

## 1. Objetivo

O modelo proposto tem como finalidade representar atividades onde um grupo de pessoas devem atuar de forma colaborativa com o propósito de resolver um problema. Esse problema pode ser dividido em etapas menores conhecidas como objetivos. Essas pessoas podem se relacionar entre si, podem se relacionar com os artefatos presentes no meio onde elas atuam. Os artefatos também possuem a capacidade de se relacionar. Cada objetivo é concluído apenas se um ou mais relacionamentos forem realizados. A conclusão do objetivo também é função de certos agentes e artefatos que devem ser presentes.

Outra finalidade do modelo consiste em representar condições que devem ser mantidas ao longo da atividade, se essas condições forem desfeitas - a atividade deve ser encerrada de imediato, caso contrário as pessoas envolvidas nesta manutenção estarão submetidas a risco de morte. Dentro desta abordagem, alguém designado para cumprir com alguma atividade pode cometer erros. Esse modelo tem como finalidade lidar com os seguintes tipos de erro: executar uma ação quando não há condições apropriadas para isso, manipular artefatos de maneira inapropriada ou inadvertida e escolher os artefatos inapropriados para cumprir com uma determinada atividade.

O modelo deve ser capaz de representar as consequências desses erros. Esse modelo está preocupado em representar dois tipos de consequências, essas são: 1 - imediatas que acontece sobre o indivíduo errante, 2 - a consequência manifesta em outro objetivo sobre o mesmo ou outro indivíduo pertencente ao grupo. Essas consequências são efeitos físicos negativos que alguém vêem a sofrer. A intensidade dessas consequências variam desde uma leve lesão a morte.

Outro aspecto deste modelo consiste representar objetivos cujo sucesso advem de certa característica aleatória presente na natureza da atividade. Essa característica aleatória consiste em um evento que possuem uma certa possibilidade de acontecer. Se esse evento acontecer, alguém sofre consequências ruins por conta disto. O modelo deve considerar relações entre erros onde as consequências se manifestam de forma indireta com eventos aleatórios.

## 2. Exemplo - Estudo de Caso

Sete profissionais de linha viva (profissionais que realizam manutenção em equipamentos elétricos energizados) são designados com o propósito de realizar a substituição de um isolador de pedestal. Os papéis desses profissionais são; 1 supervisor, 5 executores. A manutenção deve ser executada apenas sobre as seguintes condições: céu ensolarado e umidade relativa do ar menor que 70 por cento. Todos os profissionais devem possuir os EPIS necessários: capacete, óculos de sol, roupa isolante e anti-chamas, luvas isolantes e botas isolantes. Os profissionais que entrarão no potencial deverão estar vestidos de roupa condutiva e cabo guarda. As ferramentas necessárias para resolver esse problema são: bastão garra de diâmetro 64 x 3600 mm, sela de diâmetro 65 , colar, corda de fibra sintética, carretilha, chave com catraca, bastão universal, soquete adequado, localizador de pino, bastão com soquete multiangular. A substituição do isolador de pedestal pode ser escrita nos seguintes subobjetivos:

1. Limpar, secar e testar corda.
2. Instalar Bastão Garra na estrutura com o pedestal a ser substituído.

3. Instalar sela com colar na estrutura
4. Amarrar o bastão na parte superior da estrutura com a corda.
5. Amarrar o olhal do bastão ao cavalo da sela atrás de uma corda.
6. Instalar um segundo conjunto bastão e sela no lado oposto da estrutura.
7. Enforcar um estropo de Náilon no corpo do isolador.
8. Colocar a extremidade do estropo no gancho da corda de serviço.
9. Afrouxar os parafusos do conector que prendem a barra ao isolador.
10. Terminar de retirar os parafusos com o bastão com o soquete multiangular.
11. Elevar a barra através da corda que une a sela ao bastão.
12. Apertar o colar através da porca borboleta.
13. Segurar firmemente a corda de serviço.
14. Sacar parafusos da base da coluna.
15. Baixar o isolador ao solo
16. Içar o Isolador
17. Colocar Parafusos na base da coluna.
18. Baixar a barra para que a mesma apóie no novo isolador.
19. Colocar os parafusos do conector que prende a barra ao novo isolador.
20. Retirar Equipamentos

### 3. Modelo

#### 3.1. Definição dos Conjuntos

1.  $Entity = \{e_1, \dots, e_n\}$  - conjunto de todas as Entidades.
2.  $Agent = \{ag_1, \dots, ag_n\}$  - conjunto dos Agentes.
3.  $Artefact = \{at_1, \dots, at_n\}$  - conjunto dos Artefatos.
4.  $EntityGoal = eg = \{e_n, \dots, e_m\}$  - conjunto das Entidades que devem estar presentes para concluir um determinado objetivo  $g_i$ .
5.  $Relation = \{r_1, \dots, r_n\}$  - conjunto dos Relacionamentos.
6.  $RelationGoal = rg = \{r_n, \dots, r_m\}$  - conjunto dos Relacionamentos que devem estar presentes para concluir um único objetivo  $g_i$ .
7.  $Role = \{\rho_1, \dots, \rho_n\}$  - conjunto dos Papeis.
8.  $Goal = \{g_1, \dots, g_n\}$  - conjunto dos Objetivos.
9.  $GoalRandomic = \{gr_1, \dots, gr_n\}$  - conjunto dos Objetivos Randomicos.
10.  $GoalPrerequisite = gp = \{g_n, \dots, g_m\}$  - conjunto de Objetivos que são pré-requisitos para alcançar um outro objetivo.
11.  $Condition = \{c_1, \dots, c_n\}$  - conjunto das Condições que devem ser mantidas ao longo da execução de todos os objetivos.
12.  $ConditionToGoal = cg = \{c_n, \dots, c_m\}$  - conjunto de condições que devem ser mantidas para concluir um único objetivo  $g_i$ .
13.  $Risk = \{risk_1, \dots, risk_n\}$  - conjunto dos Riscos na ocorrência de Eventos Ruins.
14.  $Possibility = \{p_1, \dots, p_n\}$  - conjunto das possibilidades de Eventos Ruins.
15.  $Fatality = \{f_1, \dots, f_n\}$  - conjunto das fatalidades que acontecem na existência de um evento ruim.
16.  $agg = \{ag_n, \dots, ag_m\}$  - agentes que atingiram um determinado objetivo.
17.  $ago = \{ag_n, \dots, ag_m\}$  - agentes que atingiram um determinado objetivo e eram obrigados a isso.

### 3.2. Definição das Relações Entre os Conjuntos

1.  $Entity \equiv Agent \cup Artefact$
2.  $Agent$  e  $Artefact$  são disjuntos.
3.  $GoalRadomic \subset Goal$
4.  $\{agg_1, \dots, agg_n\} \subset Agent$
5.  $\{gp_1, \dots, gp_n\} \subset Goal$
6.  $\{cg_1, \dots, cg_n\} \subset Condition$
7.  $\{eg_1, \dots, eg_n\} \subset Entity$
8.  $\{rg_1, \dots, rg_n\} \subset Relation$

### 3.3. Definição dos Predicados

1.  $relationHas(r_i, e_i, e_k)$  onde  $i \neq j$  - Um determinado relacionamento  $r_i$  é composto por uma entidade  $e_i$  e  $e_k$  onde  $e_i$  não pode ser igual a  $e_j$ .
2.  $hasRole(ag_n, \rho_m)$  - Um determinado agente  $ag_n$  tem um determinado papel  $\rho_m$ .
3.  $hasObligation(\rho_m, g_j)$  - Quem assume o papel  $\rho_m$  é obrigado a concluir o objetivo  $g_j$ .
4.  $hasPermission(\rho_m, g_j)$  - Quem assume o papel  $\rho_m$  tem a permissão de concluir o objetivo  $g_j$ .
5.  $isReached(g_k)$  - O objetivo  $g_k$  foi alcançado.
6.  $stopIn(g_n, agg_m)$  - O objetivo não foi encerrado para todos os agentes que tiveram de executar.
7.  $stopIn(g_n)$  - A atividade como um todo teve de ser finalizada em  $g_n$ .
8.  $isPreRequisite(gp_i, g_j)$  - Os objetivos  $g$  pertencentes ao grupo  $gp_i$  devem ser concluídos para que haja condição de executar  $g$ .
9.  $hasCondition(g_i, cg_n)$  - Um objetivo do tipo  $g_i$  possui certas condições  $c$  que deve estar presentes e devem se manter durante toda execução deste objetivo. Essas condições  $c$  devem estar conditdas em  $cg_n$ .
10.  $hasEntity(g_i, eg_m)$  - Um objetivo  $g_i$  tem um conjunto de entidades  $eg_m$  onde todas as entidades presentes neste conjunto devem estar presentes no momento da execução desse objetivo.
11.  $hasRelation(g_i, rg_n)$  - Um objetivo  $g_i$  tem um conjunto de relacionamentos  $rg_n$  onde todos esses relacionamentos devem ser feito para que este objetivo seja concluído.
12.  $isPresent(X)$ ,  $X = cg_n, c_k, rg_k, r_k, eg_k, e_k$  - Define se  $X$  está presente no instante em análise, sendo que  $X$  pode ser  $cg_n, c_k, rg_k, r_k, eg_k, e_k$ .
13.  $tryReach(ag_i, g_j)$  - Um determinado agente  $ag_i$  tenta alcançar o objetivo  $g_j$ . Para o agente tentar alcançar um dado objetivo, o papel dele ao menos deve ter permissão para isso.
14.  $violationCondition(ag_i, g_j, c_k)$  - Um determinado agente  $ag_i$  comete uma violação de condição no objetivo  $g_j$  sobre a condição  $c_k$ .
15.  $violationRelation(ag_i, g_j, r_k)$  - O agente  $ag_i$  comete uma violação de Relacionamento no objetivo  $g_j$  por não realizar o relacionamento  $r_k$ .
16.  $violationEntity(ag_i, g_j, e_k)$  - O agente  $ag_i$  comete uma violação de Entidade no objetivo  $g_j$  por tentar alcançar esse objetivo sem ter a entidade  $e_k$  presente.
17.  $hasRisk(c_k, risk_j, f_m)$  - A condição  $c_k$  está associada a um risco  $risk_k$  com uma certa fatalidade  $f_m$ .

18.  $hasRisk(r_k, risk_j, f_m)$  - O relacionamento  $r_k$  está associado a um risco  $risk_k$  com uma certa fatalidade  $f_m$ .
19.  $consequenceOfBadEvent(g_k, ag_i, risk_j, f_m)$  - Agente  $ag$  sofre as consequências do risco  $risk_j$  com a fatalidade  $f_m$
20.  $hasPossibility(r_l, p_m)$  - Possibilidade  $p_m$  do evento  $r_l$  gerar alguma consequência ruim. A possibilidade  $p_m$  pode assumir apenas dois valores;  $true, false$ .
21.  $affects(r_k, r_n)$  - Se uma relação  $r_k$  não for feito, ou se essa relação for mal feita, então ela afeta negativamente alguma outra relação  $r_n$  com a possibilidade de algo errado inicialmente dado por  $false$  ser mudado para  $true$ .
22.  $happensBadEvent(gr_n, r_m, ag_k)$  - O evento ruim de  $g_r$  acontece em relação a um relacionamento necessário sobre um agente  $ag_k$ .

