

Definição 0

Definição 0.1

Existe  $E = \{e_1, \dots, e_n\}$ , que corresponde as entidades. Existe dois tipos de entidades, Agentes representados pelo conjunto  $Ag$  e Artefatos representados pelo conjunto  $At$  onde  $(E \equiv Ag \cup At) \wedge (ag_i \in Ag \rightarrow ag_i \notin At) \wedge (at_i \in At \rightarrow at_i \notin Ag)$

Definição 0.2

Existe relações de entidade para entidade que são mapeadas por um conjunto  $R$ . Isso é dado por meio da seguinte relação,  $isRelation(r_k, e_i, e_j)$

Definição 0.3

$\rho_{r_i}$  papel *recebe* ou seja, o papel que recebe a atribuição, de um papel  $\rho_a$ , um papel atribuidor,  $R_r = \{\rho_{r_n}, \dots, \rho_{r_m}\} \wedge R_a = \{\rho_{a_n}, \dots, \rho_{a_k}\}, (R \equiv R_a \cup R_r) \wedge (\rho_m \in R_m \rightarrow \rho_m \notin R_a) \wedge (\rho_a \in R_a \rightarrow \rho_a \notin R_m)$ .

Definição 0.4

Existe um conjunto  $D$  de relações deonticas. Existe  $D_{permission}$  que é equivalente a  $D$  e existe  $D_{obligation}$  que é um subconjunto de  $D_{permission}$ .

Definição 0.5

Existe um conjunto  $G$  de Objetivos. Existe dois tipos de objetivos,  $Ge$  e  $Gd$ , onde ambos são disjuntos.  $Ge$  contem os objetivos referentes  $\rho_r$ , ou seja - referente a quem é delegado para alguma função. Os objetivos de  $Gd$  são referentes a quem delega  $\rho_a$ . Agentes  $\rho_a$  de  $g_{d_k}$  tem a finalidade de atribuir  $\rho_m$  a determinado agente  $ag$ .

Definição 0.6

Existe um subconjunto de  $Ge$  conhecido como  $Gr$ , são objetivos de execução que podem ser afetado por eventos aleatórios com a finalidade de ocasionar um acidente. Esses eventos aleatórios possuem uma possibilidade de acontecer. Os agentes não têm controle sobre esses eventos, contudo, têm controle sobre as possibilidades.

Definição 1

$$attribution(ag, \rho_{r_i}) \wedge deonticRelation(\rho_r, g_{e_i}, d_{o_i}) \rightarrow obligation(ag, g_{e_i}) \quad (1)$$

Definição 2

$$attribution(ag, \rho_{r_i}) \wedge deonticRelation(\rho_{r_i}, g_{e_i}, d_{p_i}) \rightarrow permission(ag, g_{e_i}) \quad (2)$$

Definição 3

$$obligation(ag, g_{e_i}) \rightarrow permission(ag, g_{e_i}) \quad (3)$$

Definição 4

Se para iniciar um determinado objetivo  $g_{e_j}$  é necessário completar outros objetivos,  $g_{e_i}, \dots, g_{e_k}$ , então se pelo menos um destes objetivos for falso para  $isCompleted(g_x), x = i, \dots, k$ , então  $preCondition(g_j, (g_i, \dots, g_k))$  é falso, se não, então  $preCondition(g_j, (g_i, \dots, g_k))$  é verdadeiro.

Sendo que existe um conjunto  $g_{preRequisit} = \{g_i, \dots, g_k\}$ , então

$$(preCondition(g_j, g_{preRequisit}) \rightarrow preRequisit(g_j)) \wedge \quad (4)$$

$$\neg(preCondition(g_j, g_{preRequisit}) \rightarrow \neg preRequisit(g_j)) \quad (5)$$

### Definição 5

Sendo  $cm$  conjunto de condições que devem ser mantidos o tempo inteiro para que a atividade possa acontecer,

$$preRequisit(g_{e_i}) \wedge hasMaintainer(g_{e_i}, cm_i) \wedge isOk(cm_i) \rightarrow canStart(g_{e_i}) \quad (6)$$

### Definição 6

Se  $attribution(ag, \rho_r) \wedge deonticRelation(\rho_r, g_{e_i}, d_{p_i}) \wedge canStart(g_{e_i}) \wedge tryReach(g_{e_i}, ag)$  é verdade, então é verdade que existe  $violation(d_{p_i}, ag)$ . Contudo, se  $attribution(ag, \rho_r) \wedge deonticRelation(\rho_r, g_e, d_{p_i}) \wedge canStart(g_e) \wedge tryReach(g_{e_i}, ag)$  for falso, então  $violation(d_{p_i}, ag)$ .

