



Programa Interdisciplinar de Pós-Graduação em
Computação Aplicada

Mestrado Acadêmico

Agnaldo Martins Rodrigues

SAAPIEnS: A autoria de Objetos de Aprendizagem e
Acompanhamento Pedagógico para o Ensino de Dedução
Natural na Lógica Proposicional

São Leopoldo, 2014

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO
EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

AGNALDO MARTINS RODRIGUES

SAAPIENS: A autoria de Objetos de Aprendizagem e Acompanhamento Pedagógico para o Ensino de Dedução Natural na Lógica Proposicional

São Leopoldo

2014

Aginaldo Martins Rodrigues

SAAPIENS: Autorial de Objetos de Aprendizagem e Acompanhamento Pedagógico para o Ensino de Dedução Natural na Lógica Proposicional

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre, pelo Programa Interdisciplinar de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos – UNISINOS.

Coordenador: Prof. Dr. Cristiano André da Costa

Orientador : Prof. Dr. João Carlos Gluz

São Leopoldo

2014

Aginaldo Martins Rodrigues

R696s

Rodrigues, Agnaldo Martins.

SAAPIENS : autoria de objetos de aprendizagem e acompanhamento pedagógico para o ensino de dedução natural na lógica proposicional / Agnaldo Martins Rodrigues. – 2014.
83 f. : il. ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Programa Interdisciplinar de Pós-Graduação em Computação Aplicada, 2014.

"Orientador: Prof. Dr. João Carlos Gluz"

1. Objetos de aprendizagem. 2. Sistema de Acompanhamento e Apoio Pedagógico. 3. Sistemas multi-agentes. 4. Sistema de autoria. I. Título.

CDU 004

Dados Internacionais de Catalogação na Publicação (CIP)
(Bibliotecário: Flávio Nunes – CRB 10/1298)

SAAPIENS: Autoria de Objetos de Aprendizagem e Acompanhamento Pedagógico para o Ensino de Dedução Natural na Lógica Proposicional

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre, pelo Programa Interdisciplinar de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos – UNISINOS

Aprovado em: ____/____/____.

BANCA EXAMINADORA

Prof. Dr. João Carlos Gluz – UNISINOS (orientador)

Prof^a.Dra. Rosa Maria Vicari – UFRGS

Prof.Dr. Sandro José Rigo – UNISINOS

Dedico esse trabalho à minha família e todos aqueles que participaram de alguma maneira para o alcance desse objetivo.

AGRADECIMENTOS

Agradeço primeiramente à Deus, que está sempre comigo em todos os momentos da minha vida, me dando força para seguir sempre em frente com muita fé e força.

Agradeço ao meu orientador Prof . João Carlos Gluz pela grande oportunidade dada à mim, pela sua compreensão, pela amizade demonstrada durante essa jornada e acima de tudo pelo apoio e orientação em todos os momentos quando foi solicitado. Obrigado por ter usufruído e compartilhado do seu conhecimento, tanto nos momentos de descontração quanto nos momentos de trabalho.

À minha esposa Rosele, pelo apoio, compreensão e companheirismo para que esse sonho pudesse se realizar.

Aos meus três filhos: Morgana, Ana Caroline e João Vitor que são a essência da minha vida. Obrigado pelo apoio, carinho e pela compreensão durante esse período em que estive ausente.

Aos meus sogros Gildo e Ieda que eu considero meus segundos pais, pela amizade, apoio, carinho e pelas orações também.

Ao meu amigo e colega de projeto, Matheus Campezzato pela grande parceria formada e pela dedicação na conclusão desse trabalho.

Aos demais colegas de projeto, pelas conversas e inúmeras trocas de experiências que muito contribuíram para que esse trabalho se concretizasse.

“Não se deve ir atrás de objetivos fáceis, é preciso buscar o que só pode ser alcançado por meio dos maiores esforços.”

(AlbertEinstein)

RESUMO

Com o uso crescentedas tecnologias de informação e comunicação na área educacional, torna-se cada vez mais necessária a criação de ferramentas de apoio ao professor, capazes de auxiliá-lo no planejamento e execução das aulas, bem como no acompanhamento do processo de ensino-aprendizagem. Em alguns casos, o acesso a essas tecnologias, aliado ao conhecimento técnico limitado por parte dos docentes, pode comprometer a maneira com que os conteúdos são desenvolvidos e utilizados pelos alunos. Essa é a principal área de pesquisa da presente dissertação, que se propõe a ajudar a tornar mais proficiente esse processo, pelo menos no contexto do ensino de dedução natural na disciplina de Lógica, através da criação de uma ferramenta inteligente de autoria e acompanhamento de uso de materiais educacionais, na forma de objetos de aprendizagem. A ferramenta, denominada SAAPiEnS, torna possível que o professor elabore objetos de aprendizagem customizáveis no domínio específico de dedução natural na lógica proposicional. Com base em um conjunto de ações, regradas por estratégias de ensino, almeja-se também instrumentalizar o docente no acompanhamento das atividades realizadas pelos alunos no ambiente computacional, apoiando o processo de avaliação do nível dedesempenho alcançado pelos alunos. Isso é feito por meio de um subsistema que oferece serviços de visualização e análise de dos conjuntos de interações dos alunos com os objetos de aprendizagem. A ferramenta é disponibilizada tanto via Web quanto via local (*desktop*), visando a uma maior flexibilidade para professor e alunos. Os possíveis impactos dessa ferramenta foram avaliados através de cenários hipotéticos de uso, que ilustram as principais contribuições possíveis para esse tipo de ferramenta.

Palavras-Chave: Objetos de Aprendizagem, Sistema de Autoria, Sistema de Acompanhamento e Apoio Pedagógico, Sistemas Multi-agentes.

ABSTRACT

With the increasing use of information and communication technologies in education, it becomes necessary to create supporting tools which can assist teachers in planning and implementing lessons as well as monitoring the process of teaching and learning. In some cases, access to these technologies, coupled with limited technical knowledge on the part of teachers, can compromise the way the contents are developed and used by students. This is the main research area of this dissertation, which aims to make this process more proficient, at least in the context of teaching natural deduction in the discipline of logic, through the creation of an intelligent authoring tool of learning objects, which also allows the teacher to follow the use of these educational materials by students. The tool, called SAAPIEEnS, makes it possible for the teacher to prepare customizable learning objects in the specific domain of natural deduction in propositional logic. Based on a set of actions, regulated by a set of teaching strategies, the goal is to help teachers to monitor the activities performed by students in the computing environment, supporting the process of assessing the level of performance achieved by the students. This is done by means of a subsystem, which offers visualization and analysis service of the sets of interactions of students with learning objects. The tool supports Web and local (desktop) interfaces, giving greater flexibility for teachers and students. The possible impacts of this tool were evaluated through hypothetical usage scenarios that illustrate the main possible contributions to this type of tool.

Key words: Learning Objects, Authoring System, Monitoring System and Educational Support, Multi-agent Systems

LISTA DE FIGURAS

Figura 1 - Organização Geral da Infraestrutura MILOS	2322
Figura 2 - Abordagens envolvidas em um sistema de autoria.....	2524
Figura 3 - Um agente e seu ambiente	2726
Figura 4 - Serviços de páginas amarelas	3029
Figura 5 - Modelo de Referencia para uma plataforma de Agentes.....	3130
Figura 6:Fases do desenvolvimento da metodologia TROPOS	3231
Figura 7 - Tela de edição <i>eXe Learning</i>	4039
Figura 8 - Tela de edição do Xerte	4140
Figura 9 - Tela de Edição do UDUTU	4241
Figura 10 - Tela de Edição do CourseLab.....	4341
Figura 11 - Tela de edição do LCDS-Microsoft.....	4342
Figura 12:Modelo de Dependência Estratégica de Requisitos Iniciais	4948
Figura 13:Modelo de Razão Estratégica dos requisitos iniciais.....	5049
Figura 14: Modelo de Razão Estratégica – Requisitos Finais.....	5251
Figura 15: Arquitetura do SAAPIEnS	5352
Figura 16: Tela de entrada do parâmetros do OAH.....	5554
Figura 17 Tela de entrada de atividades extras.....	5655
Figura 18: Tela do Editor para inserção de questão e sua validação.....	5756
Figura 19: Interface de consulta das questões.	5857
Figura 20: Tela de cadastro do Banco de Regras	5958
Figura 21 Acesso à ferramenta através do Tablet.....	5958
Figura 22 Tela de apresentação do OAH	6059
Figura 23 Tela de Atividades do OAH.....	6059
Figura 24 Integração SAAPIEnS - Heráclito na plataforma MILOS.....	6362
Figura 25 - Ciclo de criação e resolução o do OAH.....	6564
Figura 26 Cadastro dos argumentos do OAH na Base de Questão	6867
Figura 27 Inserindo Atividade Extra	6867
Figura 28 Tela de Parâmetros do OAH	6968
Figura 29 Seleção de questões do OAH	6968
Figura 30 Tela do OAH proposto	7069
Figura 31 OAH Gerado	7170
Figura 32 OAH sendo utilizado.....	7271
Figura 33 Seleção de Aluno - Acompanhamento.....	7372

Figura 34 Consulta Individual do aluno	7473
Figura 35 Consulta questão consolidada	7473
Figura 36 - Resumo da resolução do OAH.....	7574
Figura 37 Novo OAH gerado	7675

LISTA DE TABELAS

Tabela 1 Operadores e símbolos lógicos	34
Tabela 2 Regras de Inferências.....	36
Tabela 3 Regras Derivadas	37
Tabela 4 Demonstração Formal.....	37
Tabela 5 Relação das Ferramentas de Autorias.....	39
Tabela 6 Características das ferramentas de autoria.....	4544
Tabela 7 Análise comparativa das ferramentas de acompanhamento	47
Tabela 8 Log de Atividades do SAAPIEEnS	64

LISTA DE SIGLAS

ADL	Advanced Distributed Learning Initiative
API	Application Programming Interface
AUML	Agent Unified Modeling Language
CSELT	Centro Studi E Laboratori Telecomunicazioni
DCMI	Dublin Core Metadata Initiative
EaD	Ensino a Distância
FIPA	Foundation for Intelligent Physical Agents
FIPA ACL	FIPA – Agent Communication Language
FIPA-OS	FIPA – Operating System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IA	Inteligência Artificial
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IEEE-LOM	IEEE – Learning Object Metadata
IEEE-LTSC	IEEE – Learning Technology Standards Committee
IMS	Inovation Adoption Learning
ISO	International Services for Standartization
JADE	Java Agent DEvelopment framework
JAVA	Linguagem de Programação Orientada a Objeto
JSF	Java – Server Faces
JSR-94	Java Specification Request 94
LOM	Learning Object Metadata
MILOS	MultagentInfraestructure for Learning Object Support
OA	Objeto de Aprendizagem
OAH	Objeto de Aprendizagem Heráclito
OBAA	Objetos de Aprendizagem Suportados por Agentes
OWL	WEB Ontology Language
RDF	Resource Definition Format
SCORM	Sharable Content Object Reference Model
UML	Unified Modeling Language
URI	Uniform Resource Identifiers
W3C	World Wide Web Consortium
Web	System of Interlinked Hypertext Documents
XML	Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO	17
2 PROBLEMA DE PESQUISA	19
2.1 Motivação	19
2.2 Objetivos.....	20
2.3 Metodologia.....	20
3 FUNDAMENTAÇÃO.....	22
3.1 Projeto Heráclito.....	22
3.2 Arquitetura MILOS	23
3.3 Sistemas de autoria	24
3.4 Agentes e Sistemas Multiagentes	26
3.5 Ontologias.....	33
3.6 Dedução Natural na Lógica Proposicional	34
4 TRABALHOS RELACIONADOS	39
4.1 Ferramentas de autoria.....	40
4.2 Análise comparativa das ferramentas de autoria	44
4.3 Ferramentas de acompanhamento educacional	45
4.4 Análise comparativa das ferramentas de acompanhamento	47
5 SISTEMA SAAPIENS	48
5.1 Requisitos	48
5.2 Arquitetura e Aspectos da Implementação	53
6 METODOLOGIA DE AVALIAÇÃO.....	6665
6.1 Experimentos de Validação da Funcionalidade.....	6665
6.2 Avaliação de Aspectos de Usabilidade e do Potencial Educacional	6665
6.3 Análise das avaliações	7674
7 CONSIDERAÇÕES FINAIS	7875
7.1 Conclusão	7875
7.2 Contribuições.....	7976
7.3 Trabalhos futuros.....	7976

1 INTRODUÇÃO

As Tecnologias de Informação e Comunicação (TIC), têm contribuído de forma relevante nos processos de ensino e de aprendizagem¹ (SANTOS, 2012). Entretanto, é necessário que o domínio no uso dessas tecnologias esteja associado a práticas pedagógicas compatíveis com o público-alvo e com o ambiente para os qual elas são direcionadas. Os professores desempenham um papel fundamental nesse processo, pois são os responsáveis pelo planejamento, criação e acompanhamento das atividades desenvolvidas e propostas aos alunos, cabendo a eles o uso adequado dessas ferramentas e tecnologias atualmente disponíveis no cenário no mundo educacional.

O grande volume de informações disponibilizado e a facilidade de acesso, tanto por parte de alunos quanto por parte dos professores, desencadeia, nos processos educativos, uma nova forma nas relações e na maneira de trabalhar conteúdos específicos. A implementação de ferramentas inteligentes e amigáveis, que auxiliem o professor no processo de criação, adaptação e disponibilização de conteúdos aos alunos, assim como no acompanhamento das ações executadas e do seu desempenho, é o propósito do presente trabalho.

Um Objeto de Aprendizagem (OA) é, segundo a definição da IEEE (IEEE-LTSC, 2002), qualquer entidade digital ou não, que pode ser utilizada, reutilizada ou referenciada durante o aprendizado apoiado por computador. Segundo Wiley et al. (2000), um OA é definido como uma entidade, seja ela digital ou não, passível de ser usada, reutilizada ou referenciada durante a aprendizagem suportada com o uso das tecnologias. No entanto, um OA, por si só, não garante o sucesso da aprendizagem. Para que o êxito aconteça é necessário que haja planejamento, com definições de como o objeto será construído, qual seu objetivo, quais as ações propostas e previstas pelo professor, além das análises que serão realizadas no uso desses OAs.

Acompanhar o desempenho dos alunos, assim como prever o grau de sucesso ou fracasso alcançado no processo educativo em ambientes de aprendizagem tem sido tema de pesquisa nos últimos anos (ROMERO e VENTURA, 2005; ROMERO-ZALDIVAR, 2012). Um fator importante que contribui para a eficácia de uma experiência de aprendizagem é a possibilidade que os professores têm de mediar e acompanhar o processo de aprendizagem em geral e, potencialmente, agir com base nos eventos observados. O conjunto de informações geradas em um domínio específico pode e deve ajudar o professor a avaliar melhor as suas práticas e a contribuir com uma formação mais eficiente no processo de ensino.

Uma ferramenta de autoria, segundo a definição da W3C (2014), é qualquer software, ou coleção de componentes de software que os autores podem usar para criar ou modificar o conteúdo e disponibilizá-lo para uso por outras pessoas. Ferramentas de autoria são, então, ferramentas de criação. Devem ser, portanto, em sua essência, de fácil utilização e compreensão, de modo que o usuário, seja ele professor ou *conteudista* (designer de conteúdos), não precise ser nenhum especialista em programação ou perito em técnicas multimídias para desempenhar o seu papel. Quando se tratar de um domínio específico,

¹ . O termo ensino-aprendizagem é a tradução da palavra russa *obuchenie*, e significa algo como “processo de ensino-aprendizagem”, incluindo sempre aquele que aprende e aquele que ensina e a relação entre essas pessoas. Pela falta de um termo equivalente em inglês, a palavra *obuchenie* tem sido traduzida ora como ensino, ora como aprendizagem e assim retraduzida para o português.

é importante que o professor saiba o conteúdo a ser desenvolvido, a maneira como este conteúdo deverá ser apresentado e quais os resultados a serem alcançados.

O Projeto Heráclito tem como objetivo a resolução de OAs inteligentes para o ensino de Lógica Proposicional, voltado particularmente para o ensino do processo de Dedução Natural nesta Lógica (PENTEADO e GLUZ, 2011; PENTEADO, 2013; MOSMANN, et al., 2012). O Heráclito é um dos componentes da arquitetura OBAA-MILOS (VICARI e GLUZ, 2012), atuando como o protótipo do Sistema de Apoio Pedagógico.

A etapa inicial do Heráclito já foi concluída e visava ao apoio pedagógico aos alunos. É composto de um conjunto de agentes pedagógicos aptos a fornecer um serviço de tutoria em Dedução Natural na Lógica Proposicional, vinculado aos Objetos de Aprendizagem Heráclito (OAH). Resultados importantes foram alcançados nessa primeira etapa (PENTEADO, 2013), na medida em que se pode comprovar que o uso da ferramenta, combinado com as estratégias de ensino desenvolvidas no trabalho, se mostraram capazes de fornecer uma importante ajuda no processo de aprendizado na disciplina de Lógica, com um índice de 69% de aprovação dos alunos. Porém, falta completar o cenário de apoio ao processo de ensino, incluindo o suporte ao professor. Esse suporte se dará na forma de ferramentas de autoria dos OAs Heráclito (OAH), além de um sistema de acompanhamento do uso desses OAs pelos alunos.

A presente proposta está organizada da seguinte forma: o capítulo 2 apresenta a problema de pesquisa, a motivação, os objetivos e a metodologia a ser aplicada no trabalho. O capítulo 3 apresenta a fundamentação teórica, juntamente com as bases tecnológicas que foram utilizadas no desenvolvimento da ferramenta SAAPIEnS, que foi o alvo da pesquisa tecnológica da presente dissertação. No capítulo 4 é apresentado a análise de trabalhos correlatos. O capítulo 5 contextualiza a análise de requisitos, arquitetura da ferramenta. no capítulo 6 aborda o cenário de uso da aplicação. Por fim, no capítulo 7, expõem-se as considerações finais e as conclusões tiradas com este trabalho, com o desenvolvimento da pesquisa, e, também, evidencia a prospecção de trabalhos futuros.

2 PROBLEMA DE PESQUISA

Este capítulo é dedicado a apresentar uma visão sobre o problema que instigou esta pesquisa, os objetivos a serem alcançados, juntamente com a metodologia a ser utilizada na solução do problema.

2.1 Motivação

A falta de ferramentas, tanto para autoria quanto para acompanhamento de uso de conteúdos digitais, em particular para materiais em formato de OAs adequadamente projetadas para serem utilizadas pelo professor em um ambiente de ensino computadorizado, seja ele Web ou não, significa que a criação de material instrucional digital poderá exigir conhecimento de pessoal técnico, o que não é razoável. Evidentemente, esse aspecto gera uma barreira para a utilização desse tipo de material digital por parte do professor.

Uma pressuposição primordial do presente trabalho é que os professores podem se tornar autores de um conteúdo digital instrucional, sem a necessidade de habilidades de programação ou de treinamento técnico extensivo na área de Tecnologias da Informação e Comunicação (TICs). Também se espera que os professores possam ter acesso às informações sobre o uso desses objetos, tais como registradas nos bancos de dados dos ambientes de ensino, sem necessitar de amplos conhecimentos técnicos sobre, por exemplo, como são armazenadas ou recuperadas as informações em bancos de dados. No entanto, o desenvolvimento de ferramentas de autoria e de acompanhamento para uso extensivo em sistemas digitais de ensino não é uma tarefa trivial. Essa tarefa se torna bem mais complexa quando as ferramentas de autoria precisam suportar a criação de cursos adaptativos ou material que envolva um tema de maior complexidade ou mesmo ambientes heterogêneos.

Nesse contexto, é importante levar em consideração que o professor desempenha um papel fundamental dentro do cenário educacional, tanto como especialista em ensino, quanto como usuário de ambientes digitais, sendo, portanto, imprescindível provê-lo de estruturas que contribuam para a melhoria do seu trabalho.

Isso leva ao problema de pesquisa geral da presente dissertação que pode ser resumido no seguinte questionamento:

“A integração entre os processos de Autoria e de Acompanhamento no uso de Objetos de Aprendizagem pode contribuir como estratégia para a melhoria do processo de ensino de tópicos específicos de uma disciplina, com aplicação particular para a disciplina de Lógica em Nível Superior, com o tópico de Dedução Natural da Lógica de Predicados.”

