­[[1]](#footnote-2)

Information Model for building Training Scenarios in Computer Simulated Maintenance Activities \*

First A. Author, Second B. Author, Jr., and Third C. Author, Member, IEEE

*Abstract*— Computer simulated scenarios for training operators in maintenance activities requires a lot of effort to be specified basically because they should consider prescribed and tolerated behaviors, violations of accepted behaviors, and their consequences. This paper presents a model containing required information for building training systems based on the notion of social organizations inspired in the Moise+ model. The requirements were captured by in site observation of operators during maintenance activities and by interviewing an engineer involved in maintenance of high-voltage electric power transmission lines. A case study presenting scenarios and possible inferences derived from the model show the potential uses of such model when one builds a training system.

# INTRODUCTION

Risky professional activities expose people to effects that can even cause death. Thus, training professionals before they have experiences in the field is an important issue. One can build physical models for training when the maintenance site is often the same, the cost to build a concrete model is not prohibitive, and knowledge learned in the physical model can be applied in real situations. Virtual reality models are another way to tackle this kind of training or, more generally, any kind of computer simulated scenarios can be used with different degrees of realism. Besides graphics, which allows the trainee to be immersed in the activity, team organization and relationships among artifacts, people, and actions are relevant information for providing realism and learning in training.

The goal of this paper is to build and assess a conceptual model. Model was built with information gathered from observing real teams in maintenance activities, interviewing a security engineer, and revising related approaches. We propose a conceptual model containing information that would allow a simulation engine to produce consequences

when the trainee violate­ some prescribed norm for the maintenance team. The conceptual model was evaluated by means of a case study that shows possible derived inferences from represented information.

Next section presents some fundamental concepts in the domain of security, practical activities, and organizations. Following, section 3 presents the proposed model while section 4 shows a case study. We present related work in section 5 and final remarks in section 6.

# FUNDAMENTALS

## A. Agents

In order to model workers involved in maintenance activities, we have used the notion of agent. The most common accepted concept says an agent is rational, capable of autonomous behavior, and is situated in an environment where he/she acts proactively trough actuators and senses the environment to perceive its state and react to changes [1]. Besides, we have based our work on the Belief, Desire, Intention model of agents evidencing that agents have mental states [2], particularly, they have goals.

## B. ARTEFACTS

O entendimento do conceito de artefato pode ser feito por analisar o que é um agente. Em termos gerais, existe duas categorias de agentes; aqueles que são goal-governed e aqueles que são goal-orientend. Goal-governed são agentes que apresentam capacidade cognitiva e, portanto, podem criar representações dos seus respectivos objetivos. Em contraste com isso, os agentes goal-oriented usam os objetivos para se orientarem [4],[12],[13].

Existe entidades que não podem ser caracterizadas nem como goal-governed e nem como goal-oriented. Essas entidades são os artefatos. Portanto, um artefato aquilo que pode ser explorado por agentes para que estes possam alcançar seus objetivos.

## C. Multiagent System

A team of agents performing an activity compose an multiagent system. Each agent has the capabilities­ of communication, coordination, and cooperation. They form an organization and, according to the Moise+ model [8][9], an organization specification has three dimensions: structural, functional, and deontic.

Para retratar elementos de uma MAS dentro do modelo conceitual proposto, os pesquisadores fizeram uso de certas estruturas conceituas presentes no Moise+ que são; papel, objetivo, relações deônticas de obrigação de permissão [8][9].

D Normative Organization

O conceito de norma em uma MAS está sujeito ao debate sendo trabalhado sobre diferentes perspectivas [6],[8],[10]. Por exemplo, no Moise+, as normas são usadas única e exclusivamente para orientar os agentes no que diz respeito aos objetivos que devem ser alcançados por um dado papel.

