PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 27/06

Structuring a Law Case for Law-Governed Open Multi-Agent Systems

Maíra Athanázio de Cerqueira Gatti
Gustavo Robizhez de Cavalho
Rodrigo Barros Paes
Carlos José Pereira de Lucena
Jean-Pierre Briot

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900

RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 27/06 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena August, 2006

Structuring a Law Case for Law-Governed Open Multi-Agent Systems *

Maíra Athanázio de Cerqueira Gatti[1], Gustavo Robichez de Carvalho[1], Rodrigo Barros Paes[1], Carlos José Pereira de Lucena[1] and Jean-Pierre Briot[2]

[1]Laboratório de Engenharia de Software – LES
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brasil
{mgatti, guga, rbp, lucena}@inf.puc-rio.br

[2]LIP6, Universit´e Pierre et Marie Curie (Paris 6)
8 rue du Capitaine Scott, 75015 Paris, France
Jean-Pierre.Briot@lip6.fr

Abstract. In an open multi-agent system, where the complexity and uncertainty are issues that must be analyzed during the development and evolution, the experience of emergent behavior makes it even more important to consider dependability attributes. It is harder to design and implement requirements related to dependability in law-governed multi-agent systems mainly because there is no control point over such systems. To this end, the use of an enforcement mechanism and an adapted dependability case approach, which we define as a Law Case, permits to control the failures and to promote the benefits and also contributes to tame the uncertainty presented by open multi-agent systems. Therefore, the goal of this work is to show how to structure a Law Case through a sample scenario in open multi-agent systems regulated by a law-enforcement mechanism.

Keywords: Multi-Agent Systems; Open Systems; Law-Governed Approach; Dependability of Open Systems.

Resumo. Em um sistema mutliagente aberto, cuja complexidade e incerteza são questões que precisam ser analisadas durante a evolução e desenvolvimento do mesmo, a experiência do comportamento emergente aumenta a necessidade de se levar em consideração os atributos de fidedignidade (*dependability*) da aplicação. O projeto e implementação de requisitos de fidedignidade possuem um grau de dificuldade elevado para serem efetuados principalmente pela falta de ponto de controle centralizado sobre um sistema multiagente aberto. Para isto, o uso de um mecanismo de leis e de uma abordagem de casos de fidedignidade adaptada, a qual chamamos Caso de Leis, permite o controle de falhas e promove os benefícios além de abrandar a incerteza presente em um sistema multiagente. Desta forma, o objetivo deste trabalho é descrever como estruturar um caso de leis através de um cenário simples em sistemas multiagentes abertos regulados por leis.

Palavras-chave: Sistemas Multiagentes, Sistemas Abertos, Governança baseada em Leis, Fidedignidade de Sistemas Abertos.

* This work has been sponsored by the Ministério de Ciência e Tecnologia da Presidência da República Federativa do Brasil

In charge for publications:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22453-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3114-1516 Fax: +55 21 3114-1530
E-mail: bib-di@inf.puc-rio.br

Table of Contents

1 Introduction

Generally, achieving sufficient dependability in system development and demonstrating this achievement in a convincing and rigorous manner is of crucial importance [8]. Dependability has been for a long time a major concern in ubiquitous computing systems which control structures as critical as railroads, planes, and nuclear plants. So, concepts and techniques are well established [1].

There are two concepts related to dependability [1][2]. The main one says that the dependability of a computing system is its ability to deliver service that can justifiably be trusted. Correct service is delivered when the service implements the system function, i.e., what the system is intended to do. The second one says that dependability is the ability to avoid service failures that are more frequent or more severe than is acceptable. Considering that a service failure is an event that occurs when the delivered service deviates from correct service, when service failures are more frequent or more severe then acceptable then we have a dependability failure. In this sense, a developer should consider and assess dependability attributes [1] while developing any system.

In an open multi-agent system, where the complexity and uncertainty are issues that must be analyzed during the development and evolution, the experience of emergent behavior makes it even more important to consider those attributes. Moreover, open systems consist of many distributed, asynchronous components that are open to interaction with their environment. The functionality of an open system is not defined by the result of evaluating an expression; instead, the relative state of components, the relative timing of actions, the locality and distribution of the computation, among others, are all critical to the correctness of the system [6]. These systems are populated by heterogeneous components, normally developed by different people using different languages and architectures [7]. Currently, no approach for developing open multi-agent system infrastructures provides means to effectively gauge those characteristics.

A mechanism to enforce behavioral rule (law enforcement mechanism) aims to provide a way to map the consequences that arise from the interactions to the qualitative or even quantitative criteria presented by the dependability attributes. An enforcement mechanism intercepts particular interactions between agents to control and audit the execution flow of conversations. The enforcement approach aims to contribute to a convergence from an unstable and totally unpredictable open multi-agent system under development to a dependable, more stable and less unpredictable one. Enforcement rules are specified as laws and norms that are associated with well-known consequences that may be subjected to any participant of the multi-agent system.

The use of an enforcement mechanism and an adapted dependability case approach, which we define as a Law Case, permits to control the failures and to promote the benefits and also contributes to tame the uncertainty presented by open multi-agent systems. A well-written Law Case will give all stakeholders (operating authority, members of staff and regulators) justifiable confidence that the system is reliable to operate and to continue in operation.

We will briefly introduce in the second section the XMLaw, which is a declarative language for implementing the law enforcement approach for regulating agents' interaction in an open multi-agent system that we are using. The third section describes the Law Cases and the fourth section presents a sample problem that details a Law Case. Finally, in the fifth section, we present the concludes and next steps.

2 Law-Governed Interaction

Law-governed architectures are designed to guarantee that the specifications of open systems will be obeyed. M-Law is an example of law-governed infrastructure provided to agent developers [17]. M-Law works by intercepting messages exchanged between agents, verifying the compliance of the messages with the laws and subsequently redirecting the message to the real addressee, if the laws allow it. If the message is not compliant, then the mediator blocks the message and applies the consequences specified in the law. This infrastructure, whenever necessary, can be extended to fulfill open system requirements or interoperability concerns. M-Law architecture is based on a pool of mediators that intercept messages and interpret the previously described laws (Figure 1).

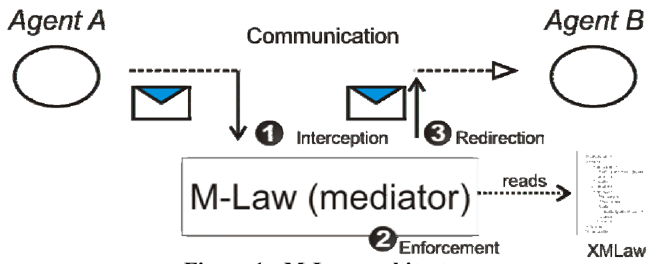


Figure 1 - M-Law architecture

M-Law was built to support law specification using XMLaw [4]. XMLaw is the description language used to configure the M-Law mediator by representing the interaction rules of an open system. These rules are interpreted by M-Law that analyzes the compliance of software agents with interaction laws at runtime [3]. XMLaw represents the structure and the relationships between important law elements (Figure 2). We selected some elements from XMLaw conceptual model to illustrate our proposal, and they will be explained below.

A law specification is a description of law elements. Law elements are interrelated in a way that it is possible to specify interaction protocols using time restrictions, norms, or even time sensitive norms.

The conceptual model uses the abstraction of *Scenes* to help to organize interactions. The idea of scenes is similar to theater plays, where actors play according to well defined scripts, and the whole play is composed of many scenes sequentially connected. *Scenes* are composed of *Protocols*, *Constraints*, *Clocks*, and *Norms*. It means that these four elements share a common interaction context through the scenes. Every *scene* specify one *protocol*. Because protocols define the interaction among the agents, different protocols should be specified in different scenes. *Scenes* also specify which agent role has permission to create scene instances.

Statically, an interaction protocol defines the set of *States* and T*ransitions* (activated by *Messages* or any other kind of event) allowed for agents in an open system. *Norms* are jointly used with the protocol specification, constraints and also temporal elements, to provide a dynamic configuration for the allowed behavior of agents in an open system.

Norms prescribe how the active distributed software agents ought to behave, and specify how they are permitted to behave and what their rights and duties are. The mediator keeps information about the set of activated norms to verify the compliance of software agents, the set of deactivated norms and any other data regarding system execution. There are three types of norms in XMLaw: obligations, permissions and prohibitions.

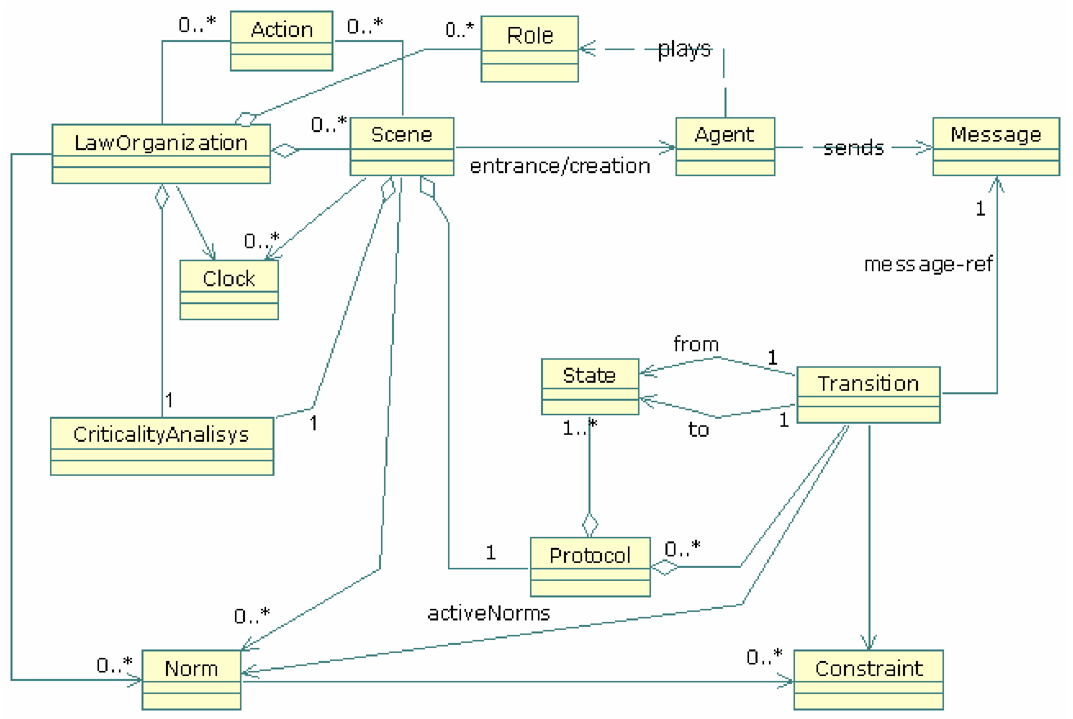


Figure 2 - XMLaw Conceptual Model

A *Norm* can be activated and deactivated by events. For example, an assembler will receive the permission upon logging in to the scene (scene activation event) and will lose the permission after issuing an order (event *orderTransition*). *Norms* also define the agent role that owns it through the attribute Assignee. Norm events and status (activated or deactivated) are referenced by other elements. For instance, as a consequence of the relationship between norms and transitions, it is possible to specify which norms must be activated or deactivated for firing a transition, i.e., a transition could only fire if the sender agent has a specific norm. Norms also have constraints and actions associated with them.

