Role4All Use Case

# Introduction

This document presents Role4All through a simple example. ….

Contents

[Introduction 1](#_Toc437422001)

[Context 1](#_Toc437422002)

[Initialization 1](#_Toc437422003)

[The Excels meta-model in Smalltalk 2](#_Toc437422004)

# Context

For this example we selected two files, an Excels document gathering some systems consumption and a Pimca model exponent a system. Moreover we know the global consumption of the system: 2 750 mW/h.

The first source of information is an array with two columns (Name and Consumption) and four rows (PC, FPGA, I7 and ARM) :

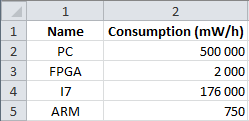


Figure 1: Excels file gathering some consumption

With this array we have a relation between some product name (PC, FPGA, …) and their consumptions.

The second source of information is a model created with Pimca.

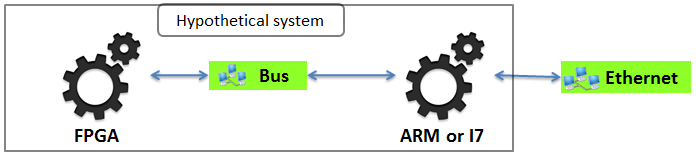


Figure 2: Pimca model of the hypothetical system

This model described a simple system including two elements, a FPGA and a processor. But we have an undetermined about the processor, it is an ARM or an I7 and we need to know which one. To solve our problem we simulate our system with an ARM and with an I7 and compare the consumption of the simulated system and of the real system. Therefore we need to create a link between our model and our Excels array that create a link between a Pimca file and an Excels file.

Role4All is one solution to create that type of link and to run simulations required.

# Initialization

Before use Role4All we need to create some elements:

* The Excels meta-model in Smalltalk (simplify for this example).
* A conversion from Excel to Smalltalk.
* The Pimca meta-model in Smalltalk (simplify for this example).
* A conversion from Pimca to Smalltalk.
* Role models.

## The Excels meta-model in Smalltalk

Excel is a complex tool base one a complex meta-model therefore we defined here our own meta-model of Excels adapted for Smalltalk.

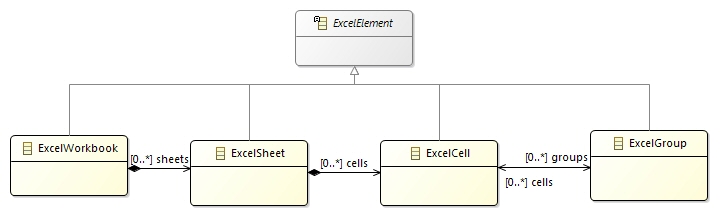


Figure 3: Simplify Excel's meta-model

Now with our Excel meta-model we can transform our Excel file to a Smalltalk model.

## Conversion from Excel to Smalltalk

To convert an Excel file to Smalltalk we use two model transformations, the first one between Excel and Json and the second one between Json and Smalltalk. In this document we will not develop this transformations we focus one the result of this transformation: the Smalltalk codes.



Figure 4: Model transformations between Excel and Smalltalk

The Smalltalk code present in the figure 4 was automatically generated from our Excels file. This code implement the Excel meta-model describe in the figure 3.

## The Pimca meta-model in Smalltalk

Pimca is a tool with lots of concepts, therefore for our example we created a Pimca meta-model with just the main concepts.

