

O LOGICAMENTE: a implementação colaborativa de Objetos de Aprendizagem de LÓGICA *

Patrick Terrematte¹, João Marcos¹, Thales Galdino¹

¹ Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte (DIMAp/CCET/UFRN)
59078-970 – Natal – RN – Brasil

terrematte@ppgsc.ufrn.br, jmarcos@dimap.ufrn.br, thalesgaldino@gmail.com

Resumo. A disciplina de Lógica é oferecida por diversos cursos de graduação com diferentes focos de aplicação. Para a maioria desses cursos, é notável como a disciplina representa um desafio pedagógico para os professores e os alunos de graduação, assim como é frequentemente alto o número de trancamentos e reprovações registrados. Diante da necessidade de uma formação sólida dos alunos na disciplina, propomos um conjunto de Objetos de Aprendizagem (OA) para o ensino de Lógica Aplicada à Computação: o LOGICAMENTE¹. As ferramentas que integram o sistema visam a ilustrar conceitos e algoritmos fundamentais da Lógica e a permitir que os estudantes façam experimentos práticos envolvendo a compreensão desses conceitos de forma interativa.

Palavras-chave: Objetos de Aprendizagem. Ensino de Lógica. Lógica.

Abstract. Logic is a subject offered to many undergraduate courses with different focal points. For most of those courses, it is remarkable how Logic represents a pedagogic challenge for both teachers and students, and the recorded number of cases of failures and of discontinuity is often high. Given the need to provide a solid basis in the subject, we propose a set of Learning Object (LO) for teaching Logic Applied to Computer Science: the LOGICAMENTE. The tools that integrate the system aim to illustrate fundamental concepts and algorithms from Logic, as well as to allow students to conduct interactive practical experiments involving the understanding of those concepts.

Key-words: Learning Objects. Teaching Logic. Logic.

1. Introdução

O uso da *Web* na educação oferece vantagens únicas que garantem a reutilização dos materiais didáticos produzidos e possibilitam metodologias para o controle de *feedback* dos alunos, para atividades dinâmicas e aprendizado colaborativo. De acordo com um relatório do Ministério da Educação dos EUA [Department of Education. US 2009], o aprendizado online tem vantagens metodológicas bem definidas sobre o ensino presencial.

No paradigma da Orientação a Objetos na computação, componentes são desenvolvidos para serem reutilizados em diversos contextos, assim também ocorre com a

*Os autores agradecem o apoio financeiro do CNPq/CAPES. Em particular, agradecimentos ao Prof. João Marcos, por toda sua atenção ao tornar possível, criar e orientar esse projeto.

¹Um protótipo do LOGICAMENTE pode ser visitado a partir do endereço <http://www.lolita.dimap.ufrn.br/logicamente>.

definição de Objeto de Aprendizagem (OA). Um OA é uma unidade que fornece um conteúdo epistemológico para estimular a reflexão do aluno no ensino uma disciplina. Um OA é qualquer entidade digital com um propósito educacional definido, que pode ser usada, reusada ou referenciada durante um processo de aprendizagem [Wiley 2002].

A ideia dos OAs é reduzir o custo na elaboração do suporte ao ensino e compartilhar material produzido. Além da reusabilidade dos recursos de OAs e a catalogação com a especificação de OAs, é possível obter outras vantagens em diversas dimensões como: *acessibilidade* através de um repositório, *interoperabilidade* de plataformas e *durabilidade* do recurso sem recodificação [Tarouco et al. 2003]. O interessante é produzir sistemas que integrem e validem esses requisitos.

Para explorar as vantagens da aplicação da *Web* no ensino de Lógica, desenvolvemos o projeto LOGICAMENTE e implementamos alguns OAs para realizar tarefas como, por exemplo, a geração automática de fórmulas com a complexidade desejada, a configuração de uma linguagem e a definição de novos conectivos, a tradução entre diferentes sintaxes, a construção de tabelas de verdade e tableaux de fórmulas, a apresentação interativa de fórmulas na forma de árvores, a implementação do método de resolução para a lógica clássica, a busca de modelos ou contra-modelos e, ainda, a utilização de um assistente para a demonstração de teoremas.²

Assim, vamos expor um conjunto interativo de OAs que reúne instrumentos de apoio ao ensino de Lógica aplicada à Computação. Nosso objetivo é produzir instrumentos integrados que ilustrem conceitos e algoritmos fundamentais da Lógica, permitindo que os estudantes façam experimentos pedagógicos práticos. Por fim, vamos apontar algumas expectativas e quais trabalhos futuros pretendemos executar para tornar o LOGICAMENTE um Ambiente Virtual de Aprendizagem de Lógica.