Constraints are restrictions over norms or transitions and generally specify filters for events, constraining the allowed values for a specific attribute of an event. For instance, messages carry information that is enforced in various ways. Constraints can be used for describing the allowed values for specific attributes. Constraints are defined inside the Transition or Norm elements. Constraints are implemented using Java code. The Constraint element defines the class attribute that indicates the java class that implements the filter. This class is called when a transition or a norm is supposed to fire, and basically the constraint analyzes if the received values are valid. For instance, a constraint can verify if the date expressed in the message is valid; if it is not, the message will be blocked.

The *CriticalityAnalysis* element is responsible for determining the agent criticality variation in order to enable M-Law to monitor the events that increase or decrease the agent criticality. The agent criticality defines how important the agent is to the organization and consequently to the system. This element is important when it is necessary to activate the fault-tolerant mechanism that can be incorporated in M-Law depending on the system domain. Basically, it allows the law designer to specify the event responsible for the agent criticality variation, its event type, the role of the agent that will have its criticality updated and the value which is a weight for this variation.

Furthermore, laws may be time sensitive, e.g., although an element that is active at time t1, it might not be active at time t2 (t1 < t2). XMLaw provides the *Clock* element to take care of the timing aspect. Temporal clocks represent time restrictions or controls and they can be used to activate other law elements. Clocks indicate that a certain period has elapsed producing clock-tick events. Once activated, a clock can generate clock-tick events. Clocks are activated and deactivated by law elements. Both are referenced to other law elements.

3 The Law Case

We define a Law Case to be

> *a structured argument providing evidence that an open multi-agent system meets its specified dependability requirements through the rationale around the law elements derivation.*

The dependability requirements of a system include the dependability attributes of interest in the particular system (e.g., security, real-time performance), and the anticipated usage (how and where) of the system. Those requirements will derive the law requirements and, consequently, the law elements. An argument for a system that's being used in a computer on a desktop in an office probably won't suffice if that same system is embedded in a spacecraft in transit to Mars.

As a dependability case, the key to a law case is the structure of the argument and the evidence that supports the argument. The law case can be formal or informal depending upon the requirements, but it must be able to convince a skeptical reviewer of its validity. It becomes a key element in the documentation of the system.

Furthermore, with the growth of the data amount derived from an analysis process, the number of claims and arguments would be very large and it would become hard the process. To this end in Figure 3 we propose a set of graphical notation elements adapted from the dependability case original proposal.

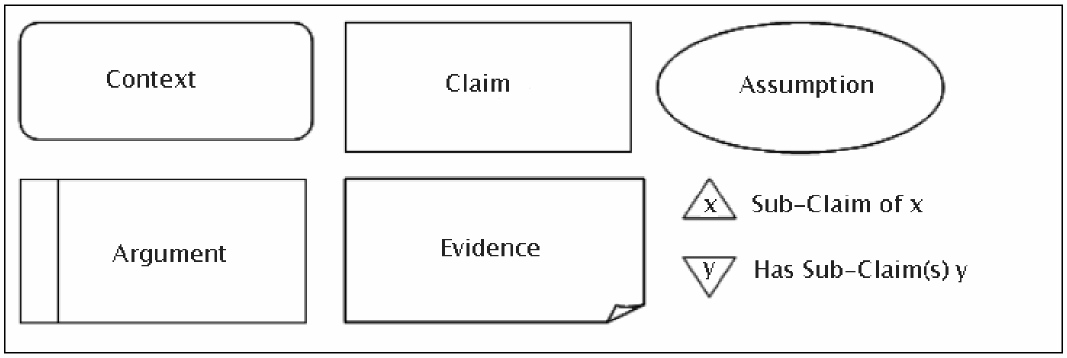


Figure 3 - Graphical Notation

Each of those graphical notation elements are structured in a conceptual model which is derived from the dependability cases original proposal application to the law-governance problems.

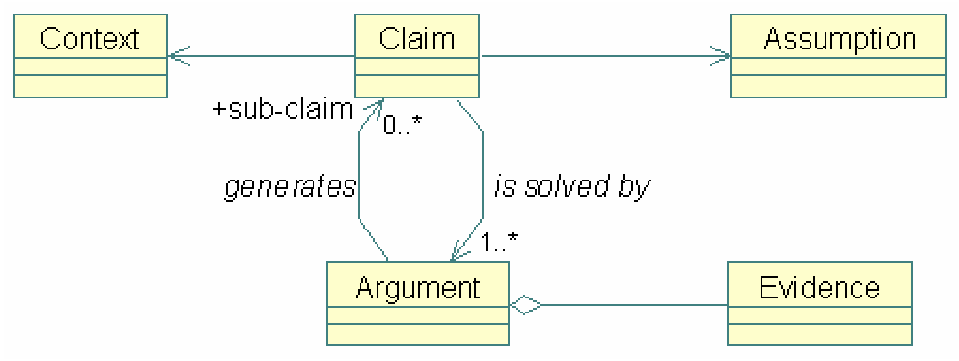


Figure 4 - Conceptual Model

From Figure 4 it is shown that a claim is based on assumptions and contexts, and it is solved by a set of arguments that are justified by evidences. Assumptions don't need to be proved, they are taken as true and helps the scenario characterization. The context represent the information that specialize the problem.

A rationale of a law case is made by making claims about a system and then showing evidence that those claims are valid. Eventually the argument gets down to ground truth. This ground truth might be a formal proof, a law of physics, or perhaps even an exhaustive enumeration of possibilities. Once every sub-claim is successfully driven down to its solution we have an argument that the original claim has been satisfied. This argument can be referred to whenever a question about the claim is raised. In particular, it can be used to identify potential problems when a change in the system is contemplated. Figure 5 describes the rationale development process of a Law Case using the conceptual model adapted. This structure allows a better comprehension and detailing of the proposal solution.

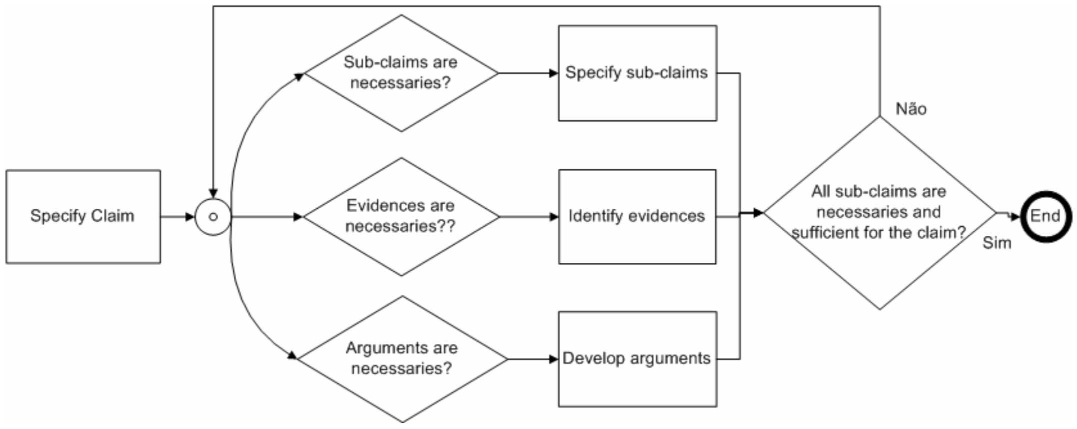

Specify Claim
Sub-claims are necessaries?
Evidences are necessaries??
Arguments are necessaries?
Specify sub-claims
Identify evidences
Develop arguments
All sub-claims are necessaries and sufficient for the claim?
Não
Sim
End

Figure 5 - Deriving the Law Case rationale

The evidence surrounding an argument and the ability to reason from it are key to making a credible and complete law case. Without evidence of a claim's correctness, there is no way to substantiate the claim. Unfortunately, evidence comes in many different forms, so it is impossible to dictate what kind of evidence or argument is appropriate for every situation.

3.1 Developing Law Cases

Before developing the law case itself it is necessary to identify it through a mechanism that ensures that the law cases necessaries to the application to be regulated were identified. And after that it is necessary to detail the law case through the mechanism proposed.

In order to identify the law cases, first the use cases and the requirements document of the system must be developed. After that, those functional requirements and non-functional requirements will guide the identification of the system threats. For each threat there would be a law case to be developed iteratively that counter the threats and protect the assets and services provided by the system (Figure 6). From this point, the law requirements derived from the law cases can be easily mapped to the XMLaw specification. Developing the threats and law cases iteratively allows the law designer to refine the law specification.

Thus, there would be at least one law case that mitigates a risk and protects a use case. To represent this situation and support the mechanism of deriving those law cases, we propose an extension of the use case diagram through the law case diagram, which is presented next section.

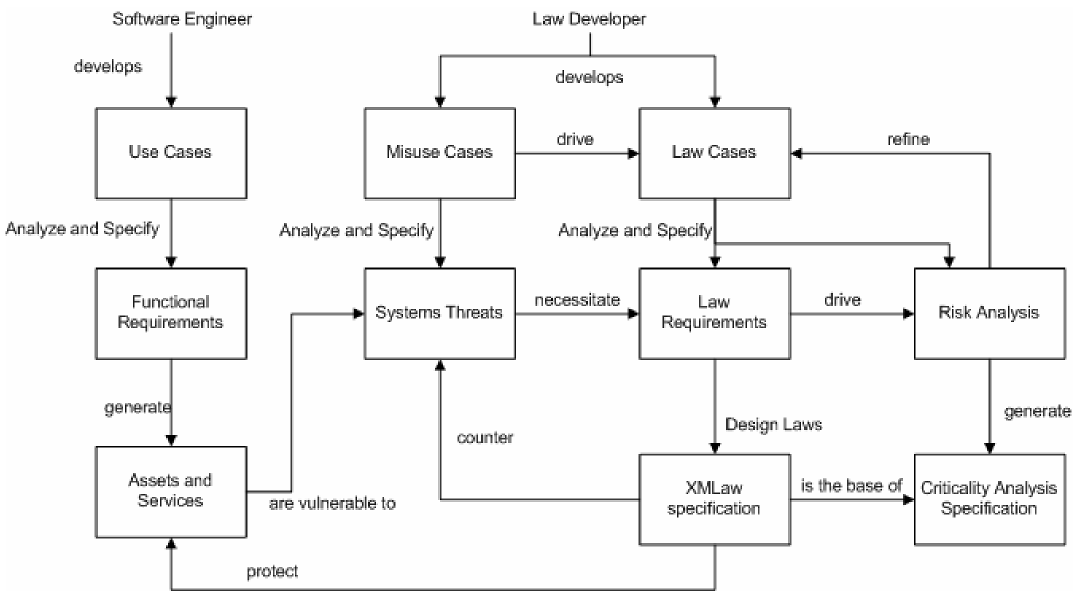


Figure 6 - Law Cases Development Process

The risk analysis phase exists in order to specify how probably it is to the risk associated to the threats arisen mitigated by the law case to occur, and the severity of the impact if it occurs. During this phase, the goal is to generate the criticality analysis specification responsible for the agents' criticality [9], since the risks would be mitigated by the criticality monitoring derived from the law case. For each event that might increase or decrease the agent criticality, it is calculated a value corresponding to this event and that contribute to this variation. This value is calculated multiplying the probability of the threat to occur and the severity caused by the impact if it occurs.

