Security Properties in Virtual Organizations

Khalifa Toumi, César Andrés, Ana Cavalli

TELECOM & Management SudParis
CNRS UMR Samovar, Evry, France
{Khalifa.Toumi, Andres.Cesar, Ana.Cavalli}@it-sudparis.eu

Abstract

In this paper we present a modeling formal framework to specify nets of virtual organizations. In these nets are defined not only the employees that are working in each organization but also there are included some properties that allow to ensure the security. This framework makes it easier to write and understand the security properties behavior. In addition to the syntax and semantics we provide a running example of two organizations to understand this approach.

Index Terms—Interoperability policy, modeling, testing, security

I. Introduction

Among those areas where the development of computer science has changed our society during the last years, the relevance of the information systems collaboration is remarkable [5]. There is a strong demand for access control of distributed shared resources in research [4], [14], [23], [2] and in particular, in the area of nets of virtual organizations [12], [25], where the classical notion of client-server architecture is obsolete and useless. A net is composed of several organizations sharing services, employees, and resources. Therefore cross-organizational interoperability is a major challenge in nets of virtual organizations [7] particularly in issues related to security. To be able to specify not only the functional aspect of a net but also those aspects that guarantee the interoperability of security policies is an industrial necessity [16], [11]. Currently a perspective that has received little attention is the establishment of a sound methodology to determine the correctness of a net with respect to all the interoperability policies that are defined on it. Formal methods [26] provide us a compound of mathematical techniques that allows the automated design, specification, development and verification of software systems. One of the advantages of using a formal representation of a system is that it allows to rigorously analyze their properties. In particular, it helps to establish the correctness of a system with respect to the specification or the fulfillment of a specific set of requirements, to check the semantic equivalence of two systems, to analyze the preference of a system to another one with respect to a given criterion, to predict the possibility of incorrect behaviors, to establish the performance level of a system, etc.

The first contribution of this paper is to provide a formal framework that allows us to represent nets of virtual organizations without any lack of precision. We show that the use of a formal model permits to provide the semantics of a net using a Labeled Transition System, in short LTS. It is worth to mention that the interoperability requirements of the net are also described in a formal way.

An additional contribution of this paper is to include some techniques of testing [19], to check the correctness of a virtual organization with respect to the required security properties. In testing there is usually a distinction between two approaches: passive and active. The main difference is that in active testing a tester can interact with the system, while in passive testing the tester simply monitors the behavior of the system. We consider that passive testing techniques are well adapted to study virtual organization correctness. For instance, the different interactions of the employees in virtual organizations can be recorded into traces, i.e., logs; and then checking its correctness with respect to the security properties can be performed by adapting some of the algorithms presented in [1].

The rest of the paper is structured as follows. In Section II related work is presented. In Section III the formal framework, syntax and semantics, regarding virtual organization is introduced. In Section IV it is presented our passive testing methodology. Next, in Section V the soundness of this technique is showed. Finally, in Section VI the conclusions and some lines of future work are presented.
II. Related work

The concept of virtual organization, interoperability testing and security policy in distributed environments form an active research area [15], [13], [21], [24], [20]. In [15], the trust chain in virtual organizations based on the ISO \ IEC 17799 best practices and maturity level concepts is well studied. In this work, the authors combine the Attribute Based Access Control (ABAC) model, with privilege management infrastructures and identity federation. They provide a tool that calculates the maturity level of the security practices of each partner. This information is used to determine the different partners and their roles. However, there are some issues that previous works do not deal, and this paper does:

1) To take into account the functional and security requirements.
2) To provide time information.
3) To use the same semantics framework

Moreover, these previous works are not focused on providing a formal model of their interoperability policies nor on testing techniques to validate their interactions. Conformance testing [10], [22], [6], [17] is a methodology that allows us to check the behavior of a system according to its specification. This kind of testing of the interoperability security policy is necessary in virtual organizations. However, conformance testing is not enough to ensure the integration of heterogeneous systems. In particular, the ISO 9646 mentions that conformance testing can increase the probability of interoperability but they cannot ensure the successful collaboration between different entities.