Partido dessas considerações optou-se por iniciar a pesquisa sobre ferramentas a partir de uma infraestrutura existente para OAs voltados ao ensino de Lógica Proposicional, no caso o Sistema de Tutoria Heráclito, e construir um conjunto de ferramentas de apoio à autoria e de suporte ao acompanhamento no uso desses objetos. Trata-se de ferramentas que serão reunidas em uma única plataforma de apoio ao uso de OAs, no caso a infraestrutura MILOS (GLUZ e VICARI, 2010).

Assim sendo, esta dissertação se propôs a desenvolver o sistema SAAPiEnS (Sistema de Autoria e Acompanhamento Inteligente de Ensino Superior - Módulo de suporte aos OAs

Heráclito), que tem como objetivo fornecer suporte ao professor através de uma ferramenta de autoria que facilite na criação, manutenção e reutilização dos OAs e de uma ferramenta que possibilite o acompanhamento contínuo do uso desses OAs pelos alunos.

2.2 Objetivos

O sistema SAAPiEnS tem como objetivo auxiliar o professor na concepção e na customização de materiais de estudo voltados para o ensino de Lógica Proposicional, incorporados na forma dos OAH (Objetos de Aprendizagens Heráclito). Tais objetos serão utilizados pelo Sistema Tutor Heráclito e contribuirão no processo de ensino através do acompanhamento do uso desses objetos, no contexto de um plano pedagógico.

Para tanto, o SAAPiEnS irá prover uma ferramenta de autoria e customização do OAH integrada à plataforma da MILOS (GLUZ, 2010), como parte do Sistema de Autoria de OAs, e uma ferramenta de acompanhamento pedagógico, também integrada à MILOS, como parte do Sistema de Apoio Pedagógico, sendo este disponibilizado pela infraestrutura.

Dessa forma, os objetivos específicos da presente proposta de dissertação são os seguintes:

- Projetar um modelo de representação e análise das informações de acompanhamento das ações e atividades realizadas pelos alunos sobre os OAH;
- Projetar um modelo de documentos web a ser utilizado pelos alunos como interface aos OAH. Este modelo deve suportar conteúdos educacionais multimídia, ativos e com tutoria inteligente na resolução de exercícios de Lógica;
- Definir as funcionalidades necessárias para os processos de autoria de objetos OAH e para o acompanhamento das atividades realizadas sobre os OAH.
- Projetar uma arquitetura de sistemas multiagentes que implemente todas as funcionalidades dos processos de autoria e de acompanhamento;
- Desenvolver o protótipo do SAAPiEnS integrado a infraestrutura MILOS através da implementação de ferramentas web para a autoria e acompanhamento;
- Avaliar as capacidades das ferramentas e o possível impacto causado pelo uso dessas ferramentas em processos de ensino através de análises e testes em laboratório;

2.3 Metodologia

Visando organizar o desenvolvimento deste projeto de pesquisa e a concretização dos objetivos destacados na seção 2.2, foi concebida a metodologia de trabalho delineada a seguir.

O projeto dos componentes da ferramenta de autoria e de acompanhamento, incluindo seus agentes e ontologias, foi dividido em vários subsistemas que foram projetados e desenvolvidos para atender especificamente às funcionalidades atuais do Sistema de Tutoria Heráclito. A compatibilidade entre os sistemas e os componentes é garantida pelo uso das mesmas ontologias bases (a ontologia de meta-dados do OBAA) e pelo uso das mesmas facilidades de comunicação (protocolos, mensagens e serviços definidos pelo padrão FIPA (seção 3.4.3)). Esse trabalho segue as diretrizes técnicas e metodológicas adotadas na

Infraestrutura da MILOS (seção 3.1), que prevê ainda o uso de agentes inteligentes e facilidades de interface para o usuário.

Em termos pedagógicos, a metodologia adotada na didática dos OAHs criada para o sistema de suporte aos objetos OBAA foi baseada no método dialético de ensino, embasado na pedagogia sócio-histórica, aplicado ao ensino de Lógica. Esses conceitos estão referenciados nos trabalhos publicados nos SBIE do ano de 2007 (GLUZ; PASSERINO; VICARI, 2007) e 2008 (GLUZ; PASSERINO; VICARI, 2010).

O protótipo do SAAPiEnS foi projetado e desenvolvido através de ciclos evolutivos de prototipagem. O protótipo inicial teve como finalidade especificar as interfaces das ferramentas deste sistema, buscando atender aos principais requisitos de usabilidade (NIELSEN e NORMAN 2012) para sistemas de autoria de materiais educacionais e de acompanhamento pedagógico. Assim como, integrar os demais agentes do sistema (incluindo *wizards*) com a interface principal e com os meios de obter as informações necessárias para criação e acompanhamento dos OAH.

Posteriormente, foram desenvolvidos os protótipos dos agentes que implementam as funcionalidades dessas ferramentas, além dos agentes de gerenciamento do sistema e da especificação da comunicação entre os mesmos. A semântica das mensagens será baseada em ontologias educacionais previamente desenvolvidas para a MILOS.

Na terceira fase de prototipação, foi realizada a modelagem e implantação da base de conhecimentos ontológica que proverá a camada de persistência do sistema, armazenando, além dos OAHs, as informações dos alunos e professores.

Os protótipos foram desenvolvidos na linguagem Java com o uso do *framework* JSF (Java Server Faces) para construção de uma interface de usuário baseada em componente para aplicações web: a plataforma JADE/FIPA como ambiente multiagente. Essas linguagens e plataformas foram escolhidas devido a sua versatilidade, eficiência, segurança e portabilidade.

Os protótipos iniciais das Interfaces da aplicação foram validados em laboratório por dois professores da área de Lógica, no intuito de verificar a adequação às necessidades e a conformidade com as análises de requisitos demandadas.

Como resultado de validação, foram realizados testes em laboratório, com a criação de um OAH e a disponibilização para resolução para um grupo de alunos. Foram monitorados os resultados da utilização dos OAH através dos Logs de uso do Sistema Tutor Heráclito, que serviram de base para as análises de acompanhamento e monitoramento.

As análises e o controle das interações são gerenciados pelos agentes da plataforma da aplicação, que possuem a tarefa de receber, analisar e processar as requisições que são persistidas na base de dados do servidor principal do SAAPiEnS.

3 FUNDAMENTAÇÃO

Este capítulo tem como objetivo apresentar o referencial teórico e técnico utilizado no trabalho, assim como as bases tecnológicas que foram necessárias para o projeto e desenvolvimento do protótipo do sistema SAAPIEnS.

3.1 Projeto Heráclito

O projeto Heráclito (PENTEADO e GLUZ, 2011; PENTEADO, 2012, 2013) tem, já na sua concepção, o propósito de contribuir para evitar o quadro de desistências e reprovações na disciplina Lógica, dos cursos de graduação Ciência da Computação e Sistemas de Informação. O projeto tem como foco o suporte à relação professor-aluno no contexto do componente curricular de Dedução Natural na Lógica Proposicional, pelo fator crítico que ele apresenta no processo de ensino de Lógica.

Os estudos realizados por Penteado (2011) comprovaram esse fato, visto que os altos índices de reprovação e desistência na disciplina estavam diretamente relacionados com as dificuldades apresentadas pelos alunos nesse conteúdo especificamente. Esses estudos resultaram no projeto Heráclito, como uma ferramenta que integra, por exemplo, as seguintes tecnologias: Sistema Tutores Inteligentes, Sistemas Multiagentes e Objetos de Aprendizagem. O foco da ferramenta é o auxílio aos alunos no ensino de resolução de problemas da lógica proposicional, através da aplicação das regras de dedução natural. Os objetos de aprendizagem utilizados no Heráclito (OAH) incorporam características, componentes tecnológicos e conceitos pedagógicos importantes, os quais contribuíram de forma significativa para que os alunos conseguissem atingir os resultados esperados, apropriando-se, então, do conhecimento desejado. Baseado na abordagem da pedagogia sócio-histórica e no processo dialético de ensino, o Heráclito, dentro do contexto do ensino de lógica, conseguiu transpor esses conceitos através da adaptação dos conceitos na Negociação Pedagógica (FLORES et al., 2005), aplicada com sucesso em outros ambientes de ensino inteligentes. A arquitetura cliente-servidor projetada no Sistema Heráclito é integrada com a infraestrutura da MILOS (GLUZ e VICARI, 2010) e é compatível com o padrão dos metadados do OBAA (VICARI et al., 2009, 2010), tendo sido, também, desenvolvida uma versão móvel do aplicativo para *Tablets* (MOSMANN et al., 2012).

O sistema Heráclito foi aplicado em um processo de ensino real em sala de aula, tendo sido obtidos bons resultados nesses experimentos empíricos (PENTEADO, 2012). O Heráclito tem diferenciais importantes que o destacam de outros sistemas similares (PENTEADO, 2011, 2012), particularmente pelo fato do sistema ter sido concebido e desenvolvido com o conceito de agentes pedagógicos inteligentes.

O Agente Mediador é o responsável por abordar de forma dialética e sócio-histórica a mediação que ocorre nos processos de ensino-aprendizagem. Ele desempenha o papel do professor ou tutor dentro do Sistema Heráclito, oferecendo um apoio pedagógico ao aluno em seu processo de aprendizagem da Dedução Natural na Lógica Proposicional. Sua atuação pode-se dar em várias situações da aprendizagem, através de intervenções de auxílio. Em caso do aluno cometer um erro ou repetir tal erro, o agente Mediador seleciona estratégias pedagógicas apropriadas para auxiliá-lo na busca da solução.

O Agente Especialista de domínio de ensino do Heráclito também apresenta um fator inovador, pois ele pode não apenas gerar automaticamente provas de dedução natural, similares às desenvolvidas por seres humanos, mas também pode finalizar provas parciais, ou seja, completar as provas que estão sendo desenvolvidas pelo aluno. Assim, cada passo feito pelo aluno pode ser avaliado em termos de sua viabilidade em alcançar o objetivo de finalização de uma prova, pois o agente também é capaz de identificar redundância de passos, o que prejudicaria a resolução do problema. Isso traz ao Heráclito um grande diferencial, permitindo a utilização de estratégias e táticas pedagógicas que auxiliam o aluno a direcionar seus esforços na superação das dificuldades de identificação e organização dos passos necessários para demonstrar um determinado argumento.

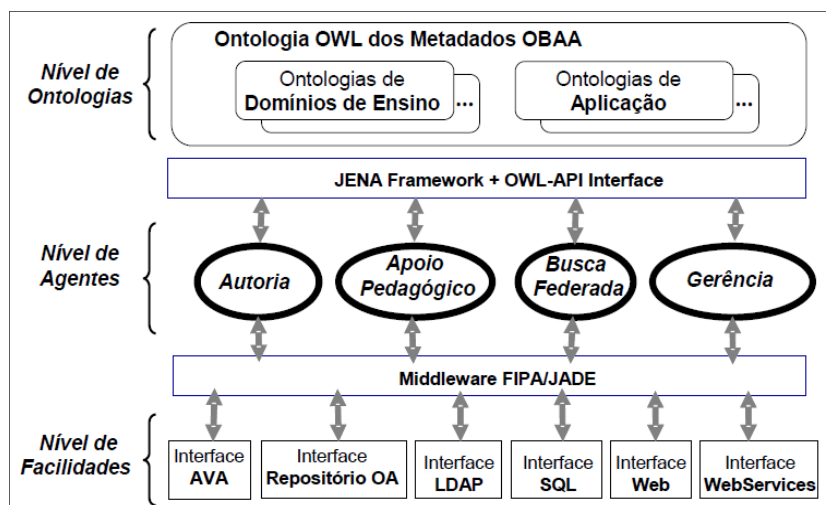
Todo o processo de melhoria é contínuo e requer muito estudo esforço de pesquisa. Os objetivos mais gerais da pesquisa envolvida no projeto Heráclito, agora direcionados ao apoio ao professor, terão sua continuidade realizada através do presente trabalho de pesquisa, não apenas pela complementação dos objetivos iniciais, mas também pela adição de novos recursos e tecnologias.

3.2 Arquitetura MILOS

A infraestrutura MILOS (*Multiagent Infrastructure for Learning Object Support*) é composta por um conjunto de tecnologias que tem como objetivo principal dar suporte e efetividade na adoção da proposta dos metadados definidos pelo padrão OBAA (GLUZ e VICARI, 2010) no uso de objetos de aprendizagem.

A arquitetura de software da infraestrutura MILOS é baseada no uso de agentes de software e de ontologias, que visa suportar os requisitos de adaptabilidade, interoperabilidade e acessibilidade previstos pelo padrão OBAA. A arquitetura da MILOS está dividida em três grandes níveis de abstração (ver Figura 1):

Figura 1 - Organização Geral da Infraestrutura MILOS



Fonte: VICARI e GLUZ (2011).

O primeiro nível, de ontologias, tem como papel principal representar os conhecimentos necessários a serem compartilhados pelos agentes, especificando as propriedades e ações do domínio da aplicação. Todas as ontologias da MILOS são construídas usando a ontologia OBAA como vocabulário de termos básicos (VICARI e GLUZ, 2011). As ontologias definidas nesse nível são obrigatoriamente definidas em OWL (W3C, 2013).

As ontologias desempenham um papel estático na arquitetura, uma vez que as ontologias não oferecem nem implementam os elementos ativos do sistema.

Assim, o segundo nível da MILOS, formado por agentes de *software*, desempenha o papel ativo na arquitetura. É nessa camada que são implementadas todas as funcionalidades e serviços disponibilizados pela infraestrutura. Esses agentes fazem uso das ontologias para dar o suporte necessário e atender aos requisitos de acordo com os subsistemas definidos pela plataforma. Os sistemas definidos para essa camada são:

- Sistema de Busca: implementa e dá o suporte às atividades de localização dos OA, em repositórios e federações de catálogos de OA;
- Sistema de Apoio Pedagógico: implementa e dá o suporte às atividades de uso dos OA;
- Sistema de Autoria: implementa e dá o suporte às atividades de autoria de um OA;
- Sistema de Gerência: dá suporte às atividades de armazenamento, gerenciamento, publicação/distribuição multiplataforma de OA.

No terceiro nível, estão definidos todos os serviços que dão suporte à interoperabilidade da infraestrutura, através de uma plataforma de comunicação entre os agentes, chamada FIPA/JADE (BELLIFEMINE et al., 2008). A camada de serviços oferece facilidades de interface para a Web permitindo que os agentes e sistemas multiagentes da MILOS possam ser acessados por ambientes e aplicações educacionais, além de permitir também que esses agentes tenham acesso aos repositórios de OAs, serviços de diretórios, plataformas de EAD, bancos de dados e servidores Web (VICARI e GLUZ, 2011).

3.3 Sistemas de autoria

Um sistema de autoria é um conjunto de ferramentas e aplicações que permitem aos autores de recursos digitais de aprendizagem integrarem diferentes componentes para criação de conteúdos interativos e multimídia (W3C, 2014). Entre as ferramentas existentes, podemos citar alguns exemplos tais como:

- Editores HTML WYSIWYG (what you see it's what you get);
- Ambiente de desenvolvimento Integrado (IDES e SDKs);
- Editores de Wikis e Blogs;
- Gerenciadores de Conteúdos (CMS);
- Sistemas Gerenciadores de Aprendizagem (LMS);
- Ferramentas de autoria multimídia;

- Ferramentas para desenvolvimento WEB e Mobile;
- Gerenciadores de Websites.

O conceito de ferramentas de criação de conteúdos é um termo genérico que se refere a um software para produção de conteúdo digital. O termo “Authoring Tools” é comumente referenciado como um software para autores de conteúdo ou sistemas de autoria. Essa dualidade terminológica refere-se a duas abordagens: a primeira, que destaca o lado tecnológico, e a outra, os aspectos pedagógicos. Na figura 2, são apresentadas as características que envolvem cada um dos aspectos mencionados.

Figura 2 - Abordagens envolvidas em um sistema de autoria



Fonte: Elaborado pelo autor.

Quanto aos aspectos tecnológicos, são características que estão diretamente relacionadas às principais funcionalidades das ferramentas de autoria:

- Usabilidade: facilidade no uso da ferramenta, interface amigável e de fácil adaptação por parte do autor;
- Recursos: características técnicas e componentes funcionais que auxiliam na criação, sendo fundamentais para o processo de autoria;
- Aprendizado: tempo e recurso despendido por parte do autor para o domínio da ferramenta;
- Ambiente e Infraestrutura: o conjunto de elementos tecnológicos que possibilitam a execução das atividades e o tráfego das informações;
- Disponibilidade: característica de acesso facilitado por parte do autor;
- Custos: quanto ao tempo despendido e os recursos necessários no processo de criação;
- Padrões Internacionais: adota padrões que proporcionam maior integração.

Nos aspectos pedagógicos, os fatores que são importantes referem-se à fase da concepção do OA e consistem nos seguintes:

- Planejamento: é a fase na qual o autor dedica-se à elaboração conceitual do OA e da sua aplicabilidade, definindo-se os objetivos finais;
- Concepção: envolve a criação, edição e customização do OA propriamente dito;
- Estratégias Pedagógicas: relativo ao plano pedagógico de utilização dos recursos didáticos. Tais recursos devem estar articulados coordenadamente, tendo cada recurso (material instrucional) uma função específica no plano, mas formando um conjunto único (OA) a partir de uma orientação pedagógica e de uma concepção de ensino, visando ao desenvolvimento das competências necessárias ao aluno;
- Acompanhamento: refere-se ao controle e monitoria das atividades executadas, assim como às análises de resultados, obtidas pelo aluno durante o processo de interação com o OA.

Independentemente dos aspectos e das abordagens existentes, o objetivo principal dessas ferramentas é facilitar a produção de materiais educacionais, proporcionando aos autores autonomia para criar seus próprios conteúdos. Embora exista uma diversidade tecnológica de recursos à disposição, o nível de conhecimento exigido por parte do autor ainda é alto e trabalhoso. O processo de planejamento e construção de materiais didático-pedagógicos em formato digital envolve uma série de requisitos essenciais que, muitas vezes, acabam inviabilizando o seu desenvolvimento por parte do autor, fazendo com que muitos projetos acabem sendo descontinuados ou deixados de serem usados. Em algumas situações, a solução é o desenvolvimento e a utilização de uma ferramenta customizada, que atenda a uma determinada demanda em um domínio específico, como é o caso do SAAPIEnS, objeto desse trabalho.

3.4 Agentes e Sistemas Multiagentes

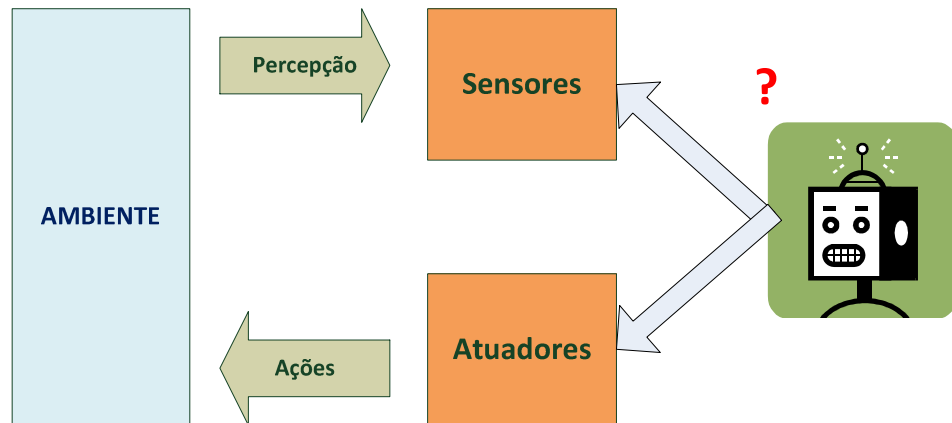
As pesquisas que envolvem agentes inteligentes de software têm aumentando significativamente nos últimos anos. Um dos motivos desse crescimento é o fato de que os agentes são objeto de estudo de diversas áreas além da computação, tais como psicologia, filosofia e sociologia. Porém, o campo que mais se utiliza desses conceitos é a Inteligência Artificial Distribuída, que é um subcampo da Inteligência Artificial (WEISS, 1999).