### 3.4. Definição das Relações de Implicabilidade

Todo agente que é obrigado a alcançar um determinado objetivo deve ter a permissão para realizar essa ação.

$$hasObligation(\rho_m, g_j) \rightarrow hasPermission(\rho_m, g_j) \quad (1)$$

Um agente que tenta alcançar um objetivo sem que pelo menos uma das condições necessárias para isso esteja presente, comete uma violação conhecida como violação de condição.

$$\begin{aligned} hasCondition(g_i, cg_n) \wedge \neg isPresent(c_k) \wedge (c_k \in cg_n) \wedge tryReach(ag_m, g_i) \rightarrow \\ violationCondition(ag_m, g_j, c_k) \wedge \\ hasCondition(g_i, cg_n) \wedge isPresent(cg_n) \wedge tryReach(ag_m, g_i) \rightarrow \\ \neg violationCondition(ag_m, g_j, c_k) \end{aligned} \quad (2)$$

Se um agente não executa, ou não tem condições de executar, um dado relacionamento  $r_k$  necessário para concluir o objetivo, então o agente cometeu uma violação de relacionamento. Contudo, se o agente alcançar o objetivo, então a não violação acontece.

$$\begin{aligned} hasRelation(g_i, rg_n) \wedge \neg isPresent(r_k) \\ \wedge (r_k \in rg_n) \wedge tryReach(ag_m, g_i) \rightarrow violationRelation(ag_m, g_i, r_k) \\ \wedge \\ hasRelation(g_i, rg_n) \wedge isPresent(rg_n) \\ \wedge tryReach(ag_m, g_i) \rightarrow \neg violationRelation(ag_m, g_i, r_k) \end{aligned} \quad (3)$$

Se ao tentar alcançar um determinado objetivo pelo menos uma das entidades necessárias para isso está ausente, então acontece um uma violação de entidade. Contudo, se todas as entidades estiverem presente, então a violação não acontece.

$$\begin{aligned} hasEntity(g_i, eg_n) \wedge \neg isPresent(e_k) \wedge (e_k \in eg_n) \wedge tryReach(ag_m, g_i) \rightarrow \\ violationEntity(ag_m, g_i, e_k) \\ hasEntity(g_i, eg_n) \wedge isPresent(eg_n) \wedge tryReach(ag_m, g_i) \rightarrow \\ \neg violationEntity(ag_m, g_i, e_k) \end{aligned} \quad (4)$$

Uma violação de condição tem como consequências a ocorrência do risco (sobre o agente que cometeu a violação) relacionado a condição que não estava presente. Esse risco pode apresentar diferentes graus de fatalidade.

$$\begin{aligned} & violationCondition(ag_m, g_i, c_k) \wedge hasRisk(c_k, risk_j, f_m) \rightarrow \\ & consequenceOfBadEvent(g_i, ag_m, risk_j, f_m) \end{aligned} \quad (5)$$

Uma violação de relacionamento tem como consequência a ocorrência do risco (sobre o agente que cometeu a violação) relacionado ao relacionamento que não foi feito. Esse risco pode apresentar diferentes graus de fatalidade.

$$\begin{aligned} & violationRelation(ag_m, g_i, r_k) \wedge hasRisk(r_k, risk_j, f_m) \rightarrow \\ & consequenceOfBadEvent(g_i, ag_m, risk_j, f_m) \end{aligned} \quad (6)$$

Uma violação de relacionamento tem como consequência o aumento da possibilidade de acontecer algo de errado em objetivos que podem ser afetados por características aleatórias. Assim sendo, esse outro objetivo tem uma nova possibilidade de dar errado que é necessariamente maior a possibilidade antiga.

$$\begin{aligned} & violationRelation(ag_m, g_i, r_k) \wedge affects(r_k, r_n) \wedge hasPossibility(r_n, false) \rightarrow \\ & hasPossibility(r_n, true) \end{aligned} \quad (7)$$

Uma violação de entidade gera a interrupção imediata da atividade no objetivo de ocorrência.

$$violationEntity(ag_m, g_i, e_k) \rightarrow stopIn(g_i) \quad (8)$$

Em um objetivo onde existe uma certa possibilidade de dar errado, essa possibilidade é vinculada aos relacionamentos que devem ser feitos para concluir este objetivo. Assim sendo, um determinado agente sofrerá as consequências ruins pela ocorrência do evento.

$$\begin{aligned} & happensBadEvent(gr_n, r_m) \wedge hasRisk(r_k, risk_j, f_m) \wedge tryReach(ag_m, g_i) \\ & \rightarrow consequenceOfBadEvent(gr_n, ag_m, risk_j, f_m) \end{aligned} \quad (9)$$

A ocorrência de um evento ruim gera imediata interrupção das atividades.

$$consequenceOfBadEvent(g_k, ag_m, risk_j, f_m) \rightarrow stopIn(g_k) \quad (10)$$

Se o objetivo não é encerrado para todos os agentes que tentam executar (onde parte desses agentes são todos aqueles que são obrigados a alcançar o objetivo), então o objetivo foi alcançado.

$$\neg stopIn(g_k, agg_n) \wedge (ago_n \subset agg_n) \rightarrow isReached(g_k) \quad (11)$$

### 3.5. Definindo Conceitos de Norma, Violação e de Sanção

Sistemas multiagentes normativas não podem ser descritos sem considerar três conceitos importantes; norma, violação e sanção.

1. **Norma:** Usado para determinar o comportamento dos agentes.
2. **Violação:** Ocorre quando o agente realiza algo que é proibido.
3. **Sanção:** Consequências punitivas que acontecem ao agente ao realizar algo proibido.

Essas definições são usadas em diversos estudos na área de sistemas multiagentes [Dastani et al. 2009], [Boissier 2011], [Barot et al. 2013], [López y López and Luck 2004] e [von Wright 1969]. Nesse estudo, a violação é tratada nos predicados  $violationCondition(ag_i, g_j, c_k)$ ,  $violationRelation(ag_i, g_j, r_k)$  e  $violationEntity(ag_i, g_j, e_k)$ . Esses predicados levam em consideração o domínio do problema deste estudo bem como o que é tido por violação dentro do contexto acadêmico, pois são eventos que acontecem mediante violação de uma norma. As relações 2, 3, 4 não determinam apenas a violação, mas também definem qual é o comportamento normativo adequado, uma vez que deixam evidentes qual dever ser o comportamento do agente para que a violação associada seja considerada falsa. As relações 5, 6 são sanções. Isso, pois definem consequências negativas aos agentes tendo em vista a ocorrência da violação. A relação de implicabilidade 7 não é uma sanção, pois define consequências ruins para outros agentes.

### 3.6. Justificando a Existência das Relações de Implicabilidade

#### 3.6.1. Relação 1

A relação 1 existe com base em estudos sobre lógica deontica (definir o estudo) onde um indivíduo só pode ser obrigado a fazer algo se esse indivíduo tiver permissão para isso. Caso contrário, há uma contradição em relação a semântica dos predicados, pois é dado como absurdo alguém ser obrigado a fazer algo sem ter permissão para isso.

#### 3.6.2. Relação 2

Representar condições que devem ser mantidas ao longo de toda a execução da atividade bem como as consequências de praticar uma determinada atividade dado a ausência de pelo menos uma das condições é um aspecto sobre o qual este modelo se propõe a representar. Para definir a ocorrência da violação se faz necessário saber quais são as condições vinculadas ao objetivo, pois caso contrário não é possível identificar quais condições devem ser mantidas ao longo da tentativa de se alcançar um objetivo e, por consequência, qualquer afirmação feita sobre essas condições não possui uma sustentação lógica rigorosa. A consideração dessas relações acontece por meio predicado  $hasCondition(g_i, cg_n)$ .

Outro aspecto relevante consiste saber se a condição necessária para tentar alcançar o objetivo não está presente durante a tentativa. Se isso não for feito, não é possível saber se a condição está presente durante a tentativa de alcançar o objetivo e qualquer afirmação derivada das condições não apresenta sustentação lógica. Isso é

feito por intermédio do predicado  $isPresent(c_k)$ . Ao saber que é verdade o negado de  $isPresent(c_k)$ , é possível saber que a condição  $c_k$  não está presente. Ainda sim, definir em uma única expressão o predicado  $hasCondition(g_i, cg_n)$  e o predicado  $isPresent(c_k)$  não é o suficiente, pois  $cg_n$  é um conjunto e  $c_k$  é um elemento que pode ou não pertencer ao conjunto  $cg_n$ . Assim sendo, isso deve ser levado em consideração por agregar a seguinte relação a expressão:  $c_k \in cg_n$ .

Os predicados considerados até o momento são necessários, porém não são suficientes. Isso, pois apenas com eles não é possível definir qual é a situação do agente em relação ao objetivo. Por exemplo, é possível considerar um cenário onde pelo menos uma das condições necessárias para atingir o objetivo não está presente. Contudo, não é razoável, considerando o conceito de violação de condição empregado neste estudo, afirmar que o agente cometeu uma violação de condição sobre esse cenário sendo que não há informação sobre a situação da tentativa do agente alcançar o objetivo. Assim sendo, se o agente tentar alcançar o objetivo mesmo com pelo menos uma condição inexistente, então esse agente cometeu uma violação de condição, contudo se o agente não tenta alcançar o objetivo em análise então ele não cometeu violação alguma. Então, a informação no que diz respeito ao agente deve ser considerada nessa relação de implicabilidade e isso acontece por meio do predicado  $tryReach(ag_m, g_i)$ .

Para definir a ausência de violação, todas as condições devem estar presentes. Para representar isso, a relação 2 exibe uma segunda relação de implicabilidade que considera o negado de uma violação de condição dada a presença de todas as condições necessárias para o objetivo.

### 3.6.3. Relação 3

Para levar em consideração violações sobre se o agente sabe ou não manipular algum tipo de relação, se faz necessário buscar por relações de implicabilidade que verificam as seguintes circunstâncias: relações que devem ser feitas para que um determinado objetivo possa ser atingido  $hasRelation(g_i, rg)$ , se a relação em análise é uma relação que pertence a todas as relações necessárias para findar esse objetivo  $r_k \in rg_g$ , verificar se a relação em análise está presente quando o agente realiza a atividade  $isPresent(r_k)$  e levar em consideração o ato do agente tentar executar a atividade  $tryReach(ag_m, g_i)$ . A violação acontece quando, na reunião de todas essas circunstâncias, uma das relações não se faz presente  $\neg isPresent(r_k)$ . Para essa situação,  $violationRelation(ag_m, g_i, r_k)$  tem que ser verdadeiro. Se todas as relações  $rg_n$  estão presentes, então a violação de relação não acontece e isso é formalizado por meio da segunda relação de implicabilidade presente na 3.

Considerando que uma violação de relacionamento acontece quando um agente tenta alcançar um objetivo sem realizar um ou mais dos relacionamentos necessários para que esse objetivo seja satisfeito com sucesso, é possível demonstrar a necessidade das circunstâncias presentes na relação de implicabilidade deste modelo por analisar o que acontece na ausência de cada uma delas. Seguindo por essa linha, supondo uma situação onde não se verifica as relações relevantes para um objetivo, logo não é possível dizer se a ausência de uma violação em específico é relevante para o objetivo em análise. Assim sendo, não é possível afirmar quando acontece uma determinada violação.

Supondo uma situação onde não se verifica as relações que foram concluídas durante o ato da manutenção. Neste cenário é possível saber quais relações são necessárias para alcançar um determinado objetivo, mas deste conjunto não é possível definir qual dessas relações foram alcançadas. Se não se sabe afirmar qual das relações um agente conseguiu realizar, não se sabe qual dessas relações um agente deixou de fazer. Não sabendo esta última, não se sabe se ocorreu uma violação dessa relação. Assim sendo, só é possível afirmar algo sobre a ocorrência de uma violação de relação se for possível analisar se as relações necessárias para um dado objetivo foram ou não cumpridas.

Supondo uma situação onde não se verifica se um agente tentou realizar o objetivo. Nesta situação não é possível analisar quem cometeu a violação. Além disso não é possível considerar se a presença ou a ausência das relações são relevantes ou não para alcançar o objetivo, pois não se pode localizar a ocorrência dessas relações nos momentos onde elas devem acontecer (que é quando o agente tenta alcançar o objetivo dessas relações). Além disso, desconsiderar a tentativa do agente envolve descaracterizar a ocorrência da violação uma vez que esta, por conseguinte, considera a existência de um agente durante a ocorrência de uma violação.

#### 3.6.4. Relação 4

Para verificar violações onde um agente tenta executar um determinado procedimento sem ter todas entidades (ex. ferramentas, colegas) para isso, é necessário construir uma expressão de implicabilidade que considere quais são as entidades importantes para a execução do objetivo  $hasEntity(g_i, eg_n)$ , considere quais as entidades que estão presentes no ambiente durante o ato da execução  $isPresent(eg_n), isPresent(e_k), e \in eg_n$  e que considere se um agente tentou alcançar o objetivo  $tryReach(ag_m, g_i, e_k)$ . Para situação onde está ausente uma das entidades necessárias para concluir um dado objetivo, então acontece uma violação de entidade. Contudo, para situação onde todas as entidades necessárias para alcançar um objetivo estão presentes durante a tentativa que o agente tem para isso, então a violação não acontece.

