

# Proposta de um Modelo para Representar Cenários de Acidentes Usando Conceitos de Normas, Sanções e Violações

Jonathan M. Samara  
Orientador Prof. Dr. Cesar A. Tacla

Universidade Tecnológica Federal do Paraná

19 de fevereiro de 2019

- Muitas pessoas são submetidas a atividades que apresentam algum risco.
- Exemplos de serviços assim; elétrica, petroquímica, telecomunicações, transportes.
- Existe uma série de possíveis causas em um acidente.
- A computação pode contribuir com esse tipo de problema.
- Criar representações para tratar esse problema.

- Uma maneira de representar cenários assim está relacionado com sistemas multiagentes normativos.
- Dentro deste tipo de atinge é possível observar os conceitos de; norma, violação e sanção.
- Portanto, entender como usar agentes normativos para criar representações específicas a problemas envolvendo trabalho, acidente e risco é uma motivação para esse estudo.
- Realizar a análise dos raciocínios que podem ser construído é outra motivação.

- Contribuição para três campos; Representação Computacional, Sistemas Multiagentes e Segurança.
- Representação Computacional e Sistemas Multiagentes.
  - Existem modelos para representar Sistemas Multiagentes, porém não apresentar características específicas (conceitos, predicados, regras) para tratar cenários de acidentes.
  - Existem modelos para representar SMA Normativos, mas apresentam aspectos altamente genéricos, não tendo características específicas (conceitos, predicados e regras) para tratar cenários de acidentes e riscos.
  - Existem modelos para representar cenários de acidentes (am ambiente de trabalho) usando *SMA*, contudo são complexos. Além disso, esses modelos não focam no erro dos agentes bem como as consequências dos mesmos sobre os demais colegas.

- Segurança

- O interesse da pesquisa é computacional.
- O modelo aqui proposto trata de conceitos relacionados a riscos e situações inesperadas.
- Portanto, apesar do interesse ser principalmente computacional, há um diálogo com o campo da Segurança.

- Conceber uma representação que tenha maior expressividade (quando comparado com os demais modelos) ao representar o seguinte cenário; "Agentes trabalham de forma colaborativa para atingir um dado objetivo geral. Esses agentes podem cometer erros submetendo a si mesmos bem como a outros a acidentes ocasionando grandes danos a integridade física do(s) acidentado(s). Não apenas isso, mas essa representação deverá ser capaz de considerar as relações existentes entre as entidades (agentes, artefatos tais como ferramentas), condições ambientais com as violações, sanções e acidentes. Questões envolvendo caráter possibilístico dos acidentes frente a ação dos agentes também deverão ser verificadas nessa representação."

# Introdução - Objetivos Específicos

- 1 Representar um SMA.
- 2 Representar normas, violações e sanções sendo que as violações são erros cometidos pelos agentes e as sanções estão relacionadas com os acidentes e suas respectivas consequências.
- 3 Representar situações onde um agente é submetido a um evento ruim (integridade física do agente é negativamente afetada de alguma maneira) mesmo que ele não tenha cometido erro algum.
- 4 Representar como se dá as relações entre as entidades, condições ambientes e conjunto com as violações e sanções.
- 5 Representar as possibilidades da ocorrência de um evento atrelado a um dado risco.

- Um agente é um sistema computacional que está situado em um dado ambiente e que apresenta comportamento autônomo com a finalidade de atingir um dado objetivo que a ele é designado.
- Ambiente
  - Acessibilidade vs Inacessibilidade; Quanto mais acessível for o ambiente mais o agente consegue ter informações claras, precisas e atualizadas.
  - Determinístico vs Não-Determinístico; Quanto mais determinístico for o ambiente mais uma ação possui um comportamento claro e determinístico sem incertezas do estado resultante.
  - Episódico vs Não-Episódico; Um ambiente tende a ser o mais episódico possível tanto quando o desempenho do agente estiver associado a um episódio discreto e específico no ambiente.
  - Estático vs Dinâmico; Um ambiente é estático se não houver outros processos em paralelo aos eventos associados ao agente.
  - Discreto vs Contínuo; Um ambiente é discreto se existe um número finito de ações e percepções.



# Fundamentação Teórica - Agentes

- Uma entidade autônoma é uma entidade que apresenta a capacidade de agir por si mesma.
- Um termostato é uma entidade autônoma.
- Programas de Servidor Daemon são exemplos de entidades autônomas.
- Esses exemplos são apropriados para serem retratados por agentes, contudo não são bons exemplos de agentes inteligentes.
- Um agente inteligente deve apresentar as definições já tratadas e mais as seguintes propriedades; reatividade, proatividade e habilidades sociais.
- Existe duas categorias de Agentes; puramente reativos (tomam decisões considerando apenas o instante presente).
- Existe duas categorias de Agentes; possuem estado; possuem uma dada estrutura de dados interna que são considerados quando o agente toma uma certa decisão.

