



## MODELAGEM DE AGENTE PESSOAL

José Eduardo Costa Lazaretti<sup>1</sup>; Aparecido Vilela Júnior<sup>2</sup>; José Valderlei da Silva<sup>3</sup>

**RESUMO:** As atividades computacionais realizadas pelas pessoas precisam de uma forma de interação simples e essa simplicidade poderá ser alcançada com um bom design de interação ou com processamento interno do software que poderá ser realizado considerando a pessoa e seu contexto de uso. Essa pesquisa considera os agentes de software (agente pessoal) uma entidade que tem como objetivo auxiliar o ensino-aprendizagem e por essa razão o objeto de estudo é a modelagem de um agente pessoal baseado em um modelo de usuário e considerando o seu contexto de uso.

**PALAVRAS-CHAVE:** interação, agente pessoal, contexto.

### 1 INTRODUÇÃO

É fato que uma grande parcela da população utiliza sistemas computacionais para desenvolverem atividades como estudar, ensinar, tomar decisões, etc. Neste sentido, as pessoas precisam de software com uma boa interação e também que de certa forma auxiliem durante a execução de uma tarefa. Acredita-se que os agentes de software seriam uma boa alternativa para o auxílio ao usuário, uma vez que esses agentes são softwares altamente inteligentes e independentes, com a capacidade de tomar de decisões antes mesmo de serem consultados.

Para [1], os agentes de software têm atuação autônoma e ativa e possuem habilidades de comunicação com pessoas e outros agentes de software além de adaptabilidade. Um agente pode ser concebido aplicando técnicas de Inteligência Artificial (IA). Na IA em cada seqüência de percepção possível, um agente racional ideal deve saber que se sua ação **maximizará sua medida de desempenho**, baseado na evidência de sua seqüência de percepção e no conhecimento que ele traz consigo [6].

Acreditando na idéia que um agente de software venha a auxiliar uma pessoa, questionamos: “*A partir de um modelo de usuário (pessoa), contendo suas características e necessidades em um determinado contexto de uso, como podemos suprir essas necessidades através de um agente de software?*”

O que buscamos é responder essa questão, tendo como premissa que os agentes poderão trabalhar em contextos pré-definidos e considerando a sua atuação como um agente que auxilia o ensino ou que auxilia a aprendizagem.

Assim, essa proposta de pesquisa considera o uso de um modelo de usuário para a criação de um modelo de agente que irá representá-lo. Além disso, procura determinar quais linguagens e ferramentas estão disponíveis para cumprir esse objetivo.

---

<sup>1</sup> Acadêmico do Curso de Engenharia de Software do Centro Universitário de Maringá – UNICESUMAR, Maringá – Paraná. Bolsista do Programa Institucional de Bolsas de Iniciação Científica (PIBIC/CNPq-Cesumar). eduardo1793@yahoo.com.br

<sup>2</sup> Orientador, Professor Mestre do Curso de Engenharia de Software do Centro Universitário de Maringá – UNICESUMAR. aparecido.vilela@unicesumar.edu.br

<sup>3</sup> Coorientador, Professor Mestre do Curso de Engenharia de Software do Centro Universitário de Maringá – UNICESUMAR. jose.valderlei@unicesumar.edu.br



## 2 DESENVOLVIMENTO

Iniciamos o estudo buscando por definições comuns de agentes de software, as quais são bastante irregulares e, em alguns casos, se diferem bastante umas das outras.

Ao se tratar de atividade educacional, os agentes são ditos pedagógicos. Os agentes pedagógicos são entidades cujo propósito fundamental é a comunicação com o estudante. Agentes pedagógicos são considerados animados quando esses são implementados com recursos de animação. (...) Os agentes pedagógicos animados são representados por personagens animados que interagem com os estudantes. (FONTES, L. M. O. *et al*, 2012)

O Agente é uma entidade autônoma que possui percepções do seu ambiente através de sensores e age sobre o mesmo utilizando-se dos executores. Por exemplo, nos agentes humanos, os olhos e ouvidos, são sensores; as mãos e a boca são executores. (RUSSELL, S. J.; NORVIG, P., 2002)

Além disso, segundo DE BARROS, A. B. P. (2009), na construção de softwares o fundamentado em agentes, deve-se decidir a forma de como será construído as percepções e as ações.

Por fim escolhemos adotar a seguinte definição:

Agentes são processos de computação autônomos e ativos que possuem alguma habilidade de se comunicar com pessoas e/ou outros agentes e de adaptar seu comportamento. Em resumo, agentes são pequenos programas de computador de inteligência artificial (IA).

Tendo como base a definição de agentes de software a qual entendemos melhor se encaixar no contexto de estudo, seguimos adiante com uma breve pesquisa sobre agentes de softwares ou softwares com inteligência artificial os quais já são usufruídos pelo sistema educacional.

Nos deparamos ainda, com um estudo sobre como um agente pedagógico animado poderia ser usado para apoiar a aprendizagem baseada em problema. Verificamos também que este tipo de abordagem de agentes de software possui conceitos interessantes, os quais poderiam auxiliar em nosso estudo.

Munidos de uma boa definição de agente de software e dos conceitos abordados no agente pedagógico estudado, sentimos a necessidade de nos aprofundar um pouco no estudo de banco de dados relacional (SQL) e não-relacional (MongoDB). Como o agente de software precisa possuir uma base de dados confiável e de rápido acesso, precisaríamos definir qual dos dois tipos estudados seria o mais aconselhável a se usar para se montar essa estrutura.