Contudo, para o problema sobre o qual trabalha esse estudo, é mais apropriado trabalhar os aspectos normativos do modelo tendo como base o que está presente em [5]. Esse estudo aborda essa problemática com base nos seguintes tipos de regras; Count-As (que caracteriza os elementos implicantes em uma violação) e Sanction-As (que caracteriza as consequências da ocorrência de uma violação). Essas duas abordagens foram exploradas no modelo proposto neste estudo.

## D. RISKS

### Diversas são as causas que contribuem para a ocorrência de um acidente [11]. Tendo em vista essa situação é que se verifica a ideia de BATU e BCTU. Aquele significa Atividades Limites Toleradas Durante o Uso. Este significa Condições Limites Toleradas Durante o Uso [7].

O conceito de risco é derivado de BATU e BCTU. Isso, pois quando o profissional trabalha dentro do limite do aceitável, esse trabalhador assume o risco de se acidentar.

# CONCEPTUAL MODEL FOR RISKY ACTIVITIES

This model was built upon the following knowledge engineering activities: in locus observation, interviews, gathering of information in technical and security documentation, and related work review.

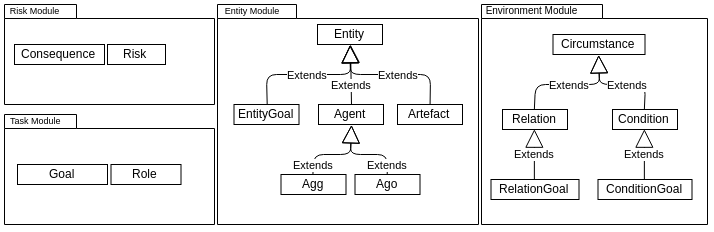
In locus observation and interviews were used to get acquainted with the domain of maintenance (particularly, in electrical energy) and to understand the problem from the users’ point of view. Important concepts and relationships were defined and elicited in this phase. A complete maintenance procedure was defined (exemplified in the case study). Technical and security information available in private documents of an electrical energy company was used as complementary information. At the end of this initial phase, we obtained a list of important terms and their relations used in training activities.

With all this information in hands, we proceeded a review of related work searching for existing models that could represent all information we defined as relevant such as [3],[4],[5],[9],[13]. Finally, combining our information needs with existing models, we produced the model presented in the next sections.

## MODULES OF INFORMATION

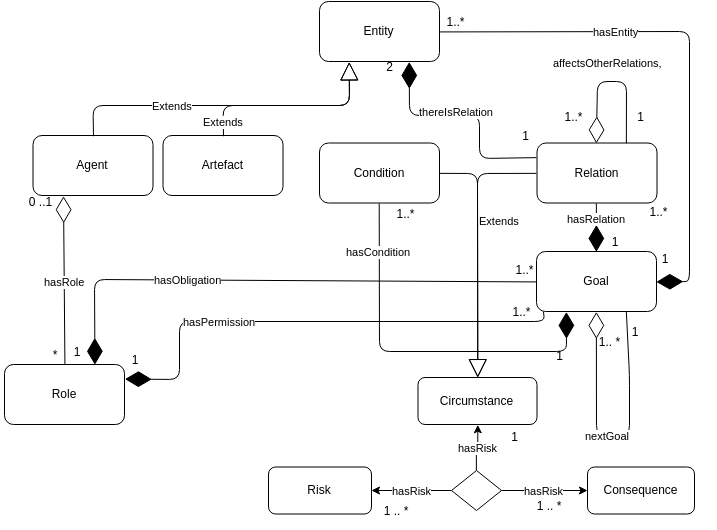
The model has the following modules: risk, tasks, entity, and environment.

O módulo de entidade apresenta os conjuntos que representam agentes e artefatos. O módulo de atividades apresenta os conjuntos usados para representar objetivos e funções. O módulo de risco apresenta os conjuntos relacionados ao risco bem como as consequências. O módulo de ambiente contém os conjuntos que representam os relacionamentos que podem acontecer entre as entidades bem como as condições que devem ser presentes para que uma dada ação possa acontecer. Essa abordagem geral dos módulos está demonstrada na figura 1.

