

SMART ITS: um Sistema Tutor Inteligente para Flexibilização e Adaptação de Currículos

Darlan Anschau, Jerusa Marchi, Elder R. Santos

¹ Departamento de Informática e Estatística
Programa de Pós-Graduação em Ciência da Computação
Universidade Federal de Santa Catarina – UFSC

darlan.anschau@gmail.com, jerusa.marchi@ufsc.br, elder.santos@ufsc.br

Abstract. *The flexibilization and personalization of resources and activities is a theme for intense researches in the Intelligent Tutoring Systems area. The Smart ITS multiagent architecture proposal utilizes the MATHEMA conceptual model to adapt a course tailored to the learner, by considering the learner knowledge in relation to the course domain knowledge, in which the Tutor Agent is specialist. The Tutor Agent's main objective is to adapt dynamically, with Temporal Logic beliefs, the sequencing of the Learning Objects that satisfy the learner's needs. The Smart ITS implementation was developed in Jacamo and follows the Agents & Artifacts paradigm.*

Resumo. *A flexibilização e personalização de recursos e atividades é tema de intensas pesquisas na área de Sistemas Tutores Inteligentes. A proposta de arquitetura multiagente do Smart ITS utiliza do modelo conceitual MATHEMA para adaptar um curso sob medida ao Aprendiz, considerando o conhecimento do mesmo em relação ao domínio de conhecimento do curso em que o Agente Tutor é especialista. O objetivo principal do Agente Tutor é adaptar dinamicamente, com crenças de Lógica Temporal, a sequência de Objetos de Aprendizagem para satisfazer as necessidades do Aprendiz. A implementação do Smart ITS foi desenvolvida no Jacamo e segue o paradigma de Agentes & Artefatos.*

1. Introdução

A área da Inteligência Artificial na Educação (IA-ED) emprega técnicas de Inteligência Artificial (IA) para construção de modelos úteis em programas computacionais educacionais, visando melhorias na aprendizagem [Frigo et al. 2007]. Um usuário Aprendiz geralmente deseja que um sistema educativo lhe forneça condições de adquirir e expandir o seu conhecimento. Para isto, duas abordagens de sistemas fundamentam esta área [Grubišić et al. 2015]: os Sistemas Tutores Inteligentes (STI), que preocupam-se com o conteúdo de um domínio específico de conhecimento, e os Sistemas Hipermídia Adaptativos Educacionais (SHAE), fazendo uso de melhorias provindas do desenvolvimento *Web* para apresentação e navegabilidade de novos recursos disponíveis.

Nos últimos anos houve um crescimento significativo de *Massive Open Online Courses* (MOOC), que apesar de serem colaborativos, até o momento, contém modelos educacionais simples que obteriam grandes melhorias com a utilização de métodos já maduros da IA-ED [Pinkwart 2016]. Neste sentido, pela área da educação ser complexa,

os STIs possibilitam maior abstração por fazer uso de Sistemas MultiAgentes (SMA) para tratar problemas educacionais de forma distribuída [Pozzebon et al. 2008].

Os STIs estão se tornando especialistas em adaptatividade e o conhecimento está se tornando um verbo e não mais um substantivo [Roll and Wylie 2016]. Sob estes aspectos um STI deve ser *adaptativo*, de acordo com a identificação das preferências do Aprendiz; e, por consequência, *personalizado*, para atender as demandas impostas pelas necessidades do Aprendiz e tornando possível acompanhar sua trajetória. A representação de conhecimento, tanto no modelo de Aprendiz quanto do modelo de domínio, são fundamentais para os STIs na busca pela personalização, resultando na melhora da aprendizagem [Sani and Aris 2014, Victorio-Meza et al. 2014].

O objetivo deste artigo é o desenvolvimento de uma arquitetura de STI baseada em agentes na qual seja permitido a um Agente Tutor fazer uso da representação de conhecimento em Lógica Temporal, conhecimento este que será usado para promover o sequenciamento de Objetos de Aprendizagem (OA) em uma estrutura de curso, possibilitando assim uma maior adaptatividade e personalização do sistema ao Aprendiz.

O restante deste artigo está organizado desta forma: a Seção 2 compara esta implementação com trabalhos relacionados; na Seção 3, o STI MATHEMA é descrito como base neste trabalho; a Seção 4 apresenta brevemente um conjunto de ferramentas e tecnologias que dão suporte à construção do STI; na Seção 5, a arquitetura proposta é apresentada; na Seção 6, é apresentado um experimento a partir da implementação; e, por fim, a Seção 7 traz algumas considerações finais e perspectivas futuras.