2. Caracterização do problema no ensino de Lógica

“Logic: The art of thinking and reasoning in strict accordance with the limitations and incapacities of the human misunderstanding.”

— AMBROSE BIERCE

The Collected Works of Ambrose Bierce (1911), Vol. 7, The Devil's Dictionary, 196.

Os currículos convencionais das nossas universidades privilegiam os alunos com um estilo de aprendizado intuitivo, verbal, dedutivo, refletivo e sequencial, em detrimento daqueles alunos, muitas vezes a maioria, que aprendem de outras formas [Barland et al. 2000, Felder and Silverman 1988]. Somando isso à falta de formação básica sólida dos estudantes que entram hoje nas nossas universidades e aos recursos precários de aprendizagem não-convencional disponíveis em nossas instalações, não há de surpreender que haja uma elevada taxa de reprovação e trancamento da disciplina de Lógica Aplicada à Computação em diversas universidades federais e centros tecnológicos.

Para resolver esses problemas no ensino da Lógica, investimos em um estilo de aprendizagem mais sensorial, visual, indutivo, ativo e global, visando a formar profissionais mais aptos a atender de forma eficiente às necessidades reais da comunidade.

Em uma busca pela internet, podemos encontrar alguns sistemas de apoio ao ensino de Lógica, tais como aqueles disponíveis na Tabela 1. No entanto, a maioria deles

²Para uma melhor compreensão dos conceitos de Lógica implementados no LOGICAMENTE, cf. [Barwise and Etchemendy 2002, van Dalen 2004].

Tabela 1. Lista de algumas ferramentas de Apoio ao Ensino de Lógica

#	NOME	CONTEÚDOS	LINGUAGEM	WEB
1	<i>Akka</i>	Lógica Prop. e 1ª Ordem; Modelos de Kripke	Java Applets	*
2	<i>Apros</i>	Lógica Prop. e 1ª Ordem; Dedução Natural	Java Web Start	*
3	<i>ASA-CalcPro</i>	Lógica Prop; Tableaux; Tabela de Verdade	Java	-
4	<i>Comp. Aristotelian Logic</i>	Lógica aristotélica	PHP	✓
5	<i>Expression Evaluator</i>	Lógica Prop. e 1ª Ordem; Analisa expressões	Perl/cgi	✓
6	<i>JAPE</i>	Lógica Prop. e 1ª Ordem; Dedução Natural	Java	-
7	<i>LoTREC</i>	Lógica Prop. e 1ª Ordem; Modal	Java Applets	*
8	<i>Pandora</i>	Lógica Prop. e 1ª Ordem; Dedução Natural	Java Applets	*
9	<i>Possible World Creation</i>	Lógica Prop.; Modal	Flash	✓
10	<i>ProofWeb</i>	Demonstrador Web de Teoremas; Dedução Natural	Ocaml	✓
11	<i>Tarski's World</i>	Lógica Prop. e 1ª Ordem; Semântica; Tableaux;	Java	*
12	<i>Fitch</i>	Lógica Prop. e 1ª Ordem; Dedução Natural	Java	*
13	<i>Boole</i>	Lógica Prop. ; Tabela de Verdade	Java	*
14	<i>The Daemon Proof Checker</i>	Lógica Prop. e 1ª Ordem; Contramodelos; Dedução Natural	C#	-
15	<i>Theorem Proving System</i>	Lógica Prop. e 1ª Ordem; Dedução Natural	Lisp	✓
16	<i>Tree Proof Generator</i>	Lógica Prop. e 1ª Ordem e Tableaux	JavaScript	✓
17	<i>Truth Table Constructor</i>	Lógica Prop.; Tabela de Verdade	Java Applet's	*
18	<i>Tableau3</i>	Lógica Prop. e Tableaux	Java Applets	*
19	<i>DiagVenn1.0</i>	Lógica Prop. e 1ª Ordem; Diagramas de Venn	Java Applets	*
20	<i>Modelos de Kripke</i>	Lógica Prop. e 1ª Ordem; Modelos de Kripke	Java Applets	*
21	LOGICAMENTE	Lógica Prop. e 1ª Ordem; e Outros	PHP	✓