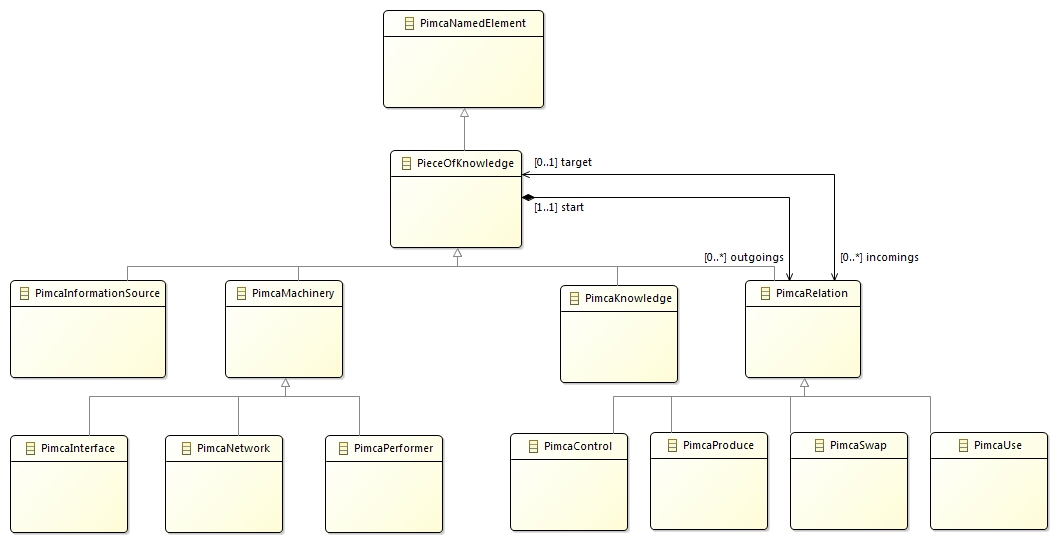


Figure 5: Simplify Pimca meta-model

Now with our Pimca meta-model we can transform our Pimca model to a Smalltalk model.

## Conversion from Pimca to Smalltalk

To convert an Pimca file to Smalltalk we use two model transformations, the first one between Pimca and Json and the second one between Json and Smalltalk. In this document we will not develop this transformations we focus one the result of this transformation: the Smalltalk codes.