Previous observations motivate several researchers to focus on interoperability testing [3], [21], [20]. Several issues can be discussed related to this type of testing [20]:

1) Functional: testing should be used to check that the different systems apply correctly the communication standards.
2) Semantic: the correct comprehension of the data and the expected interactions should be validated.
3) Security in order to verify that their collaboration respects some rules.
4) Timeliness to check timing requirements.

Different solutions aim to focus on one or more issues. For instance, in [21] the authors provide a tool for interoperability testing that is used to test the share of medical images. In [3], authors propose a new interoperability testing framework that permits to interface some conformance testing tools for generating interoperability tests. They offer also an algorithm for checking the correctness of the communicating systems. Previous works are based on active testing. However, it cannot be used during the real communication between employees since it will disturb the communication in the net. Moreover, in these approaches the testing of interoperability policy (permissions, activities and roles) is not discussed.

III. Formal model to represent virtual organizations

In this section we introduce the formalism to specify virtual organizations with timed restrictions. To do this, we give a preliminary notation regarding the definition of time intervals. In this paper intervals are used to represent time information and therefore contain real values greater than or equal to zero.

Definition 1: We say that \( \bar{p} = [p_1, p_2] \) is a time interval if \( p_1 \in \mathbb{R}_+, p_2 \in \mathbb{R}_+ \cup \{\infty\} \), and \( p_1 \leq p_2 \). We consider that \( \mathcal{T} \) denotes the set of time intervals.

Let us note that time intervals will be used to express time constraints, associated with the performance of actions, in the definition of the organization concept. The idea is that if a time interval \([p_1, p_2] \in \mathcal{T}\) is associated with a task, then it is expected that this task takes at least \( p_1 \) time units and at most \( p_2 \) time units to be performed. Intervals like \([0, p_2]\), \([p_1, \infty]\), or \([0, \infty]\) denote the absence of a temporal lower/upper bound and the absence of any bound, respectively.

The formalism used to represent specifications of virtual organizations is given in the next definition. Basically, a virtual organization represents its internal structure, that is a set of roles, a set of actions that the employees of this organization can perform, the time that is needed to perform each action, and the mapping permission function to relate actions with roles.

Definition 2: Let \( I \) be the set of all possible organization identifiers, \( A \) be the set of all possible actions that an organization can perform, and \( \mathcal{R} \) be the set of all possible roles.

A virtual organization, in short an organization, is a tuple \( E = (id, A, R, time, p) \) where \( id \in I \) is the organization identifier, \( A \subseteq A \) is the set of actions that employees can perform, \( R \subseteq \mathcal{R} \) is the set of roles, \( time : A \rightarrow \mathcal{T} \) is a function that assigns a time interval to each action of the organization, and finally \( p : R \rightarrow p(A) \) is a function that assigns rights to each role for performing a set of actions. We will denote the set of all virtual organizations by \( \mathcal{E} \).

A virtual organization is an easy way to formally represent the organization of a company independent of employed people. Following we present our running example to introduce this concept.

Example 1: In the first part of the Figure 1 the structure of an organization identified by \( id_1 \) is described. There are three types of actions that an employee can perform in \( id_1 \): \( a_1, a_2, a_3 \). The time function denotes the time that a task needs to be completed. For instance, \( time(a_1) = [1, 4] \)
means that a person who performs the action $a_1$ in the organization $id_1$ needs at least 1 time unit and at most 4 time units to complete this action.

In addition, in $id_1$ there are two different roles $r_{1a}$ and $r_{2a}$. To each role is assigned the set of actions that employed people in this role are allowed to perform. For instance, we have that any employee mapped in role $r_{1a}$ can perform the tasks $a_1$ and $a_2$, while any employee mapped in role $r_{2a}$ can perform the tasks $a_3$ and $a_3$. If a person performs the action $a_1$ and after this, she performs $a_3$, the amount of time spent to complete both tasks will belong to $\text{time}(a_1) + \text{time}(a_3)$. ☐

The next definition introduces the concept of context of an organization. In our approach, a context is a representation at a given time of the virtual organization $E$. In particular, a context for $E$ is a tuple $V_E = (H, x)$, where $H \subseteq \mathcal{H}$ is the set of employees of $E$, and $x: H \rightarrow \mathcal{V}(R)$ is a function that assigns to each person of $H$ a set of roles. We denote by $\mathcal{V}$ the set of all possible contexts.