2. Trabalhos Relacionados

Trabalhos recentes na área de STI influenciaram a atual pesquisa pela influência do sistema MATHEMA: [Silva et al. 2011] apresentam um STI que integra SMA à Sistemas Gerenciadores de Aprendizagem (SGA) *Modular Object-Oriented Dynamic Learning Environment* (MOODLE); e [Silva et al. 2014] discorrem sobre modelos de construção de STI em SMA, com domínio no modelo conceitual MATHEMA.

Em relação à seleção de OAs, [Junior and Silveira 2015] basea-se em SMA e no padrão *Sharable Content Object Reference Model* (SCORM), de [ADL 2009], para utilizar um novo tipo de OA denominado ILO (*Intelligent Learning Object*). O propósito do trabalho é a seleção dinâmica de novos objetos, a partir de raciocínio sobre dados provindos do curso e da interação com o OA mostrado ao Aprendiz. No entanto, esta solução ainda não é considerado um Sistema de Recomendação (SR), por se limitar a substituir OAs para reforçar um mesmo conteúdo e, portanto, não usa de planejamento para uma sequência personalizada de curso. Neste mesmo sentido, a especificação Tin Can [Rustici 2013] provê compatibilidade com a sua sucessora SCORM para fornecer facilidade no tratamento dos resultados de interações e diagnósticos das interações com OAs. O padrão Tin Can oferece boa extensibilidade e autonomia de controle das atividades e artefatos, que possibilitam interações e consultas facilitadas pela sua API.

Em sequenciamento de OAs, destaca-se o trabalho de [Garrido et al. 2016], que também faz uso de planejamento e, da mesma forma, faz necessário a conversão do plano sequenciado para metadados dos OAs. Contudo, tal planejamento não integra o raciocínio BDI, nem contém suporte à Lógica Temporal, e tampouco possui uma estrutura de curso

como apresentada no presente artigo. Diferente de [Garrido et al. 2016], a proposta apresentada neste artigo visa uma solução genérica que pode integrar suporte a diversas ferramentas para o sequenciamento de OAs por meio do modelo conceitual MATHEMA.

3. STI MATHEMA

Na arquitetura do STI MATHEMA [de Barros Costa 1997], do qual este artigo se baseia para a arquitetura Smart ITS, um agente é uma entidade composta por três sistemas: Sistema Tutor é responsável pela interação com o Aprendiz; Sistema Social é responsável pela cooperação entre agentes tutores; e Sistema de Distribuição é responsável pela manipulação de mensagens. A partir da arquitetura do Sistema Tutor, o STI MATHEMA utiliza do Modelo de Sobreposição para a modelagem do Aprendiz.

A criação do Modelo do Aprendiz consiste na coleta de informações relevantes e na inferência do estado cognitivo do Aprendiz, de maneira acessível e proveitosa para o sistema [Chrysafiadi and Virvou 2013]. Neste sentido, o Modelo de Sobreposição (*Overlay*, em inglês), assume que o Aprendiz possui conhecimento insuficiente, porém correto sobre um determinado domínio [Victorio-Meza et al. 2014]. Assim sendo, o conhecimento do Modelo do Aprendiz é um subconjunto que representa o nível de conhecimento dentro do Domínio de Conhecimento do curso no qual o Agente Tutor é especialista. A função do Agente Tutor é ampliar o nível de conhecimento do Modelo do Aprendiz, ou por outro ponto de vista, aproximá-lo do Domínio de Conhecimento que ele possui.

No STI MATHEMA, a construção do Modelo do Aprendiz é pautada pela definição do seu modelo conceitual [Silva et al. 2011]. Este modelo integra estruturas hierárquicas referentes ao domínio de conhecimento e estratégias pedagógicas, auxiliando significativamente na resolução de problemas devido às suas estruturas bem definidas.

3.1. Modelo Conceitual MATHEMA

A modelagem do conhecimento do domínio no STI MATHEMA [de Barros Costa 1997] é organizada em duas formas de visualização da sua estrutura conceitual: a visão externa e a visão interna. Em sua visão externa, um domínio específico é separado em subdomínios representados em três dimensões de conhecimento: o *Contexto (C)*, caracteriza diferentes pontos de vista sobre o domínio e é responsável pela definição de diferentes interpretações; a *Profundidade (P)*, refere-se a um contexto particular do qual é realizada a estratificação em diferentes níveis epistemológicos; e a *Lateralidade (L)*, que diz respeito ao conhecimento de suporte a um dado objeto do domínio.