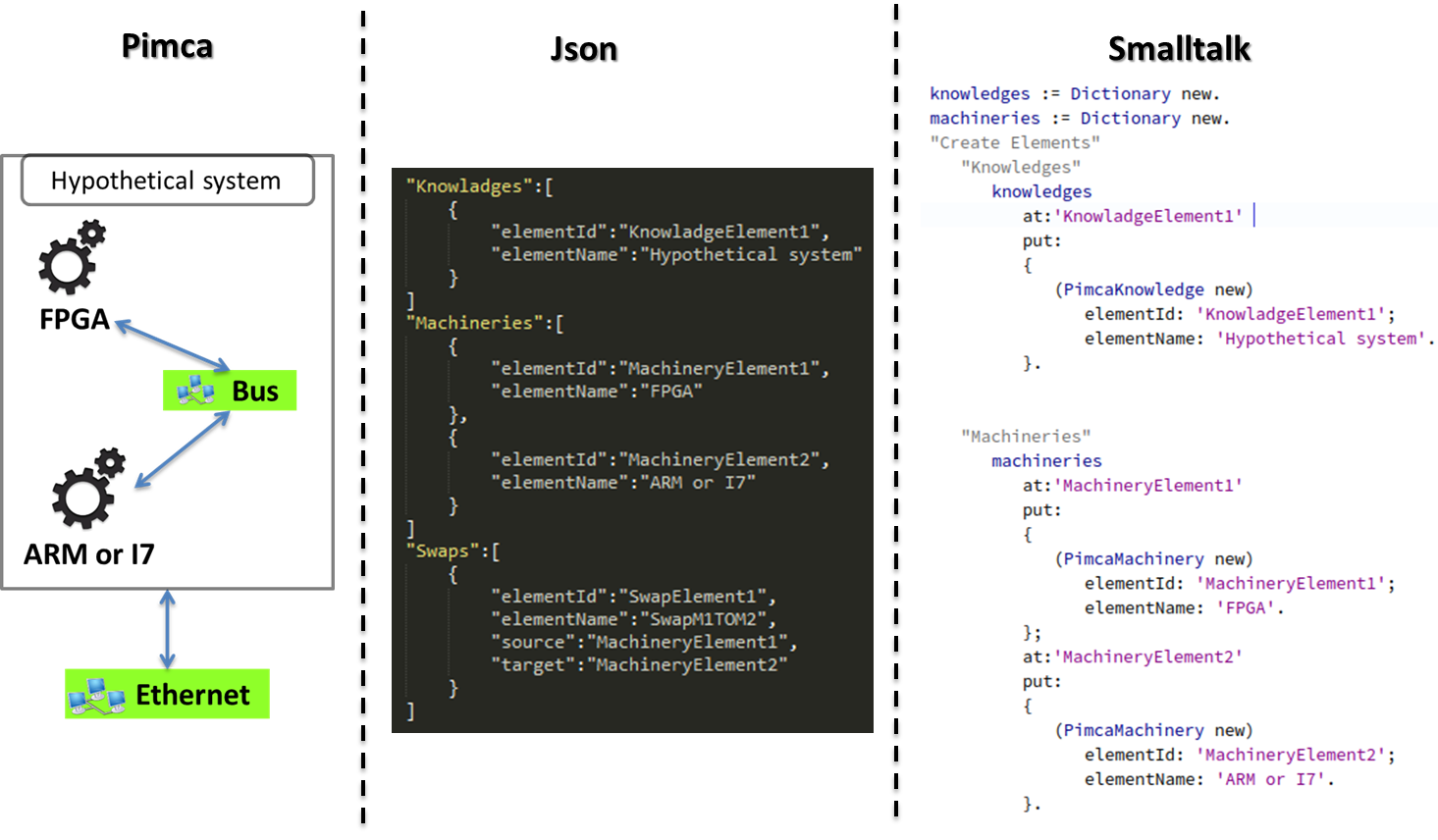


Figure 6: Model transformations between Excel and Smalltalk

The Smalltalk code present in the figure 6 was automatically generated from our Pimca model. This code implement the Pimca meta-model describe in the figure 5.

## Role models definition

A Role model includes 3 stapes:

* Role type definition
* Connection Role-Element
* Synchronization

### Role type definition

In our example we create 2 main Role types: RolePlatform and RoleCPU, they are subclass of the class Role defined in Role4All. Moreover we specify our main roles for each tools, with 2 tools (Pimca and Excel) we have 2 specifics roles per main roles. Finally we have 4 specific roles: ExcelRolePlatform, ExcelRoleCPU, PimcaRolePlatform and PimcaRoleCPU. The following figure explains the relation between main role, specific role and model element.

