

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, 1 observador e 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, locador 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. Enforçar 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. Baixaer 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. $AgentGoal = agg = \{ag_1, \dots, ag_n\}$ - conjunto dos Agentes que devem executar um determinado objetivo.
4. $Artefact = \{at_1, \dots, at_n\}$ - conjunto dos Artefatos.
5. $EntityGoal = eg = \{e_n, \dots, e_m\}$ - conjunto das Entidades que devem estar presentes para concluir um determinado objetivo g_i .
6. $Relation = \{r_1, \dots, r_n\}$ - conjunto dos Relacionamentos.
7. $RelationGoal = rg = \{r_n, \dots, r_m\}$ - conjunto dos Relacionamentos que devem estar presentes para concluir um único objetivo g_i .
8. $Role = \{\rho_1, \dots, \rho_n\}$ - conjunto dos Papéis.
9. $Goal = \{g_1, \dots, g_n\}$ - conjunto dos Objetivos.
10. $GoalRandomic = \{gr_1, \dots, gr_n\}$ - conjunto dos Objetivos Randomicos.
11. $GoalPrerequisite = gp = \{g_n, \dots, g_m\}$ - conjunto de Objetivos que são pré-requisitos para alcançar um outro objetivo.
12. $Condition = \{c_1, \dots, c_n\}$ - conjunto das Condições que devem ser mantidas ao longo da execução de todos os objetivos.
13. $ConditionToGoal = cg = \{c_n, \dots, c_m\}$ - conjunto de condições que devem ser mantidas para concluir um único objetivo g_i .
14. $Risk = \{risk_1, \dots, risk_n\}$ - conjunto dos Riscos na ocorrência de Eventos Ruins.
15. $Possibility = \{p_1, \dots, p_n\}$ - conjunto das possibilidades de Eventos Ruins.
16. $Fatality = \{f_1, \dots, f_n\}$ - conjunto das fatalidades que acontecem na existência de um evento ruim.
17. $agg = \{ag_n, \dots, ag_m\}$ - agentes que atingiram um determinado objetivo.
18. $ago = \{ag_n, \dots, ag_m\}$ - 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 ρ_m .
3. $hasObligation(\rho_m, g_j)$ - Quem assume o papel ρ_m é obrigado a concluir o objetivo g_j .
4. $hasPermission(\rho_m, g_j)$ - Quem assume o papel ρ_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. $hasNextGoal(g_i, g_j)$ - Define qual é o proximo objetivo que o agente deve buscar depois que o objetivo g_i for alcançado.
13. $isPresent(X)$, $X = cg_n, c_k, rg_k, r_k, eg_k, e_k$ - Define se X está presente no instante em análise, sendo que X pode ser $cg_n, c_k, rg_k, r_k, eg_k, e_k$.
14. $tryReach(ag_i, g_j)$ - Um determinado agente ag_i tenta alcançar o objetivo g_j .
15. $violationCondition(ag_i, g_j, c_k)$ - Um determinado agente ag_i comete uma violação de condição no objetivo g_j sobre a condição c_k .
16. $violationRelation(ag_i, g_j, r_k)$ - O agente ag_i comete uma violação de Relacionamento no objetivo g_j por não realizar o relacionamento r_k .
17. $violationEntity(ag_i, g_j, e_k)$ - 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.
18. $hasRisk(c_k, risk_j, f_m)$ - A condição c_k está associada a um risco $risk_k$ com uma certa fatalidade f_m .

19. $hasRisk(r_k, risk_j, f_m)$ - O relacionamento r_k está associado a um risco $risk_k$ com uma certa fatalidade f_m .
20. $hasRisk(e_k, risk_j, f_m)$ - A entidade e_k está associada a um risco $risk_k$ com uma certa fatalidade f_m .
21. $consequenceOfBadEvent(g_k, ag_i, risk_j, f_m)$ - Agente ag sofre as consequências do risco $risk_j$ com a fatalidade f_m .
22. $hasPossibility(gr_n, p_m)$ - Possibilidade p_m do evento gr_n gerar alguma consequência ruim.
23. $affects(r_k, gr_n, p_n)$ - Se uma relação r_k não for feito, ou se essa relação for mal feita, então ela afeta negativamente algum objetivo com caráter aleatório mudando a possibilidade para p_n de um evento ruim acontecer.
24. $happensBadEvent(gr_n, r_m, ag_k)$ - O evento ruim de gr 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 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.

$$\begin{aligned} & violationRelation(ag_m, g_i, r_k) \wedge affects(r_k, gr_n, p_n) \wedge hasPossibility(gr_n, p_m) \rightarrow \\ & hasPossibility(gr_n, p_n) \wedge (p_n > p_m) \end{aligned} \quad (7)$$

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

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 eletricitista deve medir a condutividade de um bastão universal é uma situação que pode exemplificar a aplicação desta relação. Se o eletricitista 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 11

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. Raciocínios

3.7.1. Raciocínio 1

Considerando que existe uma dada situação onde há uma relação para duas entidades; $relationHas(r_k, e_i, e_l)$. Supondo que essa relação está presente em uma condição que ocasiona violação de relacionamento dado por 3.