***Figura 1****: Estrutura geral do Modelo estruturado em Módulos*

## *RELATIONS*

*A figura 2 exibe como se dá a relação entre as classes deste Modelo.*



***Figura 2****: UML - Diagrama de Classes do Modelo.*

Uma dessas relações é; hasRole(agent,role) cuja finalidade dessa estrutura é baseada em estudos da lógica deôntica e baseada em modelos como o MOISE+. A finalidade dessa relação consite associar o agent a um dado papel role. As relações de obrigação e permissão são dados pelo seguintes predicados; hasObligation(role,goal), hasPermission(role,goal).

O predicado hasRelation(goal,relation) apresenta as relações que devem ser realizadas para que um dado objetivo possa ser alcançado.

O mesmo ocorre com hasCondition(goal,condition), contudo esse predicado faz referência as condições ambientes necessárias para realização dos procedimentos.

O predicado thereIsRelation(relation,entityA,entityB) é usado para especificar uma relação entre duas entidades.

O predicado affectsOtherRelations(relationA,relationB) define que a, na ocorrência de um erro em *relationA* afeta *relationB. Outros predicados serão melhor explicados na seção que se segue.*

## RISK AND CONDITION

O cerne deste modelo diz respeito aos conceitos e relacionamentos que retratam a ocorrência de violações bem como suas respectivas consequências.

Com base no estudo de caso e nos arcabouços analisados, os pesquisadores entendem que ocorrê trềs tipos de violações, essas são; violação de entidade, violação de relação e violação de sanção.

A violação de entidade ocorre quando um agente tenta alcançar um dado objetivo sem que todas as entidades (demais agentes e artefatos) estejam devidamente presentes no local da manutenção.

*violationEntity(agi,gm,ek),* *agi - agente, gm – objetivo, ek - entidade*

A violação de relação oc­orre quando um agente tenta alcançar um dado objetivo e não consegue estabelecer apropriadamente uma relação entre duas entidades. Para exemplificar é possível conceber um cenário onde um profissional deve remover um dado parafuso com uma chave de fenda. Assim sendo, exite três entidades (um agente e dois artefatos), que são; profissional, chave de fenda e parafuso. Nessa situação uma violação de relação ocorreria se a chave de fenda é inapropriada para o parafuso (pois a relação chave de fenda – parafuso não pode ser efetivada), ou se o profissional não consegue manipular essa ferramenta.

*violationRelation(agi,gm,rk), agi - agente, gm – objetivo, rk - relação*

A violação de condição acontece quando um agente tenta executar uma atividade sem ter as condições apropriadas para isso. Isso pode ser exemplificado com base na seguinte situação; para executar uma dado procedimento operacional, um eletricista necessita que a umidade relativa do ar se mantenha em um nível inferior a 70 % u.r. Nesse caso, a umidade 70 % u.r é uma condição necessária para realizar esse procedimento. A violação, portanto, acontece quando o profissional tentar realizar o procedimento com a umidade relativa do ar superior a 70 % u.r (ou seja, a condição necessária não está mais presente).

violationCondition(agi,gm,ck) - *agi - agente, gm – objetivo, ck - condição*

Outro ponto a ser considerado como contido no cerne deste modelo reside no fato de que tanto relações (representados pelo conjunto *Relations*) como condições (representados pelo conjunto *Conditions)* são circunstâncias (conjunto Circumstance). Assim sendo, toda circunstância possui um risco associado como uma dada consequência. Essa situação é representada pelo predicado que se segue.

*hasRisk(crts,risk,csk)*

## REASONING WITH THE MODEL

Tanto os conjuntos como os predicados foram formulados com base em um rigoroso correspondente semântico. Assim sendo, averiguando o sentindo desses termos, é possível agrupá-los em relações de inferência formulando, portanto, regras que definem como ocorre a transição de estados do modelo.