Regarding the notation, from now on people identifiers will be denoted by $h_i$, actions by $a_i$, and organizations by $E_i$.

Example 2: We present the notion of context in our running example. In the right part of Figure 1 we model the following situation: There are seven employees in the organization $id_1$, labeled by $h_1, \ldots, h_7$. Three of them are assigned to the first role of $id_1$, that is, $r_{1a}$. Two employees are assigned to the second role, by means $r_{2a}$, and the rest of employees are assigned to both roles $r_{1a}$ and $r_{2a}$. ☐

Next we focus on representing an environment with different organizations, i.e., a net of virtual organizations, and the relationships among them. We use the concept of Virtual Private Organization, in short VPO to represent the external relationships of the different organizations. A VPO is basically a set of rules to assign temporally rights, i.e., temporally roles, in an organization to people that do not work in this organization. These VPOs are associated with each organization. Before providing the formal definition we present the VPOs extending our running example. Readers interested in learning more about the use of VPOs and their advantages are referred to [8].

Example 3: In the above part of Figure 2 a new virtual organization, by means $id_2$, is defined. Let us consider the organization $id_1$ that was previously presented. Next we represent that some employees of $id_1$ can work in $id_2$. On the one hand the employees of $id_1$, $h_1$ and $h_5$ will have the role $r_{2b}$ of $id_2$ and on the other hand the employee $h_2$ will have the roles $r_{1b}$ and $r_{2b}$. Let us remark that each organization has assigned a set of VPOs in a net, and any VPO assigns temporal rights to people that are not employees of this organization. The below part of the Figure 2 represents the VPO previously presented. ☐

Following, we introduce the notion of a net of virtual organizations in our model as a set of virtual organizations and a set of VPOs.
Definition 4: Let $E_1$ and $E_2 \in \mathcal{E}$ be two virtual organizations. We define a Virtual Private Organization for $id_2$ in $id_1$, in short $\text{VPO}(id_2 \rightarrow id_1)$ as tuple $(H, \chi)$ where $H \subseteq H$ is a set of people identifiers and $\chi : H \rightarrow \varphi(R_i)$ is a function that maps $H$ in some roles of $id_1$. The set of all possible VPOs is denoted by $\text{SetVPO}$.

We say that a set of organizations, or simply net, is a tuple $N = (\mathcal{E}_1, \nu)$ where $\mathcal{E}_1 \subseteq \mathcal{E}$ is a set of organizations and $\nu : \mathcal{E}_1 \times \mathcal{E}_1 \rightarrow \text{SetVPO}$ is the function that assigns an unique VPO for each pair of organizations. We denote by $\nu(E_1, E_2) = \emptyset$ the absence of VPO between $E_1$ and $E_2$. We assume that for all $E_1$ and $E_2 \in \mathcal{E}_1$ we have that $\nu(E_1, E_2) = \emptyset = \nu(E_1 = E_2)$. Finally, the set of all nets will be denoted by $N$.

Previous definition let us represent nets of different organizations taking into account the following restriction: An organization $id$ cannot map people of this organization in its VPOs. That is, the definition of $\text{VPO}(id \rightarrow id)$ is not allowed. Moreover, this restriction also imposes that for each pair of virtual organizations $id_1$ and $id_2$ in the net, there exists two VPOs. These are $\text{VPO}(id_1 \rightarrow id_2)$ and $\text{VPO}(id_2 \rightarrow id_1)$, and they are located in $id_1$ and $id_2$ respectively.

Example 4: We need to add the VPO that maps people from $id_2$ in $id_1$ to our running example to have the net correctly defined. In particular, we define that the employee $h_8$ is mapped in role $r_{1a}$.

\[ \text{VPO}(id_2 \rightarrow id_1) = h_8 = \{ r_{1a} \} \]

Next we present the notion of context of a net. The idea is similar to the previous context definition in a virtual organization, but taking into account that there is not only one organization but a set.