* Parcialmente disponível online.

são sistemas *desktop*, cujo acesso é dificultado em diferentes sistemas operacionais, ou aplicativos *Web* baseados em *Java Applets*, que não guardam informações sobre a sessão do usuário. Esses sistemas são restritivos por precisarem de uma Máquina Virtual Java (JVM) e costumam ter um velocidade de execução baixa.

Um outro problema é que esses sistemas abrangem um domínio muito restrito do conteúdo de Lógica (apenas um ou dois conteúdos são implementados por cada sistema) e, além disso, não possuem atividades que desafiem e estimulem o usuário.

3. O Ambiente Virtual de Aprendizagem: o LOGICAMENTE

“La puissance de vision qui fait le poète,
et la puissance de déduction qui fait le savant,...”
— HONORÉ DE BALZAC
La Recherche de l'Absolu (1834).

Um Ambiente Virtual de Aprendizagem (AVA) é um sistema computacional concebido para oferecer suporte de ensino e aprendizagem em um mecanismo de acompanhamento educacional. Um AVA deve prover um conjunto de ferramentas, tais como avaliações, comunicação interativa, transferência de conteúdo, gerenciamento de atividades de alunos, ferramentas de acompanhamento de exercícios, verbetes colaborativos, *wikis* e módulos de aprendizagem específicos para cada conteúdo.

O protótipo do LOGICAMENTE foi implementado inicialmente no segundo semestre de 2006, sob a idealização e orientação do professor João Marcos, como um projeto final da disciplina de Lógica Aplicada à Computação. O objetivo era aplicar uma metodologia da Aprendizagem Baseada em Problemas (*Problem Based Learning*) para a implementação de alguns dos principais conceitos de Lógica. Nos semestres seguintes, o projeto continuou sendo desenvolvido a partir do trabalho colaborativo e voluntário de estudantes de computação, dando oportunidade à apresentação de trabalhos que exploravam tanto aspectos computacionais quanto os aspectos da informática aplicada à

educação [Vilela et al. 2008, Barros et al. 2008, Araújo et al. 2008]. O protótipo de tela inicial pode ser visto na Figura 1.

Atualmente, o LOGICAMENTE passou a ser desenvolvido com o auxílio de bolsistas de desenvolvimento tecnológico e de iniciação científica e tornou-se um projeto de mestrado, tendo como principal objetivo torná-lo um AVA para acompanhamento e ensino de temas relacionados à área de Lógica Aplicada à Computação.

Pretendemos, por conseguinte, implementar um conjunto de módulos que deverão representar um laboratório de conceitos de Lógica. O objetivo específico será a implementação de módulos interativos e OAs, envolvendo diversos conteúdos de Lógica.



Figura 1. Protótipo do LOGICAMENTE disponível *online* atualmente

3.1. Os Objetos de Aprendizagem do LOGICAMENTE

Atualmente, dispomos de nove módulos implementados, a saber: o *Settings*, o *Generator*, o *Formula Reader*, o *Truth-Table Constructor*, o *Tree Interaction*, o *Substitution Master*, o *Prenex Converter*, o *Skolemizer* e o *Resolution Game*, que serão descritos a seguir.

- *Settings*: nesse módulo, podemos escolher qual o estilo de notação desejamos utilizar. No momento, podemos optar pelo estilo de notação *Infix*, *Polish* e *Functional*. A notação *Infix* é a mais comum – $\alpha \wedge \beta$ –, ao passo que a notação *Polish* representa a Notação Polonesa em que são omitidos os parênteses e os conectivos precedem a fórmula que eles regem – $K\alpha\beta$ – e, por fim, a notação *Functional* expressa os conectivos como funções – $f(\alpha, \beta)$. Além disso, existe a opção de adicionar novos conectivos definindo o seu símbolo, sua aridade e ordem de associação. O objetivo é criar novos conectivos e salvá-los em um banco de dados para definir novas lógicas. Na figura 2(a), podemos ver o Protótipo do módulo *Settings*.