Em sua visão interna, cada uma destas dimensões é tomada como um plano: plano pedagógico, plano de problemas e plano de suporte, respectivamente. Cada conhecimento associado a um subdomínio é organizado em um ou mais currículos. O currículo leva o nome de *curriculum*. Cada *curriculum* consiste de um conjunto de Unidades Pedagógicas (UP), onde cada UP é associada a um conjunto de Problemas: $curriculum = \{up_1, up_2, \dots, up_n\}$. Por sua vez, cada Problema contém Unidades de Interação (UI), utilizadas para a apresentação dos problemas ao Aprendiz, e são efetivamente as unidades que possuem a responsabilidade pela interação com o Aprendiz. Estas UI são equivalentes aos Objetos de Aprendizagem na nossa proposta.

4. Tecnologias e Ferramentas

Dentre o conjunto de lógicas temporais, a *Propositional Temporal Logic* (PTL), descrita por [Fisher 2011] - ou ainda denominada *Propositional Linear Temporal Logic* (PLTL) -, é formalmente descrita com expressividade suficiente para a especificação de crenças temporais em sequência num determinado domínio. Porém, para a semântica deste trabalho também utilizou-se de uma representação do passado, sendo uma adaptação apresentada no Quadro 1, da linguagem METATEM [Barringer et al. 1995] e seus operadores lógicos na *Propositional MetateM Logic* (PML). Este tipo de representação é utilizada pelo Agente Tutor para descrever suas crenças dentro das semânticas de Passado, Ação e Plano.

Quadro 1. Conversão dos operadores da Lógica Temporal PML para predicados

Símbolo	Predicado	Semântica Genérica
<i>Operadores Unários de Futuro</i>		
$\bigcirc\varphi$	next	φ é satisfeito no próximo momento
$\diamond\varphi$	poscon	φ é satisfeito agora ou num futuro momento
$\square\varphi$	future	φ é satisfeito para todo o futuro
<i>Operadores Unários de Passado</i>		
$\bullet\varphi$	prev	φ é satisfeito no momento anterior
$\blacklozenge\varphi$	precon	φ é satisfeito em algum momento no passado
$\blacksquare\varphi$	past	φ sempre satisfeito no passado
<i>Operadores Binários de Futuro</i>		
$\varphi\mathcal{U}\psi$	until	φ é satisfeito até que ψ ocorra
<i>Operadores Binários de Passado</i>		
$\varphi\mathcal{S}\psi$	since	φ foi satisfeito desde que ψ ocorreu

Da mesma forma, tem-se feito uso de OAs para criação, manutenção e atualização das informações de domínio, cuja descrição é feita a partir de metadados elaborados por especialistas humanos. Para o uso de Objetos de Aprendizagem, o antigo paradigma de Programação Orientada a Agentes (POA) [Shoham 1993] não pode ser puramente aplicado. Neste contexto, a implementação da arquitetura Smart ITS utiliza do paradigma Agentes & Artefatos (A&A) [Ricci et al. 2008], onde agentes são entidades ativas que podem ter a capacidade de deliberação e de comunicação com outros agentes da sociedade artificial. Por outro lado, artefatos são entidades com funções simples e servem como ferramentas, ou utensílios, dos quais cabe aos agentes empregar maiores utilidades.

Na implementação são utilizados o interpretador de AgentSpeak(L) em Java, Jason [Bordini et al. 2007], a biblioteca para artefatos, Cartago [Ricci et al. 2006], e o organizador Moise, integrantes do *framework* Jacamo [Boissier et al. 2013]. O Jacamo possibilita ao STI a representação temporal do sequenciamento de OAs adaptado ao Aprendiz.

5. Arquitetura de STI Considerando a Temporalidade do Aprendiz

A estrutura do Smart ITS, conforme Figura 1, é baseada no STI MATHEMA e contém em seu ambiente três tipos de agentes artificiais: Agentes Tutores, Agentes de Interface e Agentes Aprendizes. O Agente Tutor possui crenças com predicados que expressam os operadores de Lógica Temporal apresentados para a identificação do estado atual do

Aprendiz em relação ao Domínio de Conhecimento. Esta identificação ocorre em relação ao modelo conceitual MATHEMA, para então propiciar a seleção e sequenciamento de Objetos de Aprendizagem (OAs). Portanto, a principal função do Agente Tutor é preencher um currículo com OAs que satisfaçam as necessidades do Aprendiz.

