Collaborative Role Modeling in a Cybersecurity Context

# Introduction

<http://www.isaca.org/Knowledge-Center/Blog/Lists/Posts/Post.aspx?ID=296>

<http://whatis.techtarget.com/definition/cybersecurity>

<https://www.dhs.gov/topic/cybersecurity>

Cybersecurity is the body of technologies, processes and practices designed to protect networks, computers, programs and data from attack, damage or unauthorized access. Cybersecurity is a complex concept including several elements like:

* [Application security](http://searchsoftwarequality.techtarget.com/definition/application-security)
* Information security
* Network security
* End-user education.

The diversity of elements involves the co- operation of the various services (Developer, Network manager, Managers, etc.). Each services use some specific tools adapt for their skills and others more common but for some specific tasks. Role4All allows simplifying this co-working with two main services:

* Specific point of view of a complex tools
* Synchronization of various tools

One of the most problematic elements of cybersecurity is the quickly and constantly evolving nature of security risks. Indeed today the most secure system without management is sure for a very short time (less than one year). The traditional approach has been to focus most resources on the most crucial system components and protect against the biggest known. Such an approach is insufficient in the current environment. The National Institute of Standards and Technology (NIST), for example, recently issued updated guidelines in its risk assessment framework that recommended a shift toward continuous monitoring and real-time assessments. Unlike the pivot model solution, Role4All is a dynamic federation tool and allows to update some elements of a system group without interact with the others.

The following parts present the role-based framework Role4All according to a cyber-attack example.

# Application to the cybersecurity context

A cyber terrorist wants infected a system named TEST with a homemade virus. To do his mission he needs some information about the system (conception, consumption, etc.). For our example we simplify the system to two elements: a platform (FPGA) and a processor (ARM) with an Ethernet connection. To collect information the terrorist hacks the mailbox of a member of the project TEST. 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 TEST is compose of two elements, a platform and a processor. The terrorist limits his investigation to two sources (Raspberry Pi and FPGA) and two processors (ARM and I7). Due to the worksheets of each product the terrorist can create an array to create a relation between some product name (Raspberry Pi, FPGA, ARM, and I7) and their consumptions.

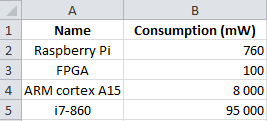


Figure 1: Excels file gathering some consumption

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

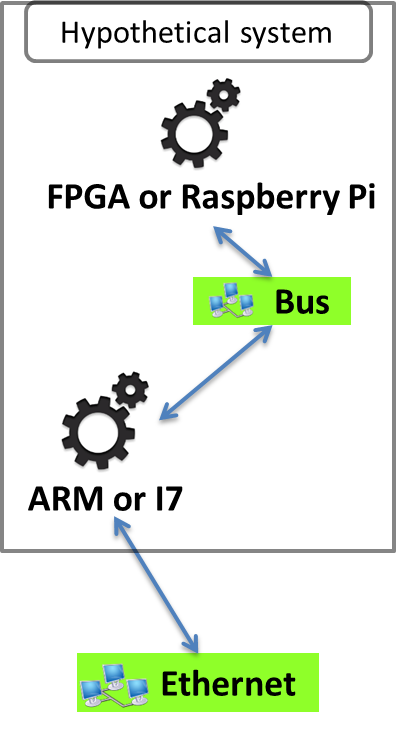


Figure 2: Pimca model of the hypothetical system

This model describes a simple system including two elements, a platform and a processor. But we have an undetermined about the 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.

# Federation Based on Roles

## The Necessity of Federation

In most projects we use several specific tools to model systems or manipulate data, and sometimes these tools use the same model or data. Therefore a collaborative process is necessary, it is called interoperability. Unification is a solution to create interoperability between two tools that consists on creating a pivot model shared by the tools. This solution is simple to implement if all the tools are linked to the same pivot model. But the unification creates two main problems, the first emerges when the pivot model is modified, indeed by definition of a pivot model all tool models linked with it need to be updated too. The second one comes when a new tool is added to the project, it is necessary to connect the new tool with the pivot model without modifying the pivot model (cf problem one). This task can be a real problem if the tool model and the pivot model are strongly discordant. Moreover the tool models progress faster than standards considering that we cannot use a pivot model for a dynamic interoperability [thesis of Maud RIO]. Therefore other solutions were imagined to create a dynamic interoperability, like federation systems. Role4All is an example of federation system, in Role4All a role model defines as a pivot model allows the interoperability between several tools.