The probability is estimated as the US Department of Defense proposes in [10]. It can be classified as frequent (80% to 100%), probable (60% to 80%), occasional (40% to 60%), remote (20% to 40%), improbable (0% to 20%). The impact by it turns can be classified as catastrophic]0,75;1], critical]0,5;0,75], marginal]0,25;0,5], or negligible]0;0,25].

3.2 Law Cases Diagram

The law case diagram extends the use case diagram and it is based on the security use cases [12]. The law case diagram is composed by actors, use cases, law cases and risks based on threats.

The actors are the systems users and it includes human beings, software agents or other systems. The main goal of the law cases is to mitigate the risks, help implementing the dependability requirements and also law requirements, and allows that the use case threaten by the risk to be implemented successfully. The figure below shows the graphical notation of the law cases diagram elements:

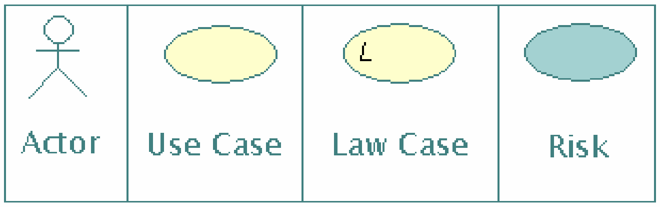


Figure 7 - Law Cases Diagram : Graphical Notation

3.3 Template for Law Cases

This sub-section presents the template to be used when developing a law case. The template contains the name of the law case, its author, the date of its creation or modification, the name of the use case to be protected, the name of the risk to be mitigated, the probability of its risk occurrence, the impact if the risk occurs, the pre-conditions, pos-conditions and the rationale based on dependability cases to derive the law requirements.

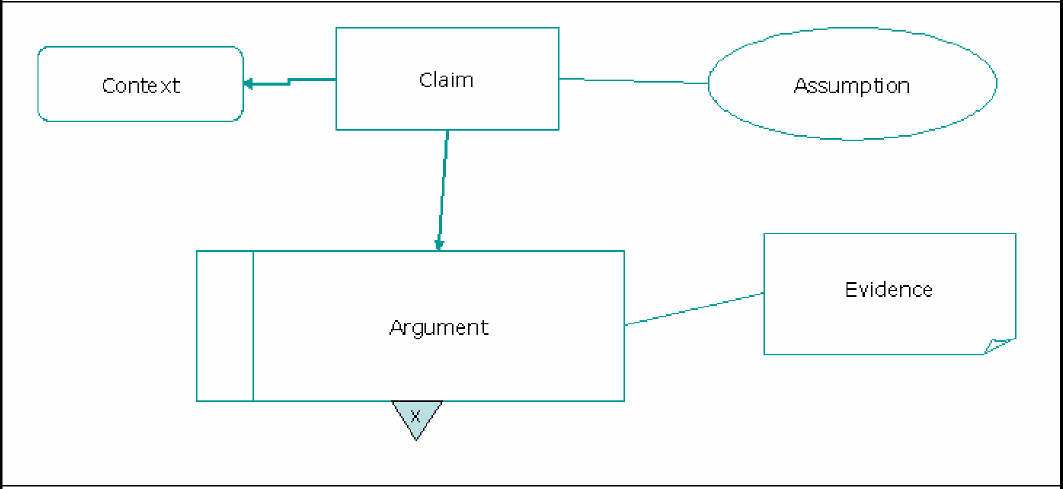
Law Case: <law_case_name>	
Author: <author_name>	**Date**: <creation_or_modification_date>
Use Case: <use_case_to_be_protected>	
Risk: <risk_to_be_mitigated>	
Probability: <occurrence_probability>	**Impact**: <impact_of_occurence>
Pre-conditions: <activation_pre-conditions>	
	
Pos-conditions: <pos-conditions_if_activated>	

Figure 8 - Law Case Template

4 Sample Problem

Perhaps the easiest way to describe Law Cases notation is through a detailed example. For this purpose we have chosen the Special System of Settlement and Custody called SELIC to exemplify the approach proposed. This system was chosen because of its unique characteristic of being an open governed distributed system regulated by a set of rules. Thus, it can be easily and directed mapped to an open law-governed multi-agent system. The SELIC works as a security's negotiation interactions mediator.

"SELIC is the central depository of securities issued by the National Treasury and the Central Bank of Brazil. The system also receives the negotiations records from the secondary market and promotes the respective settlement, counting with the modules which the National Treasure or Central Bank carries auctions out. Concerning to the negotiations, the system takes the purchase or sale commands in full or part, definitive or committed, by the necessaries proceedings to the financial movement and of custody related to the settlement of those operations, which are done one by one in real time. By the SELIC means, it is also carry the settlement of the open market operations and of the rediscount with public securities, as a consequence of the monetary politic direction" 1.

There are two types of purchase or sale defined in SELIC: definitive operation and committed operation. The committed operation is the operation of security's purchase or sale with the commitment of repurchase or resale. Between the committed operation defined by SELIC, there are three types of operations:

1. Committed Operation with Unitary Price Define: it is the purchase or sale operation with the commitment of repurchase or resale **with** a defined price of future settlement.
2. Committed Operation with Return Unitary Price Open: it is the purchase or sale operation with the commitment of repurchase or resale **without** a defined price of future settlement.
3. Anticipated Committed Operation: it is the purchase or sale committed operation which the repurchase or resale **anticipation** is partial or total.

The Unitary Price (UP) is the price of a security unit which is being negotiated. And a return operation is a repurchase or resale operation. The scenario presented in this work deals with the purchase and sale of public securities associated to a committed operation. A committed operation occurs when the financial institutions (like banks) take money for a day in order to do not cash out to zero or to negative. They usually do it with the borrowed money by other financial institution that, by its turn, demands public securities as guarantee. This kind of operation is called committed because it is done with the commitment of he security repurchase in the following day, at the moment that the money is back to the bank which lent it.

There are several requirements that rules the interaction on behalf of all institutions in a committed operation, as the several types of messages that could be sent and the several behavioral that should be implemented according to the messages specified, including norms and constrains.

As the goal of this work is not to develop the complete system with all its rules to all scenarios, but prove the statement concepts in the proposal solution, an interaction sample scenario was taken from the system. This scenario encloses all the law elements

1 Translated definition took from http://www.andima.com.br/selic/oquee.asp

necessaries to the concepts prove. This interaction sample scenario contains three entities: two financial institutions and the SELIC. Figure 9 shows this scenarios and below there is an example of interaction.

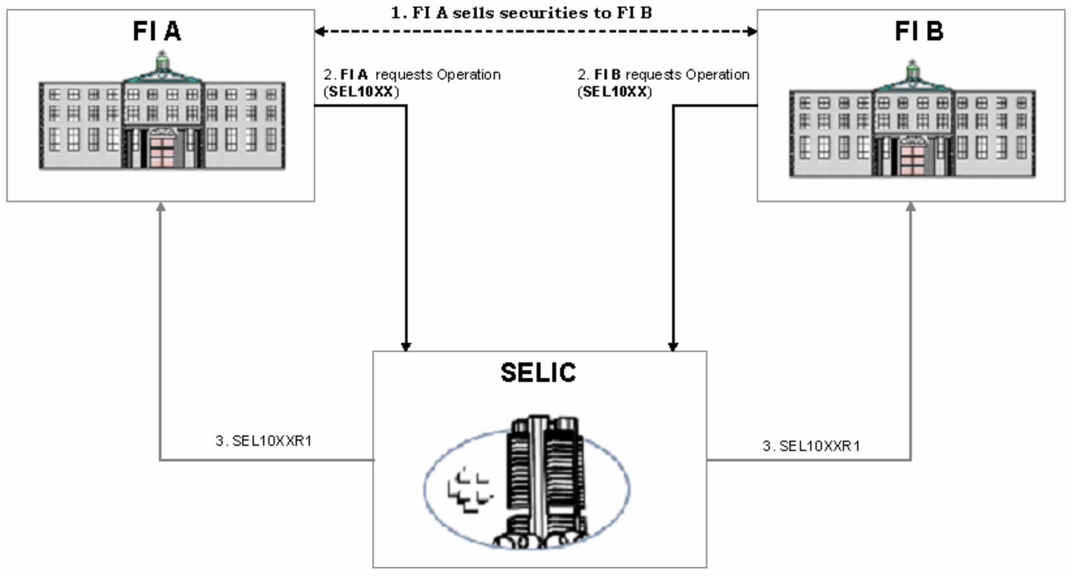

1. FI A sells securities to FI B
FI A
FI B
2. FI A requests Operation (SEL10XX)
2. FI B requests Operation (SEL10XX)
SELIC
3. SEL10XXR1
3. SEL10XXR1

Figure 9 - SELIC Example

The financial institution A (FI A) needs to sell securities to the financial institution B (FI B) and takes the commitment of repurchasing them in the following day. It works like if FI A was taken a loan from FI B for a day.

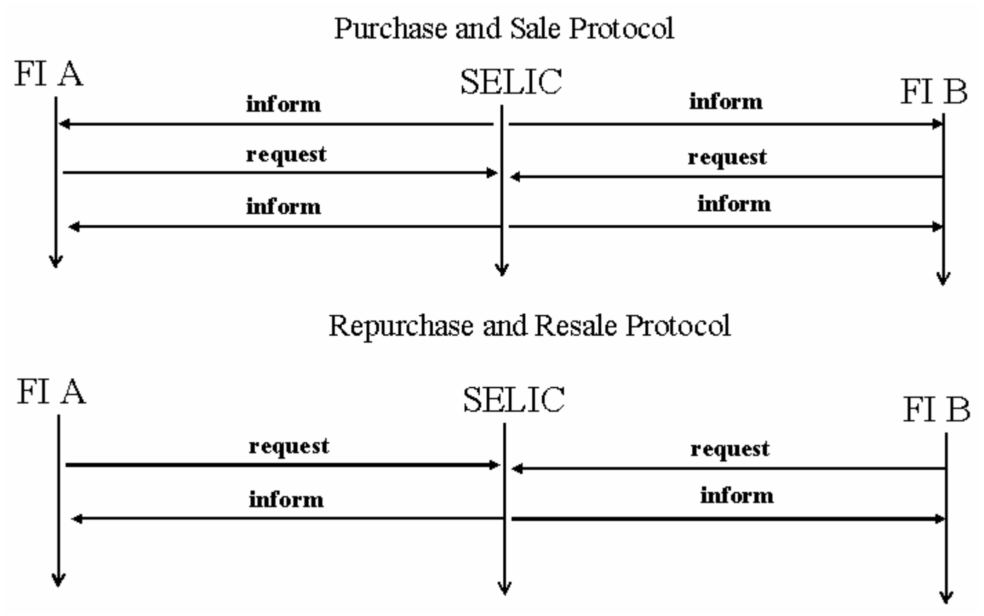


Figure 10 - SELIC: FIPA Protocol

The Figure 10 shows the scenario protocol according to the FIPA standard and can be describes from the following steps:

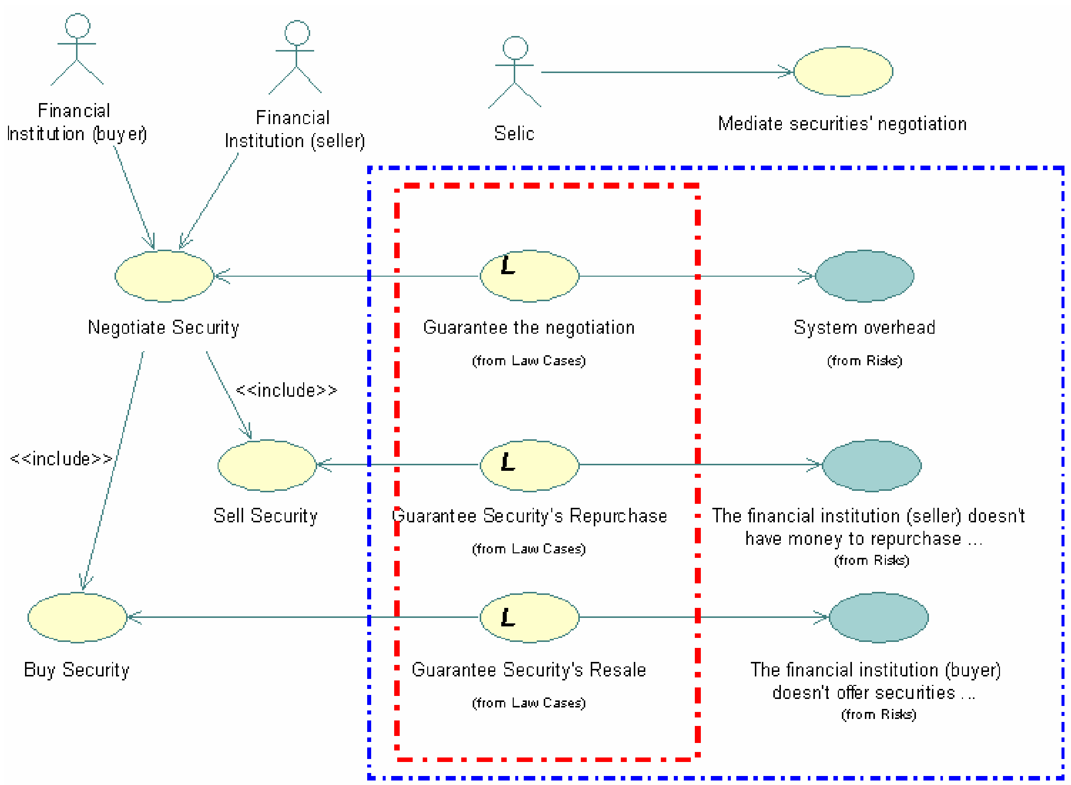
- The SELIC notifies the financial institutions that the operations are open for negotiations (*inform*);

- The FI A requests the securities' sale to SELIC (*request*);
- The FI B request the securities' purchase to SELIC (*request*);
- The SELIC updates the deposit account of both institutions and informs the operation status (*inform*);
- In the day after, the FI A requests the securities' purchase to SELIC (*request*);
- The FI B requests the securities' sale to SELIC (*request*);
- Once again, the SELIC updates the deposit account of both institutions and informs the operation status (*inform*).

While those steps are executed, some constraints are also executed. As it is a committed operation, when the securities are sold, the seller acquires the obligation of repurchasing the securities in the following day. A fine will be applied to the seller every day while it doesn't repurchase the securities. After 10 days without repurchasing the securities, the financial institution is prohibited of repurchasing them again. And, concerning to the buyer, it is obligated to resale the securities. While it doesn't resale the securities, the buyer will be fined daily. After 10 days, it will be prohibit to interacts in the system.

4.1 Law Cases Diagram and Law Cases Description

The law case diagram is composed of three actors: the buyer and the seller financial institutions, and the selic. Each actor represents a role in the XMLaw specification.



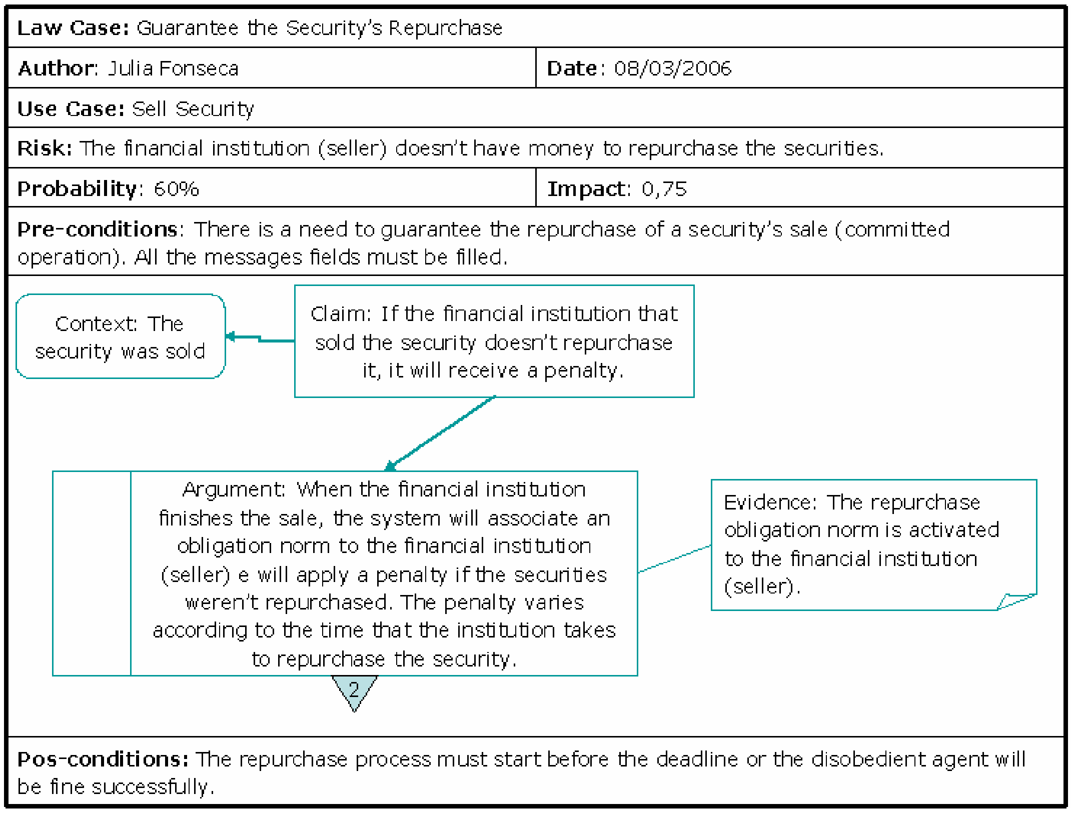
Financial Institution (buyer)
Financial Institution (seller)
Selic
Mediate securities' negotiation
Negotiate Security
L
Guarantee the negotiation
(from Law Cases)
System overhead
(from Risks)
<<include>>
<<include>>
Sell Security
Guarantee Security's Repurchase
(from Law Cases)
The financial institution (seller) doesn't have money to repurchase ...
(from Risks)
Buy Security
Guarantee Security's Resale
(from Law Cases)
The financial institution (buyer) doesn't offer securities ...
(from Risks)

Figure 11 - SELIC: Law Cases Diagram

The SELIC specification is detailed in the use case “Negotiate Security” that includes two use cases: “Buy Security” and “Sell Security” (according to the scenario described before). The use case “Buy Security” states the agent behavior that plays the buyer role. And the use case “Sell Security” states the agent behavior that plays the seller role (Figure 11).

According to the sample scenario requirements and threats, there are three macro risks that can be identified that (if succeeded) would prevent the use cases to achieve their goals: (i) system overhead that would increase the selic agent response time, (ii) the buyer financial institution, after a purchase operation, would not resale the securities, (iii) the seller financial institution would not have money to repurchase the securities that has just sold in a committed operation.
In order to mitigate those risks, three law cases were developed: “Guarantee the negotiation” that mitigates the risk (i), “Guarantee the Security’s Resale” that mitigates the risk (ii), and “Guarantee the Security’s Repurchase” that mitigate the risk (iii).

Figure 12 shows the law case “Guarantee the Security’s Repurchase”. The risk associated to this law case is related to the fact that the financial institution which plays the seller role may not have money to repurchase the securities that have sold. And this law case intends to guarantee that the securities will be repurchased during a committed operation.



Law Case: Guarantee the Security’s Repurchase	
Author: Julia Fonseca	**Date**: 08/03/2006
Use Case: Sell Security	
Risk: The financial institution (seller) doesn’t have money to repurchase the securities.	
Probability: 60%	**Impact**: 0,75
Pre-conditions: There is a need to guarantee the repurchase of a security’s sale (committed operation). All the messages fields must be filled.	
Pos-conditions: The repurchase process must start before the deadline or the disobedient agent will be fine successfully.	

Context: The security was sold

Claim: If the financial institution that sold the security doesn’t repurchase it, it will receive a penalty.

Argument: When the financial institution finishes the sale, the system will associate an obligation norm to the financial institution (seller) e will apply a penalty if the securities weren’t repurchased. The penalty varies according to the time that the institution takes to repurchase the security.

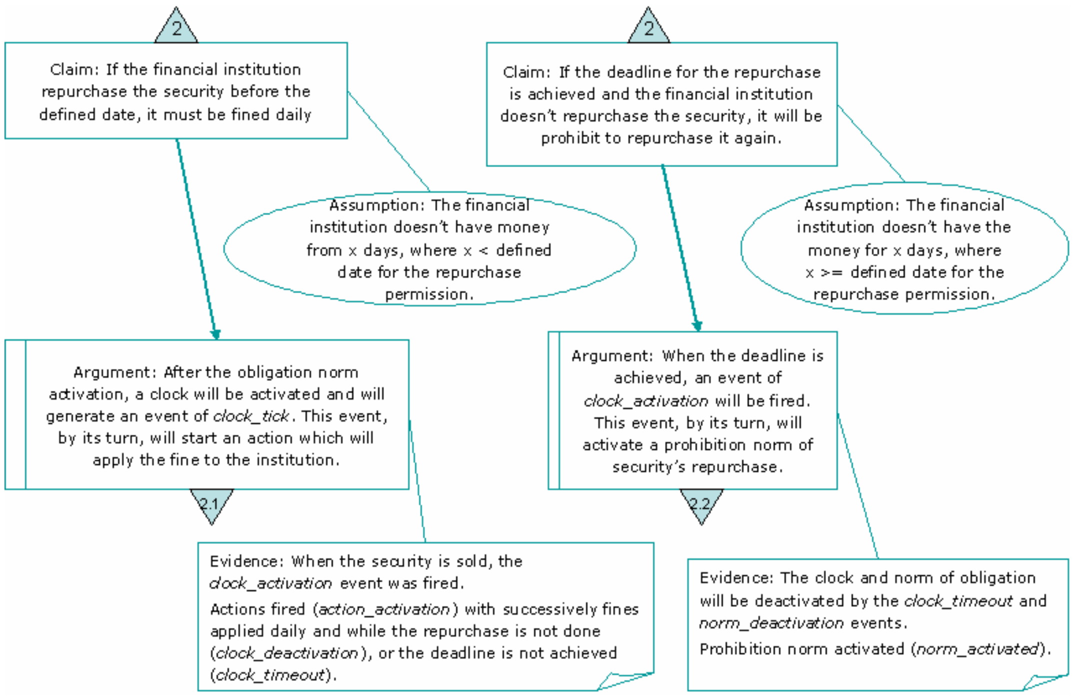
Evidence: The repurchase obligation norm is activated to the financial institution (seller).