Se uma violação de entidade é definida como sendo a tentativa de um agente alcançar um determinado objetivo sem ter todas as entidades para isso, então é possível verificar a necessidade de todas as circunstâncias por analisar o que acontece na ausência delas. Supondo não ser possível verificar quais entidades são necessárias para alcançar um determinado objetivo, então não é possível afirmar quais entidades devem ser presentes durante a tentativa do agente em alcançar um determinado objetivo. Por consequência, também não é possível identificar quais entidades estão ausentes logo, neste caso é um absurdo fazer qualquer afirmação sobre presença ou ausência sobre violação por ausência de uma entidade. Supondo não ser possível afirmar se uma entidade necessária para conclusão de um determinado objetivo não está presente durante a tentativa de alcançar esse objetivo. Então, é um absurdo afirmar que aconteceu ou não aconteceu um evento no que diz respeito a ausência de uma entidade sem mesmo saber se essa entidade estava presente. Supondo não ser possível definir se um agente tentou ou não executar o objetivo. Então também não é possível saber se a entidade estava presente no momento em que deve estar presente para definir a ocorrência de uma violação, além de descaracterizar a semântica da violação de uma entidade usada por este modelo uma vez que está



considerar necessariamente a existência de um agente pela ocorrência da violação.

### 3.6.5. Relações 5 e 6

O propósito da relação 5 consiste representar uma sanção por cometer uma violação de condição. Uma violação dessa natureza gera consequências físicas. Essas consequências são necessariamente a ocorrência dos riscos a vida por executar a atividade sem todas as condições necessárias. Assim sendo, uma relação de implicabilidade que tem como por comprometimento representar uma sanção desta natureza deve necessariamente considerar o tipo de violação (pois assim somente assim é possível identificar quais riscos devem ser considerados) e deve considerar esses riscos. Assim sendo, o predicado  $hasRisk(c_k, risk_j, f_m)$  tem como por finalidade representar as relações entre a condição faltante, a natureza do risco e o grau de fatalidade associado. O predicado  $consequenceOfBadEvent(g_i, ag_m, risk_j, f_m)$  define que o evento ruim realmente aconteceu sobre esse agente.

A relação 6 se equadra na mesma natureza. Contudo, em vez de considerar uma violação de condição, a relação 6 considera violações do tipo de relacionamento. Essa relação é necessária pois visa representar as consequências negativas de não se fazer um relacionamento necessário para a manutenção.

### 3.6.6. Relações 7

! \*\* REVISAR \*\* !

Uma das finalidades deste modelo consiste representar o seguinte cenário: agente A comete um erro, contudo as consequências ocorrem apenas um tempo depois sobre o agente B. Esse modelo considerou a violação de relacionamento como causa deste tipo de situação pois no caso em estudo foi possível identificar certas relações que se não forem feitas não geram consequências imediatas para os profissionais envolvidos no meio, contudo alguém pode sair prejudicado depois.

Um ambiente hipotético onde um eletricista deve medir a condutividade de um bastão universal é uma situação que pode exemplificar a aplicação desta relação. Se o eletricista não realizar a relação entre a entidade condutivímetro com a entidade bastão universal, nenhum profissional sai prejudicado de forma imediata. Contudo, se o bastão estiver com isolamento comprometido, o agente que for manipular a ferramenta poderá ser submetido a uma corrente elétrica elevadíssima, gerando morte imediata.

Os pesquisadores optaram por representar esse tipo de situação considerando uma dada aleatoriedade ao evento. Assim sendo, o modelo trata que em um dado tipo de objetivo (conhecido por objetivo randômico) tem uma certa possibilidade de acontecer um evento ruim com o profissional que tenta alcançá-lo. Isso tem como por objetivo representar as possibilidades que é associado a eventos ruins vir a se tornar uma realidade durante a execução de um dada atividade. Assim sendo, um relacionamento que não foi feito aumenta as possibilidades desses eventos relacionados aos riscos acontecerem.

Para formalizar isso se faz necessário considerar o risco propriamente dito, se faz

necessário quais objetivos podem ser afetados dado a ausência de um relacionamento  $affects(r_k, gr_n, p_n)$ , se faz necessário considerar uma certa possibilidade do objetivo dar errado independente da ação dos demais agentes, algo que é feito por meio do predicado  $hasPossibility(gr_n, p_m)$ . Esse predicado acontece tanto antes bem como depois do  $\rightarrow$ . Contudo, depois de  $\rightarrow$ , a possibilidade possui um novo valor (advém do predicado  $affects(r_k, gr_n, p_n)$ ), onde deve ser necessariamente maior que o anterior.

### 3.6.7. Relações 8

Sem a presença das entidade necessária para concluir um determinado objetivo, não é possível concluir o objetivo. Caso contrário, essa entidade não é, portanto, necessária para cumprir com objetivo. É possível considerar um cenário onde os agentes adaptam outras entidades para sanar a entidade faltante. Contudo, esse modelo não trata esse tipo de situação com a finalidade de evitar maiores complexidades. Para uma relação de implicabilidade que tem como por finalidade representar esse tipo de situação, deve necessariamente considerar a ocorrência da violação  $violationEntity(ag_m, g_i, e_k)$  e representar o encerramento da atividade, feito pelo predicado  $stopIn(g_i)$ .

### 3.6.8. Relações 9

A relação 9 tem como por finalidade representar a ocorrência de um evento associado a um risco em um objetivo com possibilidades randomicas. Os eventos ruins são associados aos riscos e esses, por sua vez, são associados a relacionamentos específicos que são vinculados ao objetivo. Isso abre a necessidade de um predicado que associe esses objetivos ao risco  $happensBadEvent(gr_n, r_m)$ . A relação também deve considerar o risco vinculado com o relacionamento, caso contrário não é possível saber qual evento vinculado a um risco e este, por sua vez, vinculado ao relacionamento gerou uma situação ruim ao agente. Assim sendo, o predicado  $hasRisk(r_k, risk_j, f_m)$  deve ser levado em consideração ao executar o procedimento de manutenção. Além disso, é necessário saber se o agente estava tentando alcançar o objetivo no instante que o evento ruim acontece, isso é feito por meio do predicado  $tryReach(ag_m, g_i)$ . O predicado  $consequenceOfBadEvent(gr_n, ag_m, risk_j, f_m)$  demonstra que o evento realmente aconteceu e afetou o agente de certa forma com certa intensidade. Essa relação de implicabilidade representa que um evento ruim pode acontecer a um dado agente mesmo que este não tenha cometido nada de errado. Tendo como base a relação 7, a possibilidade dessa situação acontecer aumenta com base no erro de outros agentes.

### 3.6.9. Relações 10

O modelo considera que os agentes não continuam a trabalhar tendo em vista que um dos colegas morreu ou ficou ferido ao executar uma dada atividade. Portanto, é necessário considerar uma relação de implicabilidade onde a ocorrência um evento ruim gera interrupção das atividades profissionais.

### 3.6.10. Relações 37

O modelo deve considerar a condição de quando objetivo é definitivamente alcançado, caso contrário os agentes não serão capaz de cumprir com os próximos objetivos para os quais foram designados. Para construir esse critério é necessário considerar se não houve interrupção para todos os agentes que tentaram alcançar o objetivo. Isso é feito por meio do predicado  $\neg stopIn(g_k, agg_n)$ , em que  $agg_n$ . Não só isso como é necessário considerar se todos os agentes que são obrigados a alcançar o objetivo realmente tentaram fazer isso. Assim sendo, é necessário considerar um subconjunto de  $agg_n$  que é  $ago_n$ .

### 3.7. Máquina de Estados

A figura dada por 1 apresenta a estrutura do modelo em formato de diagrama de classes.

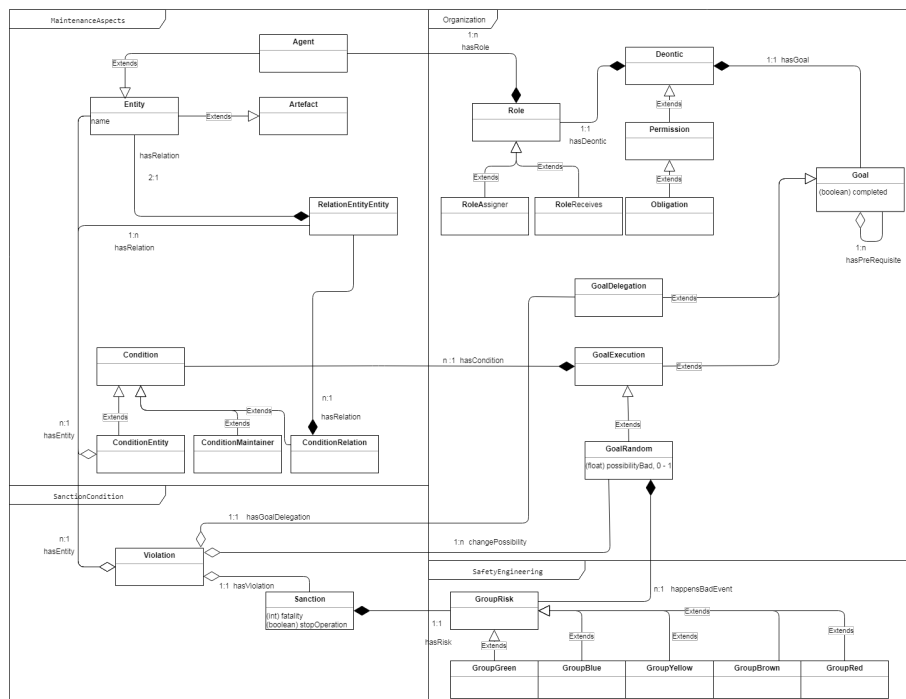


Figura 1.

A figura 2 apresenta um diagrama de estados do modelo.

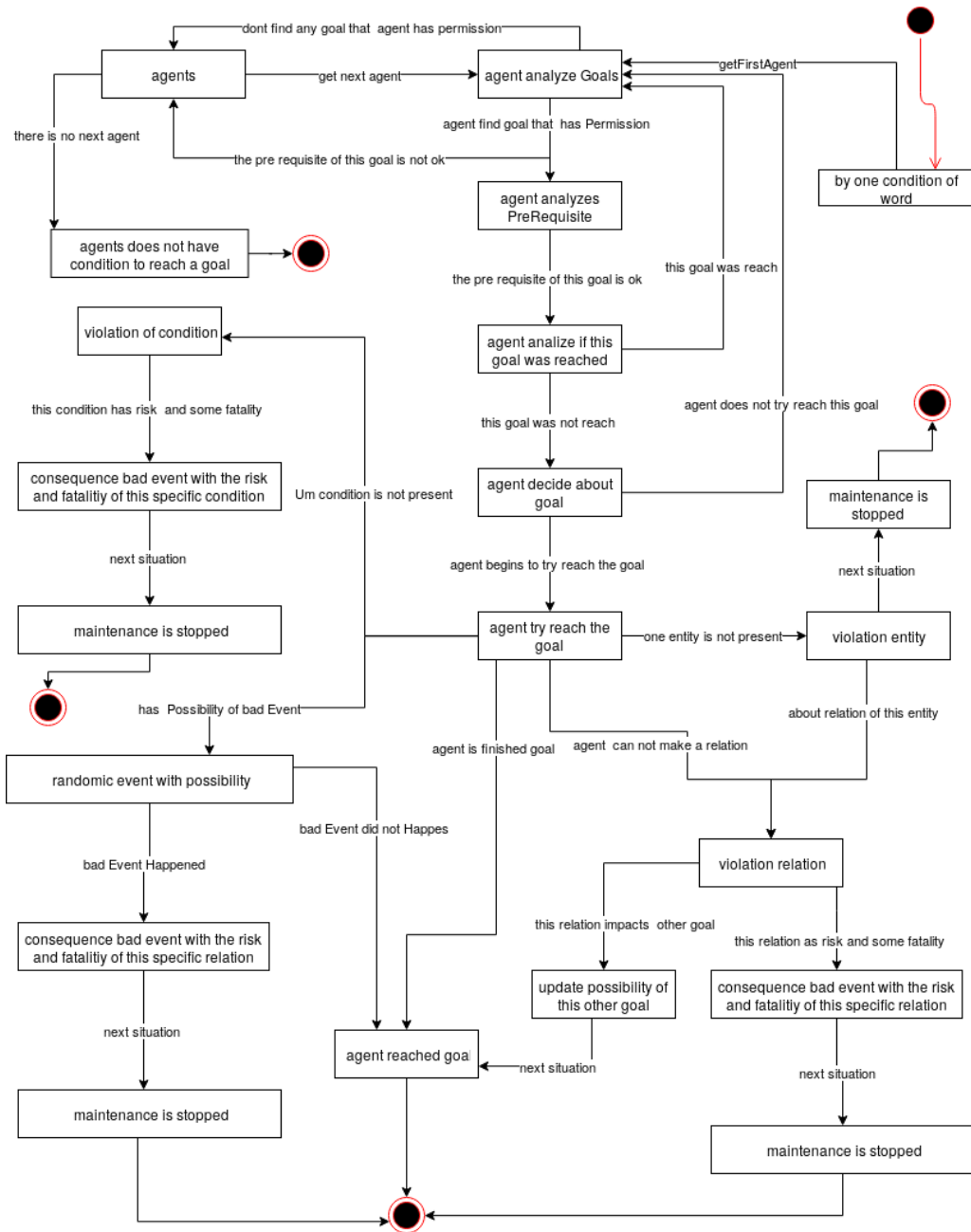


Figura 2.

