

## Um modelo de Multiagentes Normativos para Descrição de Personagens de Jogos Sérios

Jonathan Morris Samara<sup>1,2</sup>, Cesar Augusto Tacla<sup>1</sup>,  
Sebastião Ribeiro Junior<sup>2,5</sup>, Bruno Soares de Souza<sup>2</sup>  
Klaus de Geus<sup>3</sup>, Rafael Bee<sup>3</sup>,  
Sergio Scheer<sup>4</sup>

<sup>1</sup> Pós-Graduação em Engenharia Elétrica e Informática Industrial  
Universidade Tecnológica Federal do Paraná - UTFPR  
Av. Sete de Setembro 3165, CEP 80230-901 Curitiba PR.

<sup>2</sup>Divisão de Sistemas Elétricos - Institutos Lactec  
Rodovia BR-116, Km 98, nº 8813, Jardim das Américas. Curitiba, Paraná.

<sup>3</sup>Companhia Paranaense de Energia  
Rua Coronel Dulcídio, 800 - Batel - CEP 80420-170 - Curitiba - PR.

<sup>4</sup>Centro de Estudos de Engenharia Civil - Universidade Federal do Paraná - UFPR  
Rodovia BR-116, Km 98, nº 8813, Jardim das Américas. Curitiba, Paraná.

<sup>5</sup>Departamento de Engenharia Elétrica - Universidade Federal do Paraná - UFPR  
Rodovia BR-116, Km 98, nº 8813, Jardim das Américas. Curitiba, Paraná.

**Abstract.** *Research in the field of serious games has lately received significant attention of the academic community. Due to its importance in education and training, normative concepts and deontic logic have also played an important role in the field. Normative agents include computational representations of social, structural and deontic relations in the context of organizations in multiagents systems. The aim of this work is to discuss the development of an ontology used to create serious gaming characters as normative agents in serious games inspired by the definition of Moise+, in the context of an application aimed at training of live-line maintenance activities in energy substations. The implementation of this representation, using Moise+ and Unreal Engine 4, is presented, as well as the construction of an ontology that allowed the development of the game. Results indicate that the use of this specification is satisfactory for this kind of application. The results indicate the need to represent objects, tools and problems that must be solved by the characters in the game.*

**Resumo.** *A investigação no campo de jogos sérios está atraindo grande atenção da comunidade acadêmica. A lógica deôntica e sistemas normativos possuem uma grande importância no desenvolvimento de jogos sérios. Agentes normativos englobam representações computacionais das relações sociais, estruturais e deônticas no contexto de organização em sistemas multiagentes. O objetivo desse trabalho consiste em desenvolver uma ontologia para especificar personagens e suas missões como agentes normativos em jogos sérios inspirando-se na definição de organização de Moise+, no contexto de uma aplicação destinada ao treinamento de eletricitistas de linha viva que atuam em*

*subestações. Foi feita a implementação de um protótipo de um jogo sério a partir da ontologia, com o uso do Moise+ e a Unreal Engine. Os resultados indicam que o uso da ontologia de especificação é satisfatória para esse tipo de aplicação. Os resultados também indicam a necessidade de representar objetos, ferramentas e problemas que devem ser resolvidos pelos personagens do jogo.*

## **1. Introdução**

As relações normativas e organizacionais são fundamentais para o funcionamento das atividades no âmbito profissional. Com base nisso, tratar essas relações em sistemas de treinamento, como Jogos Sérios, é algo fundamental para preservação dessas relações. A representação computacional de organizações é uma área de estudo que está contida em Sistemas Multiagentes Normativos [Boella et al. 2006]. As relações de objetivos, missões, grupos, funções, permissões e obrigações são definidas em modelos lógicos usados para representar estruturas organizacionais de agentes. Moise + é um modelo que tem como finalidade realizar esse tipo de representação [Hübner et al. 2002b]. Portanto há um interesse científico em entender como usar representações relacionadas a sistemas multiagentes normativos para criar personagens de jogos sério.

Esse trabalho tem por objetivo propor e avaliar por meio de um estudo de caso um modelo constituído de elementos necessários para o desenvolvimento de jogos sérios, mais especificamente, jogos sérios de treinamento de trabalhadores que realizam manutenção em linhas vivas. Uma particularidade do modelo é sua inspiração em sistemas normativos para especificar as missões, papéis, comunicação entre papéis e a normas por meio de operadores deônticos. Isto permite avaliar a conformidade da ação do trabalhador frente as normas operacionais e de segurança. A ontologia (ou modelo) deverá funcionar como uma ferramenta que guia, por meio de parâmetros de entrada, o desenvolvimento dos personagens do jogo. Como resultado final, o modelo permite especificar agentes e artefatos que devem constituir o jogo. Não há o propósito, nesse estudo, de desenvolver um gerador automático de jogos, mas sim um sistema que oriente os desenvolvedores a implementar o jogo. O modelo será uma ontologia de especificação de jogos sérios.