Figure 12 - Law Case: Guarantee the Security's Repurchase

It is also necessary to remember that, when developing the law case, the rationale responsible for deriving the law requirements and, consequently, the law elements must be developed and will give the structure argument for the law case. From the three law cases defined, this one is a good sample example took from the scenario that shows the norm, clock, action and criticality analysis elements derivation and was chosen to be described in this paper.

The main claim of this law case is *"If the financial institution that sold the security doesn't repurchase it, it will receive a penalty"*. And the argument is based on the fact that the system will associate an obligation norm to the financial institution (seller) and will apply a penalty if the securities were not purchased. The penalty varies according to the time that the institution takes to repurchase the security. This argument has as an evidence the obligation norm activation to the financial institution (seller), and the two sub-claims (Figure 13). The two sub-claims considers two situations:

- If the financial institution repurchase the security before the limit established date, the institution must be fined daily.
- If the limit established date for the repurchase is achieved and the financial institution doesn't repurchase the security, the institution will be prohibit to repurchase the security again.


2
Claim: If the financial institution repurchase the security before the defined date, it must be fined daily
Assumption: The financial institution doesn't have money from x days, where x < defined date for the repurchase permission.
Argument: After the obligation norm activation, a clock will be activated and will generate an event of *clock_tick*. This event, by its turn, will start an action which will apply the fine to the institution.
2.1
Evidence: When the security is sold, the *clock_activation* event was fired.
Actions fired (*action_activation*) with successively fines applied daily and while the repurchase is not done (*clock_deactivation*), or the deadline is not achieved (*clock_timeout*).
2
Claim: If the deadline for the repurchase is achieved and the financial institution doesn't repurchase the security, it will be prohibit to repurchase it again.
Assumption: The financial institution doesn't have the money for x days, where x >= defined date for the repurchase permission.
Argument: When the deadline is achieved, an event of *clock_activation* will be fired. This event, by its turn, will activate a prohibition norm of security's repurchase.
2.2
Evidence: The clock and norm of obligation will be deactivated by the *clock_timeout* and *norm_deactivation* events.
Prohibition norm activated (*norm_activated*).
Figure 13 - Law Case (cont.): Guarantee the Security's Repurchase

The first sub-claim suposes that the financial institution does not have money, however the institution may have it before the limit established date. The argument assures that there will be the application of a penalty and that, after the norm activation event, a clock will be activated and will generate a *clock_tick* event which by its turn will start an action that is going to fine the institution.

On the other hand, the second sub-claim suposes that the financial institution wouldn't have money until the limit established date. The argument guarantee that there will be also the application of a penalty for this situation. The penalty is that financial institution will not be able to repurchase the securities anymore after the limit established date. The clock deactivation event will activate a prohibition norm that assures it.

Sub-claims were defined for each argument of the first sub-claims and can be accessed from the 2.1 and 2.2 references element. They were defined in order to derive the criticality monitoring specification of the envolved agents with the norm and clock activation event by the law case.

The Figure 14 details the rationale of the sub-claim derived from the 2.1 argument. The claim is that each norm and clock activated increase the financial institution that sold the securities at the moment that the purchase and sale operation is completed. And the argument states that the scene criticality monitoring module will detect the norm activation event to the seller financial institution agent and will recalculate its criticality according to the weight value resultant from the multiplication of the probability of the event occurrence (20%) by the impact if the event occurs (0,1) specified in the main document of the law case (Figure 12). And the same process will happen when the clock activation event occurs.

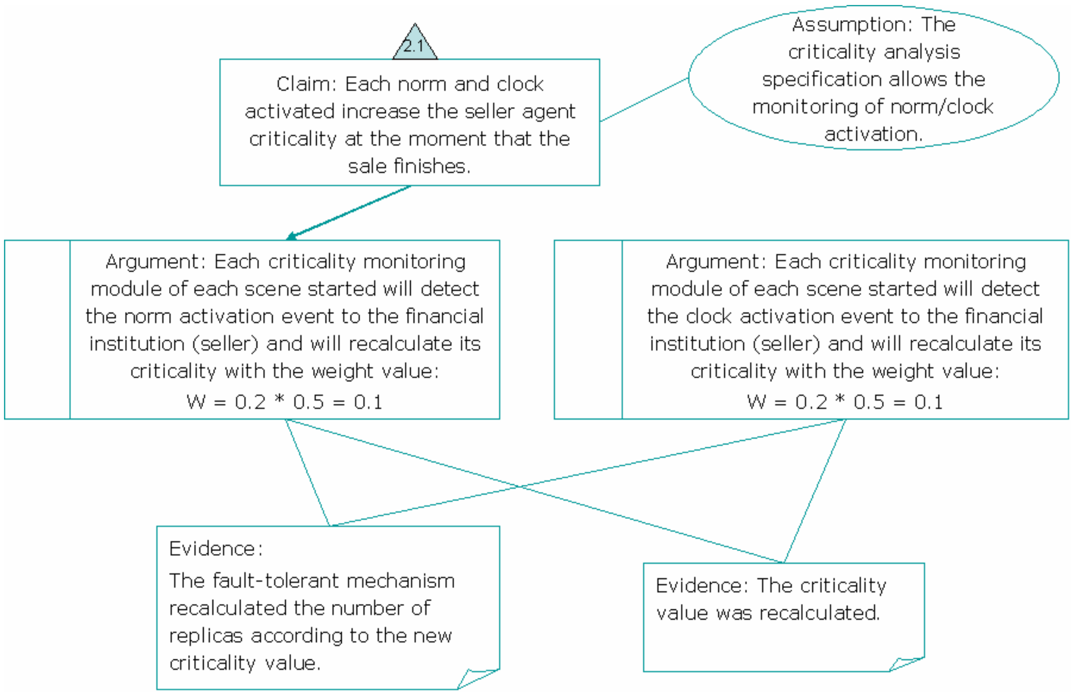

2.1
Claim: Each norm and clock activated increase the seller agent criticality at the moment that the sale finishes.
Assumption: The criticality analysis specification allows the monitoring of norm/clock activation.
Argument: Each criticality monitoring module of each scene started will detect the norm activation event to the financial institution (seller) and will recalculate its criticality with the weight value: W = 0.2 * 0.5 = 0.1
Argument: Each criticality monitoring module of each scene started will detect the clock activation event to the financial institution (seller) and will recalculate its criticality with the weight value: W = 0.2 * 0.5 = 0.1
Evidence: The fault-tolerant mechanism recalculated the number of replicas according to the new criticality value.
Evidence: The criticality value was recalculated.

Figure 14 - Law Case (cont.): Guarantee the Security's Repurchase

As a result of the law cases detailing, the sample example ilustraded is composed by two scenes: one defines the interaction protocol for the process of a committed security's purchase and sale operation with open unitary price, and the other one defines the interaction protocol for the process of security's repurchase and resale related to the committed operation. Below we present the partial law specification derived from the law cases. The full specification is presented in the attechement A of this paper.

```
<CriticalityAnalysis>
        <Increases>
            <Increase event-id="norm_obligation_of_buying_again"
            event-type="norm_activation" value="0.1">
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Increase>
            <Increase event-id="clock_obligation_buy_again" event-
         type="clock_activation" value="0.1">
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Increase>
            ….
        </Increases>

</CriticalityAnalysis>
```

Table 1 - XMLaw specification

5 Related Work

There are many software systems developed with a lack of attention to the several development levels details of a solution like data missing, not documented assumptions, tests absence, that turns it difficult to identify and solve errors.

Although use cases [14] are a popular modeling approach for engineering functional requirements, they are often misused when it comes to engineering dependability requirements because requirements engineers unnecessarily specify dependability architectural mechanisms instead of dependability requirements.

Dependability requirements should be based on services that work correctly no matter the threats that they might be threaten. In this context, there are three relevant approaches that deal with dependability requirements that are presented in the next sub-sections: dependability cases [5], security use cases [12], and risk analysis for dependability requirements concerning the law elements derivation [11].

5.1 Dependability Cases

A dependability case is a structured argument providing evidence that a system meets its specified dependability requirements. A dependability case is made by making claims about a system and then showing evidence that those claims are valid. Given a claim it is necessary to articulate a strategy to use to prove that the claim is true that might leads to sub-claims, at least one for each of the dependability requirements, with a strategy and sub-claims for each. Eventually the argument gets down to ground truth. This ground truth might be a formal proof, a law of physics, or set of test results.

Once every sub-claim is successfully driven down to its solution we have an argument that the original claim has been satisfied. This argument can be referred to whenever a question about the claim is raised. In particular, it can be used to identify potential problems when a change in the system is contemplated.

This is similar in spirit to the definition of a safety case [15]. The dependability case broadens the concept of a safety case to the whole milieu of dependability. Notice that if the only dependability attribute of interest is safety, then a dependability case becomes a safety case.

In particular, the development of a dependability case can help in deciding what kinds of tests are most important for confirming that critical areas of a system have been designed and coded correctly.

Comparing the dependability case approach with our approach we can point out two differences. First, a law case is not only the structured argument providing evidence, it is included in a process that gathers use cases and helps its derivation. And second, its conceptual model is an adaptation of the dependability cases conceptual model because it concerns the law requirement derivation support and proposes a well graphical notation in order to make the documentation easier to understand.

5.2 Security Use Cases

Security use cases should be used to specify requirements that the application shall successfully protect itself from its relevant security threats. A security use case is build from a set of misuses cases [13] of the application. The main different between both is that the security use case achieves its goal if the misuse case doesn't.

Basically, a security use case description contains the use case identification, the use case path, the security threats, the preconditions, the pos conditions and the flow of the misuser interactions interposed with the system interactions and system actions. This flow derives the security requirements.

The main difference from security use cases to law cases is the rationale around the decisions. A law case allows the structure argument providing evidence that a system meets its specified law requirements through a documented rationale and supports the law elements derivation, while a security use case only allows the security requirements derivation and doesn't have a structure document of proof.

5.3 Risk Analysis for Dependability Requirements

This approach proposed a method that considers risk analysis concepts to provide a structured way to specify, implement, monitor and maintain systems requirements and the existent challenges, opportunities, threats and limitations identified through the development of the solution.