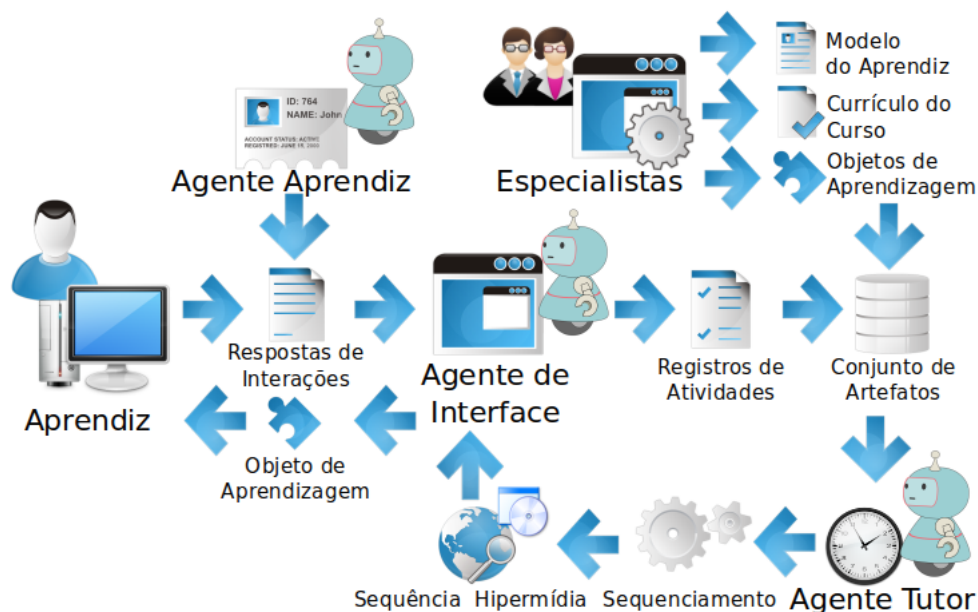


Figura 1. Visão Geral da Arquitetura Smart ITS

O Agente de Interface é responsável pelo controle das informações do curso, por meio do *Learning Management System* (LMS), e serve de canal de comunicação entre o Agente Tutor e o Agente Aprendiz. Ele referencia os OAs conforme o plano do currículo produzido pelo Agente Tutor, apresentando as atividades selecionadas e armazenando dados da interação com o Aprendiz. Por fim, o Agente Aprendiz representa o Aprendiz físico, o usuário foco do sistema, na forma de um agente artificial.

A base para o sequenciamento de atividades na estrutura do currículo é a verificação da diferença entre o conhecimento prévio do Aprendiz e o Domínio de Conhecimento. A diferença é determinante para decidir qual conteúdo deve ser apresentado posteriormente e, para tal, as crenças do Agente Tutor possuem predicados que equivalem a operadores temporais inspirados na *Propositional MetateM Logic* (PML). Desta forma, o Agente Tutor pode acessar os registros de atividades do Aprendiz e diagnosticar a situação dele no curso, planejar sobre as futuras ações sobre o domínio estruturado no modelo conceitual MATHEMA - com diferentes pontos de vista de acordo com a corrente pedagógica construtivista-, e elaborar uma sequência de ações no curso individualmente através de atividades com OAs que melhor servem ao Aprendiz no momento.

Para criar uma sequência de aprendizagem, o Agente Tutor faz uso de três artefatos que armazenam informações importantes à dinâmica de execução do sistema: Artefato de Domínio, Artefato de Objetos de Aprendizagem e Artefato do Aprendiz. O Artefato de Domínio contém os dados do Domínio de Conhecimento. Este conhecimento está estruturado conforme o modelo conceitual MATHEMA, em Contextos, Profundidades e Lateralidades. A criação e manutenção deste artefato é feita pelo Especialista de Domínio

ou Designer Instrucional; O Artefato de Objetos de Aprendizagem contém, para cada ação com OA, os dados de identificação do repositório, anotações temporais, Estilo de Aprendizagem e triplas *Resource Description Framework* (RDF) para uso de outros metadados. A criação e manutenção deste artefato é responsabilidade do Designer de Conteúdo; Por fim, o Artefato do Aprendiz contém três tipos de dados sobre o Aprendiz: dados pessoais; dados sobre o modelo cognitivo e preferências do Aprendiz, suportando estilos de aprendizagem; e dados de conhecimento prévio, que são obtidos pelo Agente Tutor e mantidos como experiências, passadas oriundas da interação com o sistema.

O Agente Tutor adiciona crenças segundo as semânticas de Passado, Ação e Plano que atribui aos artefatos do Smart ITS. A Semântica de Passado é inferida a partir das sentenças do Artefato do Aprendiz que são convertidas para crenças lógicas de passado.

