”*

A full discussion is beyond the scope of this tutorial, however this platform was used to generate a Simple Regression Tree Learner trained on co2.csv. The Knime workflow then exported the model using the PMML writer node. Please refer also to files co2Uptake.knwf (the Knime workflow) and Knime-Screenshot.PNG (a screenshot of the workflow).

Troubleshooting

Data Type in PMML

In case you encounter the following errors when trying to get a prediction:

ThingworxStorage\logs\ScriptLog.log TypeError: Cannot read property "rows" from undefined

Thingworx_Analytics_Server\data\logs\thingpredictor.log: java.lang.NumberFormatException: For input string: "212.5"

This can be due to the PMML model being generated from Knime with DataField conc having dataType=”integer”. DataField conc needs to be changed to have dataType=”double” as per the file PMML\co2UptakeFromKnime.xml provided with this tutorial.

Reference to PredictionThing

Error in the application log is
[message: Execution error in service script [predictUptake] :: TypeError: Cannot call method "RealtimeScore" of null (predictUptake#38)]
Resolution: This seems to be a problem with finding the prediction microservice in script predictUptake.
Try commenting out this line (with // prefix):
var predictiveScores = Things[ms.name].RealtimeScore({
and uncommenting this line:
var predictiveScores = Things["AnalyticsServer_PredictionThing"].RealtimeScore({

Acknowledgements

Thanks to Christophe Morfin and Laurent Germain for helping with this effort.