Definition 5: We will define a context of a net $N = (\mathcal{E}_1, \nu)$, denoted by $\bar{\mathcal{N}}$, as a vector of net contexts $\bar{\mathcal{N}} \in V^{\mathcal{E}_1}$, where each organization of $\mathcal{E}_1$ has an unique position. We will denote by $\bar{\nu}[\bar{\mathcal{N}}]$ the context for the organization $(id, A, R, \text{time}, p)$ and by $\bar{\nu}$ the set of all possible contexts of the nets.

We will say that a context $\bar{\mathcal{N}}$ is valid for the net $N$ if the following condition holds: For any $E_1$ and $E_2$ in $\mathcal{E}_1$, with $E_1 \neq E_2$, and $\bar{\nu}[\bar{\mathcal{N}}id_2] = (H_2, x_2)$ we have that $\nu(E_1, E_2) = \text{VPO}(id_2 \rightarrow id_1) = (H, x)$, we have that $H \subseteq H_2$.

In previous definition a valid context is presented, as an environment where for a VPO that appears in a net, if it is defined from $id_2$ to $id_1$, that is $\text{VPO}(id_1 \rightarrow id_2) = (H, x)$, we have that all employees belonging to this VPO, that is $H$, must be employees of $id_1$.

We finalize this Section presenting the semantics of this model. Basically, we give a characterization of the meaning of passing of time in a net of virtual organizations. To carry out this task we make a translation of the net into a LTS within an unbounded number of states. Let us remark that, in general, we will not build the associated LTS. We will use it to reason about the traces.

Definition 6: A Labeled Transition System, in short LTS, is defined by a tuple $(\mathcal{Q}, q_0, A, H, \tilde{g}, \rightarrow)$ where we have that $\mathcal{Q}$ is a set of states, $q_0 \in \mathcal{Q}$ is the initial state, $A \subseteq A$ is a set of actions, $H \subseteq H$ is a set of people identifiers, $\tilde{g}$ is a tuple in $R^{H}$, and the relation $\rightarrow \subseteq Q \times A \times H \times R \times Q$ represents the set of transitions. From now on, a transition $(q_1, q_2)$ that belongs to $\rightarrow$ will be denoted by $q_1 \rightarrow q_2$.

In our LTSs we consider that each person $h_i \in H_1$ has associated an unique position of the tuple $\tilde{g}$. This position will be denoted by $\tilde{g}[h_i]$.

The semantics of computing $t$ time units in a context $V_{\nu} = (H, \chi)$, where we have that the organization $E$ is $(id, A, R, \text{time}, p)$, is defined by its associated LTS $(\mathcal{Q}, q_0, A, H, \tilde{g}, \rightarrow)$, being $\tilde{g} = (\tilde{g}_1)$, that is all the positions has the value 0, and we apply the rule 1 presented in Figure 3 in order to generate the elements of $\rightarrow$ (let us remark that any new generated state will be in $Q$).

We denote by $\text{LTS}(V_{\nu})$ the LTS associated with this context, and by $\text{SetLTS}$ the set of all LTSs.

The semantics of computing $t$ time units in a context of a net $\bar{\mathcal{N}}$, where $\bar{\mathcal{N}} = (\mathcal{E}_1, \nu)$ is a tuple of $\text{LTS}$ $\bar{\mathcal{N}} = \bar{\mathcal{N}} = \bar{\mathcal{N}}[\mathcal{E}_1]$, $\mathcal{E}_1 \subseteq \text{SetLTS}$, each organization $E = (id, A, R, \text{time}, p)$ belonging to $\mathcal{E}_1$ has an unique position in $\mathcal{E}_1$. This position will be represented by $\mathcal{E}_1[id]$.

Let $\text{LTS}(\bar{\mathcal{N}}[id]) = (\mathcal{Q}, q_0, A, H, \tilde{g}, \rightarrow)$ be the LTS associated with the organization $E$. We define $\bar{\mathcal{N}}[id]$ as a LTS $(\mathcal{Q}', q_0, A, H', \tilde{g}', \rightarrow')$ where: Initially we have that $\mathcal{Q} \subseteq \mathcal{Q}'$. The set $H'$ must also represent those employees that do not belong to $id$. This is formally represented as:

\[ H' = H \bigcup H_2 \]

\[ E_1 \in \mathcal{E}_1, id_1 \neq id \]

\[ (H_2, \chi) = \nu(E_1, E) \]

Finally we have that $\rightarrow \subseteq \mathcal{Q}' \times \mathcal{Q}'$, where the new transitions of $\rightarrow$ are generated by applying the rule 2 presented in Figure 3.