- Arquitetura de Agentes Inteligentes;
  - Agentes baseados em Lógica; realizam deduções para efetuar uma dada decisão.
  - Agentes reativos; realizam o mapeamento de uma situação sobre uma ação.
  - Agentes BDI; apresenta estados de crenças, desejo e intenções.
  - Agentes em camadas; tomada de decisão acontece por meio de uma estrutura em camadas que gera um nível de abstração extremamente elevando sobre questões do ambiente.

- Um agente pode ser estruturado em termos *goal-governed* e *goal-oriented*.
- Os agentes *goal-governed* têm capacidades cognitivas e podem representar os seus respectivos objetivos.
- Os agentes *goal-oriented* são programados para alcançar um dado objetivo.
- Artefatos não se enquadram nessas duas características. Essas entidades são exploradas pelo agente para que eles possam alcançar um dado objetivo.
- Em termos mais formais, artefato apresenta as seguintes propriedades; interfaces de uso, instruções de operação, funcionalidade e estrutura de comportamento.

- Interface de uso são as operações que podem ser invocadas pelos agentes.
- Instruções de operação consiste nas descrições de como fazer uso dessas operações.
- Funcionalidade consiste no propósito definido pelo programador.
- Estrutura de Comportamento; consiste nos aspectos internos do artefato a fim de providenciar as funcionalidades.

- Cartago é um *framework* usado para especificar as relações entre agentes e artefatos e é composto por três blocos; *agent bodies*, *artefact* e *workspace*.
- *Agent Bodies*; Possibilita a inteligência do agente se relacionar com o meio.
- *Artifacts*; São os tijolos lógicos do Cartago. Cada *Artifact* contem um id único, um nome lógico (para o agente poder se referenciar) e um nome completo que inclui o nome dos *workspaces* onde estão logicamente contidos.
- A localização lógica dos artefatos se dá em um *workspace* e tem como finalidade definir uma topologia do ambiente de trabalho.

- Um sistema multiagente é constituído por agentes autônomos que interagem visando um propósito em comum tendo como consequência um comportamento global.
- Uma organização com essas características, portanto, apresenta em comum; cultura, memória, história, distribuição de atividades e capacidade de distinguir um agente em específico.
- Uma organização de sistema multiagente deve conter relações sociais no que tange a agentes, institutos específicos e grupos sociais.
- Por ser uma organização, uma *SMA* apresenta; Divisão de tipos de atividades, integração, composição, estabilidade/flexibilidade, coordenação, recursividade, representação multinível e causalidade, potenciais e diferenciais, regras e gramáticas.

- MOISE+ é um formalismo de SMA. Analisar o MOISE+ permite compreender uma *SMA*.
- MOISE+ define três tipos de especificação; estrutural, funcional e deôntica.
- A especificação estrutural;

$$\rho_a \sqsubset \rho_b \quad (1)$$

$$\begin{aligned} link(\rho_s, \rho_d, auth) &\rightarrow link(\rho_s, \rho_d, com) \\ link(\rho_s, \rho_d, com) &\rightarrow link(\rho_s, \rho_d, acq) \end{aligned} \quad (2)$$

$$\begin{aligned} link(\rho_s, \rho_d, t) \wedge \rho'_s \sqsubset \rho'_s &\rightarrow link(\rho'_s, \rho_d, t) \\ link(\rho_s, \rho_d, t) \wedge \rho'_d \sqsubset \rho'_d &\rightarrow link(\rho_s, \rho'_d, t) \end{aligned} \quad (3)$$

$$\rho_a \bowtie \rho_b \wedge \rho_a \neq \rho_b \wedge \rho_a \sqsubset \rho' \rightarrow \rho' \bowtie \rho_b \quad (4)$$

$$gt = \langle R, SG, L^{intra}, L^{inter}, C^{intra}, C^{inter}, np, ng \rangle \quad (5)$$

- Especificação funcional;

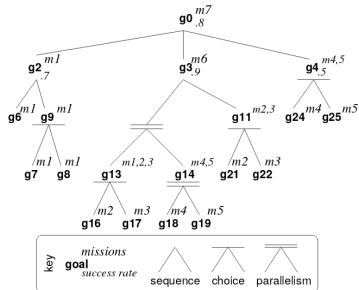


Figura: Arvore de objetivos definido pelo modelo MOISE+

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

(6)



- Especificação Deontica;

$$obl(\rho, m, tc) \rightarrow per(\rho, m, tc)$$

$$obl(\rho, m, tc) \wedge \rho \sqsubseteq \rho' \rightarrow obl(\rho', m, tc) \quad (7)$$

$$per(\rho, m, tc) \wedge \rho \sqsubseteq \rho' \rightarrow per(\rho', m, tc) \quad (8)$$