### 3.8. Modelando

Essa subseção tem como finalidade especificar o estudo de caso em interesse dentro da estrutura deste modelo.

A tabela 1 apresenta todos os agentes que fazem parte da manutenção. A tabela 2 apresenta todas as funções que deverão ser exercidas pelos agentes. A tabela 3 define a relação  $hasRole(ag_n, \rho_m)$  onde  $ag_n$  é representado pela coluna agente e  $\rho_m$  é representado pela coluna papel.

<b>símbolo</b>	<b>significado</b>
agente1	Um dos agentes participantes da manutenção
agente2	Um dos agentes participantes da manutenção
agente3	Um dos agentes participantes da manutenção
agente4	Um dos agentes participantes da manutenção
agente5	Um dos agentes participantes da manutenção
agente6	Um dos agentes participantes da manutenção
agente7	Um dos agentes participantes da manutenção

**Tabela 1. Os agentes que constituem uma manutenção**

<b>papel</b>	<b>descrição</b>
supervisor	Atribui papel a outros profissionais
executor1	Tem como por finalidade executar certas atividades manuais vinculadas a manutenção
executor2	Tem como por finalidade executar certas atividades manuais vinculadas a manutenção
executor3	Tem como por finalidade executar certas atividades manuais vinculadas a manutenção
executor4	Tem como por finalidade executar certas atividades manuais vinculadas a manutenção
executor5	Tem como por finalidade executar certas atividades manuais vinculadas a manutenção

**Tabela 2. Os papéis relevantes para a ocorrência da manutenção**

<b>agente</b>	<b>papel</b>
agente1	supervisor
agente2	executor1
agente3	executor1
agente4	executor2
agente5	executor3
agente6	executor4
agente7	executor5

**Tabela 3. Relação  $hasRole(ag_n, \rho_m)$**

A tabela 4 apresenta todos artefatos que fazem parte da descrição deste estudo de caso. Todos esses artefatos, em conjunto com os agentes, constituem todas as entidades definidas pelo modelo para este estudo de caso.

artefato	descrição
capacete	EPI usado pelo profissional para proteger a cabeça
oculos	Óculos usado para evitar dificuldades de enxergar presentes em dias claros
roupagem	Consiste em roupas isolantes e anti-chamas
luva	Luvas Isolantes
bota	Botas Isolantes para evitar que o profissional seja eletrocutado
bastaoGarra	bastão isolante que possui uma ferramenta em estrutura de garra. 64 X 3600 mm
sela	Possui diametro 65 mm, é fixada na torre para sustentar o bastão.
colar	Estrutura que fica fixa na sela, bastão isolante é travado no colar.
corda	Corda Isolante.
carretilha	Carretilha que, em conjunto com a corda, é usada para mover material na vertical.
bastaoUniversal	Bastao isolante que permite o acoplamento de multiplas ferramentas.
soquete	Usado na manipulação de parafusos.
locador	Usado como pino direcional em alinhamento de furo de parafusos, auxiliado na inserção de pinos e parafusos.
bastaoGarra	Bastão Universal que possui uma garra.
isoladorVelho	Isolador de pedestal danificado a ser substituido
isoladorNovo	Isolador de pedestal novo que será posicionado no local do isolador velho.
torre	Estrutura metálica onde fica fixo o isolador
condutor	Em formato de cabo, fica fixo sobre o topo do isolador.e é por onde passa grandes quantidades de energia elétrica.
estropo	pano firme usado para segurar Isolador quando estiver suspenso
pano	pano usado para limpar ferramentas
glicerina	substância usada para limpar as ferramentas adequadamente
condutivimetro	Medidor de corrente de fulga sobre o bastão universal.
parafuso	Parafusos prendem o conector condutor-Isolador e também prendem o Isolador a base
conector	Estrutura que tem como por finalidade manter condutor,cabeçote do isolador em conjunto.

**Tabela 4. Definindo todos os artefatos presentes na manutenção**

A tabela 5 apresenta os objetivos dados pela coluna *objetivo* bem com sua descrição. Essa tabela também apresenta os conjuntos  $gp_i$  dado pela coluna pré-requisitos. Assim sendo, essa tabela também apresenta a relação entre os objetivos e seus respectivos pré-requisitos, ou seja, a relação  $isPresentRequisite(gp_i, g_j)$ .

objetivo	pré-requisito	Descrição
gSupervisor	g0	Atribui objetivos aos demais agentes.
g0	Ø	Vestir os AP'Is
g1	gSupervisor	Limpar, secar e testar ferramentas com material isolante.
g2	g1	Medir a corrente de fulga de ferramentas isolantes
g3	g2	Instalar sela com colar na estrutura
g4	g3	Passar o bastão garra por dentro do olhal do colar.
g5	g4	Amarrar o bastão garra na parte superior da estrutura com a corda, fixar no condutor
g6	gSupervisor	Amarrar o olhal do bastão garra ao cavalo da sela atrás de uma corda.
g7	g6	Instalar sela com colar no outro lado da estrutura estrutura
g8	g7	Passar o bastão universal por dentro do olhal do colar
g9	g8	Pender carretilha no bastão Universal.
g10	g9	Amarrar o bastão universal na parte superior da estrutura com a corda;
g11	g10	Amarrar o olhal do bastão universal ao cavalo da sela atrás de uma corda.
g12	g11,g5	Rotacionar estrutura olhal garra em 45 graus.
g13	g12	Enforcar um estropo de Náilon no corpo do isolador velho.
g14	g13	Colocar a extremidade do estropo no gancho da corda de serviço.
g15	g14	Afrouxar os parafusos do conector que prendem a barra ao isolador.
g16	g15	Terminar de retirar os parafusos com o bastão com o soquete multiangular.
g17	g16	Elevar o condutor através da corda que une a sela ao bastão.
g18	g17	Apertar o colar através da porca borboleta.
g19	g18	Sacar parafusos da base da coluna.
g20	g19	Segurar firmemente a corda de serviço,baixar o isolador ao solo
g21	g20	Passar Estropo no Isolador Novo
g22	g21	Colocar a extremidade do estropo no gancho da corda de serviço.
g23	g22	Içar o Isolador
g24	g23	Colocar Parafusos na base da coluna.
g25	g24	Baixar o condutor para que a mesma apóie no novo isolador.
g26	g25	Colocar os parafusos do conector que prende a barra ao novo isolador.
g27	g26	Retirar Equipamentos

**Tabela 5. Define e descreve os objetivos bem como os respectivos pré-requisitos**

A tabela 6 apresenta  $c_k$  dado pela coluna condição e pela coluna descrição. Essa tabela define a relação  $hasRisk(c_k, risk_j, f_m)$  onde  $risk_j$  é descrito pela coluna risco e  $f_m$  é descrito pela coluna fatalidade.

condição	descrição	risco	fatalidade
umidade70	Umidade Relativa do Ar deve ser inferior a setenta por cento.	eletrocutado	morte
noVento	Não deve haver vento durante os procedimentos de manutenção.	eletrocutado	morte
noChuva	Não deve haver chuva durante o ato da manutenção	eletrocutado	morte
sol	O dia deve estar ensolarado	eletrocutado	morte

**Tabela 6. Define as condições necessárias para que a manutenção tenha possibilidade de acontecer**

A tabela 7 apresenta três relações onde uma delas é  $relationHas(r_l, e_i, e_k)$  onde  $r_l$  é definido pela coluna *relacionamento*,  $e_i$  e  $e_k$  pelas entidades envolvidas. A outra

relação é dada por  $hasRisk(r_k, risk_j, f_m)$  onde  $risk_j$  é dado pela coluna risco e  $f_m$  é dado pela coluna fatalidade. A terceira relação é dada por  $hasPossibility(gr_n, p_m)$



relacionamento	entidades envolvidas	risco	fatalidade	possibilidade
relXCapacete	X,capacete	nenhum	nenhum	false
relXOculos	X,oculos	nenhum	nenhum	false
relXRoupagem	X,roupagem	nenhum	nenhum	false
relXLuva	X,luva	nenhum	nenhum	false
relXBotas	X,bota	nenhum	nenhum	false
relXPano	X,pano	nenhum	nenhum	false
relPanoGlicerina	pano,glicerina	nenhum	nenhum	false
relPanoCorda	pano,corda	nenhum	nenhum	false
relPanoBastaoUniversal	pano,bastaoUniversal	nenhum	nenhum	false
relPanoSoquete	pano,soquete	nenhum	nenhum	false
relPanoBastaoUniversal	pano,bastaoGarra	nenhum	nenhum	false
relXSela	X,sela	nenhum	nenhum	false
relXColar	X,colar	nenhum	nenhum	false
relXBastaoGarra	X,bastaoGarra	nenhum	nenhum	false
relTorreSela	torre,sela	nenhum	nenhum	false
relSelaColar	sela,colar	nenhum	nenhum	false
relColarBastaoGarra	colar,bastaoGarra	nenhum	nenhum	false
relBastaoGarraCondutor	bastaoGarra,condutor	eletrocutado	morte	false
relXBastaoUniversal	X,bastaoUniversal	nenhum	nenhum	false
relCordaBastaoUniversal	corda,bastaoUniversal	nenhum	nenhum	false
relCordaCarretilha	corda,carretilha	nenhum	nenhum	false
relBastaoUniversalCarretilha	bastaoUniversal,carretilha	nenhum	nenhum	false
relBastaoUniversalColar	bastaoUniversal,colar	nenhum	nenhum	false
relBastaoUniversalEstopo	bastaoUniversal,estopo	nenhum	nenhum	false
relCordaEstopo	corda,estopo	eletrocutado	morte	false
relEstopoIsoladorVelho	estopo,isoladorVelho	nenhum	nenhum	false
relXChaveCatraca	X,chaveCatraca	nenhum	nenhum	false
relChaveCatracaBastaoUniversal	chaveCatraca,bastaoUniversal	nenhum	nenhum	false
relChaveCatracaParafuso	chaveCatraca,parafuso	eletrocutado	morte	false
relParafusoConector	parafuso,conector	eletrocutado	morte	false
relXBastaoSoquete	X,bastaoSoquete	nenhum	nenhum	false
relSoqueteParafuso	soquete,parafuso	eletrocutado	morte	false
relXCorda	X,corda	eletrocutado	morte	false
relXIsoladorVelho	X,isoladorVelho	nenhum	nenhum	false
relXIsoladorNovo	X,isoladorNovo	nenhum	nenhum	false
relCordaBastaoGarra	corda,bastaoGarra	nenhum	nenhum	false
relBastaoGarraSela	bastaoGarra, sela	nenhum	nenhum	false
relXCarretilha	X,carretilha	nenhum	nenhum	false
relBastaoUniversalCorda	bastaoUniversal,corda	nenhum	nenhum	false
relBastaoUniversalTorre	bastaoUniversal,torre	nenhum	nenhum	false
relEstopoCorda	estopo,corda	eletrocutado	morte	false
relEstopoIsoladorNovo	estopo,isoladorNovo	nenhum	nenhum	false
relBastaoUniversalSela	universal,sela	nenhum	nenhum	false
relBastaoGarraTorre	bastaoGarra,torre	nenhum	nenhum	false
relBastaoUniversalEstopo	bastaoUniversal,estopo	nenhum	nenhum	false
relXColar	X,colar	nenhum	nenhum	false
relParafusoTorre	parafuso,torre	eletrocutado	morte	false
relCondutivimetroCorda	condutivimetro,corda	nenhum	nenhum	false
relCondutivimetroBastaoUniversal	condutivimetro,bastaoUniversal	nenhum	nenhum	false
relCondutivimetroBastaoGarra	condutivimetro,bastaoGarra	nenhum	nenhum	false
relCondutivimetroSoquete	condutivimetro,soquete	nenhum	nenhum	false