### Definição 7

#### 7.1

Existe  $c_r$ , onde  $c_r$  é um conjunto de relações das entidades. Cada  $g_{e_i}$  possui um  $c_{r_i}$  dada pela relação  $hasConditionRelation(g_{e_i}, c_{r_i})$ . Se  $permission(ag, g_{e_i}) \wedge canStart(g_{e_i}) \wedge hasConditionRelation(g_{e_i}, c_{r_i}) \wedge isRelationOf(r_j, c_{r_i}) \wedge \neg know(ag, r_j) \wedge tryReach(ag, g_{e_i})$  é verdade, então é verdade  $violation(r_j, g_{e_i}, ag)$ . Contudo  $violation(r_j, g_{e_i}, ag)$  é falso para a condição de  $permission(ag, g_{e_i}) \wedge canStart(g_{e_i}) \wedge hasConditionRelation(g_{e_i}, c_{r_i}) \wedge isRelationOf(r_j, c_{r_i}) \wedge \neg know(ag, r_j) \wedge tryReach(ag, g_{e_i})$  ser falsa.

#### 7.2

Existe  $c_r$ , onde  $c_r$  é um conjunto de relações das entidades. Cada  $g_{e_i}$  possui um  $c_{r_i}$  dada pela relação  $hasConditionRelation(g_{e_i}, c_{r_i})$ . Se  $permission(ag, g_{e_i}) \wedge canStart(g_{e_i}) \wedge hasConditionRelation(g_{e_i}, c_{r_i}) \wedge isRelationOf(r_j, c_{r_i}) \wedge \neg execute(ag, r_j) \wedge tryReach(ag, g_{e_i})$  é verdade, então é verdade  $violation(r_j, g_{e_i}, ag)$ . Contudo  $violation(r_j, g_{e_i}, ag)$  é falso para a condição de  $permission(ag, g_{e_i}) \wedge canStart(g_{e_i}) \wedge hasConditionRelation(g_{e_i}, c_{r_i}) \wedge isRelationOf(r_j, c_{r_i}) \wedge \neg execute(ag, r_j) \wedge tryReach(ag, g_{e_i})$  ser falsa.

### Definição 8

Se  $attribution(ag, \rho_r) \wedge deonticRelation(\rho_r, g_{e_i}, d_{o_i}) \wedge canStart(g_{e_i}) \wedge \neg tryReach(ag, g_{e_i})$  então existe uma violação dada como  $violation(d_{o_i}, ag, g_{e_i})$ . Se não  $violation(d_{o_i}, ag, g_{e_i})$  é falso.

### Definição 9

Existe um conjunto  $c_e$ , condition entity, que consiste nas entidades necessárias para que um determinado objetivo possa ser alcançado. Para cada  $g_{e_i}$  existe uma dada relação em que  $hasConditionEntity(g_{e_i}, c_{e_i})$ . Sendo assim,  $permission(ag, g_{e_i}) \wedge tryReach(ag, g_{e_i}) \wedge canStart(g_{e_i}) \wedge hasConditionEntity(g_{e_i}, c_{e_i}) \wedge isEntityOf(e_k, c_{e_i}) \wedge \neg isInMomentOfGoal(e_k, g_{e_i}) \rightarrow violation(g_{e_i}, e_k, ag)$ . Contudo, é verdadeiro  $\neg violation(g_{e_i}, e_k, ag)$  se  $hasConditionEntity(g_{e_i}, c_{e_i}) \wedge tryReach(ag, g_{e_i}) \wedge canStart(g_{e_i}) \wedge hasConditionEntity(g_{e_i}, c_{e_i}) \wedge isEntityOf(e_k, c_{e_i}) \wedge \neg isInMomentOfGoal(e_k, g_{e_i})$  é falso. A relação  $isInMomentOfGoal(e_k, g_{e_i})$  indica se entidade estará disponível no momento que o agente  $ag$  tenta alcançar o objetivo  $g_{e_i}$ .