Os bancos de dados do tipo relacional são altamente confiáveis e precisos, uma vez que seus dados estão todos divididos em várias tabelas, as quais são vinculadas pelo uso de chaves primárias e estrangeiras, evitando por exemplo a redundância de informações e perda de dados. Apesar da confiança e precisão, os bancos de dados relacionais são menos eficientes que os não relacionais no quesito busca de dados quando a quantidade de registros é muito grande, ou seja, a busca se torna lenta e possivelmente problemática quando se precisa trabalhar com um grande volume de dados.



Os bancos de dados do tipo não relacional, por outro lado, possuem registros que estão sempre agrupados, o que gera uma grande repetição de informações. Apesar da redundância de informações, os bancos de dados no-SQL (não relacionais) permitem buscas com uma maior velocidade ao se trabalhar com grande massa de dados.

Notamos através desse estudo, que os bancos de dados relacionais e não relacionais atendem tipos diferentes de intenções, ou seja, eles são banco de dados complementares os quais devem ser escolhidos baseando-se no contexto final do software que será construído, e não em gostos pessoais dos desenvolvedores.

Uma vez que nossa pesquisa pretende trabalhar com pequena massa de dados, optamos por trabalhar com o banco de dados do tipo relacional, pois esse tipo de banco nos garantirá uma maior efetividade no que diz respeito a pouca redundância e confiabilidade na persistência dos dados.

Depois de amadurecer o conceito de agentes de software, estudar um modelo já existente de software, e ainda, entender o funcionamento de bancos de dados relacionais e não relacionais, iniciamos a construção do nosso próprio modelo de agente educacional para ensino-aprendizagem, o qual pretende basear-se em um modelo de usuário já existente.

### 3 CONSIDERAÇÕES FINAIS

Visando demonstrar a eficácia da abordagem de agentes de software no ensino-aprendizagem, o ideal seria além do estudo teórico, um teste prático em um ambiente propício. Entretanto, devido ao grande volume de informações à serem consideradas para a criação do agente e ao pequeno intervalo de tempo para a construção do artigo, não foi possível um aprofundamento prático em relação ao tema.

Contudo, o conteúdo do artigo é sólido e apresenta uma ótima solução para o complemento do ensino-aprendizagem de qualquer tipo de conhecimento.

### REFERÊNCIAS

BANYON, D. (2011). **Interação Humano Computador** (2 ed.). (H. C. Sousa, Trad.) São Paulo, SP: Pearson Prentice Hall.

CÔGO, Felipe Roseiro. **Uma abordagem para o uso do conceito de Folkauthority em sistemas de recuperação de informação**. Maringá. 2012.

DE BARROS, A. B. P. (2009). **Utilização de Agentes na Engenharia de Software**.

FONTES, L. M. O.; NETO, F. M. M.; DINIZ, F. A.; CARLOS, D. G.; JÚNIOR, L. J.; DA SILVA, L. C. N. **Um Agente Pedagógico Animado de Apoio à Aprendizagem Baseada em Problema**. IEEE-RITA, Vol. 7, Num. 4, Nov 2012.

GIL, A. C. (2002). **Como elaborar projetos de pesquisa** (4a. ed.). São Paulo, SP: Atlas.



IANNI, V. Introdução aos bancos de dados NOSQL. **DEVMEDIA**. Disponível em: <<http://www.devmedia.com.br/introducao-aos-bancos-de-dados-nosql/26044>> Acesso em 29 de maio de 2014 às 10:37.

MAES, P. (1994). **Agents that reduce work and information overload** (Vol. 37). New York, NY, USA: ACM.

**mongodb**. Disponível em: <<http://www.mongodb.com/>> Acesso em 12 de março de 2014 às 8:38.

NÉRI, E. L. **Agentes de Software: Delegando Decisões a Programas**. RAE-Eletrônica, Vol. 4, Num. 1, Art. 3, Jan/Jul 2005.

PEREIRA, C. NoSQL e o conceito de banco de dados não relacional. **FXP Labs**. Disponível em: <<http://www.fxplabs.com.br/blog/nosql-conceito-de-banco-de-dados-nao-relacional/>> Acesso em 29 de maio de 2014 às 10:58.

PEREIRA, R. **Folkauthority: A aplicação do conceito de autoridade cognitiva por meio de folksonomia**. Maringá. 2008.

PERELLA, F. **NoSQL e as vantagens na utilização do MongoDB**. Disponível em: <<http://pt.slideshare.net/fabioperrella56/no-sql-e-as-vantagens-na-utilizacao-do-mongodb>> Acesso em 12 de março de 2014 às 8:23.

PREECE, J.; ROGERS, Y.; SHARP, H. (2012). **Design de Interação. Além da interação homem-computador**. (3 ed.). (V. Possamai, Trad.) Porto Alegre, RS: Bookman.

RICHARD, N.; YAMADA, S. (2007). **An Adaptive, Emotional, and Expressive Reminding System**.

RUSSELL, Stuart J.; NORVIG, P. (2002). **Inteligencia Artificial**. (2 ed.). São Paulo: CAMPUS.

STEPPAT, N. Bancos de dados não relacionais e o movimento NoSQL. **Caelum**. Disponível em: <<http://blog.caelum.com.br/bancos-de-dados-nao-relacionais-e-o-movimento-nosql/>> Acesso em 29 de maio de 2014 às 10:45.