A Semântica de Ação é inferida a partir do Artefato de Objetos de Aprendizagem, onde uma ação representa um desejo a ser realizado por uma atividade com um OA. Desejos são crenças que o agente quer que sejam satisfeitas e o desejo mais propício é selecionado como intenção. Ações são anotadas com pré e poscondições para que sirvam de parâmetro, respectivamente, entre o Domínio do Aprendiz (passado) e o que o Aprendiz ainda não sabe no Domínio de Conhecimento (futuro) para que o Agente Tutor possa então determinar os melhores OAs a se usar nas atividades.

A Semântica do Plano é inferida pelo Agente Tutor a partir do Artefato de Domínio. O plano, portanto, é uma sequência de ações envolvendo características do Aprendiz com base na estrutura do modelo conceitual MATHEMA. Por meio dos contextos do plano em BDI, o Agente Tutor sequencia OAs Aprendiz de forma personalizada.

Existem dois tipos de espaços de trabalho no Smart ITS em que os agentes podem interagir com outros agentes e artefatos: o Espaço de Trabalho de Elaboração do *curriculum* (ETEC) - onde ocorre a comunicação entre Agente Tutor e Agente de Interface; e o Espaço de Trabalho de Interação Pedagógica (ETIP) - onde ocorre a comunicação entre Agente de Interface e Agente Aprendiz. Tal como ocorre em um agente BDI, o STI proposto trabalha com ciclos de percepção, deliberação e ação. O ciclo deliberativo do Agente Tutor é desencadeado pelas ações do Aprendiz na forma de Agente Aprendiz após armazenados em artefato. Estes ciclos são desencadeados pelo usuário Aprendiz, e são detalhados através dos protocolos de comunicação descritos abaixo.

5.1. Protocolos de Comunicação

O primeiro protocolo de comunicação, Figura 2, descreve o ETEC. Ele demonstra o ciclo do Agente Tutor que consiste em cinco passos: (1), o Agente Tutor espera por informações do Agente de Interface referentes a sua responsabilidade na tutoria de novos Aprendizes e os respectivos Artefato de Aprendiz (AA), Artefato de Domínio (AD) e Artefato de OAs (AOA); (2), o Agente Tutor consulta os artefatos e: se não há atualizações, repete (1) e (2), de outra forma, atualiza suas crenças e representa os novos dados com predicados temporais da semântica; (3), o Agente Tutor manipula as crenças do que sabe sobre Aprendiz e compara com as que mantém sobre o Domínio de Conhecimento para então criar o Conjunto Diferença; (4), o Agente Tutor compõe um plano para cada um dos Aprendizes ao escolher uma sequência de OAs, respeitando os contextos e as regras temporais; (5), o Agente Tutor converte o plano representado em PTL para a criação do Artefato do *curriculum* (AC) que será utilizado pelo Agente de Interface.