O problema utilizado para desenvolver essa representação é a manutenção em linha viva (manutenção em equipamentos elétricos energizados). Essas atividades necessitam de uma boa capacidade de planejamento e apresentam alto risco à vida de todos diretamente envolvidos. Outro ponto de interesse está no fato de que uma mesma manutenção pode ser solucionada de várias maneiras. Sendo assim, o modelo proposto nesse estudo deverá ser capaz de orientar o desenvolvedor a levar em consideração as seguintes restrições: relações sociais e profissionais entre as pessoas pertencentes ao grupo (ex. supervisor, executor), regras que devem ser seguidas por todos os agentes e pelo jogador, criação de regras que possam ser adequadas a todos os cenários possíveis, como tratar a violação dessas regras e quais as outras representações necessárias no contexto do jogo para que as regras continuem sendo adequadas.

## **2. Fundamentação Teórica**

Os fundamentos teóricos dessa pesquisa consistem nos seguintes campos de conhecimento: Ontologia, Multiagentes Normativos e Jogos Sérios. As subseções a seguir dissertam sobre cada um desses campos.

## 2.1. Ontologias

Do ponto de vista computacional o termo ontologia faz referência a um artefato de engenharia constituído por um vocabulário específico usado para descrever parcialmente um recorte da realidade. A ontologia também é constituída por suposições explícitas relativas ao significado pretendido para as palavras do vocabulário [Guarino 1998]. Dentro desse contexto, um Engenheiro de Ontologias está preocupado em identificar as entidades, em caracterizá-las segundo uma estrutura taxonômica e definir essas relações [Guarino et al. 2009].

## 2.2. Multiagentes Normativos

Um sistema multiagente é um sistema composto por entidades autônomas chamadas de agentes que interagem em um ambiente. Existem várias categorias de agentes. A primeira categoria é dos agentes reativos. Esses agentes são especificados por meio de regras. Uma segunda categoria consiste em agentes deliberativos. Esses agentes possuem controle de seus estados internos, informações passadas e também podem definir planos para atingir metas no longo prazo. Os agentes deliberativos de raciocínio prático com arquitetura BDI (*Beliefs, Desire, Intentions*) são caracterizados por representar e controlar seus estados internos por meio de crenças, desejos e intenções. Como consequência, esses agentes são capazes de elaborar planos para atingir um determinado objetivo [Nareyek 2001] [Woolridge and Wooldridge 2001].

Normas são entendidas como um princípio que guiam membros de um grupo tendo em vista o que é adequado e aceitável. Uma maneira de trabalhar as normas e agentes consiste no uso de lógica deontica que tem como finalidade estudar as relações de obrigação, de permissão, de violação e da dinâmica da obrigação em relação ao tempo. Sendo assim, a lógica deontica pode representar as normas como regras ou condições que possibilitam o uso de raciocínios sobre isso [Boella et al. 2006].

Uma organização consiste em um conjunto de regras, relacionamentos e estruturas de autoridades e de governança que define o comportamento dos agentes [Hayden et al. 1999] [Horling and Lesser 2005]. No Moise+, uma organização é definida em três partes: Funcional (define o papel de cada agente, define as relações de ligação, define as relações de compatibilidade e define a estrutura dos grupos de papéis), Estrutural (define os objetivos e missões) e Deontica (define normas de ação sobre as missões) [Hübner et al. 2002a]. Na implementação do Moise+ no projeto JaCaMo, há dois módulos. O módulo S-Moise (parte Funcional, Estrutural e Deontica) define as relações sociais entre os agentes. O outro módulo é J-Moise, permite a integração da representação organizacional ao sistema multiagente que executa na plataforma Jason.

## 2.3. Jogos Sérios

Atualmente a definição mais usual para o conceito de jogos sérios é dada por [Susi et al. 2007]. Com base nesse autor, um jogo sério possui objetivos maiores (ex. treinamento, educação, simulação e entre outros) que apenas mero entretenimento [Breuer and Bente 2010].