- *WFF³ Generator*: esse módulo é responsável por gerar arbitrariamente fórmulas proposicionais bem formadas, de acordo com as certas propriedades inseridas pelo usuário (número de ocorrências de conectivos, número de variáveis sentenciais etc). Com esse módulo, é possível gerar conjuntos de fórmulas de acordo com a complexidade desejada e elaborar automaticamente alguns exercícios com as fórmulas geradas. Também o aluno pode gerar fórmulas, colocá-las na área de transferência e verificar as suas propriedades

³Uma **Well Formed Formula**, ou fórmula bem formada (FBF), designa uma sequência de símbolos gerados por uma determinada linguagem formal.

em outros módulos. Na figura 2(c), podemos observar a geração de três fórmulas com dois conectivos e três átomos.

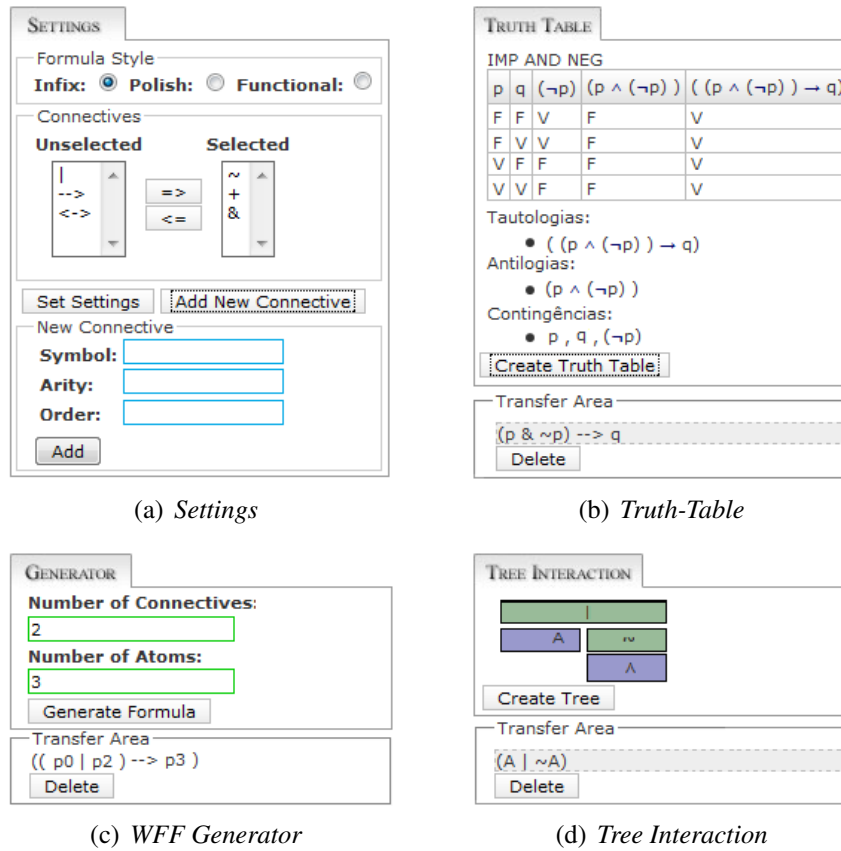


Figura 2. Protótipo de Módulos do LOGICAMENTE

- *Formula Reader*: módulo que realiza a inserção de fórmulas digitadas pelo usuário para a área de transferência. Essas fórmulas podem, assim, ser manipuladas por outros módulos.
- *Truth-Table Constructor*: possibilita a construção da tabela de verdade e o teste de satisfatibilidade de coleções de fórmulas, permite o acompanhamento de todos os estados possíveis das variáveis sentenciais e, conseqüentemente, da fórmula como um todo. Deverá dar suporte, em um segundo estágio, para a definição de novos operadores pelo usuário, bem como para avaliação de semânticas formais para certas lógicas não-clássicas que possam ser caracterizadas por interpretações em termos de tabelas de verdade. Na figura 2(b), observamos uma tabela de verdade para a fórmula $((p \wedge \neg p) \rightarrow q)$.
- *Tree Interaction*: na maior parte dos módulos do LOGICAMENTE, a estrutura de dados que corresponde às fórmulas é uma árvore, ou outras estruturas recursivas semelhantes. Esse módulo torna essas árvores visualizáveis, e permite ao usuário interagir de diversas formas com os nós e as folhas de tais árvores. Muitos outros usos são dados a esse módulo de interação, visualizar e interagir com tableaux, com derivações do tipo Resolução, com Dedução Natural etc. Na figura 2(d), observamos a geração da árvore representando a fórmula $(A \vee \neg A)$.
- *Substitution Master*: implementa uma função que recebe uma fórmula em primeira

ordem, uma variável a ser substituída e um termo, verificando, antes de mais nada, se o termo é livre para a variável na fórmula recebida⁴. Em caso afirmativo, a substituição é efetuada. Em caso contrário, o usuário é consultado sobre seu desejo de renomear convenientemente as variáveis ligadas da fórmula de modo a permitir que a substituição seja realizada. Na figura 3(c), temos a substituição da variável x pelo termo $f(a)$ na fórmula $\forall y R(x, y)$, resultando em $\forall y R(f(a), y)$.