Let us note that the previous Definition needs an amount of time $t$ to define the LTS. Basically in this LTS are represented all the possible interactions that employees of an organization can perform during $t$ time units. Following we introduce in our running example the idea of using this semantics.

Example 5: Let us present the LTS of our running example $id_1$ when $t = 2$. It is graphically depicted in Figure 4. Due to space limitation we represent the LTS in a short way. The first state of the LTS is $q_0$. From this
\[
    h_i \in H \land r_i \in x(h_i) \land a_i \in p(r_i) \land t' \in \text{time}(a_i) \land g[h_i] + t' \leq t
\]
\[
    q_i^{(a_i, h_i, t')} \Rightarrow q \land g[h_i] = g[h_i] + t'
\]

\[\text{(1)}\]

\[
    E_i \in E_i \setminus \{E\} \land (H_i, x_{t_i}) \in v(E_i, E) \land
    h_i \in H_i \land r_i \in x(t(h_i)) \land a_i \in p(r_i) \land t' \in \text{time}(a_i) \land g[h_i] + t' \leq t
\]
\[
    q_i^{(a_i, h_i, t')} \Rightarrow q \land g[h_i] = g[h_i] + t'
\]

\[\text{(2)}\]

*Figure 3. Rules to generate the elements of the LTS.*

*Figure 4. Example of LTS when } t = 2.*

... state the outgoing transitions are represented with the sets } T_1, T_2, T_3, T_4, T_5, T_6, T_7. On the one hand the arcs that go to gray filled states, that is } T_1, T_2, T_3, T_4 are generated applying the first operational rule. On the other hand the arcs that go to black filled states are computed by applying the second operational rule. Let us note that gray filled states and black filled states are not real states but sets of different states of the LTS.

For instance, the transitions belonging to } T_2 represent that the employees } h_1, \ldots, h_5 are allowed to perform the action } a_1 in } 1 \text{ time unit. The set of transitions } T_2 represents that the employees } h_1, \ldots, h_5 perform the action } a_1 in } 1 \text{ time unit belonging to } (1, 2). Let us note that the upper bound of this action is } 2 \text{ because we are building the } LTS \text{ with } t = 2. The set } T_4 \text{ represents the performing of the action } a_2 \text{ by the employees mapped in the role } r_{2a} \text{ in } 2 \text{ time units. The set } T_4 \text{ represents those transitions that begins in the goal states of the transitions of set } T_1 \text{ and performs the action } a_1 \text{ in } 1 \text{ time unit. We add the sets of transitions } T_5, T_6 \text{ and } T_7 \text{ to represent the VPD(1d_2 \rightarrow 1d_1) presented in previous example, where the employee } h_8 \text{ was mapped to the role } r_{1a}. \text{ Thus, he can perform the task } a_2 \text{ in } 1 \text{ time unit belonging to } (1, 2). \overset{\square}{\text{ }}

Let us note that within this semantics we are allowed to represent the behavior of a net. In particular, when a virtual organization is monitored and all the interactions are collected in a log it produces a trace. Next, we formally introduce the notion of trace. As usual, a trace is a sequence of visible information from a given execution.

**Definition 7.** Let } V_E \text{ be a context. We will say that the sequence } \rho = ((a_1, h_1, t_1), (a_2, h_2, t_2), \ldots, (a_n, h_n, t_n)) \text{ is a trace of } V_E \text{ if there exists the following transitions in its associated LTS:}

\[
    q_1 \xrightarrow{a_1, h_1, t_1} q_2 \xrightarrow{a_2, h_2, t_2} \ldots \xrightarrow{a_n, h_n, t_n} q_{n+1}
\]

We denote by } \cdot \text{ the concatenation of traces, by } \emptyset \text{ the empty trace, by } \mathcal{L}(V_E) \text{ the set of all traces of } V_E, \text{ and by } \mathcal{L} \text{ the set of all possible traces of any LTS. } \Box \text{ Let us remark that previous definition is made in a general way. That is, when we consider the associated LTS of a context without taking into account the time } t \text{ means for any possible } t \in \mathbb{R}_.