Essa primeira regra define que toda obrigação de um agente implica necessariamente em permissão e seus fundamentos estão atrelados a lógica deôntica (presente no MOISE+)

***if*** *hasObligation(ρm,gj)*

***then*** *hasPermission( ρm,gj)*

*Regra 1*

A *regra 2* define o que acontece na situação do agente *agm* tentar executar um dado objetivo *gi,* com um certo conjunto de condições *cgn (*onde *ck* é uma delas), contudo *ck* não está presente neste exato momento.

***if***  *hasCondition(gi,cgn)* ***and***

***not*** *isPresent(ck)* ***and***

*ck inSet cgn* ***and***

*tryReach(agm,gi)*

***then***

*violationCondition(agm,gi,ck)*

*Regra 2*

A *regra 3* apresenta as situações que causam uma dada violação de relação. Essas situações acontecem quando um certo agente agm, ao tentar executar um objetivo gi, não consegue realizar (ou faz de forma inapropriada) uma dada relação *rk* onde está contida em *rgn* sendo que este conjunto apresenta todas relações necessárias para concretizar *gi.*

***if***  *hasRelation(gi,rgn)* ***and***

***not*** *isPresent(rk)* ***and***

*rk inSet rgn* ***and***

*tryReach(agm,gi)*

***then***

*violationRelation(agm,gi,rk)*

*Regra 3*

A *regra 4* define que uma violação de entidade ocorre quando agente *agm* tentar executar gi na situação onde a entidade ek não está presente sendo que essa entidade está contida em egn, ou seja, é crucial para que gi finalizada.

***if*** *hasEntitiy(gi,egn)* ***and***

***not*** *isPresent(ek)* ***and***

*ek inSet egn* ***and***

*tryReach(agm,gi)*

***then*** *violationEntity(agm,gi,ek)*

*Regra 4*

A regra 5 define as consequências atreladas a violação de condição (sanção), onde se ocorre uma certa violação de condição *ck,* em *gi*, sobre o agente *agm* e se *ck* tem um certo risco *riskj* com uma certa consequência *csm,* então, o agente *agm,* no objetivo *gi* sobre o risco *riskj* *sofre* consequências ruins *csm.*

***if*** *violationCondition(agm, gi, ck)* ***and***

*hasRisk(ck, risk, csm)* ***and***

***then*** *consequenceOfBadEvent(gi, agm, riskj,csm)*

*Regra 5*

A regra 6 define as consequências atreladas a violação de relação (sanção), onde uma dada relação rk, que tem um um certo risco riskj com uma consequência csm ocasiona consequências ruins ao agente agm.

***if*** *violationRelation(agm, gi, rk)* ***and***

*hasRisk(rk, riskj, csm)*

***then*** *consequenceOfBadEvent(gi,agm, riskj,csm)*

*Regra 6*

A *regra 7* mostra uma outra implicação de uma violação de relação onde *rk* pode afetar uma *rn* causando a possibilidade de um evento ruim ocorrer em *rn.*

***if*** *violationRelation(agm, gi, rk)* ***and***

*affectsOtherRelation(rk, rn)*

***then*** *possibilityHappensBadEvenet(rn)*

*Regra 8*

A regra 9 apresenta uma situação onde um agente *agm* tenta alcançar um dado objetivo *gi* que contém uma relação *rk*, sendo que essa relação tem um certo risco *riskj* associado a uma dada consequência ruim *csm.* Não apenas isso, mas essa regra retrata também a existência de uma possibilidade ruim de ocorrer em rk *(*dado pelo predicado *possibilityHappensBadEvenet(rk)).* Contudo, esse predicado indica única e exclusivamente a existência de uma possibilidade de ocorrer o evento, o predicado *happensBadEvent(rk)* constata que o evento ruim associado a *rk* realmente acontece. Como resultado, isso implica na ocorrência de consequências ruins sobre o agente *agm.*

***if*** *possibilityHappensBadEvenet(rk)* ***and***

*happensBadEvent(rk)* ***and***

*hasRelation(gi, rgn)* ***and***

*hasRisk(rk, riskj,csm)* ***and***

*tryReach(agm,gi)*

***then*** *consequenceOfBadEvent(gi, agm, riskj, csm)*

*Regra 9*

A regra 10 apresenta as consequências relacionadas a uma violação de entidade que é o encerramento da atividade no exato objetivo *gi* onde a violação acontece.