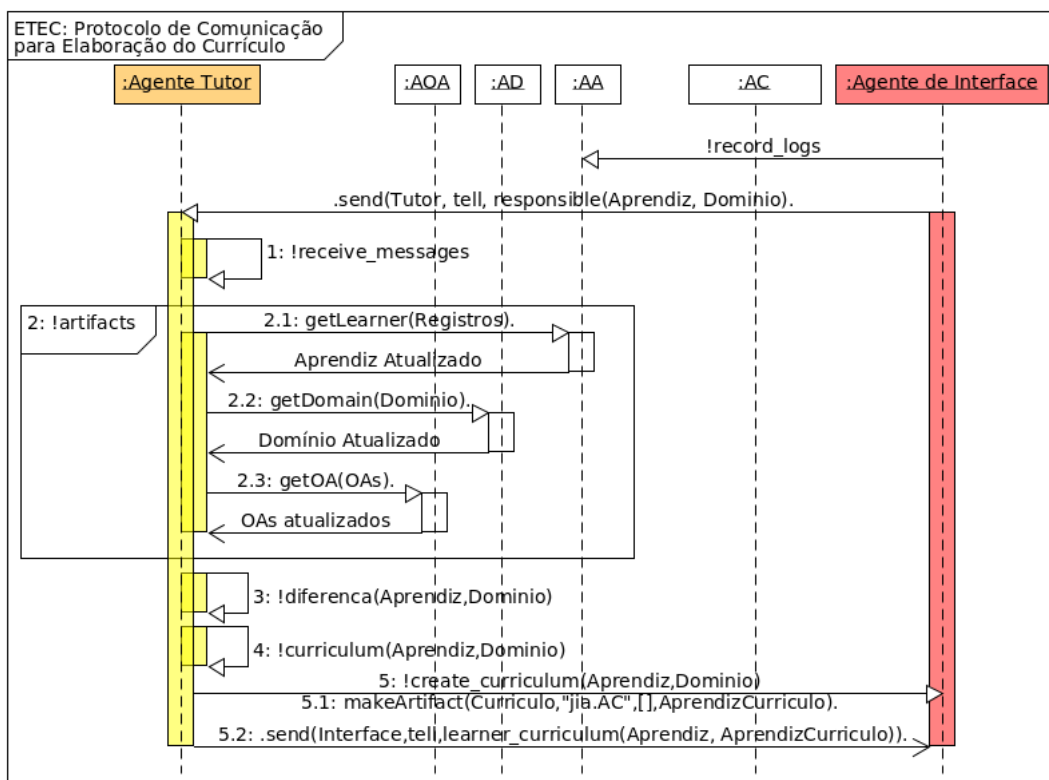


Figura 2. Protocolo do Espaço de Trabalho de Elaboração do Currículo

O segundo protocolo de comunicação, que não é o principal foco deste artigo, descreve o ETIP e corresponde às Interações Pedagógicas. Este protocolo resume-se em: (1) o Agente de Interface informa o Agente Aprendiz de um OA para interação; (2) o Agente Aprendiz inicia a Interação Pedagógica com o OA; (3) os resultados das atividades retornam ao Agente de Interface para serem armazenadas no Artefato de Aprendiz.

6. Experimento

Suponha que haja um curso de métodos algorítmicos onde Bob é o Agente Tutor, Ana uma Aprendiz e onde o Domínio D do curso contém os contextos, profundidades e lateralidades, conforme a Figura 3, com (P) atividades práticas (P) e teóricas (T);

Contextos

C_1 = Iteração

C_2 = Recursão

Profundidades

C_1P_1 = Laços de repetição (T)

C_2P_1 = Recursão de função (T)

C_1P_2 = Algoritmos iterativos (P)

C_2P_2 = Algoritmos recursivos (P)

C_1P_3 = Armazenamento iterativo (P)

C_2P_3 = Armazenamento recursivo (P)

Lateralidades

$C_1P_1L_1$ = Aula - Laços de repetição

$C_2P_1L_1$ = Aula - Chamadas de função

$C_1P_2L_1$ = Implementação - Fatorial iterativo

$C_2P_2L_1$ = Implementação - Fatorial recursivo

$C_1P_2L_2$ = Trabalho - Fibonacci iterativo

$C_2P_2L_2$ = Trabalho - Fibonacci recursivo

$C_1P_3L_1$ = Teste - Lista iterativa

$C_2P_3L_2$ = Teste - Pilha recursiva

Figura 3. Exemplo de *currículo* no modelo conceitual MATHEMA

Suponha que a Aprendiz Ana já acompanhou uma aula sobre laços de repetição ($C_1P_1L_1$) e implementou um algoritmo fatorial iterativo ($C_1P_2L_1$). O conjunto diferença é dado por todo o Domínio menos estas atividades já realizadas pela Aprendiz. Para ilustrar de forma bem simples as crenças do Agente Tutor com a finalidade de demonstrar a inferência que ele realiza, tem-se este passado descrito em sua base de crenças como $\blacksquare C_1P_1L_1$ e $\blacksquare C_1P_2L_1$, fazendo com que a profundidade 1 também esteja satisfeita, $\blacksquare C_1P_1$. A próxima lateralidade, $\circ C_1P_2L_2$, é realizar um trabalho de implementação de um algoritmo de Fibonacci iterativo. Existe um OA no Estilo de Aprendizagem de Ana que possui as pré-condições $\diamond C_1P_1$ e $\diamond C_1P_2L_1$, com a pós condição $\diamond C_1P_2$, logo, é selecionado. Esta atividade é então removida do Conjunto Diferença. Depois Ana deve realizar um teste que consiste na elaboração de uma lista com algoritmo iterativo ($C_1P_3L_1$). Ao terminar esta atividade o Aprendiz completará as três profundidades que estavam no contexto de iteração, ou seja, completando C_1 , resultando na retirada deste contexto, $\blacksquare C_1$, do Conjunto Diferença. E assim por diante, o Agente Tutor, após ter construído o Conjunto Diferença do momento atual de Ana, vai adicionando OAs para o *currículo* de Ana e subtraindo estas proposições do deste Conjunto Diferença para saber qual é a próxima proposição a ser satisfeita. Este processo ocorre até a finalização do currículo, Figura 4.