A Inteligência Artificial Distribuída (IAD) é um campo de estudo e um projeto de sistemas em que vários agentes interagem e estão logicamente distribuídos. Os agentes podem ser vistos como entidades dentro de um sistema inteligente distribuído agindo de forma autônoma, podendo cooperar entre si através de algum mecanismo de comunicação.

Existem várias definições de agentes, porém a definição empregada neste trabalho é a proposta por Wooldridge (2002), que afirma que “um agente é um sistema computacional capaz de realizar ações autonomamente para atender seus objetivos de projeto, ao invés de obedecer às regras sobre o que fazer em qualquer momento”. Outra definição que se tem a seguinte: “um agente é tudo que pode perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores” (RUSSEL e NORVIG, 2004).

A definição de autonomia está diretamente relacionada com a capacidade que o agente tem de agir em um ambiente computacional, com base em conjunto de percepções que o mesmo tem através de sensores e tomar ações apropriadas (ver figura 6).

Figura 3 - Um agente e seu ambiente



Fonte: Adaptado de Wooldridge (2002 p.16).

Segundo Wooldridge (2002), um sistema multiagente é aquele que possui dois ou mais agentes que interagem uns com os outros, geralmente por meio de troca de mensagens, através de algumas infraestruturas de redes de computadores. Para esse autor, um agente inteligente deve ser reativo, proativo e social. Para que essa interação ocorra com sucesso, os agentes devem ter a capacidade de cooperar, coordenar e negociar uns com os outros. Segundo Castro, Alencar e Silva (2005), a interação de agentes pode ser classificada como:

- Comunicação: troca de mensagens entre os agentes;
- Cooperação: habilidade de trabalhar em conjunto para um fim comum;
- Competição: aptidão para interagir com outros agentes onde o sucesso de um agente implica na falha de outros (o oposto de cooperação);
- Coordenação: execução de atividades em um ambiente com outros agentes compartilhando tarefas em comum, evitando conflitos e gerenciando recursos;
- Negociação: habilidade de interagir com outros agentes visando a alcançar um acordo entre as partes envolvidas sobre determinado problema.

O paradigma de Sistemas Multiagentes vem sendo aplicado já há algum tempo na solução em várias áreas, resolvendo problemas reais em uma escala grande de aplicações. Entre elas, podemos citar: aplicações comerciais, industriais, computação, entretenimento, educação, médica, robótica, sistema de tutores e outras mais.

Na educação, os Sistemas Multiagentes tornam-se um diferencial importante em relação às demais aplicações, pois é possível, assim, tirar o máximo proveito dos recursos advindos dessa tecnologia. A capacidade de um agente interagir no ambiente, trocar informações, propor ações educativas e realizar o acompanhamento e as mediações em situações específicas contribuem de maneira significativa nos processos educacionais. Segundo Gluz (2010):

“Os princípios dos Sistemas Multiagentes têm mostrado um potencial bastante adequado ao desenvolvimento de ferramentas computacionais de ensino, devido ao fato de a natureza do problema de ensino e aprendizagem ser mais facilmente resolvido de forma cooperativa. Ambientes de ensino/aprendizagem, baseados em arquiteturas multiagente, possibilitam suportar o desenvolvimento de sistemas mais interativos e adaptáveis, de forma mais robusta, mais rápida e com menores custos, tornando-os mais atrativos e mais eficazes do ponto de vista pedagógico.”

3.4.1 Agentes BDI

Segundo Bordini e Vieira (2003), as mais importantes arquiteturas de agentes deliberativos são baseadas em um modelo de cognição fundamentado em três principais atitudes mentais, que são as crenças, os desejos e as intenções (abreviadas por BDI, do inglês Beliefs, Desires e Intentions, respectivamente).

Crenças (*Beliefs*) representam o conhecimento sobre o mundo no qual o agente está inserido. Do ponto de vista computacional, crenças são apenas uma maneira de representar o estado do mundo, seja através de variáveis, uma base de dados relacional, ou expressões simbólicas em um cálculo de predicados.

Desejos (*Desires*) são relacionados aos estados de mundo que o agente quer provocar. Os desejos não dirigem necessariamente o agente a agir. Antes de um determinado agente decidir o que fazer, ele passa por um processo deliberativo, confrontando os seus desejos com as suas crenças e escolhendo os desejos que são possíveis, de acordo com um critério. Os desejos resultantes desse processo passam a ser intenções do agente.

Intenções (*Intentions*) correspondem a estados de mundo que o agente quer atingir. Intenções podem ser consideradas um subconjunto dos desejos, mas ao contrário destes, devem ser consistentes. Se um agente decide seguir uma meta específica, então, essa meta torna-se uma intenção e, para isso, ele necessita executar um raciocínio prático, buscando identificar quais ações devem ser executadas para satisfazer a intenção. Esse processo está diretamente associado a um planejamento que tem por objetivo realizar as intenções do agente.

Uma das abordagens mais comuns para o problema do planejamento consiste na criação de sistemas de planejamento reativo, formado por uma biblioteca de planos que contém conjuntos de ações aplicáveis a determinadas situações. A linguagem de programação AgentSpeak(L) foi projetada para a programação de agentes BDI na forma de sistemas de planejamento reativos. Ela foi primeiramente apresentada em (RAO 1995) e é uma extensão natural de programação em lógica para a arquitetura de agentes BDI, que representa um modelo abstrato para a programação de agentes e tem sido a abordagem predominante na implementação de agentes inteligentes (WOOLDRIDGE 2002). Um agente AgentSpeak(L) corresponde à especificação de um conjunto de crenças que formarão a base de crenças inicial e um conjunto de planos (BORDINI 2004). JASON (BORDINI, 2007) é uma plataforma de desenvolvimento de sistemas multiagentes baseada em um interpretador para uma versão estendida da linguagem AgentSpeak(L) e também oferece uma série de extensões que são necessárias para o desenvolvimento de tais sistemas.

3.4.2 Linguagem FIPA-ACL

A FIPA (*Foundation for IntelligentPhysicalAgents*) surge em 1996, com a finalidade de promover a interoperabilidade entre agentes heterogêneos (FIPA, 2013). Ela propõe uma arquitetura básica para agentes inteligentes e uma linguagem de comunicação de agentes chamada FIPA-ACL. Possui uma sintaxe muito semelhante ao KQML, porém o conjunto de performativas é diferente. De acordo com o documento da FIPA-ACL, cada mensagem possui uma série de parâmetros, nos quais o agente deve informar para quem a mensagem possa ser enviada e recebida corretamente pelo destino. A mensagem deve conter obrigatoriamente os campos *send*(agente emissor), *receiver*(agente receptor) e *performative* (ato performativo) preenchidos (FIPA-ACL MessageStructureSpecification - SC00061, 2002).

Os atos performativos são responsáveis pela identificação do tipo de ação que está associada à mensagem. Logo abaixo são descritos alguns dos atos performativos reconhecidos pelo padrão FIPA-ACL que estão especificados no documento FIPA-ACL CommunicativeAct Library Specification – SC00037.

- **Accept-proposal:** Informa a aceitação de uma proposta prévia para execução de uma determinada ação.
- **Agree:** Informa a concordância em executar alguma ação.
- **Cancel:** Informa a um determinado agente que não é mais preciso executar a ação que esta informada no conteúdo da mensagem anterior;
- **Cfp(call for propose):** Solicita proposta para a execução de uma determinada ação. A mensagem deve conter qual a ação que deve ser realizada;
- **Confirm:** O emissor informa ao receptor que uma dada proposição é verdadeira;
- **Disconfirm:** O emissor informa ao receptor que uma dada proposição é falsa;
- **Failure:** O emissor informa ao receptor que tentou fazer uma ação e que essa tentativa falhou. O conteúdo da mensagem é composto da ação que falhou e a razão da falha;
- **Not-understood:** O emissor avisa ao receptor que não entendeu uma ação ou ato prévio. O conteúdo da mensagem deve ser composto do ato ou da ação que não foi entendida e de uma explicação que do que não foi compreendido;
- **Propose:** Serve para que um agente emissor envie uma proposta para efetuar alguma ação, dadas certas pré-condições para o agente receptor. A descrição da ação proposta e a pré-condição devem estar presentes no conteúdo da mensagem;
- **Request:** O emissor solicita ao receptor que ele execute alguma ação. O conteúdo é composto com a ação a ser feita e da preposição;
- **Reject-proposal:** Indica a rejeição de uma proposta de execução de uma ação enviada previamente. O conteúdo da mensagem deve ser composto com a ação rejeitada e da explicação do porque da rejeição.

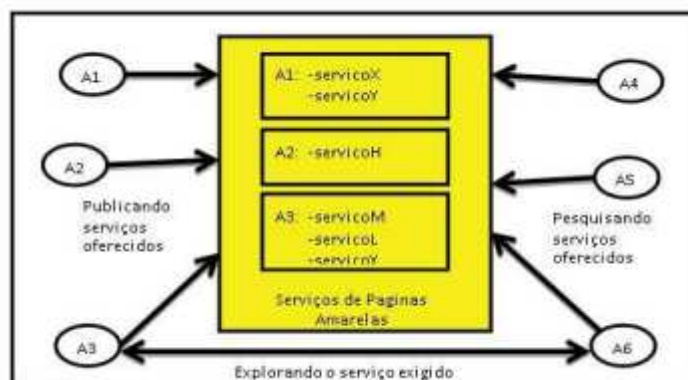
3.4.3 Plataforma JADE

A plataforma JADE (*Java Agent Development Framework*) é um ambiente de desenvolvimento de aplicações baseada em agentes, seguindo as especificações da FIPA, implementada totalmente em Java e é um software aberto, distribuído sob licença Lesser General Public License (LGPL). Ela foi desenvolvida e continua sendo atualizada pela Universidade Parma, na Itália (JADE, 2013). A plataforma JADE possui um abrangente conjunto de agentes de serviços de sistema, que facilitam e possibilitam a comunicação entre agentes, de acordo com as especificações da FIPA: serviço de nomes (*servicename*), páginas amarelas (*yellow-pageservice*), transporte de mensagens, serviço de codificação e decodificação de mensagens e uma biblioteca de interação pronta para ser usada. A comunicação entre os agentes só é possível através de troca de mensagens.

A JADE possui um ambiente de execução baseado no conceito de contêineres, que são instâncias da JADE rodando em uma *Java Virtual Machine* (JVM). O conjunto de contêineres ativos é chamado de plataforma. Numa plataforma JADE, há sempre um único contêiner principal (*main-container*) responsável por centralizar certos serviços da plataforma. Esse contêiner abriga o Agent Management System (AMS) e o *Directory Facilitador* (DF), os demais contêineres se conectam ao contêiner principal, formando um ambiente de execução completa para rodar sistemas multiagentes. De acordo com a documentação da JADE, os dois agentes especiais presentes na plataforma são (BELLIFEMINE et. al., 2010):

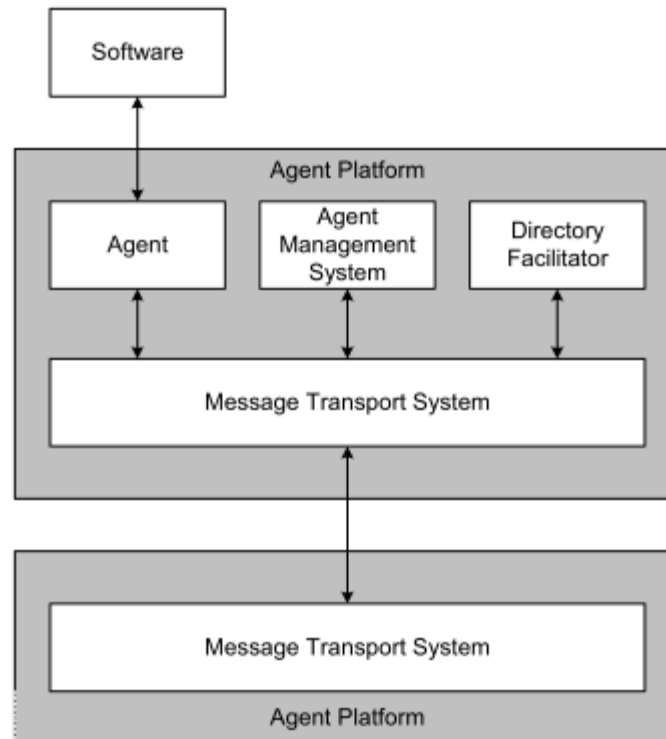
- Agente MAS: conhecido também como *white-page* ou páginas brancas; é o que realiza o controle de supervisão sobre acesso e uso da plataforma. Ele mantém um diretório de identificadores de agentes (AID), que fornece os dados sobre a localização do agente tais como o contêiner onde está situado;
- AgenteDF: conhecido também como *yellow-page* ou páginas amarelas; ele permite localizar agentes através dos serviços fornecidos por ele. Na Figura 5 podemos visualizar o serviço de páginas amarelas.

Figura 4 - Serviços de páginas amarelas



Fonte:(JADE,2013).

Figura 5 - Modelo de Referência para uma plataforma de Agentes



Fonte: Especificação Fipa SC0023J, 2002.

3.4.4 Metodologia TROPOS

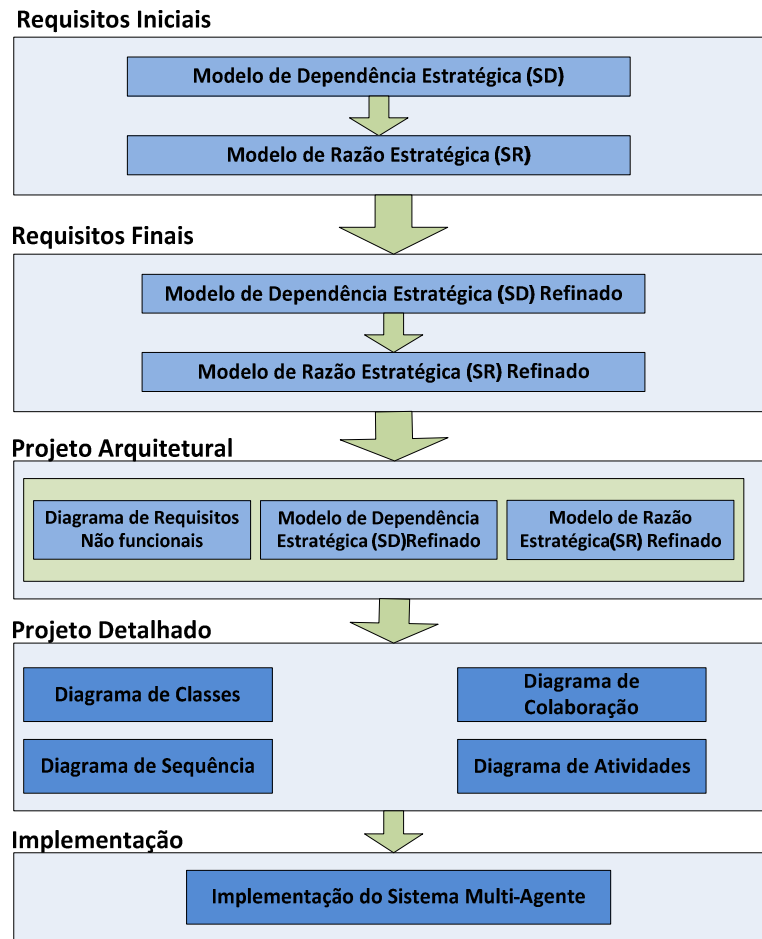
A metodologia TROPOS é uma proposta de modelagem para o desenvolvimento de sistemas orientados a agentes, baseada na análise de requisitos e fundamentada em conceitos sociais e intencionais, que visa reduzir tanto quanto possível esta incompatibilidade entre o sistema e seu ambiente (CASTRO, KOLP e MYLOPOULOS, 2002)

A metodologia TROPOS é composta por cinco fases de desenvolvimento de software: Requisitos Iniciais, Requisitos Finais, Projeto Arquitetural, Projeto Detalhado e Implementação. A idéia principal da metodologia é construir um modelo do sistema que será refinado e estendido por um processo passo a passo do nível conceitual ao executável.

Essa metodologia adota o *framework* de modelagem organizacional *i** (leia-sei *Star*) (YU, 1997) nas fases iniciais de modelagem, mantendo, porém, os conceitos intencionais derivados desse *framework* durante todas as fases de desenvolvimento (CASTRO, KOLP, MYLOPOULOS, 2002).

O processo inicia com um modelo do ambiente no qual o sistema que está sendo desenvolvido irá operar. O modelo é descrito em termos de atores, objetivos e interdependências. Através de refinamentos, este modelo é estendido para incluir tanto o sistema a ser desenvolvido quanto aos seus subsistemas, que também são representados como atores aos quais foram delegados e os objetivos a serem atingidos. Também há a definição de planos a serem executados e recursos a serem fornecidos pelos atores para o cumprimento das suas metas. Na Figura 6 é mostrada as fases que compõe o desenvolvimento da TROPOS.

Figura 6: Fases do desenvolvimento da metodologia TROPOS



Fonte: Adaptado de(SILVA, 2005)

A utilização desse *framework* possibilita a compreensão das razões internas dos atores, uma vez que as mesmas são expressas explicitamente, auxiliando na escolha de alternativas durante a etapa de modelagem do software.

O *framework* é dividido em dois modelos:

- Modelo de Dependência Estratégica(SD): fornece uma descrição intencional de um processo em termos de uma rede de relacionamentos de dependência entre atores relevantes;
- Modelode Razão Estratégica(SR): apresenta uma descrição estratégica do processo, em termos de elementos do processo e das razões que estão por detrás deles, ou seja, fornece uma análise meios fins de como as metas podem ser cumpridas através das contribuições dos demais atores.

3.5 Ontologias

Segundo Gruber (2013), uma ontologia é uma especificação explícita e formal de uma conceitualização compartilhada. Já para Guarino (1998), uma ontologia é “uma maneira de se conceitualizar de forma explícita e formal os conceitos e restrições relacionados a um domínio de interesse”. Portanto, uma ontologia visa a representar os principais conceitos de um determinado domínio, de forma direta e objetiva. Ou seja, ela deve ser formal o suficiente para ser processável por máquinas e também poder ser compartilhada por pessoas e/ou aplicações.

Considerando o contexto no qual ela está inserida, no caso do objeto proposto, definimos uma ontologia de domínio para aplicação na área educação, como representação de um conhecimento específico.

Uma ontologia, além da hierarquia de conceitos que ela deve expressar, é também um conjunto de relações, axiomas, restrições, instâncias e vocabulários. Para construirmos uma ontologia com qualidade, o domínio de conhecimento deve ser selecionado com objetividade, descrevendo-o num vocabulário essencial que evite ambiguidade nas interpretações dos seus conceitos (GRUBER, 2013).

De acordo com Guarino (1998), diferentes tipos de ontologias podem ser definidas considerando-se o seu nível de generalidade: ontologia de nível superior; de domínio; de tarefas; e de aplicação.

Uma ontologia de nível superior ou de topo descreve conceitos gerais que são independentes do domínio, representando conceitos de espaço, tempo, eventos, objetos etc.

Uma ontologia de domínio ou de tarefas especializa conceitos da ontologia de topo, descrevendo termos relacionados com um domínio genérico ou tarefas e atividades genéricas. Por exemplo: biologia, informática, meteorologia (ontologias de domínio), diagnóstico, vendas (ontologias de tarefas).

A ontologia de aplicação descreve conceitos de um domínio ou tarefa em particular, especializando-os o máximo possível.

Alguns critérios na construção das ontologias devem ser observados visando a melhorar o processo. Esses critérios, segundo Gruber (1993), são:

- Clareza e objetividade na definição dos termos, provendo, inclusive, uma documentação em linguagem natural;
- Coerência para permitir inferências consistentes com as definições contidas na ontologia;
- Extensibilidade para que a ontologia seja capaz de suportar incorporação de novos termos sem ter de revisar conceitos já definidos;

- Codificação mínima para não depender de uma tecnologia particular de representação de conhecimento e compromisso ontológico mínimo para permitir compartilhamento e reuso da ontologia desenvolvida.

As ontologias definidas e utilizadas na plataforma OBAA-MILOS foram todas definidas em OWL(W3C,2004), pelo fato daOWL ser uma linguagem expressiva e ter a capacidade de interoperabilidade semântica para troca de contextos entre diferentes aplicações em vários domínios e por ser baseada em agentes de software.