- *WFF Converter*: esse módulo realiza algumas conversões de fórmulas para o uso do método de Resolução em lógica de primeira ordem, através da manipulação dos quantificadores e reduções adequadas em conectivos. Acompanha, passo a passo, a conversão de uma fórmula para a Forma Normal Conjuntiva (FNC)⁵ e permite transformar interativamente fórmulas para a Forma Normal Prenex (FNP)⁶. O módulo também recebe fórmulas de primeira ordem na FNP e permite a aplicação do método de skolemização⁷. Na figura 3(a), a fórmula $(\forall x \exists y R(x, y) \wedge \exists y Q(y))$ é convertida na seguinte FNP: $(\forall x_0 \exists x_1 \exists x_2 (R(x_0, x_1) \wedge Q(x_2)))$. A figura 3(b) mostra a fórmula $\exists x \forall y \exists z R(x, y, z)$, na qual os quantificadores existenciais são removidos e é convertida para a fórmula $\forall y R(a_0, y, f_0(y))$.

- *Resolution Game*: implementa o método de resolução básica, assistindo o aluno na eliminação de literais complementares⁸. Na figura 3(d), podemos ver uma aplicação do método de resolução para $\{\alpha \vee \beta, \neg\alpha, \neg\beta\} \longrightarrow \square$, demonstrando que $\{\alpha \vee \beta, \neg\alpha, \neg\beta\}$ é um conjunto contraditório de fórmulas.

Todos os OAs apresentados acima deverão englobar tanto a lógica clássica de primeira ordem como a lógica proposicional. O conjunto de OAs especificados acima deverá atender aos seguintes requisitos do sistema:

- Interface do usuário: para cada OA, uma interface amigável está implementada, para facilitar e tornar agradável a sua utilização. Além disso, há no sistema um guia para o usuário inexperiente poder utilizar o programa e sanar suas dúvidas.
- Correção e otimização: analisaremos cuidadosamente a implementação de todos os OAs para manter a correção dos algoritmos implementados.

No que diz respeito às tecnologias utilizadas, para permitir o seu livre acesso na Web, o LOGICAMENTE está implementado com um conjunto de *scripts* na linguagem PHP e em *Javascript*. Sua arquitetura obedece ao padrão *Model View Control* (MVC) e utiliza um conjunto de bibliotecas do *Zend Framework*⁹ para garantir o desenvolvimento

⁴Um termo t é dito **livre para** uma variável x numa fórmula bem-formada α se nenhuma ocorrência livre de x em α cai sob o escopo de alguma quantificação $\forall y$ ou $\exists y$ em que y é uma variável em t .

⁵Uma fórmula está na **Forma Normal Conjuntiva** se é uma conjunção de subfórmulas tais que cada uma das subfórmulas é uma disjunção de literais sendo cada literal é uma proposição atômica ou a negação de uma proposição atômica, isto é, tem a forma p ou $\neg p$.

