

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

. 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.

I. 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.

II. 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 through 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-oriented. 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. [X].

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 [X], 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 conceituais presentes no Moise+ que são; papel, objetivo, relações deonticas de obrigação de permissão [X].

D Normative Organization

*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).

O conceito de norma em uma MAS está sujeito ao debate sendo trabalhado sobre diferentes perspectivas [ESTEVA], [MOSES],[LOPEZ][HUBNER]. 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 (DASTANI). 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 [(RASMUSSEN,]. 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 [FADIER].

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

III. 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 [X][X]. Finally, combining our information needs with existing models, we produced the model presented in the next sections.

A. GENERAL VIEW

B. 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.

C. 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 e nos arcabouços analisados, os pesquisadores entendem que ocorre três 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)

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, existe 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,ck)

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)

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)

D. REASONING WITH THE MODEL

Tanto os conjuntos como os predicados foram formulados com base em um rigoroso correspondente semântico. Assim sendo, averiguando o sentido 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(pm,gj)*
then *hasPermission(pm,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 *violaionEntity(agm,gi,ek)*

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 tenham um certo risco *riskj* com uma consequência *csm* ocasiona consequências ruins ao agente *agm*.

if *violaitonRelation(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 a acontece consequência ruim *csm* associada a uma dada relação *rk* que apresenta a possibilidade disso acontecer.

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 relacioadas 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 *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, pm)* **and**
 hasPermission(agn, pm) **and**
 nextGoal(gi, gj) **and**
 isReached(gi)
then *stopIn(gi)*

Regra 13

IV. 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**, $\{ag_1, \dots, ag_7\}$, 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 $\{g_1, \dots, g_{20}\}$. To define that these goals are sequential, the predicate *nextGoal(g_i, g_j)* 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_shock, death).

V. RELATED WORK

VI. CONCLUSION

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

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.