O SAAPIEnS fará o uso das ontologias definidas na plataforma MILOS, assim como as ontologias de ensino e de aplicação definidas para o Sistema Tutor Heráclito.

3.6 Dedução Natural na Lógica Proposicional

3.6.1 Conceito

Dedução natural é um dos sistemas dedutivos utilizados para construir demonstrações formais na Lógica. Formalizar significa traduzir da forma lingüística para uma notação lógica, ou seja, uma forma que é compreendida por qualquer pessoa, independente da sua língua nativa, e que também reduz o espaço ocupado pela frase escrita, tendo em vista que podemos utilizar uma notação mais econômica, a lógica.

Na notação usual da Lógica Proposicional, as proposições são usualmente simbolizadas (representadas) por letras maiúsculas do início do alfabeto: A, B, C, ... Os valores lógicos das proposições são representados de forma resumida usando V para verdadeiro e F para falso.

Assim, proposições simples como “Hoje faz sol” ou “O carro é vermelho” podem ser simbolizadas da seguinte forma:

A = “Hoje faz sol”;

B = ”O carro é vermelho”.

As proposições podem ser simples ou compostas. As proposições compostas são formadas de proposições simples conectadas através de operadores (ou conectivos) lógicos. Estes operadores ou conectivos representam as seguintes operações lógicas :

Tabela 1 Operadores e símbolos lógicos

Operador Lógico	Símbolo	Linguagem Natural
Conjunção	\wedge	e
Disjunção	\vee	ou
Negação	\sim ou \neg	não
Implicação (ou condicional)	\rightarrow	se ... então ...
Bi-implicação(ou bicondicional)	\leftrightarrow	se e somente se

Fonte: Elaborado pelo autor

Podemos então usar a seguinte representação para proposições compostas:

$A \wedge B$ = Hoje faz frio E O carro é vermelho;

$A \vee B$ = Hoje faz frio OU O carro é vermelho;

$\sim A$ = Hoje Não faz frio;

$A \rightarrow B$ = SE Hoje Faz frio ENTÃO O carro é vermelho;

$A \leftrightarrow B$ = Hoje faz frio SE e SOMENTE se O carro é vermelho;

Um argumento dedutivo é uma declaração (proposição) que afirma que uma dada conclusão é consequência de um conjunto de outras afirmações (as premissas do argumento). Tais argumentos podem ser representados formalmente da seguinte maneira:

$$P_1 \wedge P_2 \wedge P_3 \wedge \dots \wedge P_n \rightarrow Q$$

Um argumento pode não ser verdadeiro. Para mostrar que um dado argumento sempre é verdadeiro, ou seja, é válido, deve-se derivar sua conclusão das premissas o que demonstra que essa conclusão é uma consequência lógica das premissas.

Isto é importante porque muitos argumentos que usamos são tão complexos que não é possível saber diretamente se são ou não válidos. Para o saber é preciso encontrar um caminho que conduza das premissas à conclusão, usando exclusivamente raciocínios cuja validade já se conhece e que são chamados de **argumentos dedutivos**.

As proposições P_1 a P_n são denominadas de **hipóteses** ou **premissas** do argumento. A proposição é denominada de **conclusão** do argumento. A representação em linguagem natural seria:

“ $P_1 \dots P_n$ implica em Q ” ou “ $P_1 \dots P_n$ infere em Q ”, ou ainda “ Q é inferido por $P_1 \dots P_n$ ”.

Argumentos tanto podem ser válidos ou inválidos. Um argumento será válido se e somente se, sempre que suas premissas forem todas verdadeiras, então a conclusão será verdadeira. Um argumento válido não pode ter premissas verdadeiras e uma conclusão falsa, ou seja, a fórmula $P_1 \wedge P_2 \wedge P_3 \wedge \dots \wedge P_n \rightarrow Q$ deve ser uma tautologia.

Para verificar se $P_1 \wedge P_2 \wedge P_3 \wedge \dots \wedge P_n \rightarrow Q$ é uma tautologia poderias utilizar o método da tabela-verdade, porém no caso de fórmulas com muitas proposições simples esse método se torna rapidamente impraticável (complexidade de memória exponencial no número de proposições). Outra maneira de testar a validade de um argumento é através do método de **dedução** que é baseado na aplicação das **regras de dedução** (ou **regras de inferência**) que modificam fórmulas de modo a preservar seu valor lógico.

A idéia básica deste método é começar com as premissas P_1, P_2, \dots, P_n (supostamente verdadeiras) e tentar aplicar regras de dedução até terminar com a conclusão Q . Esta conclusão teria que ser, então, verdadeira uma vez que os valores lógicos são sempre preservados sob as regras de inferência.

O fato de um argumento ter sido demonstrado é simbolizado da seguinte forma:

$$P_1, P_2, P_3, \dots, P_n \models Q$$

Uma regra de dedução (ou regra de inferência) é um mecanismo particular de geração de uma nova fórmula lógica a partir de outras fórmulas lógicas tomadas como premissas.

Inferência (ou raciocínio, ou dedução) é o processo de geração de novas fórmulas lógicas a partir de um conjunto de outras fórmulas lógicas, tomadas como premissas, usando regras de inferência. Se as regras forem logicamente válidas, então pode-se garantir que as novas fórmulas lógicas serão verdadeiras se as premissas forem verdadeiras.

No sistema de Dedução Natural para a Lógica Proposicional (DNLP) empregado no Heráclito e no SAAPIEnS, as regras de dedução são divididas em dois grupos:

- Regras de inferência que se baseiam em implicações tautológicas, ou seja, implicações materiais provadamente tautológicas.
- Regras de equivalência que se baseiam nas equivalências tautológicas que permitem substituir uma fórmula pela outra, já que ambas são equivalentes.

A Tabela 2 apresenta as regras básicas de inferência do sistema de DNLP.

Tabela 2 Regras de Inferências

Inclusão de Operadores	Exclusão de Operadores
<p>Redução ao absurdo (RAA)</p> $\begin{array}{l} P \\ \dots \\ Q \wedge \neg Q \\ \hline \neg P \end{array}$	<p>Dupla negação (DN)</p> $\begin{array}{l} \neg \neg P \\ \hline P \end{array}$
<p>Prova condicional (PC)</p> $\begin{array}{l} P \\ \dots \\ Q \\ \hline P \rightarrow Q \end{array}$	<p><i>Modus Ponens</i> (MP)</p> $\begin{array}{l} P \quad P \rightarrow Q \\ \hline Q \end{array}$
<p>Conjunção (Cj)</p> $\begin{array}{l} P \quad Q \\ \hline P \wedge Q \end{array}$	<p>Simplificação (Sp)</p> $\begin{array}{l} P \wedge Q \quad P \wedge Q \\ \hline P \quad Q \end{array}$
<p>Adição (Ad)</p> $\begin{array}{l} P \quad P \\ \hline P \vee Q \quad Q \vee P \end{array}$	<p>Eliminação da disjunção (-Dj)</p> $\begin{array}{l} P \vee Q \quad P \rightarrow R \quad Q \rightarrow R \\ \hline R \end{array}$
<p>Introdução da equivalência (+Eq)</p> $P \rightarrow Q \quad Q \rightarrow P$	<p>Eliminação da equivalência (-Eq)</p> $P \leftrightarrow Q \quad P \leftrightarrow Q$

$\frac{}{P \leftrightarrow Q}$	$\frac{}{P \rightarrow Q} \quad \frac{}{Q \rightarrow P}$
--------------------------------	---

Fonte:(GLUZ,PY 2010)

Estas 10 regras são completas no sentido que permitem a manipulação de todos os operadores da lógica proposicional. Do lado esquerdo estão listadas as regras que permitem a inclusão de um dado operador em uma nova fórmula a ser adicionada no decorrer da prova. As regras do lado direito permitem adicionar uma nova prova à demonstração, onde um determinado operador foi eliminado.

Além destas regras básicas, existe também um conjunto de regras derivadas que também podem ser usadas em demonstrações. Embora todas as regras derivadas apresentadas na Tabela 3a seguir possam ser substituídas por demonstrações sem o uso delas, elas podem ser bastante úteis em determinados casos, facilitando o processo de demonstração.

Tabela 3 Regras Derivadas

<p><i>Modus Tollens (MT)</i></p> $\frac{P \rightarrow Q \quad \neg Q}{\neg P}$	<p>Silogismo Hipotético (SH)</p> $\frac{P \rightarrow Q \quad Q \rightarrow R}{P \rightarrow R}$
<p>Silogismo Disjuntivo (SD)</p> $\frac{P \vee Q \quad \neg P}{Q}$	<p>Dilema Construtivo (DC)</p> $\frac{P \vee Q \quad P \rightarrow R \quad Q \rightarrow S}{R \vee S}$
<p>Exportação (Exp)</p> $\frac{(P \wedge Q) \rightarrow R}{P \rightarrow (Q \rightarrow R)}$	<p>Inconsistência (Inc)</p> $\frac{P \quad \neg P}{Q}$

Fonte:(GLUZ,PY 2010)

Um exemplo de representação da **demonstração formal** ou **sequencia de demonstração** do argumento: $A \rightarrow B \wedge A \rightarrow B \models A \rightarrow C$ é apresentada na Tabela 4:

Tabela 4 Demonstração Formal

Argumento a ser provado : $A \rightarrow B, A \rightarrow B \rightarrow C \models A \rightarrow C$			
Passo	Fórmula	Regra	Descrição
1	$A \rightarrow B \rightarrow C$	Hip	Hipótese
2	$A \rightarrow B$	Hip	Hipótese
3	A	Hip-PC	Hipótese da Prova Condicional

4	B	MP 2,3	Modus Ponens das linhas 2 e 3
5	$B \rightarrow C$	MP 1,3	Modus Ponens das linhas 1 e 3
6	C	MP 5,4	Modus Ponens das linhas 5 e 4
7	$A \rightarrow C$	PC 3-6	Conclusão da Prova Condicional da linha 3 até a linha 6 = Conclusão final da prova

Fonte : elaborado pelo autor

Onde: na coluna 1 numera-se os passos da derivação, na coluna 2 exhibe-se o resultado do raciocínio (é nesta coluna que se apresentam as fórmulas que serão geradas ao longo da demonstração) e na coluna 3 (a coluna das regras) exibem-se as regras de dedução utilizadas na demonstração.

4 TRABALHOS RELACIONADOS

Durante a análise do estado da arte foram pesquisadas ferramentas de autoria que possuem como característica principal serem de código-aberto, livres para uso sob a licença GPL, e outras que possuem uma versão comunidade (limitada em recursos para os usuários ou a versão comercial com todos os recursos liberados). Esta pesquisa também procurou por soluções de código-aberto cujas características apresentavam similaridades com as versões comerciais. Algumas das ferramentas encontradas possuem aplicações bem específicas, como é o caso do *Multimídia Learning Tools*, cuja finalidade é mais voltado para o ensino de línguas estrangeiras, pois possuem suporte interativo a áudio, vídeo e escrita. No caso do *LessonWriter*, a sua finalidade é voltada para a alfabetização nas séries iniciais. O *CALI Author* é uma ferramenta para construção de cursos e objetos para ensino à distância com foco na área jurídica. Nas demais ferramentas, as principais características são voltadas para a criação de conteúdos de *e-learning*, apresentações, simulações e criação de cursos com atividades interativas, *quizzes* ou de colaboração de uso multidisciplinar e de conteúdo genérico. Na tabela 5, são apresentadas todas as ferramentas pesquisadas.

Tabela 5 - Relação das Ferramentas de Autorias

Ferramenta	Localização	Licença
Course-Lab	http://www.courselab.com	L
Easygenerator	http://www.easygenerator.com	L/C
eXe Learning	http://www.exelearning.org	O
LCDS (Learning Content Development System)	https://www.microsoft.com/learning/tools/lcds/default.aspx	L
Lesson Writer	http://www.lessonwriter.com/html/index.asp	L/C
Memorize	http://memorize.com/	L
MOS Solo	http://www.moschorus.com/centre/MosPub/solo_en/index.html	L/C
Multimedia Learning Object Authoring Tool	http://www.learningtools.arts.ubc.ca/mloat.htm	O
Reload Editor	http://www.reload.ac.uk/editor_eclipse.html	O
UDUTU	http://www.udutu.com	L
Xerte	http://www.nottingham.ac.uk/~cczjrt/Editor/	O
Xical	http://www.xical.org	O
CALI Author	http://www2.cali.org/index.php?fuseaction=teaching.caliauthorinfo	L
PODIUM	www.authoronpodium.com/podium	L/C

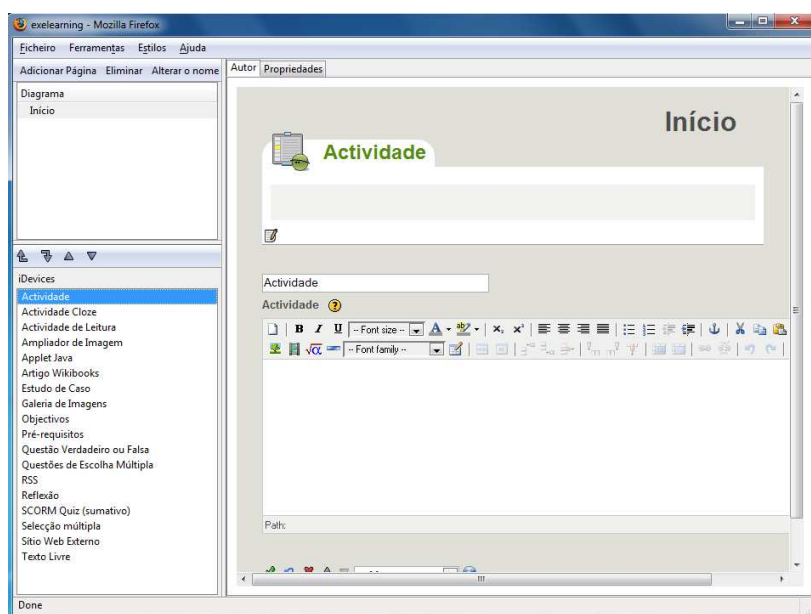
Fonte: Elaborado pelo autor.

4.1 Ferramentas de autoria

4.1.1 eXeLearning

O *eXe Learning* (EXE, 2014) é uma ferramenta de autoria desenvolvida na linguagem Python integrada ao Mozilla Firefox com licença de código aberto (openGL). Nela, os professores podem desenvolver e publicar objetos de aprendizagem, agregando diversas mídias como textos e recursos externos (*applets*, imagens, vídeos e animações em Flash), potencializando assim a aprendizagem dos seus alunos. O *eXe Learning*, na construção dos conteúdos educacionais, baseia-se no conceito de *i-devices*, que são os objetos criados e estruturados em tópicos, seções e unidades de ensino (ver figura 7). Os conteúdos criados podem ser exportados nos formatos: IMS *ContentPackage*, SCORM (ADL, 2010), página *websimple*s, além de sites *web* autocontidos ou arquivo de texto simples.

Figura 7 - Tela de edição *eXe Learning*



Fonte: elaborado pelo autor.

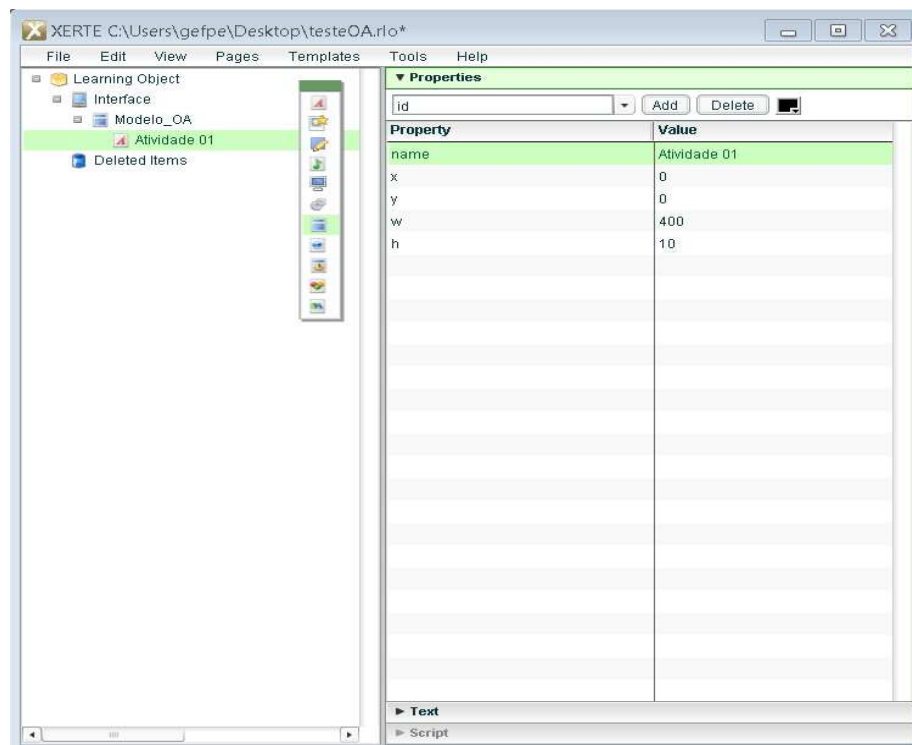
4.1.2 Xerte

O Xerte (BALL e TENNEY, 2008) é um conjunto de ferramentas de autoria para o desenvolvimento de conteúdos de aprendizagem interativa para ambientes *se-learning* com licença GPL. A ferramenta possui ambiente de criação visual, baseada em ícones que permite que objetos de aprendizagem para ser facilmente criados com o mínimo de código de *scripting*. O conteúdo pode ser compatível com o padrão SCORM 1.2 (ADL, 2010) para entrega em qualquer LMS ou AVA.

Para a criação de objetos no ambiente (ver figura 8) é necessário, por parte do criador, um conhecimento básico em linguagem HTML, porém, para criações ricas na internet com

maiores recursos de interação e apresentação, é fundamental um conhecimento mais aprofundado da linguagem Action Script (ADOBE, 2014). No caso de um usuário leigo, existe um recurso na forma de Wizards, para criação de objetos mais simples. No entanto, observou-se que a curva de aprendizado é bem maior em relação a outras ferramentas.

Figura 8 - Tela de edição do Xerte



Fonte: Elaborado pelo Autor.

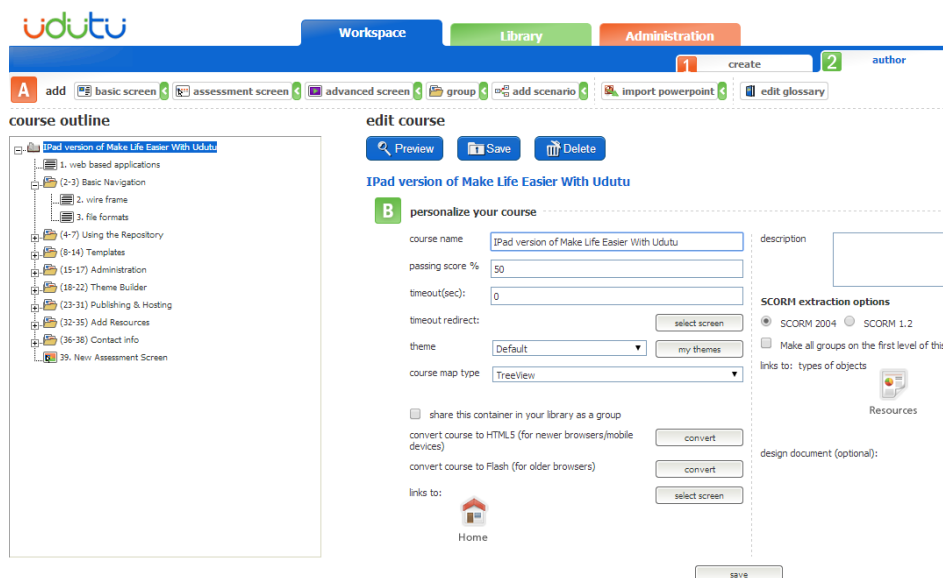
4.1.3 MyUDUTU

Essa ferramenta tem como facilidade ser utilizada de forma on-line na Web e, portanto, ela não precisa de nenhum arquivo de instalação. Na estrutura de curso apresentada pelo Udu, no painel “courseoutline” (ver figura 9), pode ser selecionado qualquer um dos itens, que corresponderá a uma tela na apresentação do pacote. Para adicionar um tipo de tela, clique na opção correspondente da aba *Workspace* (*basicsscreen*, *assesmentscreen*, *advancedscreen*, *group*, *addscenario*, *import PowerPoint* e *editglossary*). Todas as opções possuem configurações próprias, cabendo ao usuário avaliar qual é a melhor para suas necessidades. Há um formulário para espaço de trabalho com as abas *create*, *author* e *distribute*; a aba *author* é onde há o modo de edição de conteúdo, o qual é dividido em *courseoutline* (estrutura de curso) e *editcourse* (com elementos da estrutura selecionada).