⁶Uma fórmula está na **forma normal prenex** se possui a seguinte forma: $Q_1 x_1 \dots Q_n x_n \alpha$, onde α é uma fórmula livre de quantificadores, x_1, \dots, x_n são variáveis diferentes e para cada $1 \leq i \leq n$, Q_i é \forall ou \exists .

⁷Em homenagem ao matemático e filósofo Thoralf Skolem (1887-1963), dá-se o nome de **skolemização** ao método de eliminação de quantificadores existenciais em fórmulas fechadas na FNP, em que as variáveis são substituídas por funções de escolha e são introduzidos novos símbolos de constantes e funções.

⁸O método de resolução é utilizado para decidir se determinada sentença é ou não um teorema de uma fórmula na FNC. Se α e β são cláusulas-básicas, e $p_k \in \alpha$ e $\neg p_k \in \beta$, então o resultado da regra da ELC de α e β com respeito a p_k e $\neg p_k$, é $(\alpha - \{p_k\}) \cup (\beta - \{\neg p_k\})$, sendo $\gamma - \delta$ indicando o conjunto diferença de γ e δ , isto é, $\gamma - \delta \equiv \{x | x \in \gamma \text{ e } x \notin \delta\}$.

⁹Disponível em <http://framework.zend.com/>.

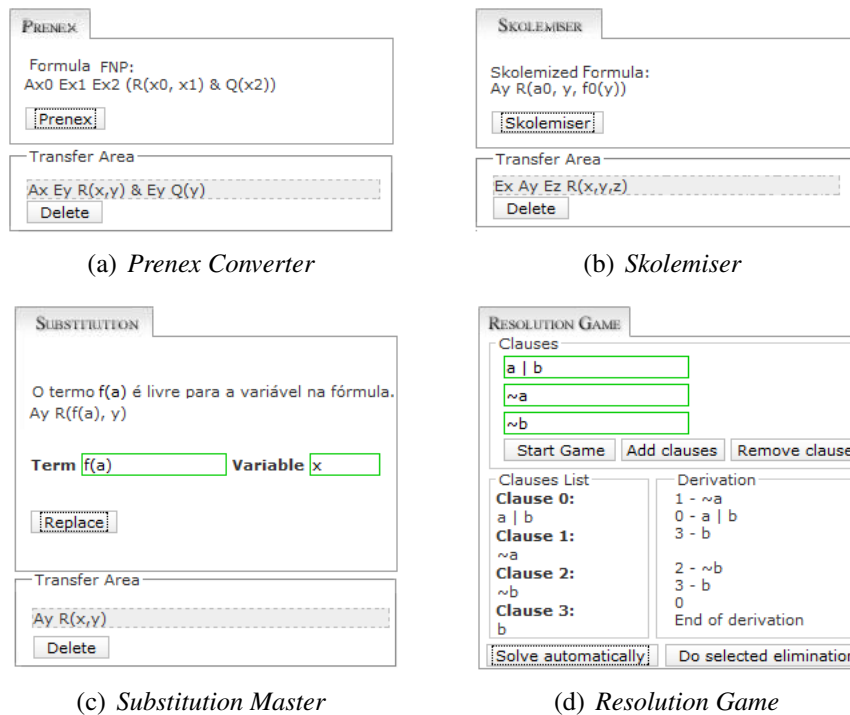


Figura 3. Protótipos de Módulos do LOGICAMENTE - 2

ágil de novos componentes. Com o objetivo de explorar ao máximo a interação do usuário com o sistema via *Web*, o LOGICAMENTE utiliza a tecnologia AJAX.

4. Trabalhos Futuros

*It is by logic that we prove,
but by intuition that we discover.*
— JULES HENRI POINCARÉ

In Science and Method (1908) translated by Francis Maitland (1914, 2007), 129.