***if*** *violationEntity(agm,gi,ek)*

***then*** *stopIn(gi)*

*Regra 10*

A regra 11 apresenta a implicação decorrente de um evento ruim (acidente), que é a interrupção da atividade no exato objetivo *gk* onde o acidente ocorre.

***if*** *consequenceOfBadEvent(gk,agm,riskj,csm)*

***then*** *stopIn(gk)*

*Regra 11*

A regra 12 mostra os critérios que devem ser considerados para que um objetivo seja dado com alcançado. Para que um objetivo *gk* seja enquadrado nessa situação, não deve ocorrer nenhuma interrupção em *gk* em todos os agentes *aggn (*agentes que têm permissão de executar *gk*  e que realmente tentaram fazer isso*)* onde *agon* (agentes que são obrigados a executar *gk*) estão contidos em *aggn.*

***if not*** *stopIn(gk, aggn)* ***and***

*(agon isSubsSetOf aggn)*

***then*** *isReached(gk)*

*Regra 12*

A regra 14 apresenta ­os critérios que definem que o agente está hábil a tentar buscar novos objetivos. Para isso, se faz necessário considerar qual é a função do agente, as permissões dessa função, qual é o próximo objetivo daquele que recentemente foi finalizado e se o objetivo atual foi alcançado.

***if*** *hasRole(agn,ρm**)* ***and***

*hasPermission(agn, ρm)* ***and***

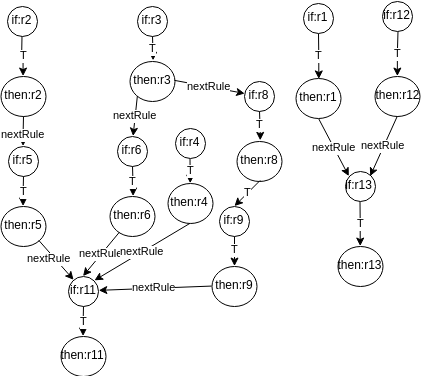
*nextGoal(gi,gj)* ***and***

*isReached(gi)*

***then*** *stopIn(gi)*

*Regra 13*

A combinação dessas regras possibilita a elaboração de raciocínios. A figura 3 apresenta esses o possíveis raciocínios que podem ser elaborados pelo sistema.



***Figura 3 -***  *Raciocínios que podem ser feitos com base nas regras*

Cada circulo corresponde a um estado que faz referência aos elementos implicantes ou aos elementos implicados da regra. Por exemplo, if:r2 representa os elementos implicantes da regra 2, ou seja*; hasCondition(gi,cgn)* ***and***

***not*** *isPresent(ck)* ***and***  *ck inSet cgn* ***and*** *tryReach(agm,gi).* Já then:r2 faz referência a; *violationCondition(agm,gi,ck).* O evento que gera transição de estados neste caso é T (if:r2 deve retornar true).

O evento nextRule é invocado quando existe o elemento presente em then:r(qualquer número) está presente no implicante de outra regra. Como, por exemplo, *violationCondition(agm,gi,ck)* está presente em if:r5, a próxima regra a ser considerada será exatamente a regra 5.

# CASE STUDY

We have taken a real case of maintenance from an electrical energy company. A team composed of one supervisor and six workers have the goal of replacing a pedestal insulator. Thus, we have seven instances of **Agent**, *{ag1, …, ag7}*, and two instances of **Roles**, *{supervisor, worker}*. The relationships among agents and roles are specified by the predicate *hasRole(ag, r).* For instance, *hasRole(ag1, supervisor)* designates that agent 1 plays the role of supervisor.