### Definição 10

$$attribution(ag, \rho_a) \rightarrow obligation(ag, g_d) \quad (7)$$

### Definição 11

Se for verdade  $obligation(ag_i, g_d) \wedge delegate(ag_i, \rho_{r_i}, ag_j) \wedge deonticRelation(\rho_{r_i}, g_{e_i}, d_p) \wedge hasConditionRelation(g_{e_i}, c_{r_i}) \wedge isRelation(r_k, c_{r_i}) \wedge \neg know(ag_j, r_k)$  então ocorre uma violação em  $violation(\rho_a, g_d, ag_i)$ . Se não,  $\neg violation(\rho_a, g_d, ag_i)$  é verdade.

### Definição 12

$$\begin{aligned} violation(d_{p_i}, ag) &\rightarrow stopOperation & (8) \\ \neg violation(d_{p_i}, ag) &\rightarrow \neg stopOperation & (9) \end{aligned}$$

### Definição 13

#### Definição 13.1

$$violation(r_j, g_{e_i}, ag) \wedge isGoalRandon(g_{e_j}) \wedge hasPossibilityBadEvent(g_{e_j}, p_1) \quad (10)$$

$$\rightarrow hasPossibilityBadEvent(g_{e_j}, p_2) \wedge p_2 > p_1 \quad (11)$$

$$\neg violation(r_j, g_{e_i}, ag) \wedge isGoalRandon(g_{e_j}) \wedge hasPossibilityBadEvent(g_{e_j}, p_1) \quad (12)$$

$$\rightarrow hasPossibilityBadEvent(g_{e_j}, p_1) \quad (13)$$

#### Definição 13.2

$$violation(r_j, g_{e_i}, ag) \rightarrow sanction(g_{e_i}, ag, risk) \rightarrow badEvent(risk, fatality, ag, g_{e_i}) \quad (14)$$

$$\neg violation(r_j, g_{e_i}, ag) \rightarrow \neg sanction(g_{e_i}, ag, risk) \rightarrow \neg badEvent(risk, fatality, ag, g_{e_i}) \quad (15)$$

#### Definição 13.3

$$hasPossibilityBadEvent(g_{e_j}, p) \wedge eventBadHappens(g_{e_j}) \wedge tryReach(ag, g_{e_j}) (16)$$

$$\rightarrow badEvent(risk, fatality, ag, g_{e_j}) (17)$$

$$hasPossibilityBadEvent(g_{e_j}, p) \wedge \neg eventBadHappens(g_{e_j}) \wedge tryReach(ag, g_{e_j}) (18)$$

$$\rightarrow \neg badEvent(risk, fatality, ag, g_{e_j}) (19)$$

$$hasPossibilityBadEvent(g_{e_j}, p) \wedge eventBadHappens(g_{e_j}) \wedge \neg tryReach(ag, g_{e_j}) (20)$$

$$\rightarrow \neg badEvent(risk, fatality, ag, g_{e_j}) (21)$$

#### Definição 14

$$violation(d_{o_i}, ag, g_{e_i}) \rightarrow stopOperation (22)$$

$$\neg violation(d_{o_i}, ag, g_{e_i}) \rightarrow \neg stopOperation (23)$$

#### Definição 15

$$violation(g_{e_i}, e_k, ag) \rightarrow sanction(g_{e_i}, ag, risk) \rightarrow badEvent(risk, fatality, ag, g_{e_i}) (24)$$

$$\neg violation(g_{e_i}, e_k, ag) \rightarrow \neg sanction(g_{e_i}, ag, risk) \rightarrow \neg badEvent(risk, fatality, ag, g_{e_i}) (25)$$

#### Definição 16

$$badEvent(risk, fatality, ag, g_{e_i}) \rightarrow stopOperation (26)$$

$$\neg badEvent(risk, fatality, ag, g_{e_i}) \rightarrow \neg stopOperation (27)$$

#### Definição 17

Para todos os agentes que receberam uma obrigação de executar  $g_{e_i}$ , ao cumprir com a condição  $tryReach(g_{e_i}, ag) \wedge \neg stopOperation$  então  $isCompleted(g_{e_i})$ , se não, então é verdade  $failed(g_{e_i})$