Os jogos sérios e os jogos de entretenimento trabalham com funcionalidades similares ao de desenvolver tarefas, manter o foco, realizar simulação e realizar comunicação. Entretanto esses critérios são trabalhados com propósitos diferentes nos dois tipos de

jogos. Para um jogo sério, a tarefa tem como finalidade resolver o problema em foco enquanto que os jogos de entretenimento visam proporcionar uma experiência lúdica ao jogador. Em termos de manter foco, para um jogo sério, esse é um elemento primordial a ser desenvolvido, já na categoria de entretenimento isso é visto como uma forma de diversão. As simulações são vistas como essenciais para a ótica dos jogos sérios enquanto que um jogo de entretenimento entende as simulações como sendo um processo simples no que tange ao rigor de reproduzir com alta confiabilidade o que acontece no mundo real. A comunicação deve ocorrer de maneira natural nos jogos sérios enquanto que a comunicação ocorre visando perfeição nos jogos de entretenimento [Susi et al. 2007].

### 3. Estudos Relacionados

O estudo [Dignum et al. 2009] apresenta esforços para integrar uma plataforma de sistema multiagentes com alguma *game engine* com o objetivo de desenvolver personagens de jogos com comportamento mais natural. Para isso, os autores apresentam uma arquitetura de integração entre uma *Game Engine* e uma Plataforma de Agentes. Outro estudo que usa a mesma linha de desenvolvimento é o [Jepp et al. 2010]. O texto apresenta um framework para relacionar uma plataforma de desenvolvimento de jogos sérios com a *game engine*. O estudo proposto neste artigo difere de [Jepp et al. 2010], [Dignum et al. 2009] na abordagem usada para integrar os agentes com o jogo. Isso pois aqueles estudos estão preocupados em manter os agentes operando nas plataformas de sistemas multiagentes enquanto esse estudo pretende estabelecer critérios de representação para implementar agentes no código do jogo.

O estudo [Pons et al. 2012] apresenta um sistema de multiagentes que dinamicamente controla e orienta a ação de um jogo conforme as ações do jogador. A abordagem usada pelos pesquisadores consiste em descrever o jogo em termos de parâmetros, critérios e objetivos. Os parâmetros são todos os elementos que constituem e caracterizam o jogo (definições da física do jogo, recursos que são usados pelos jogadores, população de entidades artificiais). Os critérios são formas de combinar os parâmetros para obter o jogo. Os critérios devem possibilitar ao desenvolvedor a manipulação de todos os elementos presentes no jogo (gerando a formação de cenários possíveis). Os objetivos pedagógicos consistem em uma parte importante do jogo, pois são definições de performance sobre condições específicas que devem ser atingidas pelo jogador. Essas definições de performance são feitas em termos do critério. Um determinado objetivo consiste, portanto, em um critério específico que deve ser alcançado. Esse critério, então, é uma configuração de parâmetros que deve ser causada pelo jogador. Há um agente para cada especificação, esses agentes são: *Parameter-agent*, *Criterion-agent* e *Constraint-agent*. O *Parameter-agent* é criado para cada parâmetro envolvido no cenário. O *Criterion-agent* é associado com cada critério usado para avaliar valores associados ao cenário. O *Constraint-agent* ocorre para cada objetivo definido em relação a cada elemento do cenário. Esse agente computa a função de criticidade especificada pelo especialista em domínio. O objetivo é estruturado na função de criticidade (ex. quando a função atingir determinado valor, então o objetivo foi alcançado). Tanto o estudo [Pons et al. 2012] como este fazem uso de sistemas multiagentes em jogos sérios. A diferença reside no objetivo, pois enquanto que este estudo tem o interesse de usar essa ciência para especificar o comportamento dos *bots* do jogo, aquele estudo tem interesse em desenvolver controladores dinâmicos em relação ao comportamento do jogador.

O estudo [Tang and Hanneghan 2011] apresenta uma ontologia para especificar a modelagem de um jogo sério. O objetivo de desenvolvimento dessa ontologia reside no fato de que o projeto de jogos de computador necessita experiência na descrição das regras, da jogabilidade e da estética que compõem a experiência interativa. Isso complica o trabalho dos desenvolvedores inexperientes. Como consequência, a existência de um meta-modelo para documentar, especificar e desenhar jogos é algo de bastante utilidade. Os pesquisadores investigaram a área de desenvolvimento de jogos com a finalidade de encontrar uma estrutura hierárquica conceitual (classes) desse domínio. Tanto o estudo [Tang and Hanneghan 2011] como o estudo desenvolvido por esta pesquisa buscaram pelo desenvolvimento de uma estrutura conceitual em termos ontológicos a fim de especificar jogos sérios. A diferença reside no fato de que esse estudo usa um modelo de organização de multiagentes para descrever um modelo organizacional de agentes (Moise+) na estrutura dos bots do jogo, já aquele estudo não tem preocupação com a parte social e normativa dos agentes (pelo menos com a mesma intensidade).