É interessante observar que há uma opção, na estrutura, para importar arquivos do *PowerPoint* e converter para o formato de apresentação do próprio Udu. Na aba *distribute*,

há o formato de compactar e, dependendo de como o curso foi definido no modo de *author*, a ferramenta gera o respectivo pacote no padrão (SCORM 2004 ou 1.2). Os pacotes são ordenados, facilmente, através do recurso de arrastar e soltar, no painel *courseoutline*. Há também um botão com uma lupa que é o *preview*, paracada tela desejada de visualização da tela no pacote em execução.

Figura 9 - Tela de Edição do UDUTU

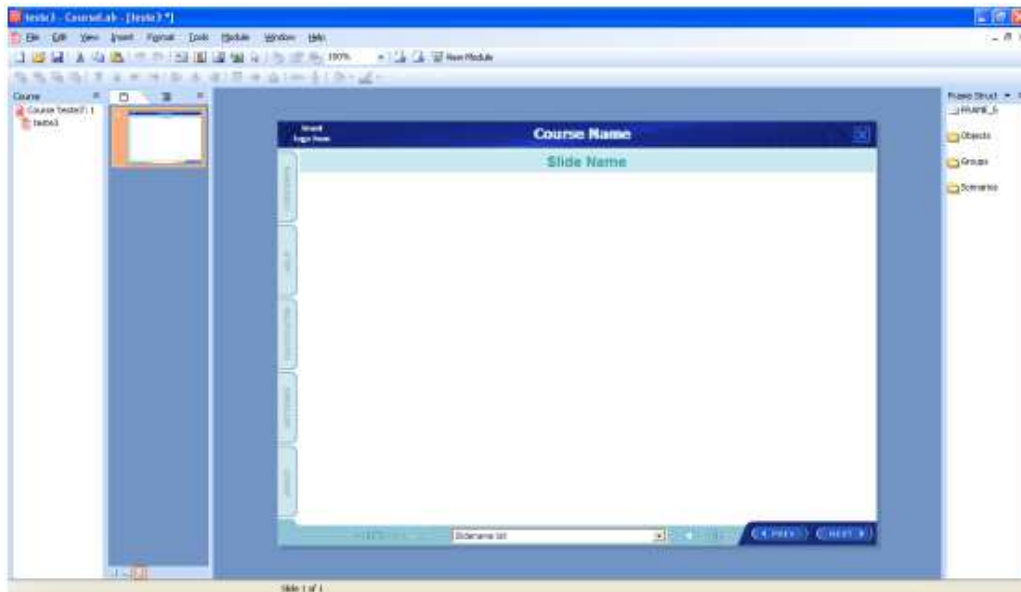


Fonte: Elaborado pelo autor.

4.1.4 CourseLAB

É uma ferramenta de autoria com ótimos recursos para criação de cursos completos e OAs. A estrutura para o material é dividida em curso, módulo e slide, podendo ter mais de um curso e o slide correspondente a cada uma das telas dos módulos. Esses slides funcionam como telas de *PowerPoint*. Pode-se observar que há três colunas com estrutura dos cursos à esquerda, telas em miniatura dos módulos e edição da tela selecionada no meio. Na coluna mais à direita, aparecem todos os elementos que podem ser inseridos com um clique, como textos, *cliparts*, endereços de sites, exercícios de vários tipos, imagens e outros. Clica-se e arrastam-se os slides dos módulos em miniatura ou os módulos à estruturação, para ordenar o pacote. Há um *preview* dos módulos e dos slides no menu *ViewModule*, onde é possível exportar os cursos no padrão SCORM 1.2, 2004 ou AICC para LMS.

Figura 10 - Tela de Edição do CourseLab



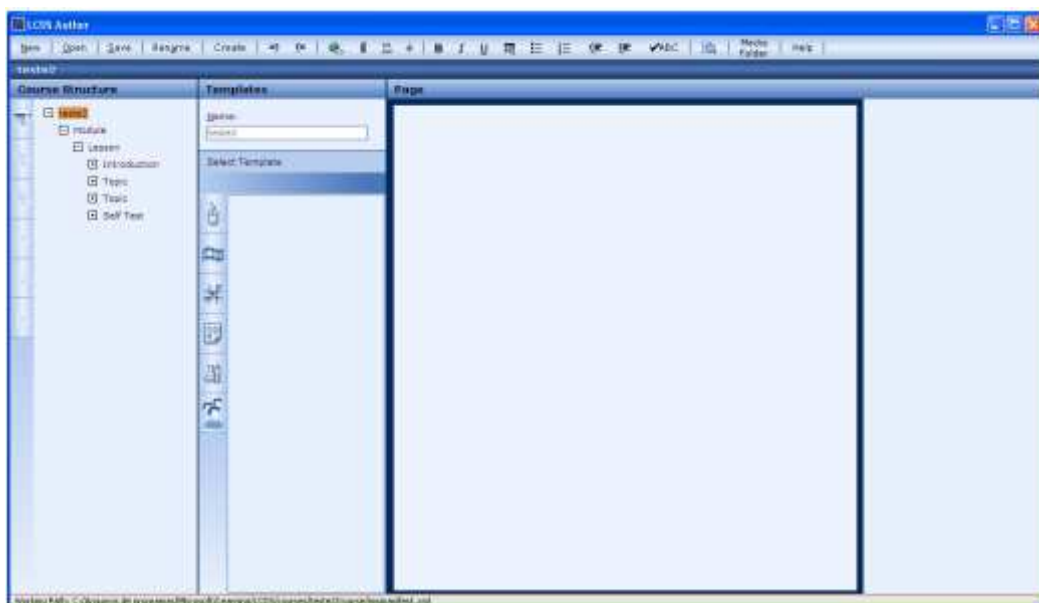
Fonte: Elaborado pelo autor.

4.1.5 LCDS

O LCDS (*Learning Content Development System*) é uma ferramenta de autoria livre de propriedade da Microsoft. Possui a estruturação hierárquica definida em Módulos e Lição, esta última sendo dividida em Introdução, Tópicos e Testes.

A estrutura dos cursos é pré-definida através de *templates*. A navegação da edição é através de ícones, onde podem ser inseridos os conteúdos desejados, tais como textos, arquivos multimídia, questionários e páginas que podem ser simulação. Apresenta recurso de *preview* do objeto, podendo exportar pacotes para LMS no padrão SCORM 1.2 (ver figura 11).

Figura 11 - Tela de edição do LCDS-Microsoft



Fonte: Elaborado pelo autor

4.2 Análise comparativa das ferramentas de autoria

Na pesquisa, procurou-se conhecer as ferramentas para edição de objetos de aprendizagem sem considerar os recursos adicionais, tais como: o preenchimento de metadados ou o empacotamento do objeto para uso em LMS (arquivos de manifestos). Em certos casos, ambas as funcionalidades foram encontradas nas ferramentas avaliadas.

Por outro lado, ainda nota-se uma dificuldade no efetivo uso dos objetos de aprendizagem produzidos em ambientes educativos. Nesse sentido, percebeu-se a necessidade das ferramentas possibilitar que os conteúdos que nelas foram produzidos, forneçam informações sobre as interações do aluno com o OA. Essa capacidade pode auxiliar o professor a preparar melhor seu material. Ao utilizar um objeto de aprendizagem com uma turma de alunos, o professor deseja saber como foi o processo de interação, quais foram os resultados obtidos, os caminhos percorridos, etc. Alguns LMS mapeiam, especificam e suportam tais informações através de plug-ins de terceiros. Entretanto, cabe ao objeto de aprendizagem implementar cada atributo, registrando-o na base de dados do LMS. Nesse sentido, apenas duas ferramentas suportam essas características. De forma mais rudimentar, a eXe Learning implementa apenas alguns dos atributos, independentemente do tipo de recurso utilizado (textos, *quizzes* ou outros componentes). Já a ferramenta *CourseLab* apresenta uma implementação mais completa, utilizando um maior número de elementos de metadados de OA, variando conforme os recursos educacionais definidos e utilizados pelo autor, porém com uma maior exigência de domínio da ferramenta, como no caso da ferramenta Xerte, devido aos recursos existentes, exige um domínio maior pela possibilidade de inserir elementos, que necessitam conhecimentos de programação por parte do autor. Como resultado, foram selecionados os seguintes critérios de análise das características dessas ferramentas:

1. Suporte a conteúdos multimídias: indica se a ferramenta tem a possibilidade de inserção de conteúdos para vários tipos de mídia, tais como arquivos textos, simulações, arquivos de vídeo, áudio, apresentações e outros;
2. Exportação para LMS: indica se a ferramenta é capaz de gerar packages para serem usados em LMS/AVAS. Os padrões estudados foram os padrões IMS, SCORM 1.2, 2004.
3. Suporte a Sistema Multiagente: indica se a ferramenta foi projetada ou tem a capacidade de interoperar em um sistema multiagente;
4. Integração com Sistema Tutor Inteligente: indica se a ferramenta tem a capacidade de interagir com um sistema de tutoria. Este critério verifica se os OAs gerados pelas ferramentas podem ser usados com apoio/sugestão de Sistema Tutor;
5. Monitoramento/Acompanhamento: verifica a capacidade das ferramentas de fornecer feedback ao autor na resolução das atividades propostas nos OAs, por parte dos alunos.

Na Tabela 6 a seguir, é apresentada a análise das ferramentas previamente analisadas. A última coluna dessa tabela apresenta as características da ferramenta SAAPInS proposta no presente, para fins de comparação.

Tabela 6 Características das ferramentas de autoria

	ExeLearning	Xerte	CourseLab	MyUdutu	LCDS	SAAPIEnS
Conteúdos Multimídia	S	S	S	S	S	S
Exporta p/LMS	S	S	S	S	S	P
Suporte aSMA	N	N	N	N	N	S
Integração STI	N	N	N	N	N	S
Monitoramento/ Acompanhamento	N	N	N	N	N	S

Legenda: Sim/Não/Parcial
Fonte: Elaborado pelo autor.

Após a utilização das ferramentas citadas, foi realizada uma análise comparativa entre as mesmas, através da criação de OAs a fim de verificar os critérios estabelecidos acima.

Na análise dos experimentos realizados, nenhuma das ferramentas apresentaram a característica de interoperabilidade em ambientes com arquitetura multiagentes e nesse item somente o SAAPIEnS atendeu à essa funcionalidade.

No critério de integração em ambiente de sistema de tutoria inteligente, as demais ferramentas não apresentaram essa característica, pois os OAs criados e disponibilizados são executados de forma *stand-alone* para serem utilizados pelos alunos, com exceção do MyUDUTU que pode ser utilizado na WEB, porém todos sem qualquer tipo de interação com um sistema de tutoria. O SAAPIEnS nesse aspecto, teve um fator positivo, pois pode operar tanto em um ambiente *stand-alone* com um suporte parcial implementado no próprio OA, quanto *on-line* quando conectado à plataforma da MILOS.

Com relação ao recurso de monitoração e acompanhamento, nenhuma das ferramentas analisadas apresentaram essa característica, pois os OAs criados apenas desempenharam um papel estático de auxílio no processo de ensino aluno, sem fornecer qualquer retorno ao professor, pelo fato de serem executados de forma isolada.

4.3 Ferramentas de acompanhamento educacional

O acompanhamento pedagógico é um elemento importante em processos de formação, que visa assegurar a qualidade do ensino pretendido no contexto referenciado (MEC, 2014). Numa perspectiva envolvendo professores, alunos e o próprio ambiente, este acompanhamento visa estimular a inovação pedagógica e fortalecer os processos de ensino dinamicamente, gerando novas concepções e práticas educativas (GADOTTI, 2000). É assumido como a possibilidade de qualificar mais o processo de ensino, por meio do planejamento e das ações. Os elementos principais que são considerados nesse contexto é a organização, planejamento do material e captação dessas experiências. Durante a fase de estudos, algumas ferramentas foram estudadas e entre elas gostaria de citar três pelo fato de apresentarem como características, um processo de acompanhamento e monitoramento da aprendizagem.

Com base nessas características foram pesquisadas ferramentas recentes de suporte ao acompanhamento pedagógico e que estejam inseridas em um contexto de ensino mediado através de ambiente digital de aprendizagem.

O *SEstatNet* (Nassar et al,2014) é uma ferramenta que usa internet como recurso, num ambiente flexível de ensino-aprendizagem . Ele é voltado basicamente para o ensino da estatística, no qual é disponibilizado um conjunto de recursos que possibilita o aprendizado. Nele o estudante planeja e realiza uma pesquisa, de forma a obter os dados que servirão de objeto de estudo. O sistema trabalha com metodologias ativas de aprendizagem, onde em cada sessão, o estudante possa aprender os principais conceitos estatísticos aplicando-os aos seus próprios dados.

O processo de aprendizagem é livre e os conteúdos não possuem uma seqüência pré-estabelecida de apresentação. Existe suporte para mapas conceituais. O controle de acompanhamento e monitoramento é realizado através das análises dos registros de uso gerados durante a interação do aluno com a aplicação. Existe um sistema de auto avaliação com auxílio de uma agente pedagógico animado que auxilia no processo. O professor tem acesso ao um conjunto de informações como gráficos e relatórios da trilha de aprendizado do aluno.

O trabalho de (CAMPANA,2008) caracteriza-se pelo desenvolvimento de uma camada multiagente implementada sobre o ambiente virtual de aprendizagem AVAUFES². O *framework* incorpora facilidades para o acompanhamento dos alunos, professores e administradores do ambiente. O desenvolvimento partiu de um estudo das deficiências encontradas nos AVAs em relação ao acompanhamento de atividades no ambiente, controle de agenda, controle de interação, conteúdos e observações pessoais. Uma característica do trabalho é a dos participantes poderem definir estratégia de acompanhamento dessas atividades, como: datas, acessos e interações com outros usuários. As facilidades de acompanhamento são definidas na interface de configuração, onde são configurados os parâmetros para que os agentes da plataforma façam o monitoramento. Tarefas como: notificações de inserção de novos conteúdos, realização e termino de atividades, aviso de chats, agendamento de reuniões síncronas, prazos limites, além de poder customizar outros tipos de eventos. O Professor além de ter o acesso ao acompanhamento dos alunos, pode ser notificado pelos agentes de todas as alterações que envolvemno ambiente.

O LOCO-Analyst (JOVANOVIC,2008) é uma ferramenta que proporciona aos professores um acompanhamento sobre os aspectos relevantes do processo de aprendizagem que ocorrem em um ambiente de aprendizagem baseado na web, assim como ajudar na melhoria dos conteúdos e na estrutura dos cursos também baseados na web. O sistema permite o acompanhamento das atividades dos alunos, preparação e disponibilização de conteúdos e contextualiza as interações sociais entre os alunos no ambiente virtual de aprendizagem. A ferramenta é baseada em tecnologias da web semântica, assim ele integra uma série de ontologias para representação formal dos objetos de aprendizagem, desempenho dos alunos, contextos e avaliações. O editor da ferramenta é extensão do Editor Reload (RELAOD,2014) e utiliza o padrão SCORM(1.2) para implementação nos LMS.

A geração do *feedback* da ferramenta é baseada na análise dos dados de rastreamento do usuário. Essas análises são baseadas na interação dos alunos (ou grupo de alunos) com um conteúdo de aprendizagem através da realização de determinada atividade (por exemplo, leitura, interrogando, conversando) com um propósito específico. O objetivo do LOCO (JOVANOVIC,2007) é facilitar a abstração de conceitos relevantes a partir de dados de rastreamento do aluno de vários sistemas e de ferramentas de e-learning.

² AVAUFES Ambiente Virtual a Distancia da Universidade Federal do Espirito Santo

4.4 Análise comparativa das ferramentas de acompanhamento

Em comparação com os trabalhos previamente analisados foi possível, verificar que o SAAPIEnS possui recursos que o diferenciou das demais ferramentas de acordo com critérios escolhidos, apresentados a seguir.

1. Processo de Interação dos alunos : capacidade de monitorar a interação dos alunos no uso da ferramenta e resolução das atividades;
2. Planejamento e Organização dos Conteúdos: conteúdos podem ser planejados de acordo com a necessidades a serem atingidas por parte dos alunos e a ferramenta apresenta uma organização na sua apresentação e execução ;
3. Desempenho dos Alunos: Verifica a capacidade das ferramentas de fornecer *feedback* ao autor na resolução das atividades propostas nos OAs;
4. Suporte a Sistema Multiagente: indica se a ferramenta foi projetada ou tem a capacidade de interoperar em um sistema multiagente;
5. Integração com Sistema Tutor Inteligente: indica se a ferramenta tem a capacidade de interagir com um sistema de tutoria. Este critério verifica se os OAs gerados pelas ferramentas podem ser usados com apoio/sugestão de Sistema Tutor;

Na Tabela 7 a seguir, é apresentada a análise das ferramentas previamente analisadas. A última coluna dessa tabela apresenta as características da ferramenta SAAPIEnS proposta no presente, para fins de comparação.

Tabela 7- Análise comparativa das ferramentas de acompanhamento

Item	SEstaNet	LOCO-Analyst	Trabalho de (CAMPANA, 2008)	SAAPIEnS
Interação dos Alunos	S	S	S	S
Organização e Planejamento dos Conteúdos	S	S	S	S
Desempenho Alunos	S	S	N	S
Suporte a Plataforma SMA	S	N	S	S
Integração com STI	N	N	N	S

Fonte: elaborado pelo autor

Das ferramentas analisadas somente o SAAPIEnS apresentou como característica, a integração entre o processo de autoria de objetos e o acompanhamento do uso destes objetos na aprendizagem. Outro diferencial do SAAPIEnS é suportar a integração com um sistema tutor inteligente, para o suporte ao aluno na execução das atividades.

O SAAPIEnS demonstrou através do comparativo das funcionalidades analisadas, que tem potencial para contribuir nos processos de autoria, uso e reuso de OAs, pois além de ser uma ferramenta de autoria também oferece recursos de acompanhamento e monitoramento, completando um cenário ideal para o desenvolvimento de conteúdos nos processos de ensino-aprendizagem.