Briefly, this analysis is based on the identification, control and assessment of relationships between cause and consequences states, events and their characteristics. With this approach, they proposed to evolve and develop a law enforcement infrastructure using risks as guidelines. The goal of this method is to provide a structured way to develop the law enforcement middleware, i.e., an infrastructure that contributes to the fulfillment of open multi-agent systems specifications and laws.

It would be a promising approach if there weren't a lack of a process to develop the rationale around the decisions and a lack of a process that integrates the functional requirements development through the non-functional and law requirements development, as the law cases approach proposed do.

6 Discussions

Law cases are the conjunction of the use case application and dependability cases and security use cases adaptation to the law requirements development. The goal of this method is to provide a process and structured way to develop and document the law enforcement middleware, i.e., an infrastructure that contributes to the fulfillment of open multi-agent systems specifications and laws.

The use of a law enforcement mechanism and of an adapted dependability case approach defined as a Law Case, permits to control the failures and to promote the benefits and also contributes to tame the uncertainty presented by open multi-agent systems. A well-written Law Case will give all stakeholders (operating authority, members of staff and regulators) justifiable confidence that the system is reliable to operate and to continue in operation.

As a future work we foresee the use of a CASE tool to support the development of the law cases. This tool would be also integrated to a (future) modeling tool of the law elements after their derivation which, by its turn, would generate the XMLaw specification automatically.

Bibliography

[1] Laprie, J. C., Arlat, J., Blanquart, J. P., Costes, A., Crouzert, Y., Deswarte, Y., Fabre, J. C., Guillermain, H., Kaâniche, M., Kanoun, K., Mazet, C., Powel, D., Rabéjac, C. and Thévenod, P.. Dependability Handbook (2nd edition) Cépaduès – Éditions, 1996. (ISBN 2-85428-341-4) (in French).

[2] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell: Dependability and its t-hreats - A taxonomy. IFIP Congress Topical Sessions 2004: 91-120.

[3] Paes, R.B; Lucena, C.J.P; Alencar, P.S.C. A Mechanism for Governing Agent Interaction in Open Multi-Agent Systems MCC nº 30/05, Depto de Informática, PUC-Rio, 31 p., 2005.

[4] Paes, R., Carvalho, G. R., Lucena, C.J.P., Alencar, P. S. C., Almeida, H.O.; and Silva, V. T.. Specifying Laws in Open Multi-Agent Systems. In: Agents, Norms and Institutions for Regulated Multi-agent Systems (ANIREM), AAMAS2005, 2005.

[5] Weinstock, C.B., Goodenough, J.B., Hudak, J.J., Dependability Cases, Technical Note, CMU/SEI-2004-TN-016, 2004.

[6] Agha, G.A. Abstracting Interaction Patterns: A Programming Paradigm for Open Distributed Systems, In (Eds) E. Najm and J.-B. Stefani, Formal Methods for Open Object-based Distributed Systems IFIP Transactions, Chapman & Hall, 1997.

[7] Fredriksson et al.. First international workshop on theory and practice of open computational systems. In Proceedings of twelfth international workshop on Enabling technologies: Infrastructure for collaborative enterprises (WETICE), Workshop on Theory and practice of open computational systems (TAPOCS), pp. 355 - 358, IEEE Press, 2003.

[8] Sommerville, I. Software Engineering. 7.ed. New York: Addison-Wesley, 2004. 759p.

[9] Gatti, M. A. C. ; Lucena, C.J.P. de ; Briot, J.-P.. On Fault Tolerance in Law-Governed Multi-Agent Systems. In: 5th International Workshop on Software Engineering for Large-scale Multi-Agent Systems, 2006, Shanghai. 28th International Conference on Software Engineering. New York, NY, USA : ACM Press, 2006. p. 21-27.

[10] MIL-STD-882C: "Standard Practice for System Safety Program Requirements", US Department of Defense, 1996.

[11] Carvalho, G.; Paes, R.; Choren, R.; Lucena, C.. Towards a Risk Driven Method for Developing Law Enforcement Middleware. Third International Workshop on Agent-Oriented Methodologies, OOPSLA 2004, Vancouver, British Columbia, Canadá, 24-28 Outubro, 2004.

[12] Firesmith, D.G.. Security Use Cases. Journal Of. Object Technology 2(3), May-June, 2003.

[13] Alexander, I.. Misuse cases: use cases with hostile intent. Software, IEEE, 20(1), Jan.-Feb. Page(s):58 – 66, 2003.

[14] Booch, J., Rumbaugh, I. J.,.The Unified Modeling Language User Guide; Addison Wesley, 1999.

[15] The Adelard Safety Case Development Manual – ASCAD, Adelard 2003.

[16] Pierce, R. H. and Baret, H.. Structuring a Safety Case for and Air Traffic Control Operations Room. Proc. Thirteenth Safety Critical System Symposium, Redmill and Anderson (Eds.). London: Springer Verlag, February, 2005.

[17] Lussier, B. et al. 3rd IARP-IEEE/RAS-EURON Joint Workshop on Technical Challenges for Dependable Robots in Human Environments, Manchester (GB), 7-9 Septembre 2004, 7p.