No estudo [Neri et al. 2012] o paradigma de agente é usado para o desenvolvimento de um jogo de estratégia de simulação. Os estudos obtêm como resultado um jogo de guerra onde todos os personagens são descritos por agentes cognitivos e reativos. Esse estudo se destaca em relação aos demais devido ao fato de que Moise+ está fortemente presente no jogo. Apesar de que tanto o estudo de [Neri et al. 2012] como este estudo fazem uso de sistemas multiagentes e de representação de conhecimento, este estudo está preocupado em desenvolver um modelo conceitual para ser implementado em qualquer *game engine* já aquele estudo não teve essa preocupação.

O estudo [Kobti and Sharma 2007] apresenta o desenvolvimento de multiagentes que são capazes de aprender e evoluir conforme as habilidades do jogador. O experimento desenvolvido pelos pesquisadores foi em um jogo simples que emprega o uso de agentes evolutivos. Os pesquisadores concluem que existe um alto potencial na utilização de agentes evolutivos nos jogos. Tanto o estudo [Kobti and Sharma 2007] como o estudo desta pesquisa fazem uso de sistemas multiagentes, porém aquele estudo visa agentes evolutivos enquanto que esse estudo está preocupado com o uso de agentes cognitivos.

#### **4. Metodologia**

Foi necessário seguir as seguintes etapas para alcançar os objetivos deste trabalho: levantamento de dados, análise dos resultados, modelagem, desenvolvimento da ontologia e desenvolvimento do jogo. A etapa levantamento de dados tem por objetivo a coleta de dados das atividades profissionais que são necessárias à construção do jogo sério. A atividade escolhida foi a manutenção de equipamentos elétricos energizados devido à disponibilidade de um engenheiro especialista da área em uma empresa de energia. O foco está nas manutenções que ocorrem em subestações de transmissão. Foram feitas três entrevistas com engenheiro especialista em manutenção em linha viva de uma certa concessionária de energia. Nas sessões foram feitas diversas perguntas com a finalidade de compreender os seguintes conceitos: ferramentas básicas, profissionais envolvidos, função de cada profissional relacionado a manutenção, tipos de manutenção existente em diferentes contextos, procedimentos envolvendo substituição de pingo aéreo, procedimentos envolvendo substituição de conector, procedimento envolvendo substituição de isolador de pedestal, quais são os perigos envolvidos e quais são os procedimentos postos em prática em situações críticas. Todas essas informações foram gravadas em um pouco

mais de 182 minutos de áudio.

Além da entrevista, os pesquisadores acompanharam a execução de um procedimento de manutenção em linha viva. Enquanto a manutenção estava sendo realizada, os pesquisadores observaram as atividades, filmavam as operações e tomavam nota dos ocorridos. Os vídeos obtidos possuem um pouco mais de 3 horas de gravação. Outro ponto que está contido no levantamento foi a consulta a documentação técnica normativa que é privativa à concessionária de energia. Nessa documentação foi possível obter informações sobre procedimentos de manutenção, conceitos, jargões, funções profissionais e hierarquias profissionais.

A análise do resultado consistiu em transferir os dados contidos em documentação, áudio e vídeo para um arquivo de texto. A informação, nesse arquivo, ficou disposta em forma de uma lista. O próximo passo consistiu em identificar quais eram as informações necessárias e prioritárias a serem representadas na ontologia. Isso consiste em responder as seguintes questões: dentre os tipos de manutenção existentes (troca do isolador de pedestal, troca do conector e troca do pingô aéreo), por qual delas o pesquisador deve começar? Quais elementos são necessários e suficientes para criar o modelo? Quais papéis são relevantes para criar um modelo consistente? O que permeia as respostas a essas indagações é a simplicidade. Sendo assim, optou-se por filtrar informações que possibilitem o desenvolvimento de um jogo operacional e funcional. Isso consiste em escolher o caso de manutenção mais simples, o número mínimo de agentes (porém suficiente) para que a manutenção tenha prosseguimento e a condição de problema mais simples possível. A modelagem é outra etapa da metodologia dessa pesquisa. Para isso, os pesquisadores buscaram pela documentação de referência do Moise+ [Hübner et al. 2002a]. Com base nas informações filtradas e com base nas especificações do Moise+, foi feita uma representação manual do procedimento de manutenção em linha viva escolhido. Nessa etapa, também, foram identificados outros elementos que devem fazer parte da representação, porém não são abrangidos pela representação organizacional de Moise+ (ex. espaço, objetos, problemas que os objetos podem ter, ferramentas).