**Tabela 7. Define os relacionamentos necessários que a manutenção aconteça**

As tabelas 8 e 9 apresentam a relação  $affects(r_k, r_n)$  onde  $r_k$  é representado pela

coluna relacionamento-errado e  $r_n$  é representado pela coluna relacionamento-afetado. A coluna nova possibilidade de algo errado tem como por finalidade representar que a possibilidade de ocorrer algum evento ruim atrelado ao relacionamento-afetado mudou de *false* para *true*.

relacionamento-errado	relacionamento-afetado	nova possibilidade de algo errado
relXCpacete	relBastaoGarraCondutor	true
relXCpacete	relCordaEstropo	true
relXCpacete	relChaveCatracaParafuso	true
relXCpacete	relParafusoConector	true
relXCpacete	relSoqueteParafuso	true
relXCpacete	relXCorda	true
relXCpacete	relEstropoCorda	true
relXCpacete	relParafusoTorre	true
relXOculos	relBastaoGarraCondutor	true
relXOculos	relCordaEstropo	true
relXOculos	relChaveCatracaParafuso	true
relXOculos	relParafusoConector	true
relXOculos	relSoqueteParafuso	true
relXOculos	relXCorda	true
relXOculos	relEstropoCorda	true
relXOculos	relParafusoTorre	true
relXLuva	relBastaoGarraCondutor	true
relXLuva	relCordaEstropo	true
relXLuva	relChaveCatracaParafuso	true
relXLuva	relParafusoConector	true
relXLuva	relSoqueteParafuso	true
relXLuva	relXCorda	true
relXLuva	relEstropoCorda	true
relXLuva	relParafusoTorre	true
relXBotas	relBastaoGarraCondutor	true
relXBotas	relCordaEstropo	true
relXBotas	relChaveCatracaParafuso	true
relXBotas	relParafusoConector	true
relXBotas	relSoqueteParafuso	true
relXBotas	relXCorda	true
relXBotas	relEstropoCorda	true
relXBotas	relParafusoTorre	true
relXPano	relBastaoGarraCondutor	true
relXPano	relCordaEstropo	true
relXPano	relChaveCatracaParafuso	true
relXPano	relParafusoConector	true
relXPano	relSoqueteParafuso	true
relXPano	relXCorda	true
relXPano	relEstropoCorda	true
relXPano	relParafusoTorre	true
relPanoGlicerina	relBastaoGarraCondutor	true
relPanoGlicerina	relCordaEstropo	true
relPanoGlicerina	relChaveCatracaParafuso	true
relPanoGlicerina	relParafusoConector	true
relPanoGlicerina	relSoqueteParafuso	true
relPanoGlicerina	relXCorda	true
relPanoGlicerina	relEstropoCorda	true
relPanoGlicerina	relParafusoTorre	true
relPanoCorda	relCordaEstropo	true
relPanoCorda	relXCorda	true
relPanoCorda	relEstropoCorda	true

**Tabela 8.** Define o impacto que o erro em um relacionamento gera em outro relacionamento por modar a possibilidade de algo errado acontecer.

relacionamento-errado	relacionamento-afetado	nova possibilidade de algo errado
relPanoBastaoUniversal	relBastaoGarraCondutor	true
relPanoBastaoUniversal	relChaveCatracaParafuso	true
relPanoBastaoUniversal	relParafusoConector	true
relPanoBastaoUniversal	relParafusoTorre	true
relPanoBastaoUniversal	relBastaoGarraCondutor	true
relPanoSoquete	relBastaoGarraCondutor	true
relPanoSoquete	relCordaEstropo	true
relPanoSoquete	relChaveCatracaParafuso	true
relPanoSoquete	relParafusoConector	true
relPanoSoquete	relSoqueteParafuso	true
relPanoSoquete	relXCorda	true
relPanoSoquete	relEstropoCorda	true
relPanoSoquete	relParafusoTorre	true
relCondutivimetroCorda	relBastaoGarraCondutor	true
relCondutivimetroCorda	relCordaEstropo	true
relCondutivimetroCorda	relChaveCatracaParafuso	true
relCondutivimetroCorda	relParafusoConector	true
relCondutivimetroCorda	relSoqueteParafuso	true
relCondutivimetroCorda	relXCorda	true
relCondutivimetroCorda	relEstropoCorda	true
relCondutivimetroCorda	relParafusoTorre	true
relCondutivimetroBastaoUniversal	relBastaoGarraCondutor	true
relCondutivimetroBastaoUniversal	relCordaEstropo	true
relCondutivimetroBastaoUniversal	relChaveCatracaParafuso	true
relCondutivimetroBastaoUniversal	relParafusoConector	true
relCondutivimetroBastaoUniversal	relSoqueteParafuso	true
relCondutivimetroBastaoUniversal	relXCorda	true
relCondutivimetroBastaoUniversal	relEstropoCorda	true
relCondutivimetroBastaoUniversal	relParafusoTorre	true
relCondutivimetroBastaoGarra	relBastaoGarraCondutor	true
relCondutivimetroBastaoGarra	relCordaEstropo	true
relCondutivimetroBastaoGarra	relChaveCatracaParafuso	true
relCondutivimetroBastaoGarra	relParafusoConector	true
relCondutivimetroBastaoGarra	relSoqueteParafuso	true
relCondutivimetroBastaoGarra	relXCorda	true
relCondutivimetroBastaoGarra	relEstropoCorda	true
relCondutivimetroBastaoGarra	relParafusoTorre	true
relCondutivimetroSoquete	relBastaoGarraCondutor	true
relCondutivimetroSoquete	relCordaEstropo	true
relCondutivimetroSoquete	relChaveCatracaParafuso	true
relCondutivimetroSoquete	relParafusoConector	true
relCondutivimetroSoquete	relSoqueteParafuso	true
relCondutivimetroSoquete	relXCorda	true
relCondutivimetroSoquete	relEstropoCorda	true
relCondutivimetroSoquete	relParafusoTorre	true

**Tabela 9. Define o impacto que o erro em um relacionamento gera em outro relacionamento por modar a possibilidade de algo errado acontecer.**

As tabelas 10, 11 apresentam a relação  $hasObligation(\rho_m, g_i)$  onde  $\rho_m$  é representado pela coluna papel e  $g_i$  é representado pela coluna objetivo.

<b>papel</b>	<b>objetivo</b>
executor1	g0
executor2	g0
executor3	g0
executor4	g0
executor5	g0
supervisor	g0
supervisor	gSupervisor
executor1	g1
executor2	g1
executor1	g2
executor2	g2
executor1	g3
executor2	g2
executor1	g4
executor2	g4
executor1	g5
executor2	g5
executor3	g6
executor4	g6
executor5	g6
executor3	g7
executor4	g7
executor5	g7
executor3	g8
executor4	g8
executor5	g8
executor3	g9
executor4	g9
executor5	g9
executor3	g10
executor4	g10
executor5	g10
executor3	g11
executor4	g11
executor5	g11
executor1	g12
executor2	g12
executor3	g12
executor4	g12
executor1	g13
executor2	g13
executor3	g13
executor4	g13
executor1	g14

**Tabela 10. Objetivos que devem ser atingidos pelo agente que assumir um dada função**

<b>role</b>	<b>g</b>
executor2	g14
executor3	g14
executor4	g14
executor2	g15
executor3	g15
executor4	g15
executor5	g15
executor2	g16
executor3	g16
executor4	g16
executor5	g16
executor1	g17
executor3	g17
executor4	g17
executor5	g17
executor1	g18
executor3	g18
executor4	g18
executor5	g18
executor1	g19
executor3	g19
executor4	g19
executor5	g19
executor1	g20
executor3	g20
executor4	g20
executor5	g20
executor1	g21
executor3	g21
executor4	g21
executor5	g21
executor1	g22
executor2	g22
executor3	g22
executor5	g22
executor1	g23
executor2	g23
executor3	g23
executor5	g23
executor1	g24
executor2	g24
executor3	g24
executor5	g24
executor1	g25
executor2	g25
executor3	g25
executor4	g25
executor1	g26
executor2	g26
executor3	g26
executor4	g26
executor1	g27
executor2	g27
executor3	g27
executor4	g27
executor5	g27

**Tabela 11. Objetivos que devem ser atingidos pelo agente que assumir um dada função**

A tabela 12 apresenta as entidades que constituem os conjuntos **eg**.

<b>entidades</b>	<b>eg</b>
capacete,óculos,roupagem,luvas,botas $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg0
pano,glicerina,carretilha,bastaoUniversal,corda,bastaoGarra, $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg1
pano,glicerina,carretilha,bastaoUniversal,corda,bastaoGarra,condutivimetro, $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg2
sela,colar $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg3
colar,bastaoGarra $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg4
corda,bastaoGarra,bastaoGarraTorre,condutor $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg5
bastaoGarra,sela $X = \{\text{agente que tenta alcanar o objetivo}\}$	eg6
sela,colar $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg7
sela,bastaoUniversal,Colar, $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg8
bastaoUniversal,carretilha, $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg9
corda,bastaoUniversal,corda,torre, $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg10
bastaoUniversal,corda,colar,sela $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg11
colar, $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg12
bastaoUniversal,estropo,isoladorVelho $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg13
bastaoUniversal,corda,estropo $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg14
chaveCatraca,bastaoUniversal,prafuso $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg15
bastaoSoquete,parafuso, $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg16
bastaoGarra,condutorcorda $X = \{\text{agente que tenta alcançar o objetivo}\}$ ,	eg17
colar, $X = \{\text{agente que tenta alcançar o objetivo}\}$ ,	eg18
chaveCatraca,bastaoUniversal,prafusobastaoSoquete,parafuso,torre $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg19
corda $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg20
estropo, isoladorNovo, $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg21
bastaoUniversal,corda,estropo $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg22
corda $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg23
chaveCatraca,bastaoUniversal,prafusobastaoSoquete,parafuso,torre $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg24
bastaoGarra,condutorcorda $X = \{\text{agente que tenta alcançar o objetivo}\}$ ,	eg25
chaveCatraca,bastaoUniversal,prafuso $X = \{\text{agente que tenta alcançar o objetivo}\}$	eg26
sela,colar,bastaoGarra,bastaoUniversal,bastaoSoquete,corda,carretilha,chaveCatraca,torre,condutor	eg27

**Tabela 12. Entidades que formam os conjuntos  $eg_n$ . Cada conjunto destes estão relacionados com um objetivo e determinam as entidades necessárias para que o mesmo tenha codição de ser alcançado.**

A tabela 13 apresenta as relações que constituem os conjuntos **rg**.