A
XMLaw Specification

```
<Laws>
    <LawOrganization id="bc" name="Banco Central">
        <!-- Role definition -->
        <Role id="IF" />
        <Role id="buyer" />
        <Role id="seller" />
        <Role id="selic" />

        <CriticalityAnalysis>
            <Weight ref="message" value="0"/>
            <Weight ref="transition" value="0"/>
            <Weight ref="role" value="0"/>
            <Increases>
                <Increase event-id="norm_obligation_of_buying_again"
                event-type="norm_activation" value="0.1">
                    <Assignee role-ref="seller" role-instance="$seller.instance"/>
                </Increase>
                <Increase event-id="clock_obligation_buy_again" event-
                  type="clock_activation" value="0.1">
                    <Assignee role-ref="seller" role-instance="$seller.instance"/>
                </Increase>

                <Increase event-id="norm_obligation_of_selling_again"
                 event-type="norm_activation" value="0.1">
                    <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                </Increase>
                <Increase event-id="clock_obligation_of_selling_again"
                 event-type="clock_activation" value="0.1">
                    <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                </Increase>
            </Increases>
            <Decreases>
                <Decrease event-id="norm_obligation_of_buying_again"
                event-type="norm_deactivation" value="0.1">
                    <Assignee role-ref="seller" role-instance="$seller.instance"/>
                </Decrease>
                <Decrease event-id="clock_obligation_buy_again"
                event-type="clock_deactivation" value="0.1">
                    <Assignee role-ref="seller" role-instance="$seller.instance"/>
                </Decrease>

                <Decrease event-id="norm_obligation_of_selling_again" event-
                    type="norm_deactivation" value="0.1">
                    <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                </Decrease>
                <Decrease event-id="clock_obligation_of_selling_again" event-
                    type="clock_deactivation" value="0.1">
                    <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                </Decrease>

                <Decrease event-id="norm_prohibition_of_buying_again" event-
                    type="norm_activation" value="0.1">
                    <Assignee role-ref="seller" role-instance="$seller.instance"/>
                </Decrease>
                <Decrease event-id="norm_prohibition_of_trading" event-
                    type="norm_activation" value="0.1">
                    <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                </Decrease>
            </Decreases>
        </CriticalityAnalysis>

        <Scene id="OpCompPUAberto" time-to-live="infinity">
            <CriticalityAnalysis>
                <Weight ref="role" value="0.1"/>
                <Weight ref="message" value="0"/>
                <Increases>
                    <Increase event-id="selic" event-type="role_activation"
                        value="0.45">
                        <Assignee role-ref="selic" role-instance="$selic.instance"/>
                    </Increase>
                    <Increase event-id="buyer" event-type="role_activation"
                    value="0.2">
                        <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                    </Increase>
                    <Increase event-id="seller" event-type="role_activation"
                    value="0.2">
                        <Assignee role-ref="seller" role-instance="$seller.instance"/>
                    </Increase>
                    <Increase event-id="SEL1054tp4Compra" event-
                    type="transition_activation" value="0.2">
                        <Assignee role-ref="selic" role-instance="$selic.instance"/>
                        <Assignee role-ref="buyer" role-instance="$buyer.instance"/>

                    </Increase>
                    <Increase event-id="SEL1054tp2Compra" event-
                    type="transition_activation" value="0.2">
                        <Assignee role-ref="selic" role-instance="$selic.instance"/>
                        <Assignee role-ref="buyer" role-instance="$buyer.instance"/>

                    </Increase>
                    <Increase event-id="SEL1054tp4Venda" event-
                    type="transition_activation" value="0.2">
                        <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="seller" role-instance="$seller.instance"/>

            </Increase>
            <Increase event-id="SEL1054tp2Venda" event-
            type="transition_activation" value="0.2">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="seller" role-instance="$seller.instance"/>

            </Increase>
        </Increases>
        <Decreases>
            <Decrease event-id="selic" event-type="role_deactivation"
            value="0.45">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
            </Decrease>
            <Decrease event-id="buyer" event-type="role_deactivation"
            value="0.2">
                <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
            </Decrease>
            <Decrease event-id="seller" event-type="role_deactivation"
            value="0.2">
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Decrease>
            <Decrease event-id="SEL1054tp2CompraConfirmada" event-
            type="transition_activation" value="0.2">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Decrease>
            <Decrease event-id="SEL1054tp4CompraConfirmada" event-
            type="transition_activation" value="0.2">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Decrease>
            <Decrease event-id="SEL1054tp2VendaConfirmada" event-
            type="transition_activation" value="0.2">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Decrease>
            <Decrease event-id="SEL1054tp4VendaConfirmada" event-
            type="transition_activation" value="0.2">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Decrease>
        </Decreases>
    </CriticalityAnalysis>
    <Creators>
        <Creator role_ref="selic"/>
    </Creators>
    <Protocol id="negotiation-protocol">
        <Messages>
         <Message id="startMsg" performative="inform">
                <Content>
                <Entry key="anuncio" value="Negociações abertas"/>
            </Content>
            <Sender role-ref="selic"/>
            <Receivers>
                <Receiver role-ref="buyer"/>
                <Receiver role-ref="seller"/>
            </Receivers>
        </Message>
        <Message id="SEL1054tp2CompraMsg" performative="request">
            <Content>
                <Entry key="CodMsg"        value="SEL1054"/>
                <Entry key="TpCompr"       value="02" />
                <Entry key="NOPRET"        value=" "/>
            </Content>
            <Sender role-ref="buyer"/>
            <Receivers>
                <Receiver role-ref="selic"/>
            </Receivers>
        </Message>
        <Message id="SEL1054tp4CompraMsg" performative="request">
            <Content>
                <Entry key="CodMsg"        value="SEL1054"/>
                <Entry key="TpCompr"       value="04" />
                <Entry key="NOPRET"        value=" "/>
            </Content>
            <Sender role-ref="buyer"/>
            <Receivers>
                <Receiver role-ref="selic"/>
            </Receivers>
        </Message>
        <Message id="SEL1054R1AguardandoVendaMsg" performative="inform">
            <Content>
                <Entry key="CodMsg"        value="SEL1054"/>
                <Entry key="NOPRET"        value="\d+"/>
                <Entry key="SitOpSEL" value="negociacao_aguardando_venda"/>
            </Content>
            <Sender role-ref="selic"/>
            <Receivers>
                <Receiver role-ref="buyer"/>
            </Receivers>
        </Message>
        <Message id="SEL1054R1CompraMsg" performative="inform">
            <Content>
                <Entry key="CodMsg"        value="SEL1054"/>
                <Entry key="NOPRET"        value="\d+"/>
            </Content>
            <Sender role-ref="selic"/>
            <Receivers>
                <Receiver role-ref="buyer"/>
            </Receivers>
        </Message>
        <Message id="SEL1054tp2VendaMsg" performative="request">
            <Content>
                <Entry key="CodMsg"        value="SEL1054"/>
                <Entry key="TpCompr"       value="02" />
                <Entry key="NOPRET"        value=" "/>
                </Content>
                <Sender role-ref="seller"/>
                <Receivers>
                    <Receiver role-ref="selic"/>
                </Receivers>
            </Message>
            <Message id="SEL1054tp4VendaMsg" performative="request">
                <Content>
                    <Entry key="CodMsg"        value="SEL1054"/>
                    <Entry key="TpCompr"       value="04" />
                    <Entry key="NOPRET"        value=" "/>
                </Content>
                <Sender role-ref="seller"/>
                <Receivers>
                    <Receiver role-ref="selic"/>
                </Receivers>
            </Message>
            <Message id="SEL1054R1AguardandoCompraMsg"
            performative="inform">
                <Content>
                    <Entry key="CodMsg"        value="SEL1054"/>
                    <Entry key="NOPRET"        value="\d+"/>
                    <Entry key="SitOpSEL"
                    value="negociacao_aguardando_compra"/>
                </Content>
                <Sender role-ref="selic"/>
                <Receivers>
                    <Receiver role-ref="seller"/>
                </Receivers>
            </Message>
            <Message id="SEL1054R1VendaMsg" performative="inform">
                <Content>
                    <Entry key="CodMsg"        value="SEL1054"/>
                    <Entry key="NOPRET"        value="\d+"/>
                </Content>
                <Sender role-ref="selic"/>
                <Receivers>
                    <Receiver role-ref="seller"/>
                </Receivers>
            </Message>
        </Messages>
         <States>
            <State id="negociacaoFechada" type="initial" label="Estado Inicial" />
            <State id="nenhumaSolicitacao" type="execution"     label="Negociação
            abertas" />
            <State id="compraSolicitada" type="execution" label="Pedido de Compra
            Solicitado" />
            <State id="vendaSolicitada" type="execution" label="Pedido de Venda Soli
            citado" />
            <State id="negociacaoExecutada" type="execution" label="Negociação Exe
            cutada (match de compra e venda)" />
            <State id="negociacaoFinalizada1" type="execution"
            label="Negociacao Concluida (1/2)" />
            <State id="negociacaoFinalizada2" type="success"
            label="Negociacao Concluida (2/2)" />
         </States>
```

```
<Transitions>
    <Transition id="start" from="negociacaoFechada"
    to="nenhumaSolicitacao" ref="startMsg" event-
    type="message_arrival" />
    <Transition id="SEL1054tp4Compra" from="nenhumaSolicitacao"
    to="compraSolicitada" ref="SEL1054tp4CompraMsg"
    event-type="message_arrival">
        <Constraint class="br.pucrio.inf.les.law.application.selic.
        constraint.ConditionNOPRET"
        semantics="NOPRETVazio" />
    </Transition>
    <Transition id="SEL1054tp2Compra"  from="nenhumaSolicitacao"
    to="compraSolicitada" ref="SEL1054tp2CompraMsg"
    event-type="message_arrival">
        <Constraint class="br.pucrio.inf.les.law.application.selic.
        constraint.ConditionNOPRET"
        semantics="NOPRETVazio" />
    </Transition>
    <Transition id="SEL1054tp4Venda" from="nenhumaSolicitacao"
    to="vendaSolicitada" ref="SEL1054tp4VendaMsg" event-
    type="message_arrival">
        <Constraint
            class="br.pucrio.inf.les.law.application.selic.constraint.
        ConditionNOPRET" semantics="NOPRETVazio" />
    </Transition>
    <Transition id="SEL1054tp2Venda" from="nenhumaSolicitacao"
    to="vendaSolicitada" ref="SEL1054tp2VendaMsg" event-
    type="message_arrival">
        <Constraint class="br.pucrio.inf.les.law.application.selic.
        constraint.ConditionNOPRET" semantics="NOPRETVazio"
        />
    </Transition>
    <Transition id="SEL1054R1AguardandoCompra1"
        from="vendaSolicitada" to="vendaSolicitada"
        ref="SEL1054R1AguardandoCompraMsg" event-
        type="message_arrival" />
    <Transition id="SEL1054R1AguardandoVenda1"
        from="compraSolicitada" to="compraSolicitada"
        ref="SEL1054R1AguardandoVendaMsg" event-
        type="message_arrival" />
    <Transition id="SEL1054tp2CompraConfirmada"
        from="vendaSolicitada" to="negociacaoExecutada"
        ref="SEL1054tp2CompraMsg" event-type="message_arrival" />
    <Transition id="SEL1054tp4CompraConfirmada"
        from="vendaSolicitada" to="negociacaoExecutada"
        ref="SEL1054tp4CompraMsg" event-type="message_arrival" />
    <Transition id="SEL1054tp2VendaConfirmada"
        from="compraSolicitada" to="negociacaoExecutada"
        ref="SEL1054tp2VendaMsg" event-type="message_arrival" />
    <Transition id="SEL1054tp4VendaConfirmada"
        from="compraSolicitada" to="negociacaoExecutada"
        ref="SEL1054tp4VendaMsg" event-type="message_arrival" />
    <Transition id="SEL1054R1AguardandoCompra2"
        from="negociacaoExecutada" to="negociacaoExecutada"
        ref="SEL1054R1AguardandoCompraMsg" event-
        type="message_arrival" />
```

```
            <Transition id="SEL1054R1AguardandoVenda2"
                from="negociacaoExecutada" to="negociacaoExecutada"
                ref="SEL1054R1AguardandoVendaMsg" event-
                type="message_arrival" />
            <Transition id="SEL1054R1CompraFinalizada"
                from="negociacaoExecutada" to="negociacaoFinalizada1"
                ref="SEL1054R1CompraMsg" event-type="message_arrival" />
            <Transition id="SEL1054R1VendaFinalizada"
                from="negociacaoFinalizada1" to="negociacaoFinalizada2"
                ref="SEL1054R1VendaMsg" event-type="message_arrival" />
    </Transitions>
    </Protocol>

    <Entrance>
        <Participant role_ref="selic" limit="1">
            <State ref="negociacaoFechada"/>
        </Participant>
        <Participant role_ref="buyer" limit="1">
            <State ref="nenhumaSolicitacao"/>
            <State ref="vendaSolicitada"/>
        </Participant>
        <Participant role_ref="seller" limit="1">
            <State ref="nenhumaSolicitacao"/>
            <State ref="compraSolicitada"/>
        </Participant>
    </Entrance>

</Scene>

<Scene id="OpRecompPUAberto" time-to-live="infinity">
    <CriticalityAnalysis>
        <Weight ref="role" value="0.1"/>
        <Weight ref="message" value="0"/>
        <Increases>
            <Increase event-id="selic" event-type="role_activation"
            value="0.45">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
            </Increase>
            <Increase event-id="buyer" event-type="role_activation"  value="0.2">
                <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
            </Increase>
            <Increase event-id="seller" event-type="role_activation"
                value="0.2">
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Increase>