5 SISTEMA SAAPIENS

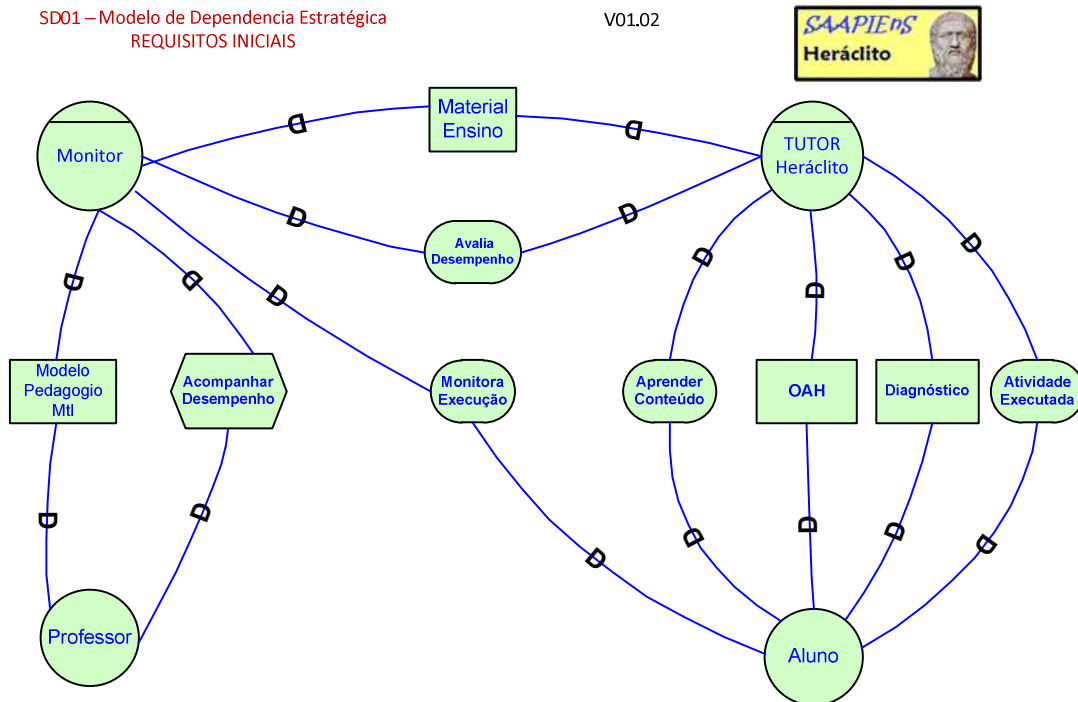
Este capítulo apresenta uma visão geral da análise de requisitos do SAAPIEnS usando a metodologia da TROPOS (CASTRO, KOLP e MYLOPOULOS, 2002) e das suas interfaces, trazendo informações sobre usabilidade e funcionalidades que o mesmo possui assim como sua arquitetura definida. Será apresentado um cenário da aplicação para melhor compreensão das suas características. O Sistema SAAPIEnS foi desenvolvido como uma ferramenta de autoria inteligente de apoio ao professor na criação, edição e customização dos Objetos de Aprendizagem Heráclito (OAHs), assim como uma ferramenta para realizar o acompanhamento e análise do desempenho dos alunos no uso dos mesmos. O Sistema utilizará as tecnologias de agentes inteligentes, na forma de *wizards*, de ontologias educacionais e fará uso análise estatísticas, criando assim um ambiente favorável de apoio ao professor no processo ensino de Lógica em ambientes digitais.

O nome SAAPIEnS é um acrônimo de **S**istema de **A**utoria e **A**com**P**anham**S** no **E**nsino **S**uperior. Heráclito de Éfeso, foi um grande filósofo e pensador grego e o fato de que, a principal aplicação do sistema é voltado a para autoria e customização de OA e apoio ao ensino da Dedução Natural da Lógica de Predicados, ele foi então designado, Sistema SAAPIEnS.

5.1 Requisitos

A análise dos requisitos do SAAPIEnS foi baseada na metodologia TROPOS (ver Seção 3.4.4). Na fase de requisitos iniciais (SD) foram identificados os atores Aluno e Professor, Tutor e o Agente Monitor. Além disso, são necessários os recursos: (Diagnóstico, Material-Didático, Material de Ensino e Modelo Pedagógico), e os objetivos (Aprender Conteúdo; Monitorar Execução, Atividade Executada e Avaliar Desempenho). O modelo de dependência estratégica dessa fase é apresentado na Figura 12.

Figura 12: Modelo de Dependência Estratégica de Requisitos Iniciais



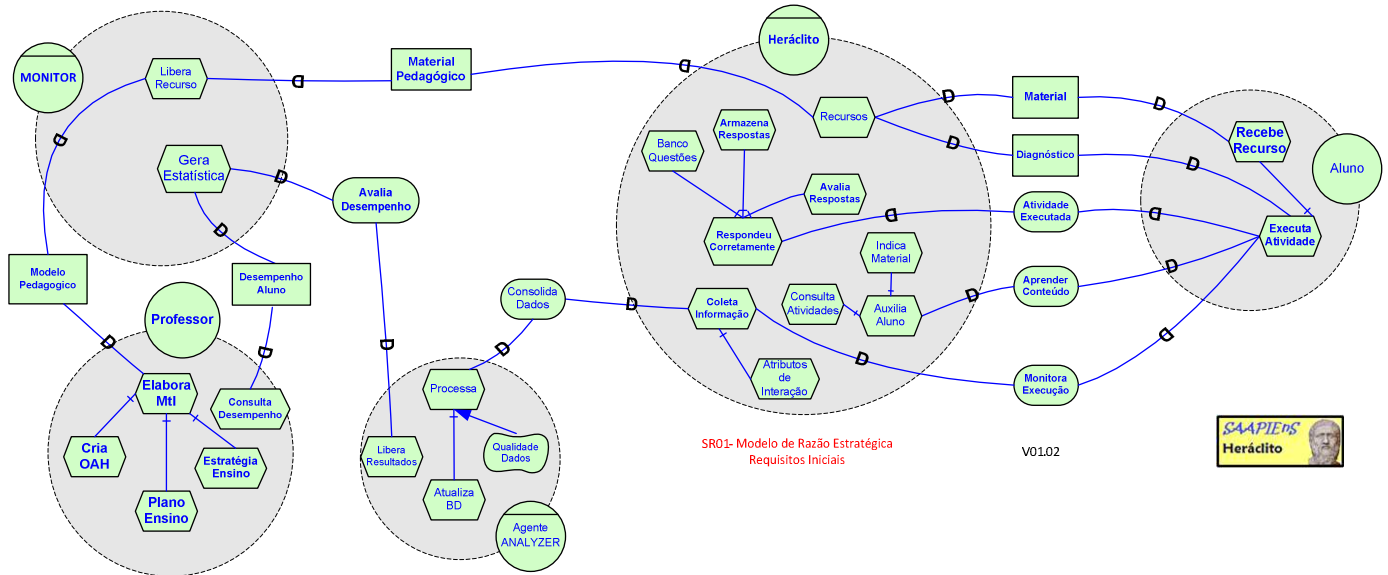
Fonte: Elaborado pelo Autor

O ator Aluno tem por objetivo aprender o conteúdo, mas para isso depende do agente Tutor para disponibilizar o material didático e os diagnósticos. O agente Tutor por sua vez tem por objetivos que a Atividade seja Executada pelo aluno. O agente *Monitor* tem por objetivo, acompanhar a execução da atividade do aluno.

O agente Tutor depende que o recurso Material de Ensino seja disponibilizado pelo agente Monitor que por sua vez tem o objetivo de Avaliar Desempenho e por sua vez depende do Tutor para passar essas informações e do Professor para criar e disponibilizar o recurso Material Pedagógico. O Professor depende do agente *Monitor* para poder acessar os dados do desempenho do aluno.

O modelo de razão estratégica (SD) explica as razões dos elementos do modelo de dependência estratégica ao adicionar novos elementos de modelagem. Assim, na Figura 13 são apresentados o agente Tutor e os atores Aluno e Professor e Monitor expandidos. Nesse modelo, novas tarefas foram adicionadas e os relacionamentos de composição de tarefas acrescentados e surge um novo componente o agente *Analyzer*.

Figura 13: Modelo de Razão Estratégica dos requisitos iniciais



Fonte: Elaborado pelo Autor

Neste modelo o Aluno tem como tarefas, receber os recursos e executar as atividades propostas pelo agente Tutor. O agente Tutor possui quatro tarefas principais que são elas: Receber os recursos do Agente *Monitor* e fornecê-los ao aluno. Auxiliar no ensino dos conteúdos, tendo como sub-tarefas indicação de material auxiliar e consulta das atividades realizadas. Monitorar a execução das atividades através da tarefa de Coleta da Informação tendo como sub-tarefa o controle dos atributos de interação de realização das mesmas. Verificar as atividades executadas tendo como sub-tarefas o armazenamento e avaliação das respostas, assim como manter um banco de questões e fazer a coleta das informações. O Agente *Monitor* por sua vez, avalia o desempenho através da tarefa de gerar estatísticas do aluno com base nos recursos de Dados Consolidados pelo Agente *Analyzer*. O agente *Analyzer* depende dos dados coletados para que ele possa avaliar a qualidade dos mesmos (objetivo *soft*), atualizá-los e gerar os resultados. O agente *Analyzer* disponibiliza os recursos de Material de Ensino, com base nos Recursos de Material Pedagógico (OAH) desenvolvido pelo Professor. Este faz o acompanhamento dos alunos através da Soft-goal, Desempenho do Aluno que servirá de base para elaboração de novos materiais alimentando o ciclo.

Na fase de requisitos finais da metodologia TROPOS é feito um refinamento dos modelos SD e SR desenvolvidos na fase de requisitos iniciais. Nessa fase, os modelos conceituais desenvolvidos na fase anterior são revisados e estendidos para incluir novos atores e tarefas, que representam tanto o sistema a ser desenvolvido quanto os seus subsistemas.

No caso do SAAPIEnS, o modelo SD dessa fase é bastante similar ao modelo SD inicial, apenas incorporando a interface *web* do sistema, que inclui suporte para as funcionalidades de autenticação e de acesso as informações do perfil do aluno. No modelo SR dessa fase são refinadas os objetivos e tarefas de cada um destes componentes, incluindo a interface web. O modelo SR da fase de requisitos finais é apresentado na Figura 14.

O ator Aluno deve se autenticar e iniciar as atividades propostas (recursos) do para o agente Tutor. Após a autenticação, o agente Tutor passa as informações do perfil do aluno para o Agente Monitor e fornece recursos de material de ensino para o aluno. Além disso, o Tutor tem o objetivo de ensinar o aluno, sendo que para isso ele deve monitorar as respostas do aluno, analisá-las e armazenar os resultados.

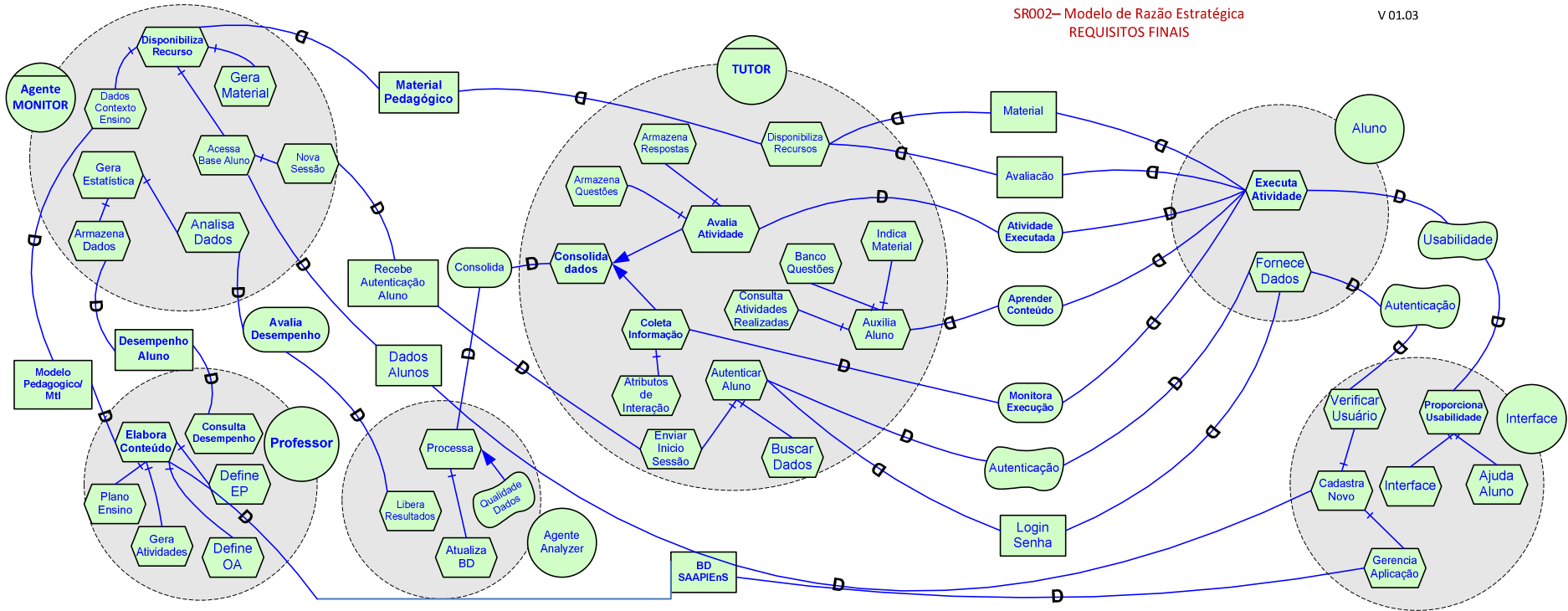
A tarefa Gerar de Novos Recursos depende da análise gerada pelo agente Monitor com base na coleta de dados feita pelo agente Tutor durante a execução das atividades feitas pelo Aluno e consolidadas e processadas pelo agente *Analyzer*.

O Professor tem o papel de consultar e analisar o desempenho dos alunos, assim como desenvolver novos conteúdos de ensino com base numa estratégia pedagógica adequada ao perfil dos alunos/turma, elaborar os planos de ensino e customizar novos objetos de aprendizagem com base nos recursos de autoria que a ferramenta disponibilizará, através dos recursos de edição e das bases de questões atualizadas e disponibilizadas pelo Sistema. Após isso o OAH será disponibilizado ao agente Monitor e ao Tutor.

Figura 14: Modelo de Razão Estratégica – Requisitos Finais

SR002— Modelo de Razão Estratégica
REQUISITOS FINAIS

V01.03



Fonte: Elaborado pelo autor

5.2 Arquitetura e Aspectos da Implementação

O protótipo inicial do SAAPIEnS foi desenvolvido na linguagem Java (ORACLE, 2014a), usando a IDE NETBEANS(ORACLE,2014b). As bases de dados foram implementadas utilizando o banco de dados MySQL (MYSQL, 2014). Outras tecnologias WEB foram necessárias como o *framework* JSF 2.2(ORACLE,2014c),JPA(ORACLE,2014d), HTML, CSS e Java script, além do uso de um servidor *webGlassFish* (ORACLE,2014c).

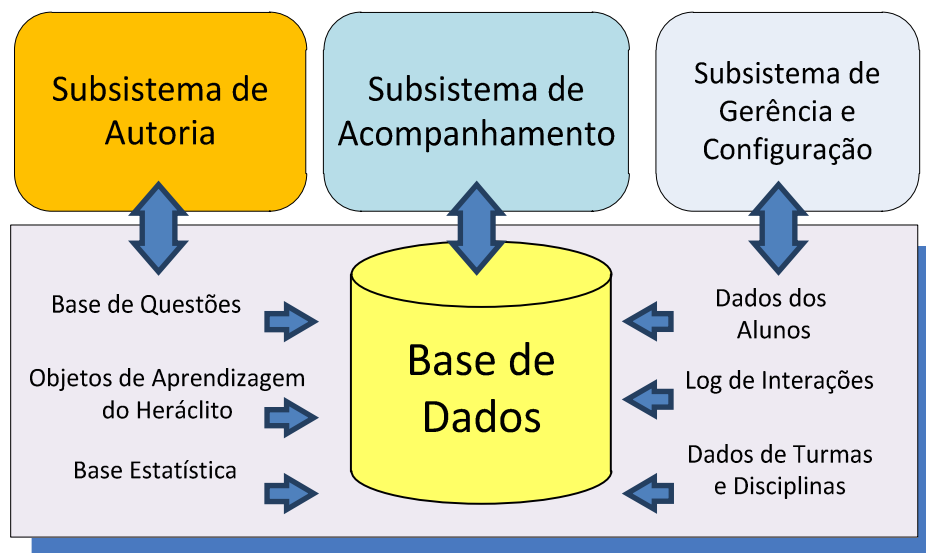
Os agentes do Heráclito foram desenvolvidos originalmente na linguagem AgentSpeak(L), com suporte para a plataforma de comunicação JADE. Estas linguagens e plataforma foram escolhidas por sua, portabilidade em função de também serem desenvolvidas em JAVA. Mais informações sobre as linguagens de programação e a plataforma JADE são apresentadas nas seções 3.4.2 e 3.4.4.

O modelo adotado para o desenvolvimento a arquitetura MVC(*Model-View-Controller*) (BRUEGGE;DUTOIT,2004) aplicada a plataforma MILOS (GLUZ e VICARI, 2010) .

5.2.1 Arquitetura

O Sistema SAAPIEnS é dividido em três subsistemas principais, sendo o primeiro, o subsistema de edição de objetos, responsável pela criação e edição dos OAHs, o segundo o subsistema de acompanhamento pedagógico responsável pela coleta de dados, controle das atividades e a geração de análises de dados. O terceiro subsistema de gerência de configuração, responsável pela parametrização e disponibilização do OA e manutenção da ferramenta conforme apresentado na Figura 15.

Figura 15:Arquitetura do SAAPIEnS



Fonte: Elaborado pelo autor

O padrão OBAA (VICARI e GLUZ, 2010) prevê a possibilidade que os OAs possam utilizar serviços externos que podem ser instalados localmente quanto remotamente através de um protocolo de acesso. Os metadados OBAA permitem que sejam definidas todas as características necessárias para que o OA utilize um dado serviço, especificando desde os endereços dos pontos de acesso ao serviço, mensagens e protocolos de interação utilizados pelo serviço, chegando até a linguagem e a ontologia empregada nas mensagens.

No caso dos OAHs, o acesso aos agentes do sistema de apoio pedagógico será disponibilizado por meio de uma interface Web Service. Os OAHs implementam um cliente de serviços, que acessa o provedor de serviços implementado pelo componente Facilidade de Acesso aos Agentes MILOS. Este provedor, por sua vez, implementa mecanismos de comunicação que convertem as requisições de serviços dos OAHs, em mensagens FIPA que podem então ser enviadas aos agentes MILOS por meio da plataforma de comunicação JADE.

5.2.2 Subsistema de Edição de Objetos

O subsistema de edição é a ferramenta principal para a criação do OAH, pois é através dele que o professor realiza as atividades de criação e disponibilização dos recursos. Entre algumas das facilidades que o sistema apresenta é a possibilidade de ser acessado via WEB, como pode ser visualizado na Figura 18.

O professor deve se autenticar no SAAPIEEnS e ter privilégio de edição, criação e disponibilização de OAH, tal como definido pelo administrador do sistema, no subsistema de gerência de configuração. No início do processo de criação do OAH o professor deve definir um conjunto de atividades didático-pedagógicas de acordo com os objetivos a serem alcançados e qual o nível de conhecimento que ele espera que os alunos alcancem. Este conjunto de recursos pode ser posteriormente alterado e customizado se necessário.

Na criação do OAH um conjunto de parâmetros precisa ser informado com a finalidade de personalizar as atividades do objeto, sendo que, a principal função desses parâmetros está relacionada ao nível de suporte que o OAH, oferece ao aluno. Os atributos definidos para um OAH são:

- Curso: identificação para qual curso será designado o OAH;
- Semestre/Ano: refere-se ao semestre de aplicação da lista;
- Data de criação: data de criação do OAH;
- Tipo de Atividade: sendo dividida em dois grupos de materiais, o primeiro relacionado aos exercícios de lógica e o segundo para materiais expositivos como apoio às atividades a serem realizadas. Os materiais ficam assim categorizados:
 - Exercício de Lógica: argumentos de dedução natural da Lógica Proposicional; tabela-verdade; avaliação de fórmulas.
 - Material Expositivo: textos;imagens;arquivos PDF;apresentação (arquivo .ppt);animações;vídeo;áudio.
- Descrição: descrição do OAH incluindo qual a sua finalidade e quais recursos serão utilizados e como as atividades devem se desenvolver;
- Número de questões: refere-se ao número de questões a serem inseridas na lista de atividades da primeira categoria do OAH;
- Seleção das Questões: o professor poderá fazer a seleção das questões que irão compor o OAH, é possível classificar as questões de acordo com o atributo visando facilitar o processo de seleção por parte do professor;
- Classificação do Objeto: um OAH pode ser classificado de em quatro níveis, 0 - Fundamental, 1-Básico , 2-Intermediário e 3-Avançado de acordo com critério estabelecido pelo professor que está criando o OAH;

Após a entrada dessas informações o subsistema de edição irá gerar automaticamente um código (ID) que será a identificação única do objeto dentro do SAAPIEEnS.

Uma vez definidos os parâmetros básicos, o professor pode customizar o perfil da tutoria/suporte para o OAH, configurando assim funcionalidades específicas do sistema de tutoria do Heráclito:

- Ajuda Tutor: parâmetro binário indicando se aluno terá suporte de tutoria ou não; Em relação a classificação é padrão a tutoria habilitada para os OAHs. Caso não seja necessário o professor pode desabilitar esse recurso;
- Dica: parâmetro binário que habilita se aluno terá ou não esse recurso de tutoria do objeto;
- Próximo Passo: parâmetro binário que indica se a tutoria fornecerá o próximo passo da solução;
- Atividades Extras/Material de Apoio: quando o professor esta configurando o OAH é possível também nesse momento, selecionar o conjunto de atividades extras que serão utilizadas pelo o aluno.