In our applicative example the cyber terrorist uses two tools (Pimca and Excel) and several concept common of two tools (FPGA, ARM, etc.). The terrorist uses Role4All to federate his tools (Pimca and Excel) through concepts that he defined himself (FPGA, ARM, Raspberry Pi and I7). In each tool, each model element can play roles defined in a role model (figure 3) in Role4All. Moreover all elements can play roles including elements of unrelated type, therefore with Role4All it is possible to federate all tool types.

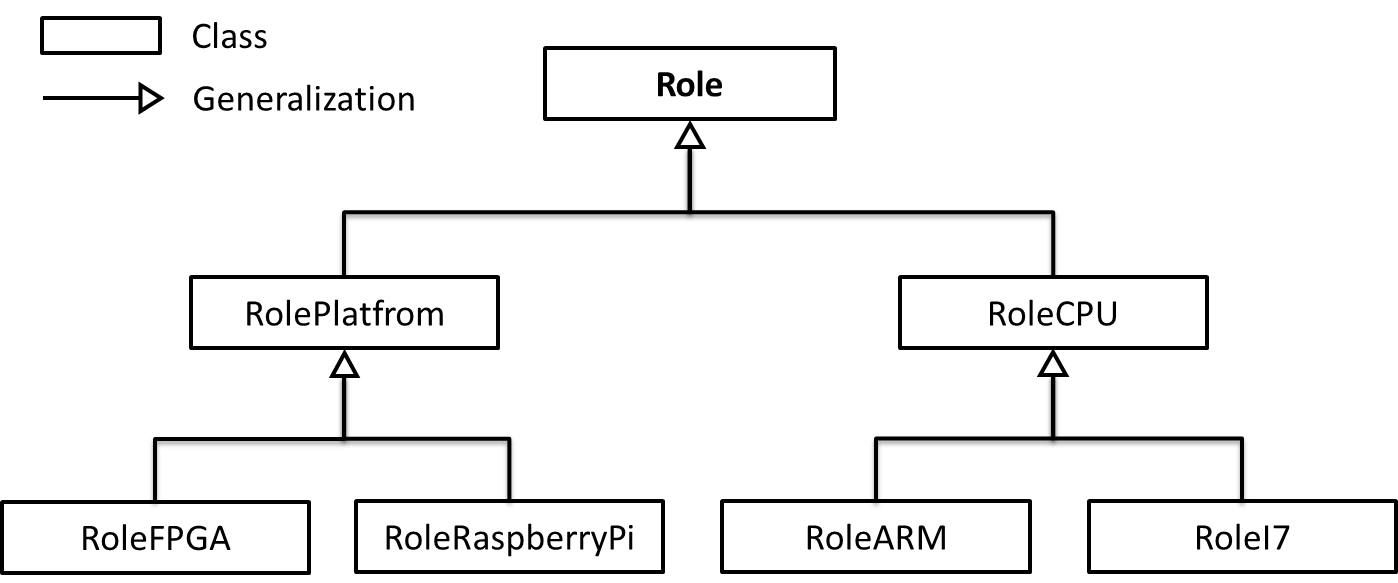


Figure 3: The role model create on Role4All by the terrorist

In our example the terrorist defines two main roles: *RolePlatform* and *RoleCPU* these roles are generalizations of the concept of Role 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*. 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 concepts like FPGA or ARM. The ability to extend or regroup roles provided by Role4All allows also creating different points of view about the same model according to the context [old article].

## Role4All a Dynamic Federation Tool

Role4All is structure around four main classes: *Player*, *Role*, *DynamicAdapter* and *PlayRelation*.

The elements extending the class *Player* are the model elements of tools, they are called “player”. In our example the terrorist uses two tools (Pimca and Excel) each tool has a meta-model whose complexity depends on the tool [ExcelMeta-model, PimcaMeta-model]. For example *PimcaMachinery* is an element of the Pimca’s meta-model and *PimcaMachinery0* is an element of a Pimca’s model (figure 2). The terrorist wants than the Pimca’s model element *PimcaMachinery0* play the role of *RoleFPGA* and he uses Role4All to create this relation. The connection between a role model element (*RoleFPGA*) and a tool model element (*PimcaMachinery0*) is formatted by an adaptor. Unlike a pivot model, a role model is independent of the tools, for example if a tool model is updated we need to adapt the adaptor but not the role model himself.

The elements extending the class *Role* are the concept create by the user through the role models, they are called “role”. Therefor say “In Role4All a role can play a role” is equivalent to say a role can be a player or the *Role* class extends the *Player* class.

The elements extending the class *DynamicAdapter* allow adapting a model element for the role played by it, they are called “adapter”. The adapters define the behavior of the relations between players and roles.

The elements extending the class *PlayRelation*

# Synchronization in Role4All

# Application to the cybersecurity context 2

# Conclusion