Cyber Security Use Case

A cyber terrorist wants infected a system named CSUC with a homemade virus. CSUC is an object detection system composes of a camera, a FPGA and various processors/CPU. For our example we simplify the system to two elements: a platform (FPGA) and a processor (ARM) with an Ethernet connection. To do his mission the terrorist needs some information about the system (conception, consumption, etc.). To collect information he hacks the mailbox of a member of the project CSUC. The terrorist catch some important information: the global consumption of the system (2 750 mW/h) and a photo of the system. According to the photo, the terrorist detect that the system CSCU is compose of two elements, a platform and a processor. The terrorist limits his investigation to two platforms (Raspberry Pi and FPGA) and two processors (ARM and I7). Due to the worksheets of each product the terrorist can create an array to connect some product name (Raspberry Pi, FPGA, ARM, and I7) and their consumptions.

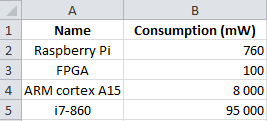


Figure 1: Excels file gathering some consumption

The second source of information (the photo) allows modeling a hypothetical system with Pimca.

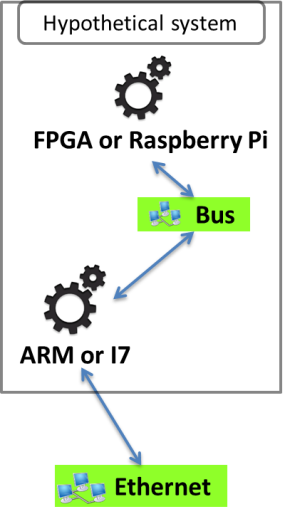


Figure 2: Pimca model of the hypothetical system

This model describes a simple system including two elements, a platform and a processor. But we have an undetermined about these elements, the processor is an ARM or an I7 and the platform is a FPGA or a Raspberry Pi. To solve our problem we simulate our systems (FPGA-ARM, FPGA-I7, etc.) and compare the consumption of the simulated systems and of the real system. Role4All is one solution to specify a hypothetical system and to run simulations required.

The cyber terrorist uses two tools (Pimca and Excel) and several concept common of these tools (FPGA, ARM, etc.). So he uses Role4All to federate his tools through concepts that he defined himself (FPGA, ARM, Raspberry Pi and I7). Each model element of each tool can play roles defined in a role model in Role4All (figure 3). The terrorist defines two main roles: *RolePlatform* and *RoleCPU,* who by definition of “role” extend the *Role* class defined in Role4All. In Role4All a role can play a role; this ability allows extending a role in order to adapt the role model without modifying the tool model or the role model. In our example the terrorist extends the role *RolePlatform* to two roles *RoleFPGA* and *RoleRaspberryPi* and the role *RoleCPU* to two roles *RoleARM* and *RoleI7*.