**Example 6:** We introduce the notion of a trace using the LTS presented in Figure 4. The following traces belongs to this LTS: } \rho_1 = ((a_1, h_1, 1)), \rho_2 = ((a_1, h_1, 1), (a_1, h_2, 1)), \rho_3 = ((a_1, h_1, 1, 8)), \text{ and } \rho_4 = ((a_2, h_2, 2)). \overset{\square}{\text{ }}

In our framework we consider that both, the specification and the implementation of the net can be modeled using the same formalism. Thus, when we refer to a trace of the implementation we assume on the one hand that this is a collected sequence of actions in the implementation, and on the other hand that this is a trace of the associated LTS of the model of this implementation.
IV. Security Properties Definition

In this section we first introduce the notion of security properties. Next, we discuss and present how to check the correctness of a trace with respect to a set of rules using passive checking techniques.

To achieve a good interoperability policy for a net, first we need to identify the critical aspects of the net. That is, those characteristics that will make the difference between a correct and an incorrect behavior. Thus, we describe a list of possible situations in the checking process where an error, i.e., an incorrect behavior, should be detected in a net of virtual organizations. Let us consider the trace: \( \rho_1 = \langle (a, h, t) \rangle \) collected in the organization \( E \). We sketch the checking process:

1) If \( a \) does not belong to the set of actions presented in the organization-scheme \( E \) return \( \text{error}_1 \), else go to 2.
2) If \( a \) cannot be done in a time \( t \) then return \( \text{error}_2 \), else go to 3.
3) If \( h \) works in the organization \( E \) go to 4 else go to 6.
4) According to the local policies of \( E \), collect in \( R \) those roles of \( E \) where \( h \) is mapped. If \( R = \emptyset \) then return \( \text{error}_3 \), else go to 5.
5) If any role of \( R \), where \( h \) is mapped can perform a then return \( \text{correct} \) else return \( \text{error}_4 \).
6) If \( h \) does not appear in any VP of \( E \) as external employee return \( \text{error}_5 \), else go to 7.
7) According to the VP policies of \( E \), collect in \( R' \) those roles of \( E \) where \( h \) is mapped. If \( R' = \emptyset \) then return \( \text{error}_6 \), else go to 8.
8) If any role of \( R' \), where \( h \) is mapped can perform a then return \( \text{correct} \) else return \( \text{error}_7 \).

Let us consider the previous set of 7 errors that we are able to detect. In Figure 5 are related these errors with the information that is needed for the checking process. For instance, \( \text{error}_1 \) is produced when the action \( a \) that appears in the trace does not belong to the set of actions that the organization \( E \) can apply. Therefore, to check this error we only need the information presented in \( E \). According to this scheme, we classify the errors in three groups: external, internal and integrity. The external errors are \( \text{error}_5, \text{error}_6 \) and \( \text{error}_7 \), which denote an external employee of the company. Therefore, the internal are \( \text{error}_3 \) and \( \text{error}_4 \) that denote a wrong behavior detected in the internal employees. Finally, the integrity errors \( \text{error}_1 \) and \( \text{error}_2 \) can be detected in internal and external employees, when they perform an irregular action in the organization. In the next definition the functions \( c_{\text{ext}}, c_{\text{int}}, \) and \( c_a \) are responsible for checking the external, the internal, and the integrity policies respectively.