The artifacts available in this scenario are security apparels (gloves, boots), tools (rope, sockets, sticks, insulators), and screws and nuts, all instances of **Artifact**.

The maintenance activity is composed by a sequence of 20 sub-goals, starting with *clean, dry and test a rope* and ending with *arranging all the equipment in the van.* For the sake of simplicity, the **goals** are named *{g1, …, g20}.* To define that these goals are sequential, the predicate *nextGoal(gi, gj)* is instantiated several times: *nextGoal(g1, g2)* represents that g1 precedes g2, *nextGoal(g2, g3), …, nextGoal(g20, NULL).*

Weather conditions must hold for the team to achieve the maintenance goal: sunny, windless, no rain, and air humidity less than 70%. If any condition is violated, involved people risk a high-voltage electrocution causing death. Instantiation of the following predicates allows the representation of each condition associated to the risk and consequence in the case the condition is violated: *hasRisk(humidityLT70, electrical\_shock, death), hasRisk(sunny, electrical\_shock, death), hasRisk(no\_rain, electrical\_shock, death), hasRisk(windless, electrical\_shcok, death)*.

O estudo de caso possibilitou reproduzir os raciocínios presentes na figura 3. Os pesquisadores conseguiram reproduzir sete situações distintas tendo como base esses raciocínios.

A primeira situação envolve a regra 3 e 8. Essa situação reproduz o que acontece quando o agente4 esquece de passar glicerina no pano. Nessa situação não há ocorrência de acidente, mas sim uma sé­rie de relações futuras passam a apresentar a possibilidade ocorrência de algum dado evento ruim.

A segunda situação envolve as regras 4 e 11. O cenário a ser reproduzido consiste na seguinte situação; os agentes 2,3 (que possuem o papel de executor 1) e o agente 4 (que possui o papel de executor 2), tentam alcançar o objetivo g1. Para o sucesso disso, se faz necessário que esses agentes tenham acesso a um pano. Contudo, ao tentar alcançar o objetivo, esse pano não estava presente a disposição dos agentes. Por consequência (como retratado na regra 11), a manutenção é imediatamente interrompida.

A terceira situação envolve as regras 2, 5 e 11. Essa situação apresenta um caso onde o eletricista tenta realizar uma dada manutenção com a umidade relativa do ar superior a 70% u.r. Por cometer essa violação, o profissional acaba sendo eletrocutado e consequentemente morto. Outra consequência é o imediato encerramento da manu­tenção.

A quarta situação envolve as regras 3, 6 e 11. Essa situação apresenta um caso onde o agente3 erra a maneira adequada de usar a ferramenta chave catrada sobre o parafuso. A consequência disso consiste no fato de que o eletricista acaba morrendo gerando o imedianto encerramento da operação.

A quinta situação retrata o que acontece com uma das relações que foram afetadas tendo em vista a ocorrência da primera situação. Essa relação é entre o Parafuso com o Conector. Isso se deve ao fato de que tentar desfazer essa relação com uma ferramenta que não é envolta por Glicerina, pode fazer com que a corrente de fulga atinga valores nocivos ao ser humano fazendo com que ele morra. A situação cinco é representada pelo encadeamento das regras 8, 9 e 11, onde se faz necessário considerar o predicado *happensBadEvent(rk)* como T para a regra 9.

A sexta situação está atrelada a condição de um dado objetivo que, para ser dado como concluído, precisa ser alcançado pelos seguintes executores; executor1, executor2, executor3 e executor5. Assim sendo, na sexta situação esses executores conseguiram cumprir com esse respectivo objetivo sem que a ocorrência de uma interrupção especifica. A regra 12, portanto, representa o momento referente a quando esse objetivo é devidamente atingido.