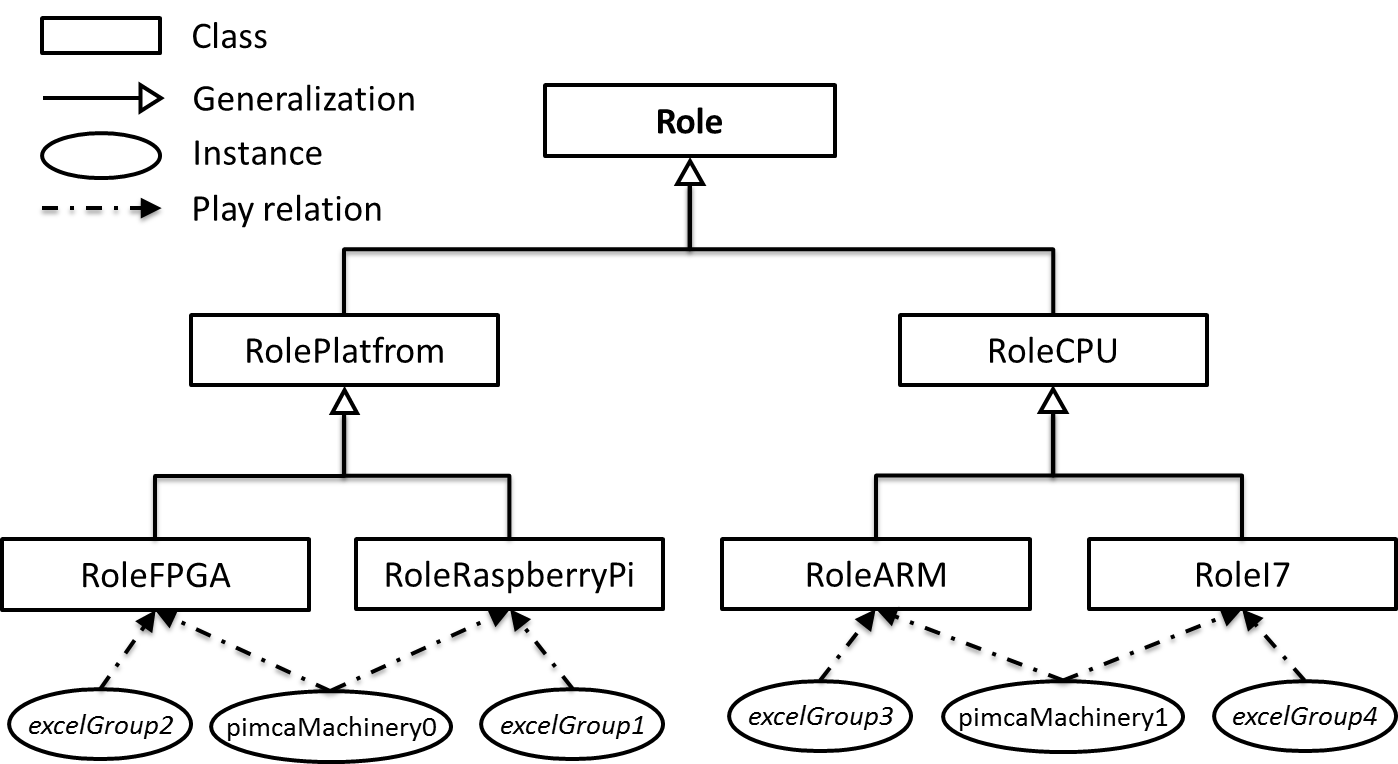


Figure 3: The role model create on Role4All by the terrorist

The role model of the figure 3 allows to the terrorist to define platform and processor’s concepts and to specify them to create new roles like FPGA or ARM. With this role model the terrorist can create a point of view on his tools and associate elements form various tools. As an example the Pimca’s element p*imcaMachinery0* and the Excel’s element *excelGroup2* (the equivalent of the line 3 of the Excel file) play the same role: *RoleFPGA*. Finally Pimca’s elements and Excel’s elements are linked through roles therefore the terrorist can create points of view on different elements of various tools. As example, due to the role model, the terrorist can manipulate the concept of FPGA instead of the couple of tools Excel and Pimca.

A point of view allows working in a specific environment on data provided by different tools (like Pimca and Excel) and sometimes the same information are included in different tools, in our example the Excel file and the Pimca model provide the same data: the ARM’s model. In this case, synchronization is necessary and Role4All provide a synchronization system through roles. So the terrorist defines some synchronization rules as an example he choose to synchronize the ARM model value on Pimca according to its value on Excel. In this specific case the “good” value will always be the value on Excel. Of course the user can create other synchronization rules adapted to the context. The figure 5 is an example of synchronization, the terrorist uses Excel to update an Excel element synchronized with a Pimca model element through a role (roleFPGA0).

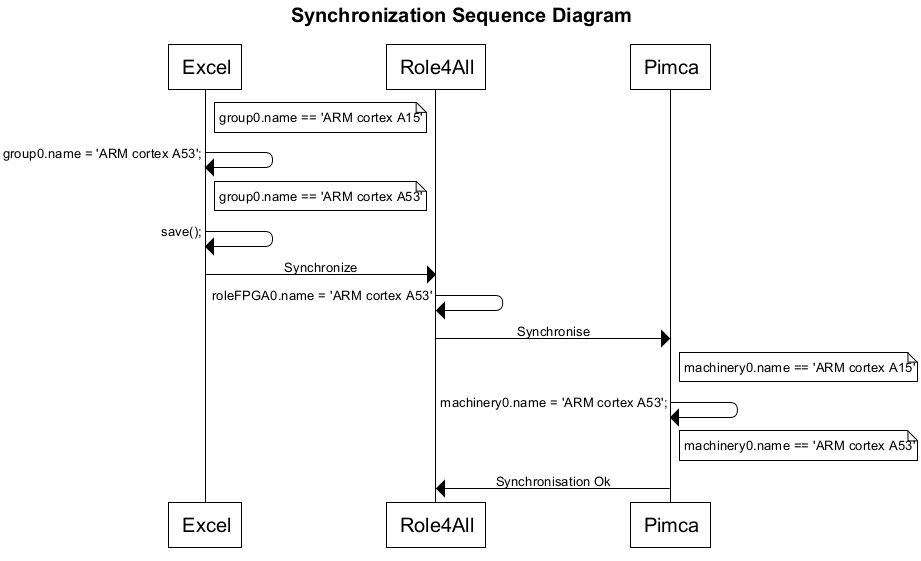


Figure 5 : A sequence diagram of a synchronization using Role4All

When the user saves his Excel file he sends a request to Role4All. Role4All detects the changes between the new model elements and the last ones and applies the suitable modifications to the Pimca model elements. The suitable modifications are defined due to the adapter between Role4All and Pimca associate with the role *roleFPGA0* (*dynamicAdapterFPGA\_Pimca0*). In this example the user use Role4All only to synchronize tools, but the main feature of Role4All is the creation of dedicated point of views. Therefore the common use case of the synchronization in Role4All uses the points of view. As an example to update the name of the role FPGA (the update performed in the figure 5) the user can use a point of view dedicated to the concept of FPGA or another one which gathering only the name of the elements. So the terrorist works with this point of view instead of tools and synchronize them due to Role4All.