­[[1]](#footnote-2)

*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.

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

Jonathan Samara, Gregory Wanderley, Emerson Paraíso, Gustavo Gimenez-Lugo, and Cesar Tacla

# 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 sites are similar, in this case, the cost to build a concrete model may be justified, 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 scenario, team organization and relationships among artifacts, people, and actions are relevant for providing realism and learning during training sessions.

The goal of this paper is to build and assess 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. This model was built with information gathered observing real teams in maintenance activities, interviewing a security engineer, and revising related approaches. The conceptual model was evaluated by means of a case study that shows inferences possible to be inferred 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 agents are rational, capable of autonomous behavior, act proactively in an environment by means of their actuators and sense 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

Very shortly, an artifact is a concrete or abstract thing that agents can use whey they pursue their goals. Artifacts are neither goal-governed nor goal-oriented things ([4],[12],[13]). A common repository, a list of attendance, a coffee-machine are all artifacts one can find in an environment.

## C. Normative Organizations

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

Structural specification allows for defining roles and their structure with inheritance (for role and sub-roles) and composition relations with cardinality for defining groups (e.g. a maintenance team composed by exactly one agent playing the role of leader and from 1 to 4 agents playing the role of technical staff). The following links constraint two related roles in the following way: acquaintance (one role can represent the other), communication (one role can communicate with the other), and authority (a role can control the other). The role compatibility link says an agent can eventually adopt compatible roles simultaneously. Thus, when an agent enters a group and adopts a role, he/she is expected to behave accordingly to the role’s constraints.

Functional specification is composed by missions (collective goals). An organization may have several missions each one being a set of goals arranged in a decomposition tree. Subgoals may be children nodes of some parent goal and their execution may be sequential or parallel, or as a choice (one or another path). For instance, if a mission g0 has two subgoals, g1 and g2, when an agent accepts g0, he/she commits to pursue the goals g1 and g2 (in sequence, in parallel or one or another accordingly to the specification).

Deontic specification represents which roles have the permission or the obligation to commit to a mission. Permissions and obligations may start at any time or at a specific time (after a certain goal of the mission is achieved).

## D. NORMATIVE ORGANIZATIONS

There are different perspectives on the utilization of norms in multiagent systems (e.g. [6], [8], [10]). For instance, in Moise+, when agents adopt roles they are obliged or have the permission to execute missions. In this work, we take inspiration from [5]. Dastani and colleagues use the following kinds of norms: count-as and sanction-as. Count-as norms specify what causes a violation of a norm and sanction-as the consequences of a violation.

## D. RISKS

Risks come from violation of prescribed actions to perform an activity and/or of conditions that must be observed to start an activity or stop an on-going activity. As noticed by [7], workers, based on real operation, make interventions adapting the actions to tolerated limits preserving at a certain extent the performance and safety of the activity. Such interventions are called boundary activities. When the worker goes out of the tolerated limits, (unforeseen) events or circumstances may happen causing an accident.

# CONCEPTUAL MODEL FOR RISKY ACTIVITIES

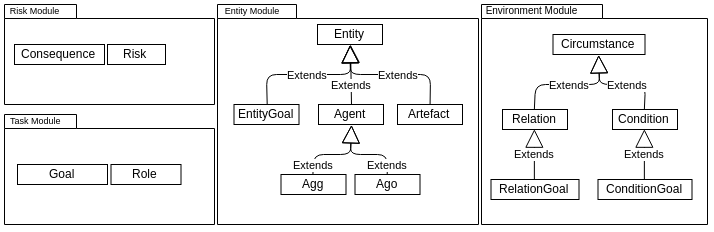
This model was built upon the following knowledge engineering activities: in locus observation, interviews, gathering of information in technical and security documentation, and related work review.

In locus observation and interviews were used by one of the authors 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). Private documents containing technical and security information of an electrical energy company were 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 the terms and their relationships in hands, we proceeded a review of related work ([3],[4],[5],[9], and [13]) searching for existing models that are able to represent all that information we defined as relevant. Finally, combining information needs with existing models, we produced the model presented in the next sections.

## GENERAL VIEW

The model has the following modules: risk, tasks, entity, and environment as the figure 1 illustrates. All the main elements of the risk, tasks, and entity modules were described in the previous section. The environment module contains classes and relationships for representing different kinds of relations that occurs between entities, goals, goals and their pre-conditions, and goals and their consequences.

1. Modules and their main classes and relationships.

## BASIC RELATIONSHIPS

We describe some relationships that put in relation instances of different modules. Variables that instantiates the aforementioned classes are written in lowercase

**Entity and task modules**. agents, when executing activities, adopt roles. This relation is represented by the *hasRole(agent, role)* predicate. Entities, for instance A and B, may have different kinds of relations. The model lets the user specify them by using the predicate *thereIsRelation(relation, entityA, entityB).*

**Task and environment modules**. Goals have conditions in order to be considered achieved (goal test), this relation is represented by the *hasRelation(goal, relation)* predicate. Goals have environmental conditions that must hold in order to pursue the goal. This is represented by the *hasCondition(goal, condition)* predicate.