Nosso objetivo imediato é incluir os seguintes OAs, já implementados, mas ainda não integrados ao LOGICAMENTE :

- *Small Counter-Model Builder*: implementa um buscador exaustivo de contra-modelos (*Bounded Model Checker*) finitos, recebendo como entrada um número de objetos, uma coleção de premissas e uma fórmula de conclusão.
- *WFF Diagnoser*: testa se uma expressão recebida é uma fórmula bem formada e sugere algumas correções. Verifica se há parênteses supérfluos na fórmula, além de informar várias propriedades das fórmulas bem formadas: seu grau de complexidade, seu conjunto de átomos básicos, de subfórmulas etc.
- *WFF Translator*: possibilita a criação de sintaxes para fórmulas e pode representar os conectivos em símbolos de diferentes notações, imagens, códigos em HTML, códigos em ASCII ou \LaTeX . Com esse módulo, é possível traduzir fórmulas de diferentes aplicativos.
- *Tableaux Constructor*: implementa o método de tableaux de lógica de primeira ordem, permitindo ao aluno visualizar a árvore de refutação e escolher que regra usar.
- *Semantic Consequence Tester*: aqui o aluno pode inserir conjuntos de fórmulas proposicionais e testar, passo a passo, a relação de consequência semântica envolvendo as

mesmas. O módulo gera um contra-modelo sempre que possível.

- *Theorem Proving Web System*: implementa um assistente de demonstração em Dedução Natural. Pretendemos desenvolver esse módulo integrando e adaptando ao nosso projeto uma ferramenta já existente, o *ProofWeb*¹⁰.

Após a integração desses OAs, o LOGICAMENTE garantirá ao aluno o retorno do andamento de sua aprendizagem ao executar suas lições designadas e seus exercícios relacionados com os OAs. Para as tarefas de gerenciamento de sessões de usuários, vamos adaptar o LOGICAMENTE à plataforma *Moodle*¹¹.

Com o objetivo de facilitar a reutilização de OAs, o LOGICAMENTE utilizará o padrão IMS Learning Design (LD) para ter seu conteúdo especificado e ter a garantia da durabilidade de seus recursos [IMS Global Learning Consortium 2003]. O IMS LD é compatível com a plataforma *Moodle* e é um *framework* utilizado para o planejamento de novos cursos, suporte de atividades, organização de ambientes de aprendizagem [de Souza Dutra and Tarouco 2006]. Na Figura 4, temos uma visão geral do LOGICAMENTE através da ferramenta de edição e reutilização de OAs, o *ReCourse*¹².

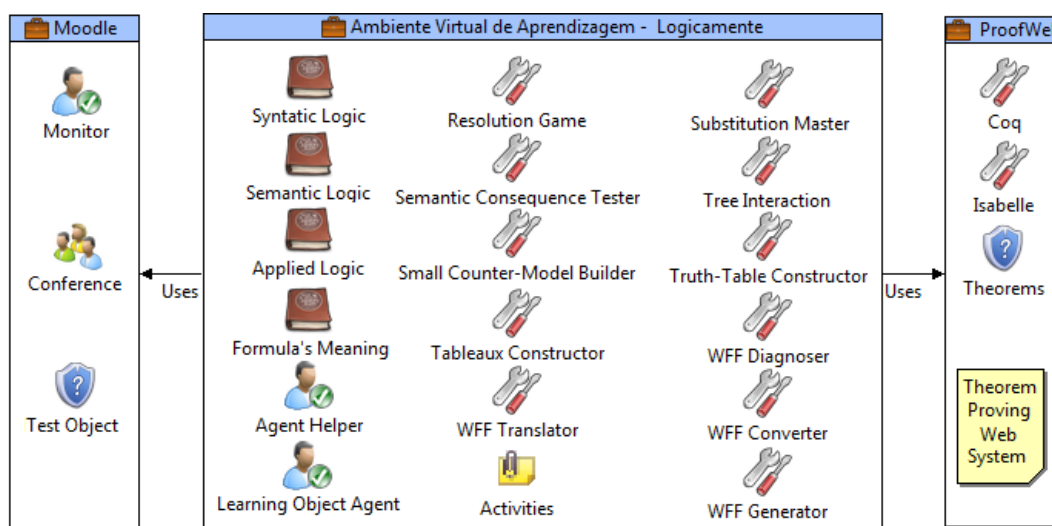


Figura 4. Diagrama de visão do AVA do LOGICAMENTE

5. Considerações finais

Atualmente, a área de ensino de Lógica possui uma grande carência por uma ferramenta computacional que explore outros estilos de aprendizagem e ensino. Ainda que existam ferramentas de ensino de Lógica, esses aplicativos não representam ambientes virtuais de aprendizagem realmente integrados e interativos. Para satisfazer esta necessidade, apresentamos o projeto LOGICAMENTE. Nós explicamos como o projeto foi inicialmente idealizado a partir do aprendizado colaborativo baseado em problemas de implementação,