<b>relacionamentos</b>	<b>rg</b>
relXcapacete relXoculos relXroupagem relXluva relXbotas	rg0
relXPano relPanoGlicerina relPanoCorda relPanoBastaoUniversal relPanoBastaoGarra relPanoSoquete	rg1
relCondutivimetroCorda relCondutivimetroBastaoUniversal relCondutivimetroBastaoGarra relCondutivimetroSoquete	rg2
relCondutivimetro	rg3
relXBastaoGarra relColarBastaoGarra	rg4
relXBastaoGarra relXCordarelCordaBastaoGarra relBastaoGarraTorre relBastaoGarraCondutor	rg5
relBastaoGarraSela relXBastaoGarra relXSela	rg6
relXSela relXColar relTorreSela	rg7
relBastaoUniversalColar relXBastaoUniversal	rg8
relXBastaoUniversal relXCarretilha relBastaoUniversalCarretilha	rg9
relXCorda relXBastaoUniversal relBastaoUniversalCorda relBastaoUniversalTorre	rg10
relXCorda relXBastaoUniversal relXColar relBastaoUniversalColar relBastaoUniversalSela	rg11
relXColar	rg12
relXBastaoUniversal relBastaoUniversalEstropo relEstropoIsoladorVelho	rg13
relXBastaoUniversal relBastaoUniversalCordarelCordaEstropo relEstropoCorda	rg14
relChaveCatracaBastaoUniversal relXChaveCatraca relXBastaoUniversal relChaveCatracaParafuso	rg15
relXBastaoSoquete relSoqueteParafuso	rg16
relXCorda relCordaBastaoGarra relBastaoGarraCondutor	rg17
relXColar	rg18
relChaveCatracaBastaoUniversal relXChaveCatraca relXBastaoUniversal relChaveCatracaParafuso relParafusoTorre relXBastaoSoquete relSoqueteParafuso	rg19
relXCorda	rg20
relXEstropo relEstropoIsoladorNovo	rg21
relXBastaoUniversal relBastaoUniversalCorda relCordaEstropo relEstropoCorda	rg22
relXCorda	rg23
relChaveCatracaBastaoUniversal relXChaveCatraca relXBastaoUniversal relChaveCatracaParafuso relParafusoTorre relXBastaoSoquete relSoqueteParafuso	rg24
relXCorda relCordaBastaoGarra relBastaoGarraCondutor	rg25
relChaveCatracaBastaoUniversal relXChaveCatraca relXBastaoUniversal relChaveCatracaParafuso	rg26
relXSela relXColarrelXBastaoGarrarelXBastaoUniversal relXBastaoSoquete relXCorda relXCarretilha relXChaveCatraca relColarBastaoGarra relCordaBastaoGarra relBastaoGarraTorre relBastaoGarraCondutor relBastaoUniversalCarretilha relBastaoGarraSela relBastaoUniversalSela relSelaColar relTorreSela relBastaoUniversalCorda relBastaoGarraCorda	rg27

**Tabela 13. Relacionamentos que formam os conjuntos  $rg_n$ . Cada conjunto  $rg_n$  está relacionado com um objetivo A relação entre  $rg_n$  e  $goal_m$  determina os relacionamentos necessários para que um dado objetivo tenha condição de ser atingido.**

A tabela 14 apresenta as condições que constituem os conjuntos **cg**.

<b>condicoes</b>	<b>cg</b>
umidade70,noVento,noChuva,sol	cg1

**Tabela 14. Todas as condições que constituem o conjunto  $cg_n$ . Este conjunto está relacionando com um ou mais objetivos e determina quais são as condições que devem ser mantidas para que o agente tenha uma situação razoavel para tentar alcançar um certo obejtivo**

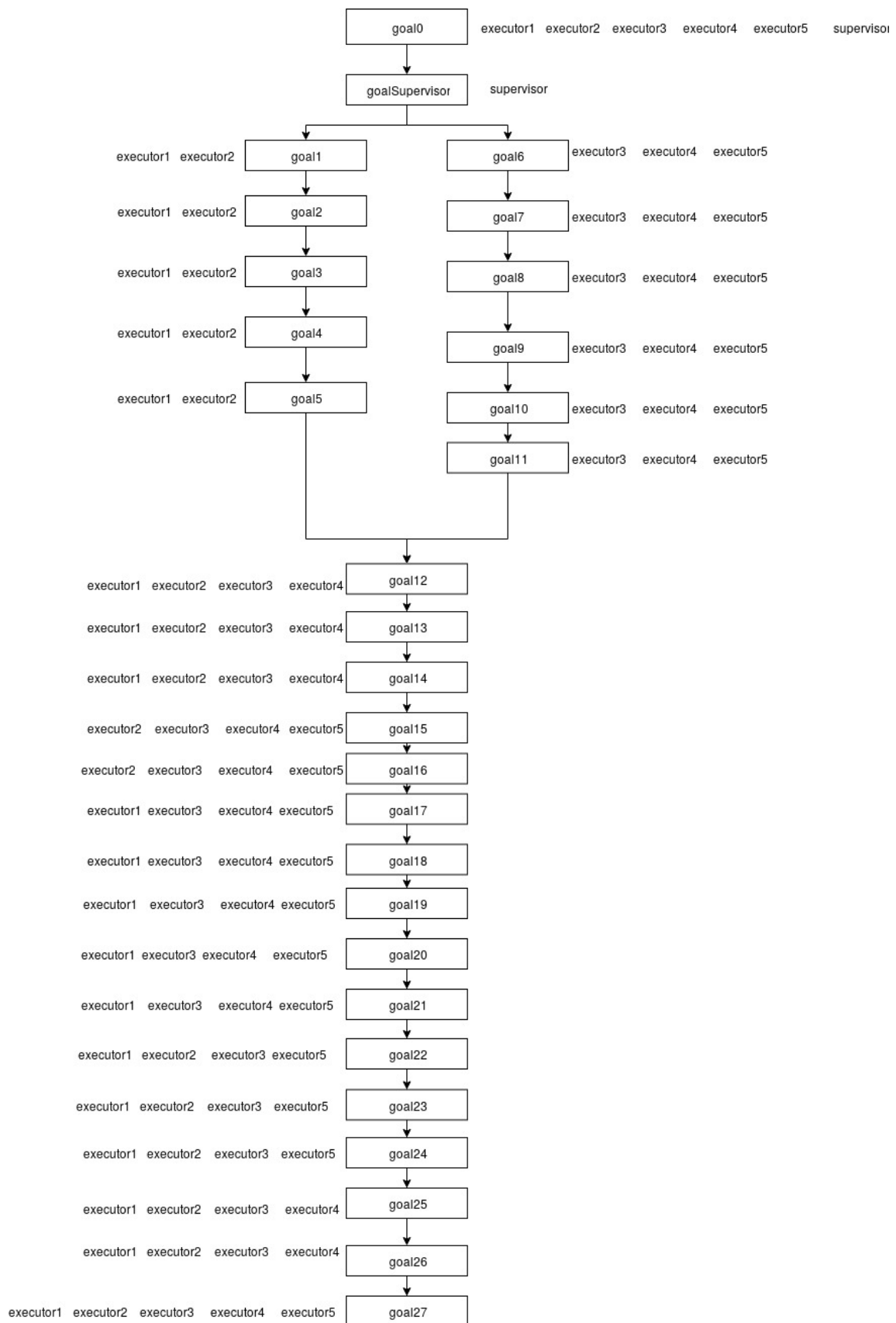
A tabela 15 define as relações  $hasRelation(g_i, rg_n)$  onde a coluna objetivo é representada por  $g_i$ ,  $hasEntity(g_i, eg_m)$ ,  $hasCondition(g_i, cg_n)$ .

objetivo	rg	eg	cg
goal0	rg0	eg0	cg1
goal0	rg0	eg0	cg1
goal1	rg1	eg1	cg1
goal2	rg2	eg2	cg1
goal3	rg3	eg3	cg1
goal4	rg4	eg4	cg1
goal5	rg5	eg5	cg1
goal6	rg6	eg6	cg1
goal7	rg7	eg7	cg1
goal8	rg8	eg8	cg1
goal9	rg9	eg9	cg1
goal10	rg10	eg10	cg1
goal11	rg11	eg11	cg1
goal12	rg12	eg12	cg1
goal13	rg13	eg13	cg1
goal14	rg14	eg14	cg1
goal15	rg15	eg15	cg1
goal16	rg16	eg16	cg1
goal17	rg17	eg17	cg1
goal18	rg18	eg18	cg1
goal19	rg19	eg19	cg1
goal20	rg20	eg20	cg1
goal21	rg21	eg21	cg1
goal22	rg22	eg22	cg1
goal23	rg23	eg23	cg1
goal24	rg24	eg24	cg1
goal25	rg25	eg25	cg1
goal26	rg26	eg26	cg1
goal27	rg27	eg27	cg1

**Tabela 15.** Define a relação entre os objetivos, conjuntos  $rg_n$ ,  $eg_n$  e  $cg_n$

A figura 3 o sequenciamento de objetivos, relacionamentos de obrigação entre objetivos e papeis que fazem parte da natureza do atividade dentro da perspectiva deste modelo.





**Figura 3.**

## 4. Raciocínios

Uma vez que o modelo foi definido e que foi implementado em um estudo de caso, é possível avaliar as conclusões possíveis dado certa condição de mundo. Essa seção demonstra como esse modelo cumpre o proposto por demonstrar certos raciocínios tendo em vista o caso de estudo em análise.

### 4.0.1. Raciocínio - 1

O raciocínio a seguir mostra o que acontece se o *agente4* esquecer de passar o pano no bastão universal *relPanoGlicerina* designados a ele no objetivo *goal1*. Todos os possíveis predicados vinculados a essa situação são;

1. *hasRole*(*agente4*, *executor2*)
2. *hasObligation*(*executor2*, *g1*)
3. *hasRelation*(*g1*, *rg1*)
4. *relPanoCorda*  $\in$  *rg1*
5. *x* = *agente4*
6. *tryReach*(*agente4*, *goal1*)
7. *affects*(*relPanoGlicerina*, *relBastaoGarraCondutor*)
8. *affects*(*relPanoGlicerina*, *relCordaEstropo*)
9. *affects*(*relPanoGlicerina*, *relChaveCatracaParaFuso*)
10. *affects*(*relPanoGlicerina*, *relParaFusoConector*)
11. *affects*(*relPanoGlicerina*, *relSoqueteParaFuso*)
12. *affects*(*relPanoGlicerina*, *relParaFusoTorre*)
13. *affects*(*relPanoGlicerina*, *relXCorda*)
14. *affects*(*relPanoGlicerina*, *relEstropoCorda*)

Com base nisso, as relações de implicabilidade resultantes são;

$$\begin{aligned} & \text{hasRelation}(\text{goal1}, \text{rg1}) \wedge \neg \text{isPresent}(\text{relPanoGlicerina}) \\ & \wedge (\text{relPanoGlicerina} \in \text{rg1}) \wedge \text{tryReach}(\text{agente4}, \text{goal1}) \\ & \rightarrow \\ & \text{violationRelation}(\text{agente4}, \text{g1}, \text{relPanoGlicerina}) \end{aligned} \tag{12}$$

$$\begin{aligned} & \text{violationRelation}(\text{agente4}, \text{g1}, \text{relPanoGlicerina}) \\ & \wedge \text{affects}(\text{relPanoGlicerina}, \text{relBastaoGarraCondutor}) \\ & \wedge \text{hasPossibility}(\text{relBastaoGarraCondutor}, \text{false}) \\ & \rightarrow \\ & \text{hasPossibility}(\text{relBastaoGarraCondutor}, \text{true}) \end{aligned} \tag{13}$$

$$\begin{aligned}
&violationRelation(agente4, g1, relPanoGlicerina) \\
&\quad \wedge affects(relPanoGlicerina, relCordaEstropo) \\
&\quad \wedge hasPossibility(relCordaEstropo, false) \\
&\quad \rightarrow \\
&\quad hasPossibility(relCordaEstropo, true)
\end{aligned} \tag{14}$$

$$\begin{aligned}
&violationRelation(agente4, g1, relPanoGlicerina) \\
&\quad \wedge affects(relPanoGlicerina, relParafusoConector) \\
&\quad \wedge hasPossibility(relParafusoConector, false) \\
&\quad \rightarrow \\
&\quad hasPossibility(relParafusoConector, true)
\end{aligned} \tag{15}$$

$$\begin{aligned}
&violationRelation(agente4, g1, relPanoGlicerina) \\
&\quad \wedge affects(relPanoGlicerina, relSoqueteParafuso) \\
&\quad \wedge hasPossibility(relSoqueteParafuso, false) \\
&\quad \rightarrow \\
&\quad hasPossibility(relSoqueteParafuso, true)
\end{aligned} \tag{16}$$

$$\begin{aligned}
&violationRelation(agente4, g1, relPanoGlicerina) \\
&\quad \wedge affects(relPanoGlicerina, relParafusoTorre) \\
&\quad \wedge hasPossibility(relParafusoTorre, false) \\
&\quad \rightarrow \\
&\quad hasPossibility(relParafusoTorre, true)
\end{aligned} \tag{17}$$

$$\begin{aligned}
&violationRelation(agente4, g1, relPanoGlicerina) \\
&\quad \wedge affects(relPanoGlicerina, relAgente4Corda) \\
&\quad \wedge hasPossibility(relAgente4Corda, false) \\
&\quad \rightarrow \\
&\quad hasPossibility(relAgente4Corda, true)
\end{aligned} \tag{18}$$

$$\begin{aligned}
& violationRelation(agente4, g1, relPanoGlicerina) \\
& \wedge affects(relPanoGlicerina, relEstropoCorda) \\
& \wedge hasPossibility(rrelPanoGlicerinaelEstropoCorda, false) \\
& \rightarrow \\
& hasPossibility(relEstropoCorda, true)
\end{aligned} \tag{19}$$

$$\begin{aligned}
& violationRelation(agente4, g1, relPanoGlicerina) \\
& \wedge affects(relPanoGlicerina, relEstropoCorda) \\
& \wedge hasPossibility(relEstropoCorda, false) \\
& \rightarrow \\
& hasPossibility(relEstropoCorda, true)
\end{aligned} \tag{20}$$

#### 4.0.2. Raciocínio - 2

O raciocínio a seguir mostra o que acontece se o pano não estiver presente no local da manutenção quando os eletricitistas forem alcançar o *goal1*. A lista a seguir exibe todos os predicados necessários para averiguar essa condição de mundo.