Os pesquisadores estavam interessados em verificar como o sistema reage uma situação que não corresponde a realidade. Para isso formularam a sétima situação que é quando o agente1, que tem papel de supervisor, tenta alcançar o objetivo de número 23. Contudo, pela modelagem o predicado hasObligation(agente1,g23) retorna falso e o mesmo ocorre para hasPermission(agent1,g23), logo é uma situação que não é possível de acontecer dentro da lógica deste modelo.

# RELATED WORK

Uma ver feita a formulação dos conceitos envolvendo acidentes e riscos, se faz necessário avaliar como essa formulação pode ser usada por certos arcabouços consolidados pela comunidade acadêmica. Essa análise é baseada nos seguintes parâmetros: *Agente* (condiz com a capacidade do arcabouço expressar os estados internos do agente), *SMA* (capacidade do arcabouço expressar os conceitos associados a uma SMA), Normas (capacidade do arcabouço expressar comportamentos normativos), Violação ­(capacidade de expressar violações as normas), Sanção (capacidade de expressar uma sanção no que tange a uma violação), P.O.A.E (Possibilidade de Ocorrer algo de errado e isso corresponde a capacidade de expressar situações de acidentes em relação a agentes que não cometeram nenhuma violação), Objetivos (capacidade de representar os alvos que devem ser atingidos pelos agentes), C.A (Condições Ambientes – faz referência as condições que são necessárias para que o agente possa atingir seus objetivos), I.AG.AR (capacidade de representar as interações entre os agentes e os artefatos) e D.C.A (Descrição de Cenários de Acidentes - capacidade de desenvolver raciocı́nios a fim de representar cenários de acidentes).

**Figura 4:**  Comparação entre os arcabouços.

Os critérios anteriormente mensionados foram definidos em uma escalada de 0 – 4, onde 0 significa que o arcabouço não é capaz de expressar o ítem no que tange ao modelo deste estudo, 1 significa que o arcabouço consegue pouca expressividade no ítem, 2 – o arcabouço consegue expressar de forma razoável o respectivo ítem, 3 – o arcabouço expressa o ítem muito bem e 4 que significa que o arcabouço consegue expressar muito bem o respectivo ítem em relação ao arcabouço.. Os modelos escolhidos para essa finalidade são; MOISE+[8][9], DASTANI [5], V3S [3], NORMMAS [4] e Cartago [13]. Esses elementos foram sintetizados no gráfico apresentado na Figura 4.

# Conclusion

Nesse estudo os pesquisadores fizeram uma análise do estudo de caso bem como uma averiguação dos arcabouços que tratam de temáticas atreladas a agentes, sma, normas, acidentes e risco. Disso, os pesquisadores conceberam um modelo conceitual para compreender a dinâmica do problema. Esse modelo conceitual permitiu formular regras e raciocínios a fim de formular conclusões no que tange a compreensão do problema. Como consequência, os pesquisadores puderem compreender o estado do problema e puderam avaliar como implementar os arcabouços disponíveis dentro do ambito acadêmico.

Como base nos resultados dessa pesquisa é possível concluir que a formulação conceitual do problema de acidentes e ricos em ambiente de trabalho foi um objetivo efetivado com sucesso. Outro ponto do qual se pode concluir consiste no fato de que foi possível obter uma visão clara dos potenciais de cada arcabouços no que tange a especificação de problemas atrelados a acidentes e a ­riscos vinculados a atividades de trabalho.

No que tange aos trabalhos futuros, é possível derivar um novo trabalho para cada arcabouço. Isso se deve ao fato de avaliar como o modelo conceitual aqui desenvolvido pode ser especificado nos moldes desses arcabouços. Como trabalho futuro pode ser posto a realização de uma análise mais específica sobre certos pontos a fim de definir estruturas conceituais mais granular.

Appendix

Appendixes should appear before the acknowledgment.