$$(9)$$

- Há estudos que tratam normas com objetivo de representar sociedades, organizações e instituições.
- Há estudos que tratam normas como forma dos agentes trabalharem de maneira coordenada a fim de atingir um dado objetivo.
- No MOISE+ normas são tratadas sobre a ótica da lógica deôntica e é usada para especificar missões aos papéis dos agentes.
- Contudo nesse estudo as normas são tratadas tendo como base o modelo do DASTANI.

```
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>  $\Rightarrow$  <i-literals> )+;
<sanctions>  := ( <i-literals>  $\Rightarrow$  <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:** Linguagem para descrever um programa de multiagentes normativos com a possibilidade de violações e sanções na notação EBNF segundo o texto [?]. Nesta notação,  $\langle ident \rangle$  é usado para denotar uma *string* e  $\langle int \rangle$  inteiros. Os termos  $\langle b-prop \rangle$  e  $\langle i-prop \rangle$  são usados para designar dois tipos de conjuntos de proposições que são disjuntos entre si

- Os erros em industria não podem ser definidos apenas nas falhas de humano.
- Consequência de um comportamento global da instituição contribuem para os erros.
- Esse comportamento advêm de uma forte pressão tendo em vista eficiência e otimização dos processos de produção.

- BATU - *Boundary Activities Tolerated during Use* (Atividades Limites Toleradas Durante o Uso). Muitas vezes a equipe adota atividades paliativas a fim de otimizar os processos de produção. Isso envolve assumir níveis de tolerância no que diz respeito ao desempenho e a segurança.
- BCTU - *Boundary Conditions Tolerated by Use* (Condições Limites Toleradas Durante o Uso). O termo condição faz referência a uma situação, um estado, circunstâncias externas às quais pessoas ou até mesmo entidades são afetados no que diz respeito a uma certa decisão atrelada a circunstâncias ambientais, materiais, humanas, de produtos e entre outras.

- Os requisitos são derivados de uma análise dos Objetivos Específicos desse estudo.
- Apresentam as seguintes características;
  - São estáticos (uma vez formulado o problema, não mudam com o decorrer do desenvolvimento da representação);
  - São articulados em um vocabulário compreensível para os pesquisadores

- Investigação de modelos que podem ser aplicados a essa situação.
- Verificação dos conceitos em interesse dentro desses modelos.
- Os pesquisadores isolaram os conceitos e suas relações nesses modelos.
- Os conceitos são adaptados ao cenário que se deseja representar.
- A estrutura resultante é escrita no seguinte formalismo; Teoria de Conjuntos para representar os conceitos e lógica de predicados para representar as relações.
- UML também é usado para essa mesmo propósito.
- Uma vez feito isso, os pesquisadores constroem regras para determinar a transição as transições de estado.

- Manutenção em Linha Viva.
- Estudo de manuais técnicos.
- Entrevista com o Engenheiro de Manutenção de Linha Viva.
- Acompanhamento de um procedimento de Manutenção.



- O cenário a ser retratado é; Método a Distância para troca de um isolador de Pedestal.
- É feito um Modelo Baseado em Cenários que consiste na descrição da atividade em linguagem natural. Esse modelo foi verificado por profissionais da área.
- Desse houve o processo de especificação nos conjuntos e predicados definidos anteriormente.

- Uma vez com a manutenção especificada, os pesquisadores geraram raciocínios para traduzir diferentes cenários de trabalho (com o propósito de provar a hipótese desse estudo).
- A validação desses raciocínios se dá em análise com o modelo de cenários, pois isso permite verificar se esses raciocínios correspondem com a realidade.

# Resultados

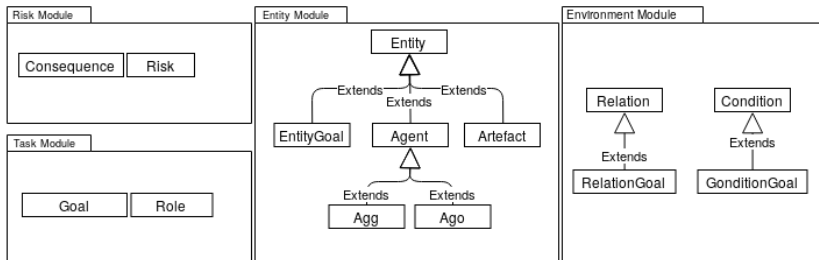


Figura: A estrutura geral das classes do modelo

$$M_{Entity} = \{Entity, Agent, Artefact, EntityGoal, Agg, Ago\} \quad (10)$$

$$(Agent \cup Artefact) \subset Entity \quad (11)$$

$$Agent \cap Artefact = \emptyset \quad (12)$$

$$M_{Task} = \{Goal, GoalPreRequisite, Role\} \quad (13)$$

$$M_{Environment} = \{Relation, ReationGoal, Condition, ConditionGoal\} \quad (14)$$

$$thereIsRelation(r_l, e_i, e_k) | r_l \in Relation \wedge e_i, e_k \in Entity \quad (15)$$

$$hasRole(ag_n, \rho_m) | ag_n \in Agent \wedge \rho_m \in Role \quad (16)$$

$$hasObligation(\rho_m, g_j) | \rho_m \in Role, g_j \in Goal \quad (17)$$

$$hasPermission(\rho_m, g_j) | \rho_m \in Role, g_j \in Goal \quad (18)$$

$$isReached(g_k) | g_k \in Goal \quad (19)$$

$$stopIn(g_n, ag_m) | g_n \in Goal, ag_m \in Agent \quad (20)$$

$$nextGoal(g_i, g_j) | g_i, g_j \in Goal \quad (21)$$