1. *hasRole*(agente2, executor1)
2. *hasRole*(agente3, executor1)
3. *hasRole*(agente4, executor2)
4. *hasObligation*(executor1, g1)
5. *hasObligation*(executor2, g1)
6. *tryReach*(agente2, goal1)
7. *tryReach*(agente3, goal1)
8. *tryReach*(agente4, goal1)
9. *hasEntity*(g1, eg1)
10. *pano* ∈ *eg1*
11.  $\neg isPresent(pano)$

$$\begin{aligned}
& hasEntity(g1, eg1) \\
& \wedge \neg isPresent(pano) \\
& \wedge (pano \in eg1) \wedge tryReach(agente2, g1) \rightarrow \\
& violationEntity(agent2, g1, pano)
\end{aligned} \tag{21}$$

$$\begin{aligned}
& hasEntity(g1, eg1) \\
& \wedge \neg isPresent(pano) \\
& \wedge (pano \in eg1) \wedge tryReach(agente3, g1) \rightarrow \\
& \quad violationEntity(agente3, g1, pano)
\end{aligned} \tag{22}$$

$$\begin{aligned}
& hasEntity(goal1, eg1) \\
& \wedge \neg isPresent(pano) \\
& \wedge (pano \in eg1) \wedge tryReach(agente4, g1) \rightarrow \\
& \quad violationEntity(agente4, g1, pano)
\end{aligned} \tag{23}$$

$$violationEntity(agente4, goal1, pano) \rightarrow stopIn(goal1) \tag{24}$$

#### 4.0.3. Raciocínio - 3

O raciocínio a seguir mostra o que acontece se o *agente5* tentar alcançar o objetivo *goal11* com a umidade relativa do ar superior a setenta por cento. A lista a seguir exibe todos os predicados necessários para averiguar essa condição de mundo.

1. *hasRole(agente5, executor3)*
2. *hasObligation(executor3, goal11)*
3. *tryReach(agente5, goal11)*
4. *hasCondition(g11, cg1)*
5. *umidade70*  $\in$  *cg1*
6.  $\neg isPresent(umidade70)$
7. *hasRisk(umidade70, eletrocutado, morte)*

$$\begin{aligned}
& hasCondition(goal11, cg1) \\
& \wedge \neg isPresent(umidade70) \\
& \wedge umidade70 \in cg1 \\
& \wedge tryReach(agente5, g11) \rightarrow \\
& \quad violationCondition(agente5, g11, umidade70)
\end{aligned} \tag{25}$$

$$\begin{aligned}
& violationCondition(agente5, goal11, umidade70) \\
& \wedge hasRisk(umidade70, eletrocutado, morte) \rightarrow \\
& \quad consequenceOfBadEvent(g11, agente5, eletrocutado, morte)
\end{aligned} \tag{26}$$

$$consequenceOfBadEvent(g11, agente5, eletrocutado, morte) \rightarrow stopIn(g11) \quad (27)$$

#### 4.0.4. Raciocínio - 4

O raciocínio a seguir mostra o que acontece se o *agente3* errar a forma adequada de realizar o relacionamento *relChaveCatracaParaFuso* no objetivo *goal15*. Os predicados envolvidos são;

1. *hasRole*(*agente4*, *executor2*)
2. *hasObligation*(*executor4*, *goal15*)
3. *tryReach*(*agente4*, *g15*)
4. *hasRelation*(*g15*, *rg15*)
5. *relChaveCatracaParaFuso*  $\in$  *rg15*
6.  $\neg isPresent(relChaveCatracaParaFuso)$
7. *hasRisk*(*relChaveCatracaParaFuso*, *eletrocutado*, *morte*)

$$\begin{aligned}
& hasRelation(goal15, rg15) \\
& \wedge \neg isPresent(relChaveCatracaParaFuso) \\
& \wedge (relChaveCatracaParaFuso \in rg15) \\
& \wedge tryReach(agente4, g15) \\
& \rightarrow \\
& violationRelation(agente4, g15, relChaveCatracaParaFuso)
\end{aligned} \quad (28)$$

$$\begin{aligned}
& violationRelation(agente4, g15, relChaveCatracaParaFuso) \\
& \wedge hasRisk(relChaveCatracaParaFuso, eletrocutado, morte) \\
& \rightarrow \\
& consequenceOfBadEvent(g15, agente4, eletrocutado, morte)
\end{aligned} \quad (29)$$

$$consequenceOfBadEvent(g15, agente4, eletrocutado, morte) \rightarrow stopIn(g15) \quad (30)$$

#### 4.0.5. Raciocínio - 5

A finalidade dessa demonstração consiste em mostrar como um agente pode ser submetido a consequencias ruins tendo em vista erros cometidos por outros profissionais. O raciocínio 1 mostra que o fato do *agente4* não conseguir realizar o relacionamento *relPanoGlicerina* resulta na violação

$violationRelation(agent4, goal1, relPanoGlicerina)$ . Essa violação, por sua vez, impacta diversas outras relações, em que  $hasPossibility(relPara fusoTorre, true)$  é uma delas. Assim sendo, antes do  $agent4$  comenter o erro, a possibilidade da ocorrência de um evento ruim acontecer era 0, se o agente realizar a relação  $relPara fusoTorre$  sem cometer violação alguma. Contudo, após a ocorrência do erro cometido pelo  $agent4$ , existe uma possibilidade de um evento ruim acontecer na relação  $relPara fusoTorre$  mesmo que tudo seja feito de acordo com os conformes. Assim sendo, a lista de predicados e o raciocínio mostra o que acontece dado a seguinte situação; o possível evento ruim presente em  $relPara fusoTorre$  se torna uma realidade;

1.  $relPara fusoTorre \in rg19$
2.  $hasRelation(g19, rg19)$
3.  $hasObligation(executor3, g19)$
4.  $hasObligation(executor4, g19)$
5.  $hasObligation(executor5, g19)$
6.  $tryReach(agent5, g19)$
7.  $tryReach(agent6, g19)$
8.  $tryReach(agent7, g19)$
9.  $hasRole(agent5, executor3)$
10.  $hasRole(agent6, executor4)$
11.  $hasRole(agent7, executor5)$
12.  $hasRisk(relPara fusoTorre, eletrocutado, morte)$
13.  $hasPossibility(relPara fusoTorre, true)$
14.  $happensBadEvent(g19, relPara fusoTorre)$

$$\begin{aligned}
& happensBadEvent(g19, relPara fusoTorre) \\
& \wedge hasRisk(relPara fusoTorre, eletrocutado, morte) \\
& \wedge tryReach(agent5, g19) \\
& \rightarrow \\
& consequenceOfBadEvent(goal19, agent5, eletrocutado, morte) \quad (31)
\end{aligned}$$

$$consequenceOfBadEvent(g19, agent5, eletrocutado, morte) \rightarrow stopIn(g19) \quad (32)$$

$$\begin{aligned}
& happensBadEvent(g19, relPara fusoTorre) \\
& \wedge hasRisk(relPara fusoTorre, eletrocutado, morte) \\
& \wedge tryReach(agent6, g19) \\
& \rightarrow \\
& consequenceOfBadEvent(goal19, agent6, eletrocutado, morte) \quad (33)
\end{aligned}$$

$$consequenceOfBadEvent(g19, agent6, eletrocutado, morte) \rightarrow stopIn(g19) \quad (34)$$

$$\begin{aligned}
& happensBadEvent(g19, relPara fusoTorre) \\
& \wedge hasRisk(relPara fusoTorre, eletrocutado, morte) \\
& \wedge tryReach(agente7, g19) \\
& \rightarrow \\
& consequenceOfBadEvent(goal19, agente7, eletrocutado, morte)
\end{aligned} \tag{35}$$

$$consequenceOfBadEvent(g19, agente7, eletrocutado, morte) \rightarrow stopIn(g19) \tag{36}$$

#### 4.0.6. Raciocínio - 6

O objetivo  $g23$  deve ser atingido pelos agentes com as funções de  $executor1, executor2, executor3$  e  $executor5$ . Isso implica dizer que os agentes;  $agente2, agente3, agente4, agente5$  e  $agente7$  devem tentar alcançar esses resultados. Considerando que  $agg23$  são todos os agentes que tentaram alcançar o objetivo e  $ago23$  os agentes que são obrigados a fazer isso, segue o raciocínio;

1.  $agente2 \in agg23$
2.  $agente3 \in agg23$
3.  $agente4 \in agg23$
4.  $agente5 \in agg23$
5.  $agente7 \in agg23$
6.  $agente2 \in ago23$
7.  $agente3 \in ago23$
8.  $agente4 \in ago23$
9.  $agente5 \in ago23$
10.  $agente7 \in ago23$
11.  $agg23 \subset ago23$
12.  $\neg stopIn(g23, agg23)$

$$\neg stopIn(g23, agg23) \wedge (agg23 \subset ago23) \rightarrow isReached(g23) \tag{37}$$

#### 4.0.7. Raciocínio - 7

O raciocínio para o caso onde  $agente1$  tente alcançar o objetivo  $g23$ .

1.  $hasRole(agente1, supervisor)$
2.  $hasObligation(agente1, g23) \rightarrow F$

Isso implica em uma afirmação falsa, então esse mundo não é possível segundo o modelo implementado para este estudo de caso.



## 4.1. Trabalhos Correlatos

Essa seção tem como finalidade realizar uma análise comparativa do modelo proposto neste estudo com os modelos atuais sobre sistemas multiagentes normativos. Cada uma das próximas seções aborda um modelo diferente.

### 4.1.1. MOISE+

O Moise+ é usado para realizar a especificação de sistemas multiagentes. Para cumprir com essa finalidade, existe três tipos de especificação que são; Estrutural, Funcional, e Deôntica.