Através do subsistema de acompanhamento o professor terá a possibilidade de verificar se a configuração do OAH está realmente de acordo como o nível de aprendizagem da turma, ou seja, através das informações e das análises disponíveis no subsistema de acompanhamento será possível ao professor fazer os ajustes necessários, detectando dificuldades e ou facilidades encontradas por parte dos alunos. Na figura 16 é apresentada, a tela de configuração dos parâmetros do OAH.

Figura 16: Tela de entrada do parâmetros do OAH

Fonte: elaborado pelo autor

No subsistema de gerência da aplicação, há uma área de recursos onde é possível ao professor incluir essas atividades extras como objetos do tipo: documentos em formato PDF, arquivos de imagens, programas, animações ou questionários, através do recurso de *Upload*

de arquivos selecionados ou mesmo informar a URI de acesso dos mesmos. O professor poderá deixar instruções referentes para cada atividade em um espaço disponível para edição. É possível vincular vários arquivos de atividades extras ao mesmo OAH, bastando para isso, selecionar o recurso na tela de parâmetros do OAH como citado anteriormente ou ainda visualizar os objetos já inseridos através da consulta de atividades. Esses recursos poderão ser tanto de cunho informativo ou utilizado como atividades de apoio, avaliação ou complementares. Na figura 17 é visualizada a interface de atividades extras.

Figura 17 Tela de entrada de atividades extras

Fonte: elaborado pelo autor

O subsistema de edição permite o acesso à base de questões de lógica, dando suporte ao professor na escolha e sugestões das atividades aos alunos. O professor pode inserir questões novas e avaliá-las com o agente especialista do Heráclito. Ao inserir um argumento de lógica, o professor pode comparar o resultado apresentado pelo Agente especialista com a sua solução desenvolvida. Uma vez realizada a validação, o professor deve classificar a questão quanto ao seu nível de dificuldade, de acordo com sua interpretação. Existem dois níveis de classificação, o primeiro categorizado em classes: 1-Fundamental, 2-Básico, 3-Intermediário e 4-Avançado. O segundo refere-se ao nível de dificuldade dentro de cada categoria, variando do 1 ao 3, sendo o nível 3 o mais difícil.

Em relação ao processo de classificação das questões realizadas pelo professor, esse atributo poderá ter seu valor modificado, devido ao fator de *ranking* da questão que é medido por um processo constante de análise. Por exemplo, uma questão definida como classe intermediária e nível 1 poderá ser re-categorizada em outro nível e classe, na medida que vai sendo utilizada nos OAH e respondida pelos alunos. Na Figura 18 é apresentado a tela de validação dos argumentos de lógica e da inserção das questões na base.

Figura 18: Tela do Editor para inserção de questão e sua validação

Olá, admin! Sair?

Home
Minha área
Cadastrar Questões
Listar Questões
Voltar

SAAPiEnS - Heráclito

OBAA
FÓRUM DE ATUALIZAÇÃO
CONTÉÚDOS DE APRENDIZAGEM

Cadastrar Questão

Argumento a resolver: *

Categoria:

Nível:

Valida no Proveedor

1	q	Hip
2	~ (~p)	Hip
3	p^q-->r^s	Hip
4	p	DN 2
5	p^q	Cj 4,1
6	r^s	MP 3,5
7	s	Sp 6

Fonte : Elaborado pelo autor

Cada professor pode ter sua visão particular sobre quais são os melhores métodos de ensino, sobre como os conteúdos devem ensinados e em que ordem. A principal característica dos OAHs, é que sua reusabilidade, vem em favor dessa visão, permitindo, no mínimo, uma fácil adaptação e customização dos materiais contidos no OAHs.

A grande vantagem do uso da base de questões, é que há um processo de compartilhamento entre os professores que utilizam a ferramenta, pois cada um terá a liberdade de inserir e avaliar novos argumentos de DNLP, gerar novos materiais de apoio, assim como propor novos OAHs, fazendo da reusabilidade uma importante característica. À medida que o professor vai tendo acesso às análises estatísticas no uso e no controle de acompanhamento dos alunos, será possível a seleção das questões que mais se adequam às necessidades dos alunos, propondo assim, atividades individuais, para grupos específicos ou turmas de alunos. Outra possibilidade, é o próprio sistema, através do agente *Analyzer*, sugerir a questão com base no histórico e no conjunto de informações disponibilizada na base de dados do SAAPiEnS. Na Figura 19 pode-se visualizar a interface de consulta das questões catalogadas.

Figura 19: Interface de consulta das questões.

SAAPIENS - Heráclito

Lista de Questões

ID	Argumento a resolver	Categ.	Nivel	Ranking Ok	Ranking Errc	N°de Util.	Uso
1	$p \rightarrow q, p \vdash q$	0	Intermediário				
17	$A \rightarrow (B \rightarrow C), A \vee \neg D, B \vdash D \rightarrow C$	1	Intermediário				
18	$A \rightarrow B, B \rightarrow (C \rightarrow D), A \rightarrow (B \rightarrow C) \vdash A \rightarrow D$	1	Intermediário				
19	$A \wedge B \vdash (A \rightarrow B)$	1	Intermediário				
20	$P \leftrightarrow Q, Q \leftrightarrow R \vdash P \leftrightarrow R$	1	Intermediário				
21	$P \leftrightarrow Q \vdash \neg P \leftrightarrow \neg Q$	1	Fundamental				
23	$P \rightarrow Q \vdash (P \vee R) \rightarrow (Q \vee R)$	1	Intermediário				
24	$\neg P \rightarrow P \vdash P$	1	Avançado				
25	$\neg P \vdash P \rightarrow Q$	1	Intermediário				
22	$\neg P \vee Q \vdash \neg (P \wedge \neg Q)$	2	Intermediário				
26	$k \rightarrow r, k \vdash r$	2	Avançado				

Fonte: Elaborado pelo Autor

Outro recurso disponibilizado é a base de regras de inferência, cuja finalidade é manter o histórico da sua aplicação na solução das atividades. Os atributos da base de regras são divididos em duas categorias, descritivos e estatísticos. Na primeira estão definidos:

- A identificação da regra propriamente dita;
- A descrição simbólica da regra;
- A descrição analítica da regra, onde constam os principais conceitos e definições.

Na categoria estatística estão definidos os atributos que armazenam a utilização que cada regra teve, tanto de maneira correta ou incorreta. Essas informações serão mantidas e atualizadas pelos agentes do sistema. É possível ainda, inserir materiais adicionais vinculados a cada regra de inferência. Esses materiais devem ser categorizados em níveis de ajuda e servem para serem utilizados, como indicação pelo sistema Tutor do Heráclito de acordo com grau de classificação do OAH, estabelecido pelo professor no momento da sua criação. Na Figura 20 é apresentada a tela de inserção na base de regras de inferência.

Figura 20: Tela de cadastro do Banco de Regras

SAAPIENS - Heráclito

Olá, admin! Sair?

Cadastrar Regra

Código da Regra: RAA

Descrição do Material: *

Classificação do Material: Ajuda Nível 1

Nível: Fundamental

URI do Material:

Nome do arquivo: Execicio_RAA.pdf

Tipo de arquivo: application/pdf

Unisinos - Universidade do Vale do Rio dos Sinos - PIPCA

Fonte: Elaborado pelo autor

Outrapossibilidade doSAAPIEnS,é sua utilização diretamente de dispositivos móveis. Como sua arquitetura de desenvolvimento foi baseada em tecnologia para WEB, é possível ao professor criar e acessar os seus OAHs através de dispositivos móveis. Na Figura 21 é apresentado o acesso à ferramenta por meio de umTablet.

Figura 21 Acesso à ferramenta através do Tablet

Saapiens V1.3.4

http://obaa.unisinos.br/saapiens/content/CRUD/atvextra/List.jsf

SAAPIENS - Heráclito

Olá, admin! Sair?

Cadastrar Atividade Extra

Tipo de atividade: Ajuda Nível 1

Classificação: Fundamental

Descrição da atividade: *

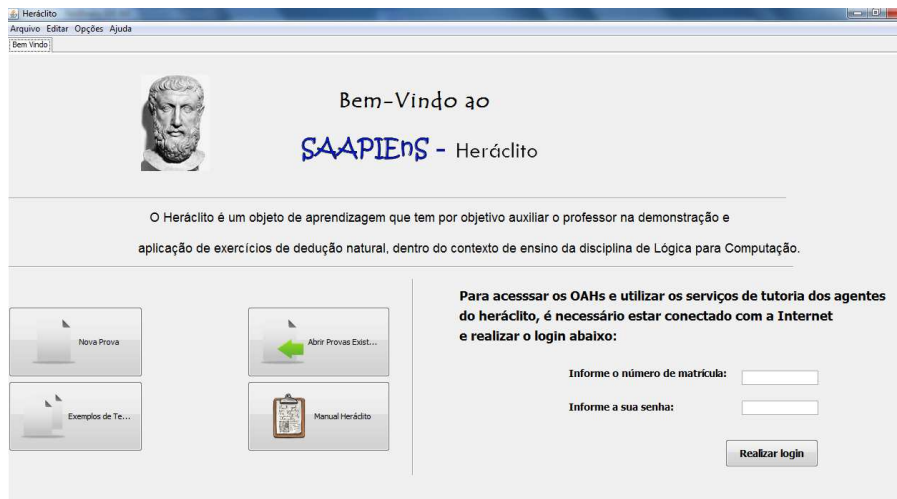
URI do Material:

19h53

Fonte: Elaborado pelo autor

Depois de realizadas as etapas para criação, é possível disponibilizar o OAH para ser visualizado e testado pelo professor. O OAH é um aplicativo independente do ambiente operacional, pois funciona como uma aplicação Java desktop. A única exigência é que, para o aluno usar os recursos providos pelos serviços de tutoria do Heráclito, o aluno necessita estar conectado à internet. No momento que é realizado o *login*, a tela de acesso do OAH é habilitada ao aluno, na qual o mesmo tem acesso às informações, referente às atividades a serem executadas. Na figura 22 é possível visualizar a tela de principal do OAH.

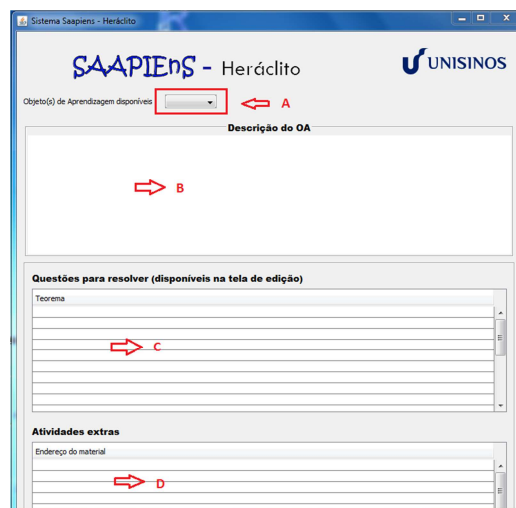
Figura 22 Tela de apresentação do OAH



Fonte: Elaborado pelo autor

Na figura 23 é apresentada a tela de acesso às atividades do OAH. Em (A) o aluno irá selecionar, dentre os objetos disponíveis, qual será trabalhado. Em (B) é apresentada a descrição do OAH e quais as tarefas e orientações designadas pelo professor. Em (C) são apresentadas as questões que compõem o objeto. O aluno pode selecionar qualquer uma das questões e terá acesso à tela do editor onde ele pode dar início a resolução. Em (D) é apresentado o endereço para acesso das atividades extras proposta pelo professor. O acesso é realizado diretamente pelo Browser que faz o acesso aos documentos disponibilizados.

Figura 23 Tela de Atividades do OAH



Fonte :Elaborado pelo Autor

5.2.3 Subsistema de Acompanhamento

O subsistema de acompanhamento é o responsável pela coleta dos dados e das atividades feitas pelos alunos e pela geração das informações estatísticas dos dados consolidados. Para tanto o subsistema é capaz de monitorar as interações entre os alunos e as atividades na resolução de exercícios de Lógica contidas nos OAHs. São os registros dessas interações que fornecem as informações necessárias para essas análises no subsistema de acompanhamento.

Em termos de arquitetura o subsistema de acompanhamento é formado por dois agentes de *software*: o agente *Monitor* e o agente *Analyzer*, descritos a seguir.

5.2.4 Agente MONITOR

Esse agente tem o papel de monitorar as interações dos alunos com os OAH e atualizar as bases de dados com os registros dessas monitorações. As ações realizadas pelo aluno durante a execução das atividades, são persistidas no Log de atividades e servem para as análises posteriores.

As ações monitoradas pelo agente *Monitor* são:

- O tempo total que o aluno está conectado realizando atividade;
- O tempo dispendido para realizar uma determinada prova seja ela parcial ou total. Aqui nesse caso, define-se como prova parcial, àquela em que o aluno não chegou ao resultado final, porém deu início a sua resolução;
- Tempo de inatividade do aluno;
- O número de passos que o aluno precisou para realizar determinada prova;
- O uso de ajuda do Tutor solicitado pelo aluno;
- O uso de dicas que o aluno teve para determinada solução;

Os recursos de apoio fornecidos pelo tutor/OAH ao aluno, nesse caso são divididos em dois níveis: O primeiro nível de ajuda Tutor, é um nível mais abrangente onde pode ser a indicação um material extra para o aluno, explicação de provas através de exemplos, etc... O segundo nível de suporte é o de Dicas, aqui o sistema fornece auxílios pontuais, como sugestão de regras e avisos quando uma regra está usada de maneira incorreta.

O agente também controla quais as regras utilizadas e a quantidade de vezes que foram utilizadas tanto corretas, quanto incorretamente na solução dos passos para todos os argumentos avaliados.

Quando o aluno estiver resolvendo as atividades *stand-alone*, ou seja, não estando conectado à plataforma, a ferramenta manterá os históricos das atividades em um arquivo temporário armazenado no computador do próprio aluno. No momento que o aluno efetuar conexão no ambiente *on-line*, esses dados serão transferidos e analisados pelos agentes atualizando assim a base de dados dos alunos.

O propósito de manter os históricos das atividades dos alunos, estando eles conectados à plataforma ou não, é para que o agente que faz o monitoramento e coleta, possua a maior quantidade possível de dados, afim de gerar um conjunto de informações relevantes para suporte ao professor.

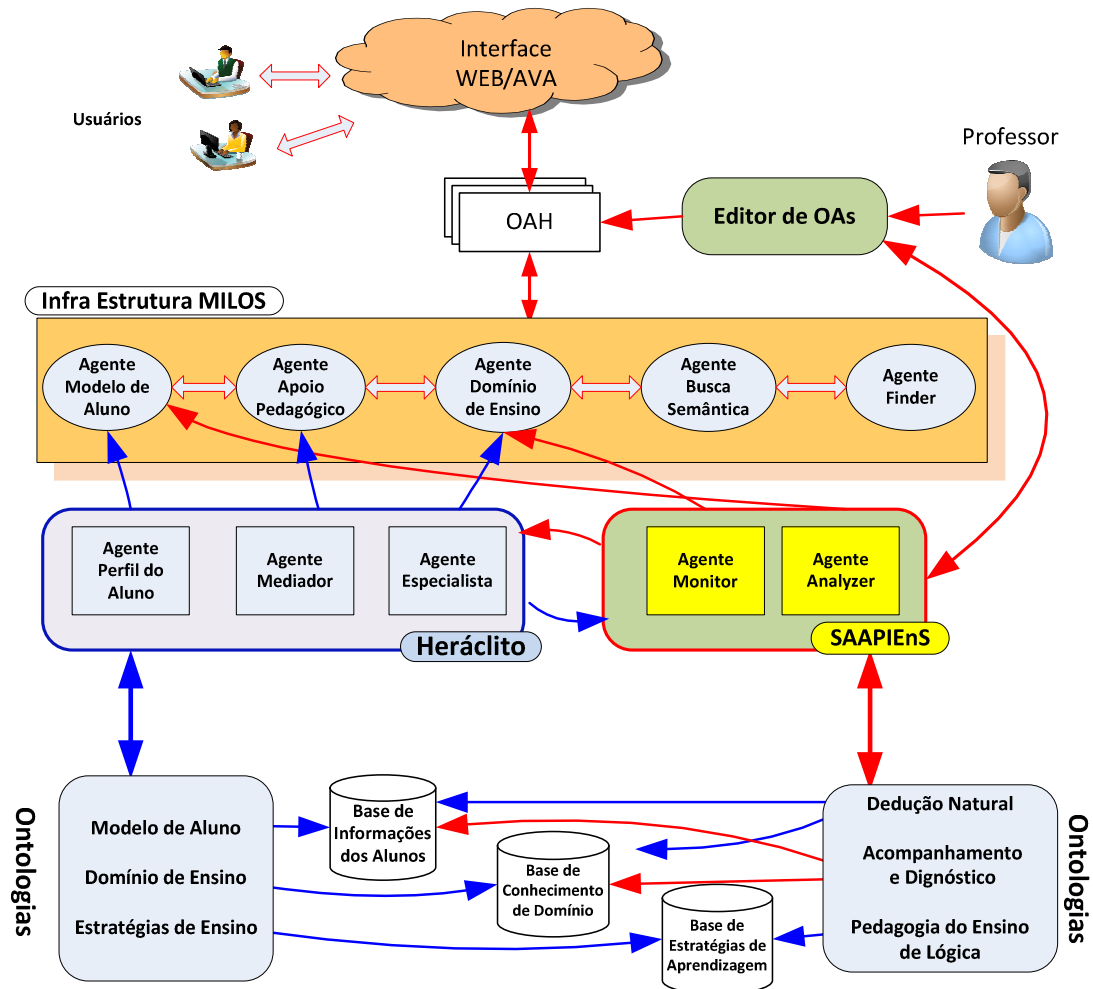
No modo *stand-alone* são considerados apenas os argumentos que estejam catalogados nos OAHs propostos, ou novos argumentos inseridos e testados pelos alunos de forma aleatória, cuja resolução tenha sido alcançada com êxito. No caso dos argumentos não catalogados nos OAH, esses poderão ser avaliados e classificados pelo agente *Analyzer* e então, inserido na base de dados. Nesse aspecto o OAH tem uma ação colaborativa, pois a medida que o alunos utilizam nos estudos, realizando novas provas e argumentos, ele vai incrementando a base de questões do SAAPIEEnS.

5.2.5 Agente ANALYZER

O agente *Analyzer* é o responsável pelo pré-processamento dos dados, consolidação para geração das análises finais. A principal função do Agente *Analyzer* é gerar os resultados que servirão de subsídios ao professor, para que o mesmo possa através dos mesmos, fazer o acompanhamento do desempenho dos alunos, assim como desenvolver ações pedagógicas visando contribuir na melhoria do processo de ensino.

Os agentes do SAAPIEEnS possuem uma interação com agentes do Heráclito na plataforma da MILOS, através de troca de mensagens e de recursos entre os mesmos. Na Figura 24, é possível visualizar o ambiente da plataforma e de interação entre os agentes do sistema.

Figura 24 Integração SAAPIEnS - Heráclito na plataforma MILOS



Fonte : Elaborado pelo autor

O *log* de atividades do SAAPIEnS é atualizado à medida que o aluno interage com o OAH. Durante o processo de resolução dos exercícios, o agente Perfil de Aluno do Heráclito, recebe as requisições e as envia ao agente especialista, responsável pela realização das provas. Feito isso o agente especialista envia o resultado ao agente Perfil de Aluno, nesse momento há a troca de mensagens entre o agente de Perfil de Aluno do Heráclito e os agentes *Analyzer* e *Monitor*, pois eles são responsáveis pela coleta, análise e persistência das informações, resultantes desse processo.