Neste caso, é possível dar como verdade a ocorrência dos seguintes predicados e relações; $hasRelation(g_i, rg_n), (r_k \in rg_n), tryReach(ag_m, g_i)$. Assim sendo, é verdade que; $tryReach(ag_m, g_i) \wedge (r_k \in rg_n) \wedge hasRelation(g_i, rg_n) \wedge relationHas(r_k, e_i, e_l)$. Portanto, é possível o seguinte raciocínio;

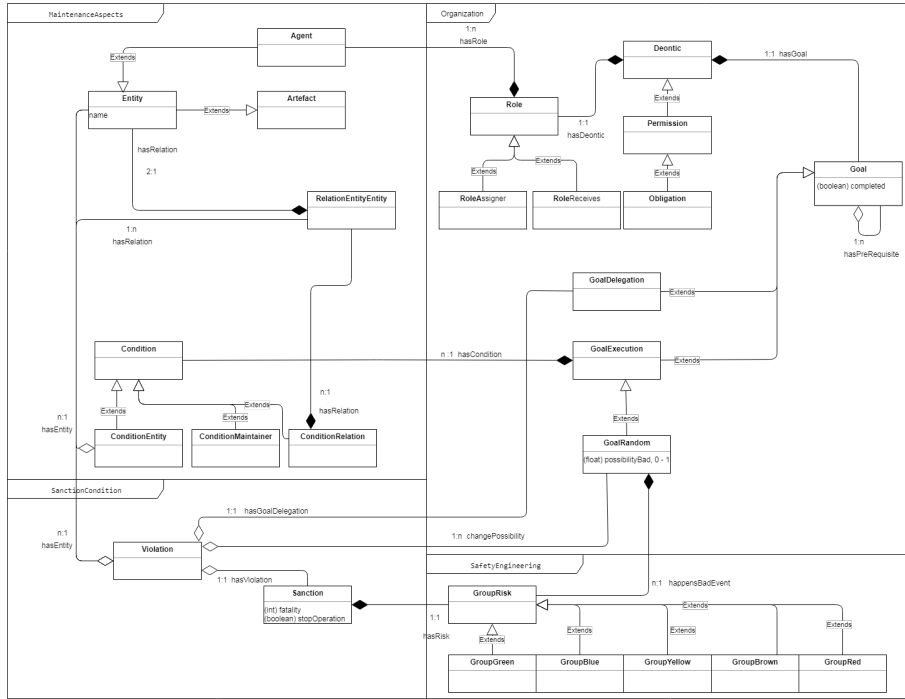
$$\begin{aligned}
 & hasRelation(g_i, rg_n) \wedge (r_k \in rg_n) \wedge tryReach(ag_m, g_i) \rightarrow T \\
 & relationHas(r_k, e_i, e_l) \rightarrow T \\
 & (r_k \in rg_n) \wedge relationHas(r_k, e_i, e_l) \rightarrow T, \\
 & relationHas(r_k, e_i, e_l) \wedge hasRelation(g_i, rg_n) \wedge (r_k \in rg_n) \wedge tryReach(ag_m, g_i) \rightarrow T
 \end{aligned} \tag{12}$$

Para um dado relacionamento que está sendo montado por um agente na tentativa de alcançar um certo objetivo, então as entidades atreladas a esse relacionamento devem estar presentes também. Considerando isso, para toda situação que se considera a relação 3, também se faz necessário considerar, para cada uma das entidades, 4. Assumindo isso, podemos concluir o seguinte;

$$\begin{aligned}
 & (relationHas(r_k, e_i, e_l) \wedge \neg violationRelation(ag_m, g_i, r_k) \rightarrow \\
 & \quad \neg violationEntity(ag_m, g_i, e_i)) \\
 & \quad \wedge \\
 & (relationHas(r_k, e_i, e_l) \wedge \neg violationRelation(ag_m, g_i, r_k) \rightarrow \\
 & \quad \neg violationEntity(ag_m, g_i, e_l))
 \end{aligned} \tag{13}$$

3.7.2. Raciocínio 2

Dado um cenário com a seguinte situação; $relationHas(r_k, e_i, e_l), hasRelation(g_i, rg_n), hasEntity(g_i, eg_m), rg_n, e_i \in eg_m$, neste caso é necessário que $e_l \in eg_m$. É possível considerar o seguinte



raciocínio

$$\begin{aligned}
 & violationEntity(ag_i, g_j, e_i) \rightarrow T \\
 & \neg isPresent(e_i) \rightarrow T \\
 & \neg isPresent(e_i) \wedge relationHas(r_k, e_i, e_l) \rightarrow \neg isPresent(r_k), \\
 & violationRelation(ag_i, g_j, r_k) \rightarrow T
 \end{aligned} \tag{14}$$

$$\begin{aligned}
 & violationEntity(ag_i, g_j, e_l) \rightarrow T \\
 & \neg isPresent(e_l) \rightarrow T \\
 & \neg isPresent(e_l) \wedge relationHas(r_k, e_i, e_l) \rightarrow \neg isPresent(r_k), \\
 & violationRelation(ag_i, g_j, r_k) \rightarrow T
 \end{aligned} \tag{15}$$

Ou seja, para esse cenário, a ocorrência de uma das violações de entidade tem como por consequência a ocorrência de violação de relacionamento.

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.
- 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.