**Definition 8:** We say that the functions \( c_{\text{ext}} : H \times A \times E \times \bigcup \{\text{SetVP0}\} \rightarrow \{\text{true}, \text{error}_1, \text{error}_2\} \) and \( c_a : A \times R_a \times E \rightarrow \{\text{true}, \text{error}_1, \text{error}_2\} \) check the external, the internal and the integrity interoperability policies of a net of virtual organizations. For any \( h \in H, a \in A, E = (\text{id}, A, R, \text{time}, p) \in E, t \in R_a, \) and \( K \subseteq \text{SetVP0} \), we have that the function \( c_{\text{ext}}(h, a, E, K) \) is defined as:

\[
c_{\text{ext}}(h, a, E, K) = \begin{cases} 
\text{error}_5 & \text{if } \forall(h, xt) \in K : h \in H \\
\text{error}_6 & \text{if } \exists(h, xt) \in K : h \in H \land \\
\quad ((xt(h) \cap R) = \emptyset) \\
\text{error}_7 & \text{if } \exists(h, xt) \in K : h \in H \land \\
\quad (xt(h) \cap R) \neq \emptyset \land a \notin p(r) \\
\text{true} & \text{other case}
\end{cases}
\]

The function \( c_{\text{int}}(h, a, E) \) is defined as:

\[
c_{\text{int}}(h, a, E) = \begin{cases} 
\text{error}_1 & \text{if } a \notin A \\
\text{error}_2 & \text{if } a \in A \land t \notin \text{time}(a) \\
\text{true} & \text{other case}
\end{cases}
\]

And the function \( c_a(a, t, E) \) is defined as:

\[
c_a(a, t, E) = \begin{cases} 
\text{error}_3 & \text{if } a \notin A \land t \notin \text{time}(a) \\
\text{true} & \text{other case}
\end{cases}
\]

Following we introduce the function \( \mathcal{m}(\rho, \mathcal{V}_N) \) that checks the correctness of a trace \( \rho \), collected in an organization \( E \) of the net \( N \), with respect to the interoperability policies described in the context \( \mathcal{V}_N \) of this net.

**Definition 9:** We define the function \( \mathcal{m} : \mathcal{L} \times \mathcal{V} \rightarrow \{\{\text{true}, \text{error}_1, \text{error}_2, \text{error}_3, \text{error}_4, \text{error}_5, \text{error}_6, \text{error}_7\}\} \) that checks the correctness of a trace with respect to the interoperability policies of a net. For any \( N = (\mathcal{E}_1, \mathcal{V}) \in \mathcal{C} \), for any context \( \mathcal{V}_N \in \mathcal{V} \), for any \( E = (\text{id}, A, R, \text{time}, p) \in \mathcal{E}_1 \) such that \( \mathcal{V}_N[\text{id}] = (H, x) \), and for any \( \rho = \langle (a_1, h_1, t_1) \rangle : \rho_1 \in \mathcal{L} \langle \mathcal{V}_N[\text{id}] \rangle \) we have that \( \mathcal{m}(\rho, \mathcal{V}_N) \) is defined as:

\[
\mathcal{m}(\rho, \mathcal{V}_N) = \begin{cases} 
\{\text{true}\} & \text{if } \rho = \emptyset \\
\{c_a(a_1, t_1, E_1)\} \cup & \text{if } \rho \neq \emptyset \land h_1 \notin H \\
\{c_{\text{ext}}(h_1, a_1, E_1, K)\} \cup & \text{if } \rho \neq \emptyset \land h_1 \in H \\
\mathcal{m}(\rho', \mathcal{V}_N) & \text{if } \rho \neq \emptyset \land h_1 \notin H \\
\{c_{\text{ext}}(h_1, a_1, E_1)\} \cup & \text{if } \rho \neq \emptyset \land h_1 \in H \\
\mathcal{m}(\rho', \mathcal{V}_N) & \text{if } \rho \neq \emptyset \land h_1 \notin H
\end{cases}
\]

where \( K = \bigcup \{v(E, E_1)\} \) and \( E_1 \in \mathcal{E}_1, E_1 \neq E \).
Figure 5. Places to detect the different interoperability errors in a net of virtual organizations.

We will say that $\rho$ is correct in a given context if $\mu(\rho, \bar{V}) = \{\text{true}\}$. Otherwise, we say that it is incorrect and the set of errors is $\mu(\rho, \bar{V}) \setminus \{\text{true}\}$. □

Example 7: Let $\rho_1 = (\langle a_1, h_1, 1 \rangle)$, $\rho_2 = \langle h_1, h_1, 1 \rangle, (a_1, h_2, 1 \rangle$, $\rho_3 = \langle a_4, h_1, 1 \rangle$, and $\rho_4 = \langle a_2, h_2, 2 \rangle$ be the four traces of the system $id_1$ previously presented in our running example. We have that all of them are correct with respect to the context of the net presented in Example 4.