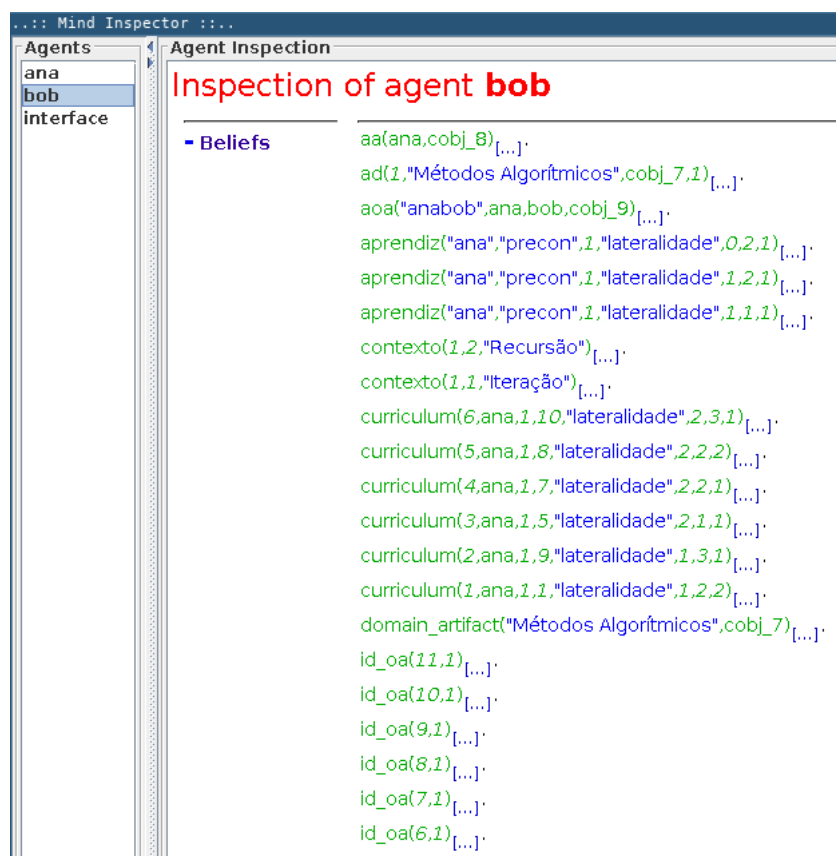


Figura 4. Parte das crenças do Agente Tutor Bob, representando a sequência de OAs no *currículo*, a partir da próxima atividade não satisfeita ($C_1P_2L_2$) no curso da Ana

Por limitações de espaço no artigo, foram ocultadas as crenças de Bob sobre: o Domínio de Conhecimento no modelo conceitual MATHEMA, o perfil de Ana com estilo de aprendizagem, detalhes dos onze OAs criados, como endereços, pré e pós-condições,

anotações de triplas RDF, além do conjunto diferença, agora convertido em *curriculum*.

Dado o exemplo apresentado, o experimento demonstra que as crenças após a deliberação do Agente Tutor Bob expressam uma sequência de OAs no *curriculum* da Aprendiz Ana. Esta sequência ocorre ao percorrer os número dos *ids*, identificadores seriais, do formato `curriculum(id, aprendiz, dominio, idoa, dimensao, contexto, profundidade, lateralidade)` em sua ordem. Devido à Lógica Temporal, estes identificadores capturam o estados/momento do curso em que o OA deve ser apresentado e são úteis para a navegação entre os OAs. Caso a situação do mundo mude, a deliberação do Agente Tutor leva em consideração os novos fatos e cria uma nova sequência.

7. Considerações Finais

Este artigo apresentou uma arquitetura STI que integra um STI MATHEMA ao ciclo de raciocínio BDI, cujo principal enfoque é o sequenciamento de OAs. O experimento da implementação demonstrou ser possível utilizar de Lógica Temporal para adaptar e flexibilizar *curricula* para atender às necessidades educacionais de Aprendizes.

Este trabalho tem como principais pontos de contribuição da adição da Lógica Temporal: estruturação de conhecimento sólida e clara, provida pelo modelo conceitual; flexibilização e adaptação do conteúdo, dadas as anotações temporais das crenças; e simplicidade de personalização, pois o ciclo de raciocínio BDI não precisa ser alterado.

Como trabalhos futuros, em primeiro momento, visa-se estender a implementação para utilizá-la em um curso real, seguindo o padrão Tin Can agregado ao SGA Moodle para uma melhor validação do modelo. Em um segundo momento, espera-se poder contemplar melhor o suporte para padrões de metadados e Estilos de Aprendizagem, aproveitando melhor dos aspectos de raciocínio em função do tempo nestes quesitos.