            <Increase event-id="SEL1056tp1Recompra" event-
            type="transition_activation" value="0.2">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
            </Increase>
            <Increase event-id="SEL1056tp3Recompra" event-
                type="transition_activation" value="0.2">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
            </Increase>
            <Increase event-id="SEL1056tp1Revenda" event-
            type="transition_activation" value="0.2">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Increase>
            <Increase event-id="SEL1056tp3Revenda" event-
            type="transition_activation" value="0.2">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Increase>
        </Increases>
        <Decreases>
            <Decrease event-id="selic" event-type="role_deactivation"
                    value="0.45">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
            </Decrease>
            <Decrease event-id="buyer" event-type="role_deactivation"
                value="0.2">
                <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
            </Decrease>
            <Decrease event-id="seller" event-type="role_deactivation"
                value="0.2">
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Decrease>
            <Decrease event-id="SEL1056tp1RecompraConfirmada" event-
                    type="transition_activation" value="0.2">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Decrease>
            <Decrease event-id="SEL1056tp3RecompraConfirmada" event-
                    type="transition_activation" value="0.2">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Decrease>
            <Decrease event-id="SEL1056tp1RevendaConfirmada" event-
                        type="transition_activation" value="0.2">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Decrease>
            <Decrease event-id="SEL1056tp3RevendaConfirmada" event-
                        type="transition_activation" value="0.2">
                <Assignee role-ref="selic" role-instance="$selic.instance"/>
                <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
                <Assignee role-ref="seller" role-instance="$seller.instance"/>
            </Decrease>
        </Decreases>
    </CriticalityAnalysis>

    <Creators>
        <Creator role_ref="selic"/>
    </Creators>

    <Protocol id="renegotiation-protocol">
        <Messages>
```

```
<Message id="SEL1056tp3RecompraMsg"
    performative="request">
    <Content>
        <Entry key="CodMsg"          value="SEL1056"/>
        <Entry key="TpCompr"         value="03" />
        <Entry key="NOPRET"          value="\d+"/>
    </Content>
    <Sender role-ref="buyer"/>
    <Receivers>
        <Receiver role-ref="selic"/>
    </Receivers>
</Message>
<Message id="SEL1056tp1RecompraMsg"
    performative="request">
    <Content>
        <Entry key="CodMsg"          value="SEL1056"/>
        <Entry key="TpCompr"         value="01" />
        <Entry key="NOPRET"          value="\d+"/>

    </Content>
    <Sender role-ref="buyer"/>
    <Receivers>
        <Receiver role-ref="selic"/>
    </Receivers>
</Message>
<Message id="SEL1056R1AguardandoRevendaMsg"
    performative="inform">
    <Content>
        <Entry key="CodMsg"          value="SEL1056"/>
        <Entry key="NOPRET"          value="\d+"/>
        <Entry key="SitOpSEL"
        value="negociacao_aguardando_revenda"/>
    </Content>
    <Sender role-ref="selic"/>
    <Receivers>
        <Receiver role-ref="buyer"/>
    </Receivers>
</Message>
<Message id="SEL1056R1RecompraMsg"
performative="inform">
    <Content>
        <Entry key="CodMsg"          value="SEL1056"/>
        <Entry key="NOPRET"          value="\d+"/>
    </Content>
    <Sender role-ref="selic"/>
    <Receivers>
        <Receiver role-ref="buyer"/>
    </Receivers>
</Message>
<Message id="SEL1056tp1RevendaMsg"
performative="request">
    <Content>
        <Entry key="CodMsg"          value="SEL1056"/>
        <Entry key="TpCompr"         value="03" />
        <Entry key="NOPRET"          value="\d+"/>
    </Content>
```

```
                <Sender role-ref="seller"/>
                <Receivers>
                    <Receiver role-ref="selic"/>
                </Receivers>
            </Message>
            <Message id="SEL1056tp3RevendaMsg"
            performative="request">
                <Content>
                    <Entry key="CodMsg"          value="SEL1056"/>
                    <Entry key="TpCompr"         value="01" />
                    <Entry key="NOPRET"          value="\d+"/>

                </Content>
                <Sender role-ref="seller"/>
                <Receivers>
                    <Receiver role-ref="selic"/>
                </Receivers>
            </Message>
            <Message id="SEL1056R1AguardandoRecompraMsg"
            performative="inform">
                <Content>
                    <Entry key="CodMsg"          value="SEL1056"/>
                    <Entry key="NOPRET"          value="\d+"/>
                    <Entry key="SitOpSEL"
                     value="negociacao_aguardando_recompra"/>
                </Content>
                <Sender role-ref="selic"/>
                <Receivers>
                    <Receiver role-ref="seller"/>
                </Receivers>
            </Message>
            <Message id="SEL1056R1RevendaMsg" performative="inform">
                <Content>
                    <Entry key="CodMsg"          value="SEL1056"/>
                    <Entry key="NOPRET"          value="\d+"/>
                </Content>
                <Sender role-ref="selic"/>
                <Receivers>
                    <Receiver role-ref="seller"/>
                </Receivers>
            </Message>
        </Messages>
         <States>
            <State id="nenhumaResolicitacao" type="initial" label="Estado
            Inicial" />
            <State id="recompraSolicitada" type="execution" label="Pedido
            de Recompra Solicitado" />
            <State id="revendaSolicitada" type="execution" label="Pedido de
            Revenda Solicitada" />
            <State id="renegociacaoExecutada" type="execution"
            label="Renegociação Executada (match de recompra e revenda)"  />
            <State id="renegociacaoFinalizada1" type="execution"
            label="Renegociacao Concluida (1/2)" />
            <State id="renegociacaoFinalizada2" type="success"
            label="Renegociacao Concluida (2/2)" />
         </States>
        <Transitions>
            <Transition id="SEL1056tp1Recompra"
            from="nenhumaResolicitacao" to="recompraSolicitada"
            ref="SEL1056tp1RecompraMsg" event-type="message_arrival"/>
            <Transition id="SEL1056tp3Recompra"
            from="nenhumaResolicitacao" to="recompraSolicitada"
            ref="SEL1056tp3RecompraMsg" event-type="message_arrival" />
            <Transition id="SEL1056tp1Revenda"
            from="nenhumaResolicitacao" to="revendaSolicitada"
            ref="SEL1056tp1RevendaMsg" event-type="message_arrival" />
            <Transition id="SEL1056tp3Revenda"
            from="nenhumaResolicitacao" to="revendaSolicitada"
            ref="SEL1056tp3RevendaMsg" event-type="message_arrival" />

            <Transition id="SEL1056R1AguardandoRecompra1"
            from="revendaSolicitada" to="revendaSolicitada"
            ref="SEL1056R1AguardandoRecompraMsg" event-
            type="message_arrival" />
            <Transition id="SEL1056R1AguardandoRevenda1"
            from="recompraSolicitada" to="recompraSolicitada"
            ref="SEL1056R1AguardandoRevendaMsg" event-
            type="message_arrival" />

            <Transition id="SEL1056tp1RecompraConfirmada"
                from="revendaSolicitada" to="renegociacaoExecutada"
                ref="SEL1056tp1RecompraMsg" event-type="message_arrival" />
            <Transition id="SEL1056tp3RecompraConfirmada"
                from="revendaSolicitada" to="renegociacaoExecutada"
                ref="SEL1056tp3RecompraMsg" event-type="message_arrival" />
            <Transition id="SEL1056tp1RevendaConfirmada"
                from="recompraSolicitada" to="renegociacaoExecutada"
                ref="SEL1056tp1RevendaMsg" event-type="message_arrival" />
            <Transition id="SEL1056tp3RevendaConfirmada"
                from="recompraSolicitada" to="renegociacaoExecutada"
                ref="SEL1056tp3RevendaMsg" event-type="message_arrival" />
            <Transition id="SEL1056R1AguardandoRecompra2"
                from="renegociacaoExecutada" to="renegociacaoExecutada"
                ref="SEL1056R1AguardandoRecompraMsg" event-
                type="message_arrival" />
            <Transition id="SEL1056R1AguardandoRevenda2"
                from="renegociacaoExecutada" to="renegociacaoExecutada"
                ref="SEL1056R1AguardandoRevendaMsg" event-
                type="message_arrival" />
            <Transition id="SEL1056R1RecompraFinalizada"
                from="renegociacaoExecutada" to="renegociacaoFinalizada1"
                ref="SEL1056R1RecompraMsg" event-type="message_arrival" />
            <Transition id="SEL1056R1RevendaFinalizada"
                from="renegociacaoFinalizada1" to="renegociacaoFinalizada2"
                ref="SEL1056R1RevendaMsg" event-type="message_arrival" />
        </Transitions>
    </Protocol>
    <Entrance>
        <Participant role_ref="selic" limit="1">
            <State ref="nenhumaResolicitacao"/>
            <State ref="revendaSolicitada"/>
            <State ref="recompraSolicitada"/>
```

```
                    <State ref="renegociacaoExecutada"/>
               </Participant>
               <Participant role_ref="buyer" limit="1">
                    <State ref="nenhumaResolicitacao"/>
                    <State ref="revendaSolicitada"/>
               </Participant>
               <Participant role_ref="seller" limit="1">
                    <State ref="nenhumaResolicitacao"/>
                    <State ref="recompraSolicitada"/>
               </Participant>
          </Entrance>

     </Scene>
   <Action id="seller-has-been-fined" class="br.pucrio.inf.les.law.application.selic.action.
     FineAction">
      <Element ref="clock_obligation_buy_again" event-type="clock_tick"/>
   </Action>

   <!-- 1 dia = 24h * 60 min * 60 seg * 1000 mlseg = 86400000 -->
   <Clock id="clock_obligation_buy_again" type="periodic" tick-period="86400000">
     <Activations>
          <Element ref="norm_obligation_of_buying_again"
            event-type="norm_activation" />
     </Activations>
     <Deactivations>
          <Element ref="SEL1056tp1Recompra" event-type="transition_activation" />
          <Element ref="SEL1056tp3Recompra" event-type="transition_activation" />
          <Element ref="SEL1056tp1Revenda" event-type="transition_activation" />
          <Element ref="SEL1056tp3Revenda" event-type="transition_activation" />
     </Deactivations>
   </Clock>

     <!-- 10 dias = 10 * 24h * 60 min * 60 seg * 1000 mlseg = 864000000 -->
   <Clock id="clock_prohibition_of_buying_again" type="once" tick-period="864000000">
     <Activations>
          <Element ref="norm_obligation_of_buying_again"
            event-type="norm_activation" />
     </Activations>
     <Deactivations>
          <Element ref="SEL1056tp1Recompra" event-type="transition_activation" />
          <Element ref="SEL1056tp3Recompra" event-type="transition_activation" />
          <Element ref="SEL1056tp1Revenda" event-type="transition_activation" />
          <Element ref="SEL1056tp3Revenda" event-type="transition_activation" />
     </Deactivations>
   </Clock>
   <Norm type="obligation" id="norm_obligation_of_buying_again">
     <Assignee role-ref="seller" role-instance="$seller.instance"/>
     <Activations>
          <Element ref="SEL1054R1VendaFinalizada" event-type="transition_activation"/>
     </Activations>
     <Deactivations>
          <Element ref="SEL1056tp1Recompra" event-type="transition_activation" />
          <Element ref="SEL1056tp3Recompra" event-type="transition_activation" />
          <Element ref="SEL1056tp1RecompraConfirmada"
             event-type="transition_activation" />
          <Element ref="SEL1056tp3RecompraConfirmada"
          event-type="transition_activation" />
    </Deactivations>
  </Norm>
  <Norm type="prohibition" id="norm_prohibition_of_buying_again">
    <Assignee role-ref="seller" role-instance="$seller.instance"/>
    <Activations>
        <Element ref="clock_prohibition_of_buying_again"
          event-type="clock_timeout"/>
    </Activations>
  </Norm>

  <Action id="buyer-has-been-fined" class="br.pucrio.inf.les.law.application.selic.
    action.FineAction">
     <Element ref="clock_obligation_of_selling_again" event-type="clock_tick"/>
  </Action>
  <!-- 1 dia = 24h * 60 min * 60 seg * 1000 mlseg = 86400000 -->
  <Clock id="clock_obligation_of_selling_again" type="periodic"
   tick-period="86400000">
    <Activations>
        <Element ref="norm_obligation_of_selling_again"
         event-type="norm_activation" />
    </Activations>
    <Deactivations>
        <Element ref="SEL1056tp1Recompra" event-type="transition_activation" />
        <Element ref="SEL1056tp3Recompra" event-type="transition_activation" />
        <Element ref="SEL1056tp1Revenda" event-type="transition_activation" />
        <Element ref="SEL1056tp3Revenda" event-type="transition_activation" />
    </Deactivations>
  </Clock>

    <!-- 10 dias = 10 * 24h * 60 min * 60 seg * 1000 mlseg = 864000000 -->
   <Clock id="clock_prohibition_of_trading" type="once" tick-period="864000000">
    <Activations>
        <Element ref="norm_prohibition_of_trading" event-type="norm_activation" />
    </Activations>
    <Deactivations>
        <Element ref="SEL1056tp1Recompra" event-type="transition_activation" />
        <Element ref="SEL1056tp3Recompra" event-type="transition_activation" />
        <Element ref="SEL1056tp1Revenda" event-type="transition_activation" />
        <Element ref="SEL1056tp3Revenda" event-type="transition_activation" />
    </Deactivations>
  </Clock>
  <Norm type="obligation" id="norm_obligation_of_selling_again">
    <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
    <Activations>
        <Element ref="SEL1054R1VendaFinalizada" event-type="transition_activation"/>
    </Activations>
     <Deactivations>
        <Element ref="SEL1056tp1Recompra" event-type="transition_activation" />
        <Element ref="SEL1056tp3Recompra" event-type="transition_activation" />
        <Element ref="SEL1056tp1RecompraConfirmada"
            event-type="transition_activation" />
        <Element ref="SEL1056tp3RecompraConfirmada"
           event-type="transition_activation" />
    </Deactivations>
  </Norm>
```

```
        <Norm type="prohibition" id="norm_prohibition_of_trading">
          <Assignee role-ref="buyer" role-instance="$buyer.instance"/>
          <Activations>
               <Element ref="clock_prohibition_of_trading" event-type="clock_timeout"/>
          </Activations>
          <Deactivations>
               <Element ref="SEL1056tp1Revenda" event-type="transition_activation" />
               <Element ref="SEL1056tp3Revenda" event-type="transition_activation" />
               <Element ref="SEL1056tp1RevendaConfirmada"
                 event-type="transition_activation" />
               <Element ref="SEL1056tp3RevendaConfirmada"
                 event-type="transition_activation" />
          </Deactivations>
        </Norm>
     </LawOrganization>
</Laws>
```