Next let us consider the set of traces in Figure 6, collected from an implementation of our net scheme at $id_1$. Following we show, by using passive testing, if there exists any vulnerability of the interoperability policies. On the one hand we have that the following errors are reported in the checking process of $\rho_5$:

- $error_2$ because there exists the sequence $(h_8, a_3, 5)$, and according to the associated time of $a_3$ it must be in $[3, 4]$.
- $error_1$ because $a_0$, in $(h_3, a_5, 2)$, is not a possible action of the net scheme $id_1$.

On the other hand, the checking process of $\rho_6$ reports only one error: $error_5$. This happens because in $(h_9, a_1, 4)$ the external employee $h_9$ does not appear in any VPO associated with $id_1$. □

Figure 6. Possible set of traces of $id_1$.

V. Formal Correctness

This section presents an implementation relation to formally define what a good implementation is with respect to a specification. The ultimate goal is to show the correctness of our approach to check interoperability properties presented in this paper. A simple timed implementation relation is considered, taking into account the time information presented in the transitions of the LTSs, but other alternative relations [18] could be easily incorporated to the framework.

Definition 10: Let $\bar{V}_N$ and $\bar{V}_N'$ be two contexts of the net $N = (E, v)$. We will define that $\bar{V}_N$ conforms $\bar{V}_N'$, denoted by $\bar{V}_N \approx \bar{V}_N'$, if for any organization $E = (id, A, R, time, p) \in E_i$ we have that $L(\bar{V}_N[id]) \subseteq L(\bar{V}_N'[id])$. □

The next result presents the relation among the specification model, the implementation, traces, and interoperability policies. Consider a trace recorded from an implementation-net and the specification model where is represented the interoperability policies of this net. We have that "if it is detected an error in the trace then the implementation is not conform with respect to the specification".

Theorem 1: Let $N$ be a net, $\bar{V}_N$ and $\bar{V}_N'$ be two contexts of this net, by means the implementation and the specification, and $\rho$ be a trace collected under $\bar{V}_N$. If we have that $\rho$ is not correct in $\bar{V}_N$ then we have that $\bar{V}_N \not\approx \bar{V}_N'$, that is, the implementation is not conform with respect to the specification. □

In order to proof this theorem, we have to show that for any context $\bar{V}_N$, the operational rules presented in Definition 6, that allow us to provide the semantics of a context as a LTS, do not generate any transition $(q, a, h, t, q')$ such that $\mu(((a, h, t), \bar{V}_N) \neq \{\text{true}\}$.

Let us note that this result shows the soundness of our methodology. In particular, this result ensures us that if we detect an error is because we have an incorrect implementation of the system. Let us point out that this technique is based on a testing approach, and testing allows us to check the presence, not the absence of error [9].
we can assume that the implementation is consistent until we find a bug.

VI. Conclusions and Future Work

In this paper we have presented a new modeling framework to represent a net of virtual organizations. In this context there are several organizations that share employees and resources among them, thus the interoperability must be taken into account. Within this new model we are allowed to represent on the one hand the organizations, mapping their employees in an internal role hierarchy. On the other hand, when we consider a collaborative environment we model the interoperability rules in sets called Virtual Private Organizations.

In addition with the modeling framework, we present a methodology, based in passive testing, that allow us to check that the interoperability rules are applied in each organization. In particular, this model allow us to identify seven different interoperability erroneous scenarios.

Besides defining the syntax of the model, we have given the semantics of the processes by using LTSs. Moreover, we have provided an implementation-view relation to define whether a system conforms this model. Finally, we have shown the correctness of this approach.

As future work we plan to extend this framework in two ways, on the one hand to apply runtime verification techniques at real time in the collected traces of the system and on the other hand to integrate the trust concept, that defines a dynamic relation between partners, in our model.

References