This following relation occurs between entities of the module Environment, being also a basic one. When a relation A affects B by some way, the user can represent this by using the predicate *affectsOtherRelation(relationA, relationB).*

## RISKS AND CONDITIONS

Entities that represent risks and conditions are the kernel of the model. The authors have decided to model three types of violations: entity, relation and sanction violation.

**Entity violation**. It happens when an agent tries to reach a goal without the presence of the required entities as the other agents and artifacts. The predicate *entityViol(agent, goal, entity*) says that the *agent* represents the agent doesn’t have the *entity* when he/she decided to pursue the *goal*.

**Relation violation**. When an agent tries to reach a goal but doesn’t successfully put the needed entities into relation. For instance, a worker should remove a screw with a screw driver. If the screw driver doesn’t fit the screw, we have a relation violation. This violation is represented by the *predicate relationViol(agent, goal, relation)*.

**Condition Violation**. When an agent tries to execute an activity without observing the required conditions. For example, a procedure to pursue a goal requires the air humidity less than 70%. The predicate *conditionViol(agent, goal, condition)* represents this *condition* was not satisfied.

**Circumstances**, the generic kind for relations and conditions, when violated may cause risks of accidents engendering consequences. The predicated *hasRisk(circumstance, risk, consequence)* represents this relation, such as circumstance may be a relation or a condition.

Central to the model is the application of **sanctions**. This is represented by the predicate *badConseqFor(goal, agent, risk, consequence)* meaning the agent suffers the *consequence* of the *risk* when trying to reach the *goal*.

It is necessary to give the modeler a way of representing the possibility of a linked relation be disturbed by other relation. For instance, when relationA linked to B is violated, B may be disturbed (see *affectsOtherRelation(relationA, relationB))*. This possible bad event is represented by the predicated *possOfBadConseqFor(relation)*.

## REASONING WITH THE MODEL

In this section, we describe fundamental inference rules of the model and the inferences one can produce using the proposed model.

The first fundamental rule inherited from the Moise+ says that if an agent performing a *role* have the obligation to execute a *goal*, he/she also have the permission to execute it.

***if*** *hasObligation(role,goal)****then*** *hasPermission(role, goal)* (R1)

­

Rule 2 produces a condition violation when an agent tries to pursue a goal without the presence of any needed condition. The predicate *isPresent(condition)* is to be interpreted as “the agent beliefs the condition holds” and *tryReach(agent, goal)* is true when an agent tries to start a goal.

***if***  *hasCondition(goal, condition)* ***and***

***not*** *isPresent(condition)* ***and***

*tryReach(agent, goal)*

***then*** *conditionViol(agent, goal, condition)* (R2)

Rule 3 produces a relation violation when an agent tries to pursue a goal without the presence of any needed relation.

***if***  *hasRelation(goal, relation)* ***and***

***not*** *isPresent(relation)* ***and***

*tryReach(agent, goal)*

***then*** *relationViol(agent, goal, relation)* (R3)

Rule 4 produces an entity violation when an agent tries to pursue a goal without the presence of any required entity.

***if***  *hasEntity(goal, entity)* ***and***

***not*** *isPresent(entity)* ***and***

*tryReach(agent, goal)*

***then*** *entityViol(agent, goal, entity)* (R4)

Rule 5 applies to the agent the consequence of violating a condition (it sanctions the agent). Such condition presents a risk with a consequence represented by the predicate *hasRisk*. Rule 6 is similar to rule 5 replacing condition by relation.

***if***  *conditionViol(agent, goal, condition)* ***and***

*hasRisk****(****condition, risk, consequence)* ***and***

***then***

*badConseqFor(goal, agent, risk, consequence)* (R5)

A regra 6 define as sanções de uma violação de relação que foi ocasionada por um agente.

***if***  *relationViol(agent, goal, relation)* ***and***

*hasRisk****(****relation, risk, consequence)* ***and***

***then***

*badConseqFor(goal, agent, risk, consequence)* (R6)

Rule 7 allows for representing the possible spread of the effect of violating some relation A linked to the relation B.

***if*** *relationViol(agent, goal, relation)* ***and***

*affectsOtherRelation(relationA, relationB)*

***then*** *possOfBadConseqFor(relation)* (R7)

Rule 8 allows to represent that what was merely a possible bad consequence really happens due to the occurrence of a trigger event, i.e. the bad event. When an agent tries to reach a goal associated to the relation that has a possibility to going wrong and it goes wrong, the agent takes the risk and suffers the consequence.