A estrutura obtida da modelagem foi representada em uma ontologia construída conforme a metodologia de desenvolvimento *Methontology* [Fernandez-Lopez et al. 1997]. O ponto de interesse consiste em identificar elementos comuns a todos os casos em análise, permitindo a criação de uma ontologia que seja a mais genérica possível, i.e. que permita a descrição de outros casos de manutenção que serão realizados futuramente. Para cada caso a ser modelado, é feita uma análise para verificar se a ontologia permite sua representação. Se não, a ontologia é modificada a fim de poder representar o caso em questão sem perder a expressividade que permita representar os casos anteriores. Boa parte do desenvolvimento da ontologia consistiu em reproduzir a estrutura lógica do Moise+ em linguagem OWL, linguagem esta escolhida para implementar a ontologia. Ainda assim há outros conceitos que fazem parte de elementos que não estão presentes na ontologia. O motivo pelo qual deva existir elementos na ontologia além da estrutura lógica do Moise+ reside no fato de que esses elementos devem fazer parte da representação do jogo. Uma vez que se encontrou todas as classes importantes para descrever o modelo, foi feito o uso de lógica de predicados para encontrar todas as relações que devem existir. Ainda nessa etapa é feita uma validação com profissionais da área. Essa validação envolve a apresentação

de um diagrama da ontologia ao Engenheiro de Manutenção a fim de verificar se os conceitos e relações representados são adequados e cobrem o esperado.

A última etapa consistiu em usar a ontologia para desenvolver o jogo. Para esse estudo os pesquisadores optaram por usar a *Game Engine Unreal 4* (porém a implementação do jogo sério pode ser feito em qualquer outra plataforma). A linguagem de programação dessa *Game Engine* é C++. Outro aspecto relevante da Unreal consiste no fato de que os *bots* do jogo são implementados por meio de uma árvore de comportamentos. Essa árvore exhibe o que deve acontecer tendo em vista estímulos do ambiente.

Os pesquisadores testaram várias consultas que podem ser feitas a ontologia. As tentativas revelaram qual é o sequenciamento de consultas que devem ser feitas com a finalidade de fornecer as informações relevantes para o desenvolvimento do jogo. A lista a seguir exhibe formalmente a transferência de conhecimento da ontologia para desenvolvimento do jogo.

- Consultar todos os objetos e problemas presentes nesses objetos. Implementar no código do jogo os objetos e os problemas que eles apresentam.
- Consultar todas as ferramentas, todos os agentes e suas respectivas restrições deônticas em relação as missões e objetivos. Implementar no código de cada agente o que eles devem fazer em determinado objetivo. Os objetivos orientam ao desenvolvedor quando os eventos associados a cada agente, a cada ferramenta, a cada objeto e a cada problema devem ocorrer.
- A estrutura de código de cada agente deve permitir que o personagem seja comandado pelo jogador em vez das regras definidas pela ontologia. Nessa condição, as regras deixam de comandar o comportamento dos agentes para supervisionar o comportamento do jogador que comanda aquele respectivo personagem.

## 5. Resultados

As entrevistas, os vídeos e a documentação técnica permitem concluir que os eletricitistas trabalham com dois métodos de atuação: ao potencial e a distância. O método ao potencial é usado quando o executor acessa o potencial com o uso de uma roupa condutiva mantendo distância do potencial da terra. O método à distância ocorre quando o eletricitista acessa o potencial por meio de um bastão isolante, estando isolado do potencial do condutor, porém se mantendo ao mesmo potencial da terra. Todas as manutenções são definidas nos moldes do método ao potencial ou do método a distância.