Acknowledgment

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression, “One of us (R. B. G.) thanks . . .” Instead, try “R. B. G. thanks”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References

1. M. Wooldridge, N. R. Jennings. “Intelligent agents: theory and practice,” *The Knowledge Engineering Review*, Cambridge University Press, v. 10, n. 2, p. 115–152, 1995.
2. A. Rao, M. P. Georgeff. “Modeling rational agents within a BDI-architecture.” *In: Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991. (KR’91), p. 473–484.
3. BAROT, C. et al. V3S: A virtual environment for risk-management training based on human-activity models. Presence, v. 22, n. 1, p. 1–19, 2013. Disponı́vel em: <http://www.mitpressjournals.org/doi/abs/10.1162/PRES a 00134>.
4. CHANG, S.; MENEGUZZI, F. Simulating normative behaviour in multi-agent environmentsusing monitoring artefacts. In: DIGNUM, V. et al. (Ed.). Coordination, Organizations, Institutions, and Norms in Agent Systems XI. Cham: Springer International Publishing, 2016. p. 59–77. ISBN 978-3-319-42691-4.
5. DASTANI, M. et al. Normative multi-agent programs and their logics. In: MEYER, J.-J. C.; BROERSEN, J. (Ed.). Knowledge Representation for Agents and Multi-Agent Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 16–31. ISBN 978-3-642-05301-6.
6. ESTEVA, M.; PADGET, J.; SIERRA, C. Formalizing a languages for institutions and norms. In: MEYER, J.; TAMBE, M. (Ed.). Lecture Notes Artificial Intelligence. [S.l.]: Springer-Verlag, 2002. v. 2333, p. 348–366.
7. FADIER, E.; GARZA, C. D. L.; DIDE1LOT, A. Safe design and human activity: construction of a theoretical framework from an analysis of a printing sector. Safety Science, v. 41, n. 9, p. 759 – 789, 2003. ISSN 0925-7535. Disponı́vel em: <<http://www.sciencedirect.com/science/article/pii/S092575350200022X>>.
8. HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. A model for the structural, functional, and deontic specification of organizations in multiagent systems. In: BITTENCOURT, G.; RAMALHO, G. L. (Ed.). Advances in Artificial Intelligence. Berlin, Heidelberg: Springer

Berlin Heidelberg, 2002. p. 118–128. ISBN 978-3-540-36127-5.

1. HüBNER, J.; SICHMAN, J.; BOISSIER, O. Moise+: Towards a ­structural, functional, and deontic model for mas organization. In: . [S.l.: s.n.], 2002. p. 501–502.
2. LOPEZ, F.; LUCK, M. Modelling norms for autonomous agents. In: . [S.l.: s.n.], 2003. p. 238 – 245. ISBN 0-7695-1915-6.
3. RASMUSSEN, J. Risk management in a dynamic society: a modelling problem. Safety Science, v. 27, n. 2, p. 18­3 – 213, 1997. ISSN 0925-7535. Disponı́vel em: <http://www.sciencedirect.com/  
   science/article/pii/S0925753597000520>.
4. RICCI, A.; VIROLI, M.; OMICINI, A. Programming mas with artifacts. In: BORDINI, R. H. et al. (Ed.). Programming Multi-Agent Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 206–221. ISBN 978-3-540-32617-5.
5. RICCI, A.; VIROLI, M.; OMICINI, A. Cartago: A framework for prototyping artifact-based environments in mas. In: WEYNS, D.; PARUNAK, H. V. D.; MICHEL, F. (Ed.). Environments for Multi-Agent Systems III. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 67–86. ISBN 978-3-540-71103-2.

1. \*Research supported by ABC Foundation.

   F. A. Author is with the National Institute of Standards and Technology­, Boulder, CO 80305 USA (corresponding author to provide phone: 303-555-5555; fax: 303-555-5555; e-mail: author@ boulder.nist.gov).

   S. B. Author, Jr., was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar. colostate.edu).

   T. C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsuku­ba, Japan (e-mail: author@nrim.go.jp). [↑](#footnote-ref-2)