¹⁰O *ProofWeb* é um software *Opensource* para o ensino de Dedução Natural que permite a interação entre alguns demonstradores de teoremas (Coq, Isabelle, Lego) e uma interface *Web*. Está disponível em <http://proofweb.cs.ru.nl/>.

¹¹O *Moodle* é um sistema de gerenciamento de aprendizagem para organizar conteúdos e atividades na educação. Uma das vantagens da plataforma é possibilitar a reutilização de OAs especificados.

¹²Disponível em <http://www.tencompetence.org/ldauthor/>.

mostramos qual seu estado atual e quais Objetos de Aprendizagem estão implementados e apresentamos nossas metas imediatas.

Os OAs desenvolvidos e apresentados visam a esclarecer as relações entre diferentes abordagens da Lógica, tais como a teoria semântica, a teoria da demonstração e as suas aplicações computacionais. Essas diferentes abordagens terão seus conteúdos associados a atividades e a desafios oferecidos no *Moodle*, objetivando um processo construtivo de ensino-aprendizagem de Lógica. O LOGICAMENTE busca explorar a potencialidade metodológica oferecida pela *Web* e, ainda, possibilitar a aplicação de diferentes estilos de ensino de Lógica, sem se restringir a estilos particulares de aprendizagem.

Referências

- Araújo, A. E. F. D., Barros, T. M., and Marcos, J. (2008). Logicamente: A implementação colaborativa de uma suíte de ferramentas on-line de apoio ao ensino de Lógica. In *EPOCA-2008*.
- Barland, I., Felleisen, M., Fisler, K., Vardi, M. Y., and Kolaitis, P. (2000). *Integrating Logic into the Computer Science Curriculum*. Helsinki.
- Barros, T. M., Araújo, A. E. F. D., and Marcos, J. (2008). A implementação colaborativa de uma suíte de ferramentas on-line de apoio ao ensino de Lógica. *VIII ERMAC-R3*.
- Barwise, J. and Etchemendy, J. (2002). *Language, Proof and Logic*. Center for the Study of Language and Information.
- de Souza Dutra, R. L. and Tarouco, L. M. R. (2006). Objetos de Aprendizagem: Uma comparação entre SCORM e IMS Learning Design. *Novas Tecnologias na Educação – CINTED-UFRGS*, 4(1):1–8.
- Department of Education. US (2009). Evaluation of evidence-based practices in online learning – a meta-analysis and review of online learning studies. Technical report, U.S. <http://www.ed.gov/rschstat/eval/tech/evidence-based-practices/finalreport.pdf>.
- Felder, R. M. and Silverman, L. K. (1988). Learning and Teaching Styles In Engineering Education. *Engr. Education*, 78(7):674–681. <http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/LS-1988.pdf>.
- IMS Global Learning Consortium, I. (2003). *IMS Learning Design Best Practice and Implementation Guide*, version 1.0 final specification edition. <http://www.imsglobal.org/learningdesign>.
- Tarouco, L. M. R., Fabre, M.-C. J. M., and Tamusiunas, F. R. (2003). Reusabilidade de objetos educacionais. *Novas Tecnologias na Educação – CINTED-UFRGS*, 1(1):1–11.
- van Dalen, D. (2004). *Logic and Structure*. Springer.
- Vilela, G., Rosan, M., and Marcos, J. (2008). Implementação de um gerador e verificador de modelos finitos para a Lógica Clássica de Primeira Ordem. *VIII ERMAC-R3*.
- Wiley, D. A. (2002). *The Instructional Use of Learning Objects*, chapter Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy, pages 3–23. AIT/AECT. <http://reusability.org/read/chapters/wiley.doc>.