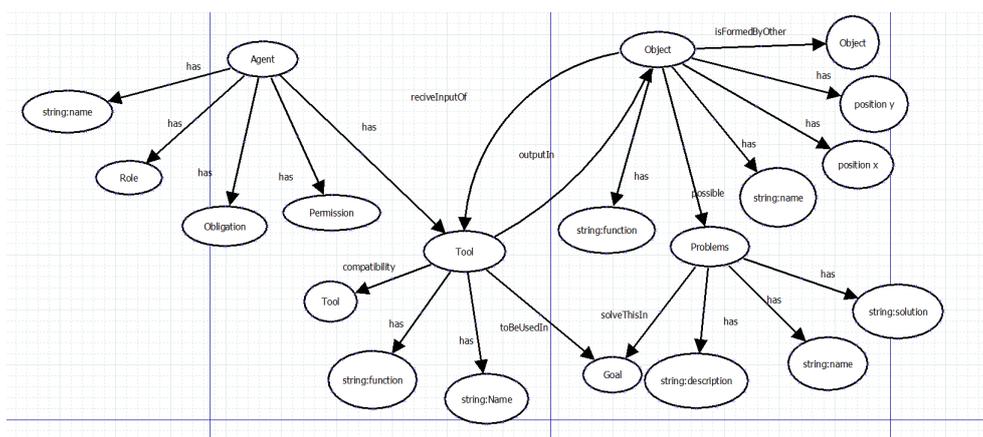
Foi identificado papéis de Supervisão, Execução e Observação. O executor realiza os procedimentos manuais. O supervisor analisa e comanda o prosseguimento das atividades e o Observador analisa a manutenção sobre o aspecto de segurança. Antes da manutenção acontecer é feita uma análise da situação e planejamento por Técnicos e Engenheiros especialistas no assunto.

Na etapa de análise de dados, optou-se por trabalhar com a representação da substituição de um conector pelo método ao potencial. Esse tipo de manutenção apresenta um número de objetivos menor quando comparado aos demais. O número de agentes envolvidos é um total de sete, um observador, um supervisor e cinco executores. Optou-se por isso tendo em vista de que essa quantidade de elementos é realística. As ferramentas, disposição dos objetos e problemas foram pensados para que esteja de acordo com uma

manutenção que possa ser executado por sete profissionais por intermédio da troca do conector por meio do método ao potencial.

A ontologia possui 5 classes principais; *Agent*, *Object*, *Problems*, *Tool* e *Organization*. A classe *Agent* contém os agentes que atuam no jogo. A classe *Object* descreve todos os objetos no espaço de ação dos agentes. A classe *Problems* apresenta todos os problemas possíveis que deverão ser corrigidos. Esse problema deve estar relacionado com o objetivo onde ocorre a correção. A classe *Tool* possui todas as ferramentas necessárias para executar a manutenção. A classe *Organization* e todas as suas respectivas subclasses apresentam as especificações lógicas definidas no modelo de representação Moise+. As relações presentes na ontologia orientam o desenvolvedor a desenvolver os *bots* do jogo como agentes.

A ontologia possui 1197 axiomas, 33 classes, 59 propriedades de objeto, 27 propriedades de dados, 170 instâncias, 101 axiomas de subclasses, 7 subpropriedades e 8 funções inversas. A lógica dessa representação é do tipo ALCRIF(D). Sendo assim, possui linguagem atributiva (o que permite fazer negação anatômica, tem interseção conceitual, possui restrições universais e possui quantificação existencial limitada), possui negação conceitual complexa, possui disjunção, possui propriedades inversas e possui propriedades do tipo dado [Baader et al. 2003].



**Figura 1. Descrição das classes Agent, Object, Problems e Tool.**

A figura 1 apresenta o esquemático das relações de predicado entre as classes *Agent*, *Object*, *Problems* e *Tool*. O termo que está contido em um círculo, quando não recebe a expressão *string*: faz referência a uma classe. Quando recebe a expressão *string*, faz referência a um determinado tipo de dado. Os termos que estão fora dos círculos são os predicados. Classe para a qual a flecha sai é o Domínio e a classe na qual a flecha chega é a Imagem.

A lista a seguir apresenta as consultas que podem ser feitas a fim de poder implementar as soluções da ontologia no jogo. O primeiro elemento consiste por onde a consulta começa. O último elemento é o que deve ser fornecido ao usuário. Os elementos intermediários são consultas internas que devem necessariamente acontecer. A legenda da lista é Objetivo - *O*, Objetivo Paralelos - *OP*, Missão - *M*, Especificação Deontica - *ED*, Função - *F*, Grupo - *G*, Compatibilidade - *C*, Link - *L*, Agente - *A*, Ferramenta - *Fe*, Objetos - *Ob*, Problema - *P*, Próximo Objetivo - *P.O*.

- $O \rightarrow OP$ .
- $O \rightarrow M \rightarrow ED$ .
- $O \rightarrow M \rightarrow ED \rightarrow F$ .
- $O \rightarrow M \rightarrow ED \rightarrow F \rightarrow G$ .
- $O \rightarrow M \rightarrow ED \rightarrow F \rightarrow C$ .
- $O \rightarrow M \rightarrow ED \rightarrow F \rightarrow L$ .
- $O \rightarrow A \rightarrow F$ .
- $O \rightarrow A \rightarrow Fe$ .
- $O \rightarrow Fe \rightarrow Ob$ .
- $O \rightarrow Fe \rightarrow Ob \rightarrow P$ .
- $O \rightarrow Fe \rightarrow Ob \rightarrow P.O$ .