***if*** *possOfBadConseqFor(relation)* ***and***

*tryReach(agent, goal) and*

*hasRelation(goal, relation)*

*happensBadEvent(relation) and*

*hasRisk(relation, risk, consequence)*

***then*** *badConseqFor(goal, agent, risk, consequence) (R8)*

Rule 9 stops an on-going goal when an entity violation occurs.

***if*** *entityViol(agent, goal, entity)*

***then*** *stop(goal) (R9)*

Rule 10 stops an on-going goal when a bad consequen­ce happens for the agent pursuing a goal.

***if*** *badConseqFor(goal, agent, risk, consequence)*

***then*** *stop(goal) (R10)*

Rule 11 determines a goal is finished when it has not been stopped. This rule must be verified for all agents playing roles that oblige them to pursue the goal.

***if not*** *stop(goal, agent)*

***then*** *reached(goal) (R11)*

Rule 12 defines an agent is able to pursue a new goal B when the precedent goal A is reached. The agent must play a role having permission or obligation to stop the current goal.

***if*** *hasRole(**agent, role)* ***and***

*hasPermission(role, goalA)* ***and***

*nextGoal(goalA, goalB)* ***and***

*reached(goalA)*

***then*** *ableTryReach(agent, goalB)**(R12)*

*A Regra 13 é define quando um agente deve encerrar suas operações em relação a um dado papel*

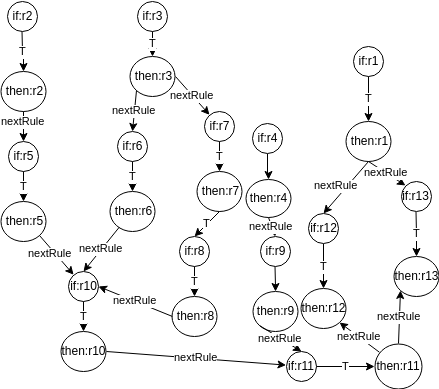
***if*** *hasRole(agent,role)* ***and***

*hasPermission(role, goalA)* ***and***

*lastGoal(goalA, agent)* ***and***

*reached(goalA)*

***then*** *stopIn(goalA) (R13)*

FIGURE 3.

# 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(agent, role).* 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 by *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 conditions are **not** present: *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 7. 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 9. 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 10. 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 10. 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 imediato encerramento da operação.

A quinta situação retrata o que acontece com uma das relações que foram afetadas tendo em vista a ocorrência da primeira 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 fuga atinja valores nocivos ao ser humano fazendo com que ele morra. A situação cinco é representada pelo encadeamento das regras 7, 8 e 10, 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

In order to compare the expressivity of the proposed model with related work, we use the following criteria:

* Agent: is it possible to represent agent’s internal states?
* MAS: is it possible to represent concepts (ex. organizations) associated do multi-agent systems?
* Norms: is it possible to represent normative behaviors?
* Violation: is it possible to represent norm violation?
* Sanction: is it possible to represent consequences of a violation?
* Bad consequences: is it possible to represent events that may produce bad consequences for agents that haven’t violated any norm?
* Goals: is it possible to represent goals to be pursued by the agents?
* Environmental conditions: is it possible to represent environment conditions that must hold in order to pursue a goal?
* Interactions: is it possible to represent interactions among artifacts and agents?
* Scenarios: is it possible to represent scenarios where accidents may happen?

1. Expressivity of related approaches.

Each criterion was evaluated in the range [0,4], such as zero means no such expressivity and 4, it can express very well the criterion. The chosen works are mostly from the multiagent systems Community: MOISE+[8][9], DASTANI [5], V3S [3], NORMMAS [4] and Cartago [13]. Figure 4 shows in a radar that any assessed approach can handle all the expressivity criteria.

# Conclusion

In this work, the authors have used a real case of maintenance activity in order to derive a conceptual model. In this knowledge engineering task, it was observed that agents, goals, norms, circumstances of activity execution, risks associated to them and consequences are the main concepts.

This conceptual model allowed us to specify rules that represent the dynamics of goals associated to the occurrence of bad events, what (may) cause risks to the workers and consequences to them (not only to the worker who violated the norm).

From this model, we could establish some criteria to analyze the expressivity of languages and frameworks that are candidates to represent these risky activities. We noticed that are some features that are not easily expressible like XXXXXXXXXXXXXXXX. By the other side, it is possible to observe the potential of the compared approaches, being the ………………………. the most suitable to this end.

It is worthwhile to mention that this conceptual model allows for maintenance technical staff to model and simulate maintenance scenarios. Finally, if someone wants to build a system for training involving circumstances, violations, associated risks and consequences, this model shows what kind of information the modeler must gather from his/her available sources.

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. Available in: <http://www.mitpressjournals.org/doi/abs/10.1162/PRES a 00134>.
4. CHANG, S.; MENEGUZZI, F. Simulating normative behaviour in multi-agent environments using 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 language 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. Available in: <http://www.sciencedirect.com/  
   science/article/pii/>.
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)