Referências

- [ADL 2009] ADL (2009). *SCORM 2004 4th Edition Sequencing and navigation*. Advanced Distributed Learning.
- [Barringer et al. 1995] Barringer, H., Fisher, M., Gabbay, D., Gough, G., and Owens, R. (1995). Metatem: An introduction. *Formal Aspects of Computing*, 7(5):533–549.
- [Boissier et al. 2013] Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. (2013). Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6):747–761.
- [Bordini et al. 2007] Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak Using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons.
- [Chrysafiadi and Virvou 2013] Chrysafiadi, K. and Virvou, M. (2013). Review: Student modeling approaches: A literature review for the last decade. *Expert Syst. Appl.*, 40(11):4715–4729.
- [de Barros Costa 1997] de Barros Costa, E. (1997). *Um modelo de ambiente interativo de aprendizagem baseado numa arquitetura multi-agentes*. PhD thesis, UFPA.
- [Fisher 2011] Fisher, M. (2011). *An Introduction to Practical Formal Methods Using Temporal Logic*. Wiley.

- [Frigo et al. 2007] Frigo, L. B. et al. (2007). *Um modelo de autoria para sistemas tutores adaptativos*. PhD thesis, UFSC.
- [Garrido et al. 2016] Garrido, A., Morales, L., and Serina, I. (2016). On the use of case-based planning for e-learning personalization. *Expert Syst. with Appl.*, 60:1 – 15.
- [Grubišić et al. 2015] Grubišić, A., Stankov, S., and Žitko, B. (2015). Adaptive courseware: A literature review. *j-jucs*, 21(9):1168–1209.
- [Junior and Silveira 2015] Junior, J. A. and Silveira, R. (2015). Seleção dinâmica de objetos de aprendizagem baseados no padrão scorm no ambiente moodle. In *Simpósio Brasileiro de Informática na Educação-SBIE*, volume 26, page 967, Maceió, AL. Sociedade Brasileira de Computação - SBC.
- [Pinkwart 2016] Pinkwart, N. (2016). Another 25 years of aied? challenges and opportunities for intelligent educational technologies of the future. *Inter. Journal of Artif. Intell. in Education*, 26(2):771–783.
- [Pozzebon et al. 2008] Pozzebon, E. et al. (2008). *Um modelo para suporte ao aprendizado em grupo em sistemas tutores inteligentes*. PhD thesis, UFSC.
- [Ricci et al. 2008] Ricci, A., Piunti, M., Acay, L. D., Bordini, R. H., Hübner, J. F., and Dastani, M. (2008). Integrating heterogeneous agent programming platforms within artifact-based environments. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pages 225–232. International Foundation for Autonomous Agents and Multiagent Systems.
- [Ricci et al. 2006] Ricci, A., Viroli, M., and Omicini, A. (2006). Cartago: A framework for prototyping artifact-based environments in mas. In *International Workshop on Environments for Multi-Agent Systems*, pages 67–86. Springer.
- [Roll and Wylie 2016] Roll, I. and Wylie, R. (2016). Evolution and revolution in artificial intelligence in education. *Inter. Journal of Artif. Intell. in Education*, 26(2):582–599.
- [Rustici 2013] Rustici (2013). Anatomy of a tin can api statement.
- [Sani and Aris 2014] Sani, S. and Aris, T. N. (2014). Computational intelligence approaches for student/tutor modelling: A review. In *Intelligent Systems, Modelling and Simulation (ISMS), 2014 5th International Conference on*, pages 72–76. IEEE.
- [Shoham 1993] Shoham, Y. (1993). Agent-oriented programming. *Artif. intell.*, 60(1):51–92.
- [Silva et al. 2011] Silva, M. T., Bittencourt, I. I., and Costa, E. (2011). Modelos para a construção de sistemas multiagentes: um estudo de caso em sistemas tutores inteligentes. *Revista Brasileira de Informática na Educação*, 19(01):74.
- [Silva et al. 2014] Silva, S. B. J., Machado, V. P., and Araújo, F. N. C. (2014). Sistema tutor inteligente baseado em agentes na plataforma MOODLE para apoio as atividades pedagógicas da universidade aberta do piauí. In *Anais dos Workshops do III Congresso Brasileiro de Informática na Educação (CBIE 2014)*. Comissão Especial de Informática na Educação.
- [Victorio-Meza et al. 2014] Victorio-Meza, H., Mejía-Lavalle, M., and Rodríguez, G. (2014). Advances on knowledge representation of intelligent tutoring systems. pages 212–216.