A partir da ontologia, os pesquisadores usaram as especificações para implementar o jogo. Foi possível obter uma versão operacional do jogo que reproduz a existência de todos os objetos, agentes, problemas e interações descritas na ontologia. As especificações do Moise+ também estão funcionando de forma operacional no jogo. A figura 2 apresenta o jogo em execução. Para uma primeira versão, os objetos foram representados como cubos, as ferramentas como cones e os problemas como "chamas azuis" (emissor de partícula simples). O que rege o comportamento dos bots são estruturas lógicas de regras baseadas no Moise+ (definida nas especificações deônticas). A lista a seguir exhibe as regras de um dos *bots* implementados.

- *agentHasRole: roleExecutor*. Define o papel do agente.
  - *hasObligation: missaoProcedimentoDistanciaRoleExecutor*. Define a obrigação relacionada ao papel (*role*).
  - *hasPermission: missaoProcedimentoDistanciaRoleExecutor*. Define a permissão relacionada ao papel (*role*).
- *hasName: Executor01*. Define o nome relacionado ao agente.
- *hasTools: bastaoUniversal*. Define todas as ferramentas que o agente deve usar.

É importante ressaltar que o modelo representacional não leva em consideração as relações dinâmicas de adoção ou remoção de *schemas* como ocorre no JaCaMo.

## 6. Discussão

Ao descrever a manutenção em linha viva em uma ontologia inspirando-se em sistemas multiagentes normativos, tornou possível a modelagem das relações sociais, estruturais e deônticas. Sendo assim as relações normativas facilitaram o processo de definição do comportamento de cada agente. Isso, pois, as relações deônticas relacionam a função do agente em relação a missão. Esta, por sua vez, implica um conjunto de objetivos que devem ser atingidos pelo agente. A lógica deôntica estruturada nessa pesquisa não é útil apenas para delimitar o comportamento dos agentes, mas também possui utilidade na verificação do comportamento do jogador. Pois através da violação das normas do jogo torna-se possível verificar se o jogador está realmente preparado para assumir o cargo de interesse. Com base nessa linha de raciocínio, uma outra utilidade dessa aplicação implica identificar quais normas o jogador tem dificuldade de seguir.

No contexto da análise da manutenção é possível avaliar que a árvore de objetivos (definida na especificação estrutural) facilitou a identificação de procedimentos que são



**Figura 2. Jogo em operação, jogador observa bot realizando um determinado procedimento**

similares em diferentes manutenções. Como consequência há uma significativa redução no esforço necessário para realizar a descrição das atividades de novos casos (desde que esses casos se enquadrem nas mesmas restrições). Os objetivos exercem uma importância central dentro da representação presente na ontologia. Por esse motivo as consultas devem ser feitas sempre pelos objetivos. Uma consequência dessa situação consiste no fato de que os jogos resultantes dessa ontologia são orientados a objetivos. A relação entre os agentes e os objetivos ocorre por intermédio de lógica deôntica entre a função dos agentes e missões que encapsulam os objetivos.

A implementação da estrutura organizacional diretamente no código do jogo (sem utilizar Moise+) se deve ao fato de que ao analisar estudos que tentaram construir interfaces síncronas entre *Game Engines* comerciais com plataformas de Agentes como Jason, encontraram muitos problemas de implementação e de funcionamento [Dignum et al. 2009] [Jepp et al. 2010]. Como [Dignum et al. 2009] e como [Nareyek 2001] demonstram, a indústria de jogos eletrônicos, quando trabalham com agentes para descrever os personagens do jogo, fazem isso por meio de uma implementação no código do jogo. Esse é o motivo que levou à proposição da ontologia para especificação de um jogo sério de treinamento em manutenção. Outra vantagem de optar por essa escolha se deve ao fato de que o modelo proposto nesse estudo torna-se independente de uma plataforma específica de desenvolvimento de agentes ou de jogos. Futuramente, geradores de código específicos para cada plataforma poderão ser criados para geração de parte dos jogos sérios.

Os resultados indicam que a lógica de descrição (usando a sintaxe OWL-RDF) foi o suficiente para representar agentes normativos em jogos sérios. O uso do modelo de representação de uma organização inspirada em Moise+ foi o suficiente para tratar todas as relações deônticas e sociais entre os agentes do estudo de caso. Porém, além disso, também foi necessário representar os agentes, as ferramentas, os objetos dispostos ao redor dos agentes e os Problemas que devem ser corrigidos pelos agentes. Todas essas representações se encontram em uma única ontologia em conjunto com as especificações sociais do Moise+. Ressalta-se que as instâncias podem ser de outro tipo de atividade

distinta do domínio elétrico.