A especificação estrutural acontece em três níveis, individual, social e coletivo. O nível individual trata de definir os papéis  $\rho$  dos agentes. Uma possível entre os papéis acontece por intermédio da hereditariedade em que se  $\rho'$  é filho de  $\rho$ . Isso implica afirmar que  $\rho'$  é uma especialização de  $\rho$ . Um exemplo apropriado para isso é o jogo de futebol onde existe o papel jogador dado por  $\rho$  e existe o papel atacante dado por  $\rho'$  [Hübner et al. 2002]. Em termos formais, essa relação é dada por;

$$\rho_a \sqsubset \rho_b$$

O nível social estabelece relações de ligação dado pelo predicado  $link(\rho_s, \rho_d, t)$ . Existe três possíveis valores para  $t$ , os quais são  $t = \{aut, com, acq\}$ . O valor *auth* significa autoridade (neste caso  $\rho_s$  exerce autoridade sobre  $\rho_d$ ), o valor *com* significa comunicação (neste caso  $\rho_s$  pode se comunicar com  $\rho_d$ ) e o valor *acq* significa conhecimento ( $\rho_s$  tem conhecimento da existência de  $\rho_d$ ) [Hübner et al. 2002]. O MOISE+ define as seguintes relações de implicabilidade

$$\begin{aligned} link(\rho_s, \rho_d, auth) &\rightarrow link(\rho_s, \rho_d, com) \\ link(\rho_s, \rho_d, com) &\rightarrow link(\rho_s, \rho_d, acq) \end{aligned} \quad (38)$$

O modelo também determina como se dá as relações de hereditariedade para o predicado de *link*, é dado por [Hübner et al. 2002];

$$\begin{aligned} link(\rho_s, \rho_d, t) \wedge \rho'_s \sqsubset \rho'_s &\rightarrow link(\rho'_s, \rho_d, t) \\ link(\rho_s, \rho_d, t) \wedge \rho'_d \sqsubset \rho'_d &\rightarrow link(\rho_s, \rho'_d, t) \end{aligned} \quad (39)$$

O nível coletivo determina a existência de compatibilidade entre os papéis [Hübner et al. 2002]. Essa é uma relação reflexiva e transitiva de determina que se um papel  $\rho_a$  possui a capacidade de realizar um determinado objetivo, então o papel  $\rho_b$  também tem essa capacidade. Em termos formais, essa relação se dá da seguinte forma [Hübner et al. 2002].;

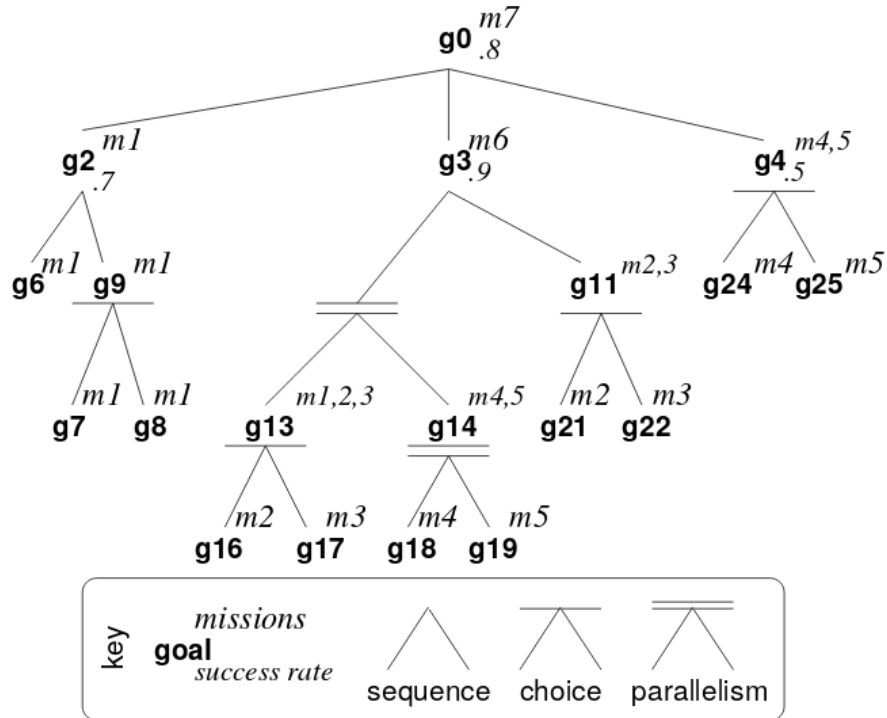
$$\rho_a \bowtie \rho_b \wedge \rho_a \neq \rho_b \wedge \rho_a \sqsubset \rho' \rightarrow \rho' \bowtie \rho_b$$

O nível coletivo também apresenta o conceito de grupo dado por  $gt$  e constituído por;

$$gt = \langle R, SG, L^{intra}, L^{inter}, C^{intra}, C^{inter}, np, ng \rangle$$

Em que  $R$  é o conjunto dos papéis não abstratos,  $SG$  são subgrupos que estão contidos neste grupo,  $L^{intra}$  consiste dos *links* intra-grupos,  $L^{inter}$  dos links inter-grupos,  $C^{intra}$  das relações de compatibilidade intra-grupos e  $C^{inter}$  das relações de compatibilidade inter-grupos. O símbolo  $np$  denota a cardinalidade mínima e máxima para uma dada função e o símbolo  $ng$  realiza o mesmo para os subgrupos [Hübner et al. 2002].

A Especificação Funcional tem como por finalidade descrever os objetivos a serem atingidos dentro de uma estrutura de árvore. A figura a seguir define como se dá esse tipo de especificação;



**Figura 4. Árvore de objetivos definido pelo modelo Moise [Hübner et al. 2002]**

A figura 4 define três tipos de relação de subobjetivos; *sequence* onde todos os subobjetivos devem necessariamente ser concluídos em seqüência, *choice* onde o agente tem a possibilidade de escolher qual objetivo ele deseja seguir e *parallelism* onde todos os objetivos devem ser concluídos, contudo sem uma seqüência definida. Como é possível observar na figura, os objetivos são agrupados em conjuntos de missões  $m$ . A relação a seguir define isso melhor;

$$m_k = \{g_n, \dots, g_m\}$$

A Especificação Deontica define predicados para estabelecer permissões e obrigações entre os papéis e as missões. Toda obrigação implica necessariamente em uma permissão. A relação a seguir estabelece isso;

$$\begin{aligned} obl(\rho, m, tc) &\rightarrow per(\rho, m, tc) \\ obl(\rho, m, tc) \wedge \rho \sqsubset \rho' &\rightarrow obl(\rho', m, tc) \end{aligned} \quad (40)$$

$$per(\rho, m, tc) \wedge \rho \sqsubset \rho' \rightarrow per(\rho', m, tc) \quad (41)$$

$$(42)$$

Onde o predicado *obl* define uma obrigação e o predicado *per* define permissão. O argumento *tc* define uma periodicidade de tempo para o qual a relação deôntica é válida.

Existe muitos ponto similares entre o modelo proposto neste estudo e o Moise. Ambos os modelos apresentam papéis  $\rho$ , apresentam objetivos  $g$  e apresentam relações deonticas de obrigação e permissão. Isso se deve ao fato, em partes, que o modelo proposto neste estudo importou muitos dos conceitos presentes no Moise tendo em vista a aplicabilidade para o domínio em interesse. Esses conceitos são importantes para o modelo deste estudo pois são grande relevância para descrever situações onde uma equipe de pessoas devem atingir certos objetivos trabalhando em cooperação.

Contudo, ambos modelos apresentam diferenças significativas. Uma dessas diferenças reside em como as relações deônticas são atribuídas, pois no Moise a relação é feita entre o papel e a missão e no modelo proposto neste estudo a relação é feita entre o papel e o objetivo. Isso se deve ao fato de que uma das finalidades deste estudo consiste na realização de uma análise de sanções e violações. Como essas análises trabalham no nível de atividades, objetos e condições (dentro do contexto deste estudo), trabalhar na ordem de objetivos trás uma análise mais aprimorada no estudo das violações e sanções.

O Moise não apresenta suporte a trativa de sanções e violações. Assim sendo, para conseguir atingir um modelo onde fosse possível analisar certos tipos de sanções e violações de interesse, foi necessário introduzir predicados que não fazem parte do vocabulário e da sintaxe do Moise.

O Moise trabalha uma estrutura lógica de interesse a descrição de grupos, *links* e compatibilidades. Esses conceitos não são trabalhados no modelo em interesse por não serem necessários ao domínio de interesse deste estudo. Assim sendo, esses conceitos trariam complexidades adicionais sem justificativa válida para isso.

#### 4.2. Normative Multi-Agent Programs and Their Logics

O texto [Dastani et al. 2009] apresenta um modelo com a finalidade de descrever agentes normativos com a capacidade de cometer uma certa violação. O modelo define sanções para as violações. O estudo estrutura o modelo como uma linguagem de programação (usando a notação EBNF) onde os problemas de sistemas multiagentes são tratados como programas escritos neste linguagem [Dastani et al. 2009]. Na perspectiva desta linguagem, um programa de sistemas multi-agentes é descrito como sendo;

```

N-MAS_Prog  := "Agents: " (<agentName> <agentProg> [<nr>])+ ;
              "Facts: " <bruteFacts>
              "Effects: " <effects>
              "Counts-as rules: " <counts-as>
              "Sanction rules: " <sanctions>;

<agentName> := <ident>;
<agentProg> := <ident>;
<nr>        := <int>;
<bruteFacts> := <b-literals>;
<effects>    := ({<b-literals>} <actionName> {<b-literals>})+;
<counts-as>  := ( <literals> ⇒ <i-literals> )+;
<sanctions>  := ( <i-literals> ⇒ <b-literals> )+;
<actionName> := <ident>;
<b-literals> := <b-literal> {", " <b-literal>};
<i-literals> := <i-literal> {", " <i-literal>};
<literals>   := <literal> {", " <literal>};
<literal>    := <b-literal> | <i-literal>;
<b-literal>  := <b-prop> | "not" <b-prop>;
<i-literal>  := <i-prop> | "not" <i-prop>;

```

**Figura 5. Linguagem para descrever um programa de multiagentes normativos com a possibilidade de violações e sanções na notação EBNF segundo o texto [Dastani et al. 2009]. Nesta notação, *< ident >* é usado para denotar uma *string* e *< int >* inteiros. Os termos *< b – prop >* e *< i – prop >* são usados para designar dois tipos de conjuntos de proposições que são disjuntos entre si**

Com base no proposto por esse modelo, um programa de sistemas multiagentes é descrito por *Agents* (agentes), *Facts* (fatos), *Effects* (efeitos), (*Count-as rules*) (regras que determinam o adequado comportamento do agente), *Sanction rules* (regras de sanção) [Dastani et al. 2009].

Os *Agents* são definidos em termos de duas *strings* e um inteiro. A primeira *string* é *agentName* e é usado para definir o nome do agente e a segunda *string* é *agentProg* e define o nome do arquivo onde se encontra especificações do respectivo agente. O inteiro *nr* é usado para definir a quantia do agente [Dastani et al. 2009].

Os *Facts* são compostos por conjuntos de literais denominados de *bruteFacts*. Esses são fatos onde ocorre uma violação que desencadeia em uma dada sanção [Dastani et al. 2009].

Os *Effects* são compostos por *effects*. Estes, por sua vez, são estruturados em termos de *b-literals* (conjuntos que contem literais onde estes, por sua vez, representam um dado estado de mundo), e *actionName* onde este, por sua vez, descreve ações que geram transições de um estado de mundo para outro estado [Dastani et al. 2009].

Os *Count-as rules* são compostos por *counts-as*. Esses, por sua vez, definem regras normativas. Isso implica em relações de implicabilidade que resultam em uma dada violação [Dastani et al. 2009].

Os *Sanction rules* são estruturados por *sanctions*. Esses, por sua vez, definem regras de implicabilidade que competem a uma tratativa da violação [Dastani et al. 2009].

A figura a seguir ilustra um exemplo de um programa de sistema multiagente

escrito nesta linguagem;

```
Agents:      passenger  PassProg  1
Facts:       {-at_platform, -in_train, -ticket}
Effects:     {-at_platform} enter {at_platform},
            {-ticket} buy_ticket {ticket},
            {at_platform, -in_train} embark {-at_platform, in_train}
Counts_as rules: {at_platform, -ticket} ⇒ {viol1},
                {in_train, -ticket} ⇒ {viol⊥}
Sanction rules: {viol1} ⇒ {fined10}
```

**Figura 6. Um programa descrito na linguagem proposta neste estudo onde um agente representa um passageiro em uma estação de trem que pode entrar com ou sem um *ticket* na plataforma e no trem [Dastani et al. 2009].**

## Referências

- Barot, C., Lourdeaux, D., Burkhardt, J., Amokrane, K., and Lenne, D. (2013). V3S: A virtual environment for risk-management training based on human-activity models. *Presence*, 22(1):1–19.
- Boissier, O. (2011). From organisation oriented programming to multi-agent oriented programming. In Klügl, F. and Ossowski, S., editors, *Multiagent System Technologies*, pages 1–1, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Dastani, M., Grossi, D., Meyer, J.-J. C., and Tinnemeier, N. (2009). Normative multi-agent programs and their logics. In Meyer, J.-J. C. and Broersen, J., editors, *Knowledge Representation for Agents and Multi-Agent Systems*, pages 16–31, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Hübner, J. F., Sichman, J. S., and Boissier, O. (2002). A model for the structural, functional, and deontic specification of organizations in multiagent systems. In Bittencourt, G. and Ramalho, G. L., editors, *Advances in Artificial Intelligence*, pages 118–128, Berlin, Heidelberg. Springer Berlin Heidelberg.
- López y López, F. and Luck, M. (2004). A model of normative multi-agent systems and dynamic relationships. In Lindemann, G., Moldt, D., and Paolucci, M., editors, *Regulated Agent-Based Social Systems*, pages 259–280, Berlin, Heidelberg. Springer Berlin Heidelberg.
- von Wright, G. H. (1969). *On The Logic and Ontology of Norms*, pages 89–107. Springer Netherlands, Dordrecht.