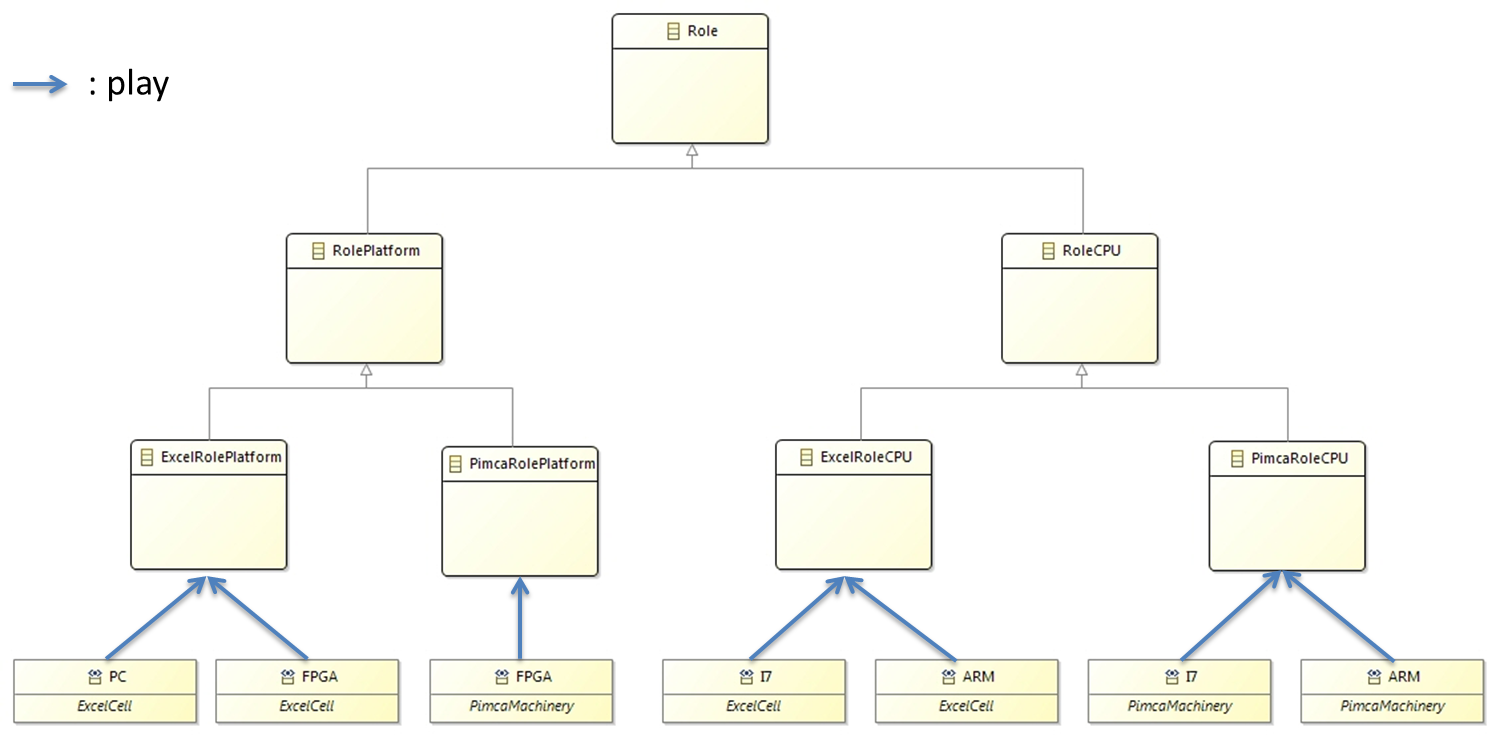


Figure 7: The relation between main role, specific role and model element

The link between our model elements and the specific role is a complex link call “play”. Model elements “play” specific roles. The action “play” creates a bidirectional reference between a model element and a role instance, this relation depend on an adaptor. We have 1 instance of specific roles per elements whose play these roles. For example the ExcelCells “I7” and “Arm” play the same specific role ExcelRoleCPU consequently we have two instance of ExcelRoleCPU: “excelRoleCPU0” and “excelRoleCPU1”.

### Connection Role-Element

The connection between a Role and a model element depend on a PlayRelation and an Adaptor. The adaptor is a class where we specify how to collect data in a model element and adapt them to for a role instance. We need to create an adaptor for each relation role-data. For example the PimcaMachinery with the id “I7” play the specific role PimcaRoleCPU therefore we have an adaptor PimcaAdaptorCPU between the role PimcaRoleCPU and the model element PimcaMachinery. The connection between Role, model element and adaptor made by a PlayRelation element is illustrate in the figure 8.

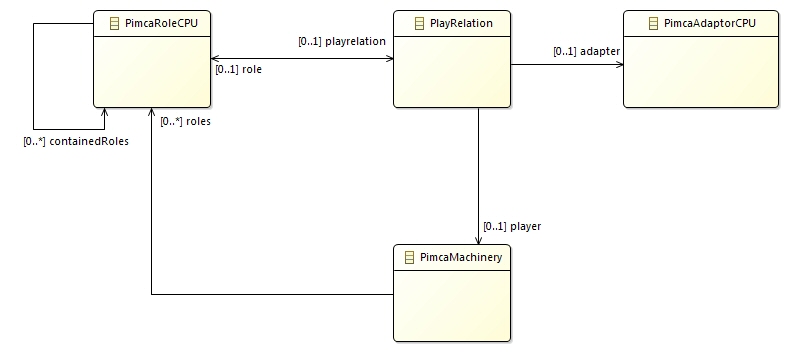


Figure 8: Relation between roles, model elements and adaptor

In the figure 8 we highlight that a Role has an instance variable call “containedRoles”, this is a list of role instance link with the current one. We use this list to synchronize some role instance in order to synchronize models element.

### Synchronization