## 7. Conclusão

O objetivo inicial deste trabalho consiste em desenvolver uma ontologia de especificação de jogos sérios que permite descrever personagens como agentes. O interesse nesse objetivo consiste em analisar quais representações são necessárias para realizar esse tipo de modelagem, quais linguagens são mais adequadas, qual é a capacidade de generalização dos resultados e qual é característica da lógica descritiva. Os resultados mostram que a lógica obtida possui alta capacidade de generalização, mostrou que os elementos do Moise+ foram suficientes para representar questões normativas do jogo, mostrou que foi necessário representar ferramentas, objetos e problemas para obter um modelo eficaz. Os próximos passos dessa pesquisa são: finalizar o desenvolvimento do jogo sério, validar o jogo por profissionais da área, implementar os outros casos de manutenção na ontologia, verificar modificações que devem ser feitas tanto na ontologia como no jogo e verificar como os profissionais de linha viva reagem ao jogo.

## Referências

- Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.
- Boella, G., van der Torre, L., and Verhagen, H. (2006). Introduction to normative multi-agent systems. *Computation and Mathematical Organizational Theory, Special issue on Normative Multiagent Systems*, 12(2-3):71–79.
- Breuer, J. and Bente, G. (2010). Why so serious? on the relation of serious games and learning. *Journal for Computer Game Culture*, 4:7–24.
- Dignum, F., Westra, J., van Doesburg, W. A., and Harbers, M. (2009). Games and Agents: Designing Intelligent Gameplay. *International Journal of Computer Games Technology*, 2009:18.
- Fernandez-Lopez, M., Gomez-Perez, A., and Juristo, N. (1997). Methontology: from ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium*, pages 33–40, Stanford, USA.
- Guarino, N. (1998). *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 1st edition.
- Guarino, N., Oberle, D., and Staab, S. (2009). What is an ontology? In Staab, S. and Studer, R., editors, *Handbook on Ontologies*. Springer, second edition.
- Hayden, S. C., Carrick, C., and Yang, Q. (1999). A catalog of agent coordination patterns. In *Proceedings of the Third Annual Conference on Autonomous Agents, AGENTS '99*, pages 412–413, New York, NY, USA. ACM.
- Horling, B. and Lesser, V. (2005). A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, pages 281–316.

- Hübner, J. F., Sichman, J. S., and Boissier, O. (2002a). A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems. In Bittencourt, Guilherme; Ramalho, G., editor, *Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages pp 439 – 448. Springer Berlin / Heidelberg.
- Hübner, J. F., Sichman, J. S., and Boissier, O. (2002b). A model for the structural, functional, and deontic specification of organizations in multiagent systems. In Bittencourt, G. and Ramalho, G. L., editors, *Advances in Artificial Intelligence*, pages 118–128, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Jepp, P., Fradinho, M., and Pereira, J. M. (2010). An agent framework for a modular serious game. In *2010 Second International Conference on Games and Virtual Worlds for Serious Applications*, pages 19–26.
- Kobti, Z. and Sharma, S. (2007). A multi-agent architecture for game playing. *2007 IEEE Symposium on Computational Intelligence and Games*, pages 276–281.
- Nareyek, A. (2001). Review: Intelligent agents for computer games. In Marsland, T. and Frank, I., editors, *Computers and Games*, pages 414–422, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Neri, J. R., Zatelli, M. R., Uez, D. M., and Callegaro, R. F. (2012). Aplicando conceitos de sistemas multiagentes na elaboração de um jogo de estratégia simulado. *RITA*, 19:34–58.
- Pons, L., Bernon, C., and Glize, P. (2012). Scenario control for (serious) games using self-organizing multi-agent systems. In *2012 IEEE International Conference on Complex Systems (ICCS)*, pages 1–6.
- Susi, T., Johannesson, M., and Backlund, P. (2007). Serious Games: An Overview. Technical report, GLS University of Wisconsin-Madison.
- Tang, S. and Hanneghan, M. (2011). Game content model: An ontology for documenting serious game design. In *2011 Developments in E-systems Engineering*, pages 431–436.
- Woolridge, M. and Wooldridge, M. J. (2001). *Introduction to Multiagent Systems*. John Wiley & Sons, Inc., New York, NY, USA.