Após a conclusão de um prova realizada pelo aluno, o agente *Analyzer* consolida os dados dos alunos de cada OAH, mantendo o histórico sempre atualizado. Na tabela 8 é apresentado o registro (*log*) de atividades que é gerado na resolução uma questão. É possível verificar a composição de cada registro do *log* de interação. Sendo os atributos: 1) Identificação do Aluno, 2) Identificação do OAH, 3) Número da Atividade do OAH, 4) Data/hora de Final da atividade, 5) Status de Inatividade (0 Ativo 1-Inativo), 6) Descrição da Prova / Hipótese Informada, 7) Status da regra utilizada corretamente, 8) Uso de Ajuda Tutor, 9) Uso de Próximo Passo, 10) Data/hora de início da atividade, 11) Status da Prova (I Iniciada / P parcial /

C concluída), 12)Número Sequencial da Prova, 13)Regra utilizada, 14)Nº de tentativas erradas, 15)Uso de Dicas do Tutor, 16) FeedBack tutor.

As informações armazenadas nos registros de atividades são utilizadas para a análise dos processos de interatividade, acompanhamento e desempenho dos alunos. É possível também exportar os registros no *log* (ver tabela 8) para serem processadas por uma ferramenta externa ao SAAPIEEnS.

Tabela 8 - Log de Atividades do SAAPIEEnS

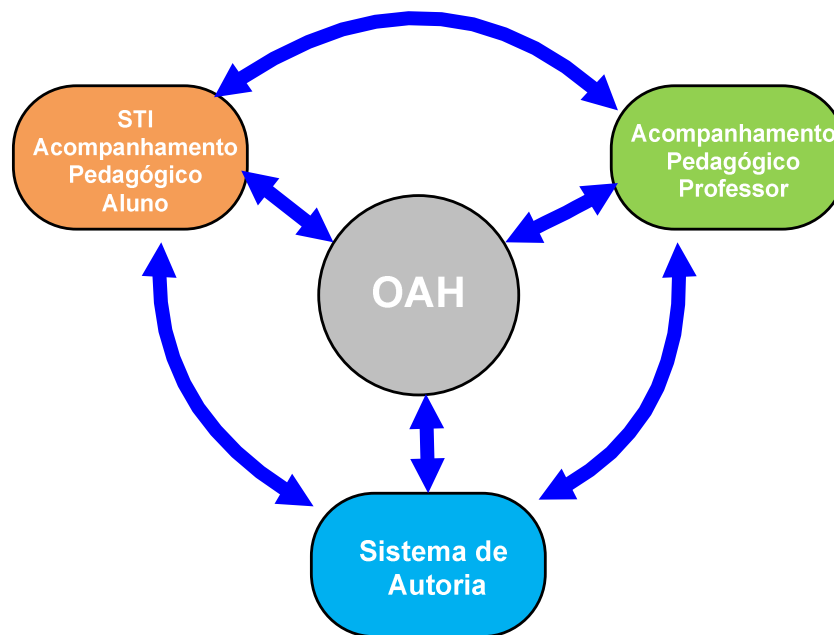
Argumentoresolvido: A->B, A->(B->C) - (A->C)															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
admin	000001	99	2014 07 20 09:31:07		I	0	0			true	0			1	0
admin	000001	99	2014 07 20 09:31:34		P	0	1	A->B	HIP	true	0			1	0
admin	000001	99	2014 07 20 09:31:34		P	0	2	A->(B->C)	HIP	true	0			1	0
admin	000001	99	2014 07 20 09:31:44		P	0	3	A	HIP- PC	true	0			1	0
admin	000001	99	2014 07 20 09:31:52		P	0	4	B	MP	true	0			1	0
admin	000001	99	2014 07 20 09:32:06		P	0	5	B->C	MP	true	0			1	0
admin	000001	99	2014 07 20 09:32:15		P	0	5	B->C	MP	false	1			1	0
admin	000001	99	2014 07 20 09:32:22		P	0	6	C	MP	true	0			1	0
admin	000001	99		2014 07 20 09:32:25	C	0	7	A->C	PC	true	0			1	0

Fonte: elaboradopeloautor

O conjunto de informações armazenadas na base de dados do SAAPIEEnS, resultado dos processos tanto de criação / customização por parte dos professores, quanto da interação na resolução dos OAHs por parte dos alunos, servem como recursos para a geração de análises, estatísticas, relatórios de controle e de acompanhamento. Os agentes do sistema, desempenham assim, o papel de monitorar e atualizar essas bases sempre que uma atividade for executada e concluída pelos alunos, permitindo ao professor realizar ações de caráter preventivo ou corretivo no processo de ensino sempre que necessário.

O ciclo de criação e resolução pode ser visualizado na figura 25, onde o OAH representa o elemento central do processo de autoria, de tutoria e suporte e do acompanhamento do progresso do aluno realizado por parte do professor.

Figura 2525 - Ciclo de criação e resolução o do OAH



Fonte: elaborado pelo autor

5.2.6 Subsistema de Gerência e Configuração

Esse subsistema é o responsável pelas ações de controle e parametrização da ferramenta, permitindo que o administrador da plataforma MILOS possa executar as seguintes tarefas gerenciais relacionadas ao suporte aos objetos Heráclito:

- Manutenção das contas de alunos e professores;
- Manutenção de cadastros gerais;
- Parametrização e habilitação de recursos da ferramenta;
- Configuração e execução de cópia de segurança;
- Importação e Exportação de Arquivos;
- Rotina de análises de consistência da Base de Dados.

6 METODOLOGIA DE AVALIAÇÃO

Neste capítulo é apresentado o processo de avaliação do protótipo do SAAPIEnS, incluindo os testes funcionais realizados com a ferramenta e a avaliação da usabilidade desta ferramenta.

A metodologia de avaliação da usabilidade foi baseada em um cenário relativamente complexo de utilização do SAAPIEnS, para que fosse possível avaliar tanto aspectos de usabilidade da ferramenta, quanto seu potencial no processo de ensino-aprendizagem.

6.1 Experimentos de Validação da Funcionalidade

Os testes e experimentos foram realizados em laboratório para verificar a funcionalidade da ferramenta de autoria e acompanhamento do SAAPIEnS.

Na fase inicial de implementação da ferramenta, foi desenvolvido o protótipo da interface do sistema para verificar se atendiam as necessidades e as especificações definidas nas fases anteriores. Os testes funcionais visaram avaliar a satisfação das especificações da ferramenta. Foram realizados testes de adequação da ferramenta de autoria aos requisitos definidos na análise de requisitos. Outro teste realizado foi verificar se os dados gerados pela ferramenta de acompanhamento teriam a precisão e acurácia necessárias para atender ao professor no uso da ferramenta. O teste de interoperabilidade foi realizado com as ferramentas de autoria e acompanhamento interagindo no ambiente da MILOS, usando o *WEB-Service* e o serviço de comunicação dos agentes da plataforma via FIPA, com o Sistema Tutor Heráclito, em ambiente local ou através de dispositivos móveis. Outro teste realizado foi o de segurança e da integridade das informações persistidas e o processo de recuperação das mesmas, através de teste contínuos em ambiente *WEB* e local. Por último foram realizados testes de conformidade visando à padronização do projeto da ferramenta.

Os testes de validação funcional foram executados por um período de um mês, seguido de um pequeno período necessário para correções nas ferramentas. Após as correções uma nova série de teste foi aplicada, com resultados positivos, certificando a operação e funcionalidade das ferramentas do SAAPIEnS.

6.2 Avaliação de Aspectos de Usabilidade e do Potencial Educacional

No cenário hipotético utilizado na avaliação educacional do SAAPIEnS, o professor necessita preparar uma atividade que envolvam exercícios de lógica para uma turma de alunos do curso de Ciência da Computação. Os exercícios a serem preparados são referentes aos conteúdos à Dedução Natural da Lógica de Predicados. Os conteúdos que são pré-requisitos para o desenvolvimento do tema da atividade é a Lógica Proposicional e as regras de Dedução Natural. Assume-se que esses conteúdos já foram previamente apresentados em sala de aula. Assume-se também que já foram ministradas práticas em sala de aula visando à fixação desses conteúdos.

Os exercícios de dedução a serem propostos serão os seguintes:

- | | |
|--|-------------|
| 1. $C, B \rightarrow A, C \rightarrow B \mid \neg A$ | Fundamental |
| 2. $\neg A \rightarrow (B \rightarrow C), \neg A, B \mid \neg C$ | Fundamental |
| 3. $P, \neg \neg (P \rightarrow Q) \mid Q \vee \neg Q$ | Básica |
| 4. $\neg A, B \rightarrow A \mid \neg \neg B$ | Básica |

5. $P \leftrightarrow Q \sim P \leftrightarrow \sim Q$	Básica
6. $(P \wedge Q) \rightarrow R, P \sim Q \rightarrow R$	Intermediário
7. $A \rightarrow (B \rightarrow C), A \vee \sim D, B \sim D \rightarrow C$	Intermediário
8. $P \leftrightarrow Q \sim P \leftrightarrow \sim Q$	Intermediário
9. $\sim (P \wedge Q) \vee (\sim P \vee \sim Q)$	Avançado
10. $A \vee \sim B, C \vee \sim D \sim A \vee B \vee C \vee D$	Avançado

No cenário o professor utiliza a ferramenta de autoria de OA para criar um OAH contendo tais exercícios. Após tal OAH é publicado e disponibilizado aos alunos. O professor começa, então, a acompanhar a execução das atividades previstas no OAH, utilizando a ferramenta de acompanhamento para monitorar o uso do OAH.

Esse acompanhamento irá ilustrar possíveis problemas e dificuldades encontradas pelos alunos no uso do OAH ou na execução das atividades. O professor pode, através da ferramenta de autoria, ajustar detalhes no OAH ou no processo de tutoria automática, para minimizar ou corrigir esses problemas.

Por fim, o professor também poderá utilizar o *feedback* oferecido pela ferramenta de acompanhamento para aperfeiçoar seus conteúdos.

6.2.1 Etapa de criação do OAH

Na aula anterior ao dia da aplicação do exercício, o professor apresenta a ferramenta de suporte aos OAH aos alunos, demonstrando suas principais funcionalidades, tais como: fazer o acesso *on-line* à plataforma e realizar uma atividade dirigida de como usá-la na resolução dos OAHs durante a aula, num processo de tutoria orientada.

Com o uso do SAAPInS o professor tem condições de construir o OAH para ser disponibilizado aos alunos. No planejamento do OAH o professor definiu um conjunto de atividades abrangendo o conteúdo de DNLN e prepara também um material de apoio para utilização pelos alunos.

Após realizar *login* de acesso, o professor pode consultar a base de questões e verificar se as questões catalogadas já constam na base, cadastrando as questões escolhidas para o exercício. As questões foram previamente escolhidas e catalogadas na base do SAAPInS, abrangendo 4 (quatro) níveis de dificuldades, sendo: 2 (duas) do nível Fundamental, 3 (três) Básicas, 3 (três) Intermediárias e 2 (duas) Avançadas. Convém ressaltar, que esses níveis foram definidos inicialmente, obedecendo alguns critérios básicos sendo eles: (a) N° de Hipóteses; (b) N° de passos na sua resolução; (c) N° de regras utilizadas e por último, (d) Tempo médio de resolução. Esses critérios tiveram como parâmetro, estudos realizados com professores da área, comparando as suas próprias resoluções, com a do Tutor do Heráclito ficando assim as questões assim categorizadas na base. A Figura 25 ilustra esse processo de cadastramento de argumentos.

Figura 26 Cadastro dos argumentos do OAH naBase de Questão

The screenshot shows a web browser window with the URL `obaa.unisinos.br/saapiens/content/CRUD/bdq01/Create.jsf`. The page header includes the logo for SAAPIENS - Heráclito and OBAA (Paraná de Atendimento de Direitos de Aprendizagem). The user is logged in as 'Olá, Ludovico Parda! Sair?'. The main content area is titled 'Cadastrar Questão' and shows a success message: 'Questão criada com sucesso.'. Below this, the form contains the following fields: 'Argumento a resolver: *' with the value 'AvEvB, Cv-EvD |- AvBvCvD', 'Categoria:' with a dropdown menu showing '01', and 'Nível:' with a dropdown menu showing 'Avançado'. A 'Salvar' button is located below the form.

Fonte: Elaborado pelo autor

Após o cadastramento e seleção das questões, o professor realiza a inserção de um material comatividades extras, contendo um texto sobre regras de inferências de dedução natural, além de exemplos de resolução utilizando as regras derivadas e outra atividade extra, incluindo umaapresentação em PowerPoint contendo um tutorial básico de dedução. A Figura 26 mostra como é a interface de inclusão de atividades extras.

Figura 27 Inserindo Atividade Extra

The screenshot shows a web browser window with the URL `obaa.unisinos.br/saapiens/content/CRUD/atvextra/List.jsf`. The page header includes the logo for SAAPIENS - Heráclito and OBAA. The user is logged in as 'Olá, Ludovico Parda! Sair?'. The main content area is titled 'Editar Atividades Extras' and shows the following fields: 'Tipo de atividade:' with a dropdown menu showing 'Ajuda Nivel 1', 'Classificação:' with a dropdown menu showing 'Básico', 'Descrição da atividade:' with a text area containing 'Tutorial BASICO de LOGICA Exercicios resolvidos e uso das Regras de Inferencia.', 'URI do Material:' with a '+ Material' button and 'Upload' and 'Cancel' buttons, 'Nome do arquivo:' with a text input field containing 'AtividadeExtra001.pdf', 'Tipo de arquivo:' with a text input field containing 'application/pdf', 'Data de conclusão:' with a text input field containing '30/07/2014', and 'Data de criação:' with a text input field containing '27/07/2014'. A 'Salvar' button is located at the bottom of the form.

Fonte : Elaborado pelo autor

Para finalizar a etapa de criação do OAH professor seleciona a opção de Parâmetros do menu do Editor de OAH, onde pode ser preenchido, além da descrição deste objeto e dalista de questões que o irão compor, as informações sobre atividades extras e opções de tutoria (ver Figura 27).

Figura 28 Tela de Parâmetros do OAH

Parâmetros do OBJETO de APRENDIZAGEM

Matrícula do Professor: Ludovico Pardal

Data Criação: 29/07/2014

Descrição do OAH:

Este OAH tem por finalidade realizar uma validação para a resolução de questões de Dedução Natural na Logica de Predicado

O OAH esta dividido em duas tarefas:

O uso do Sistema Tutor Heraclito para resolução das questões ;

Material adicional para consulta dos alunos da atividade.

Questões do OA

<input type="checkbox"/>	Argumento a resolver	Categoria	Nível	Número de Utilizações	Uso em Avaliação	Uso em Exercício	Tempo Médio Solução
<input type="checkbox"/>	$A \rightarrow B, A \rightarrow (B \rightarrow C) \vdash (A \rightarrow C)$	0	Fundamental				
<input type="checkbox"/>	$(C \rightarrow D) \rightarrow C \vdash (C \rightarrow D) \rightarrow D$	1	Básico	43	0	2	5.05
<input type="checkbox"/>	$\neg A, A \vee B \vdash B$	2	Básico				
<input type="checkbox"/>	$(P \wedge Q) \rightarrow R, P \vdash Q \rightarrow R$	0	Básico				
<input type="checkbox"/>	$A \rightarrow (B \rightarrow C), A \vee \neg B \vdash C$						

Fonte: Elaborado pelo autor

Depois de preenchidos os parâmetros do OAH (ver Figura 27) o professor seleciona (do banco de questões) quais questões deverão ser resolvidas pelos alunos neste OAH específico (ver Figura 28). Isso encerra o processo de criação do OA, assim logo após este objeto pode ser disponibilizado para acesso aos alunos através do Heráclito (ver Figura 29).

Figura 29 Seleção de questões do OAH

Parâmetros do OBJETO de APRENDIZAGEM

<input type="checkbox"/>	Argumento a resolver	Categoria	Nível	Número de Utilizações
<input type="checkbox"/>	$A \rightarrow (B \rightarrow C), A \vee \neg D, B \vdash D \rightarrow C$	0	Intermediário	
<input checked="" type="checkbox"/>	$A \rightarrow B, B \rightarrow (C \rightarrow D), A \rightarrow (B \rightarrow C) \vdash A \rightarrow D$	1	Intermediário	
<input checked="" type="checkbox"/>	$A \wedge B \vdash (A \rightarrow \neg B)$	1	Intermediário	
<input checked="" type="checkbox"/>	$P \leftrightarrow Q, Q \leftrightarrow R \vdash P \leftrightarrow R$	1	Intermediário	
<input type="checkbox"/>	$P \leftrightarrow Q \vdash \neg P \leftrightarrow \neg Q$	1	Intermediário	
<input type="checkbox"/>	$\neg P \vee Q \vdash (P \wedge \neg Q)$	1	Intermediário	
<input type="checkbox"/>	$P \rightarrow Q, P \rightarrow \neg Q \vdash \neg P$	2	Intermediário	

Tipo de Atividade: Atividade de DNLP

Classificação da Atividade: Intermediário

Ajuda do: Sim

Fonte: Elaborado pelo autor

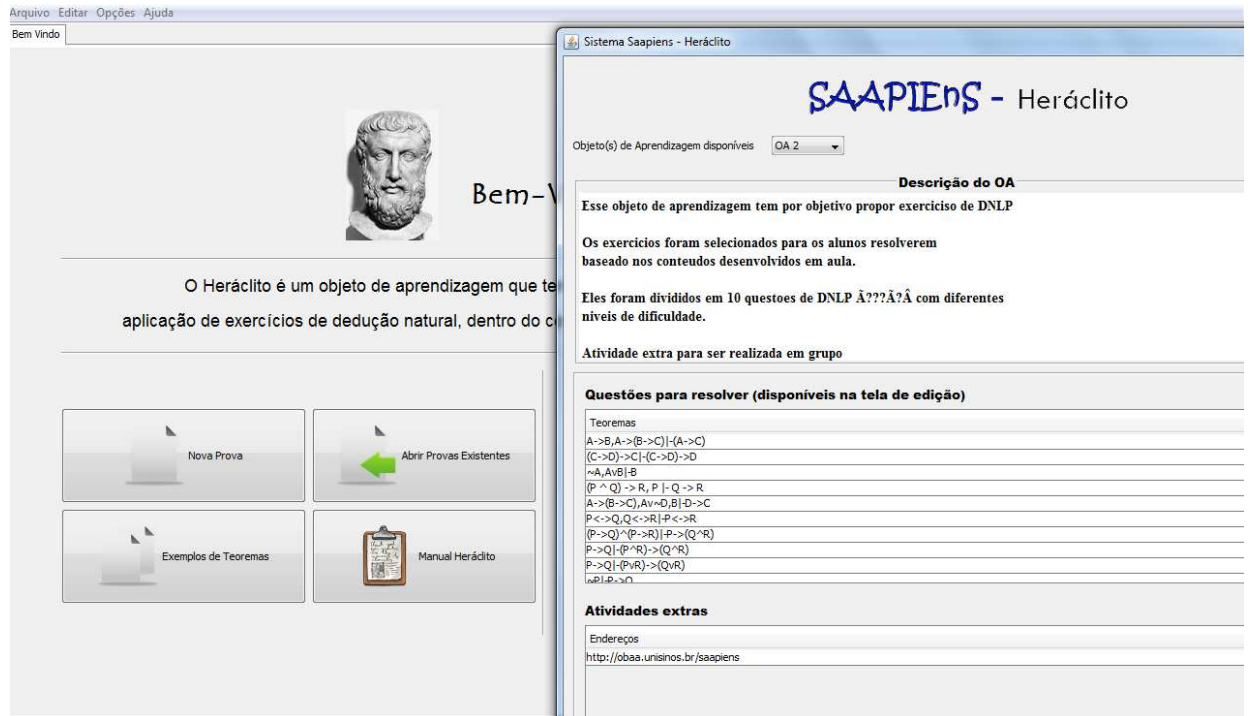
Figura 30 Tela do OAH proposto

Fonte: Elaborado pelo autor

6.2.2 Etapa de uso do OAH

Quando os alunos se autenticam no Heráclito, é apresentado na tela inicial deste sistema os OAH disponíveis. Após a seleção do objeto a ser estudado, então são apresentadas na tela do Heráclito as atividades previstas, as questões a serem resolvidas bem como as atividades e materiais extras que foram disponibilizados. A Figura 30 mostra como ocorre esse processo. Ao selecionar qualquer uma das questões, o editor de provas de dedução natural (ver Figura 31) é apresentado para que se possa dar início ao processo de resolução das questões.

Figura 31 OAH Gerado

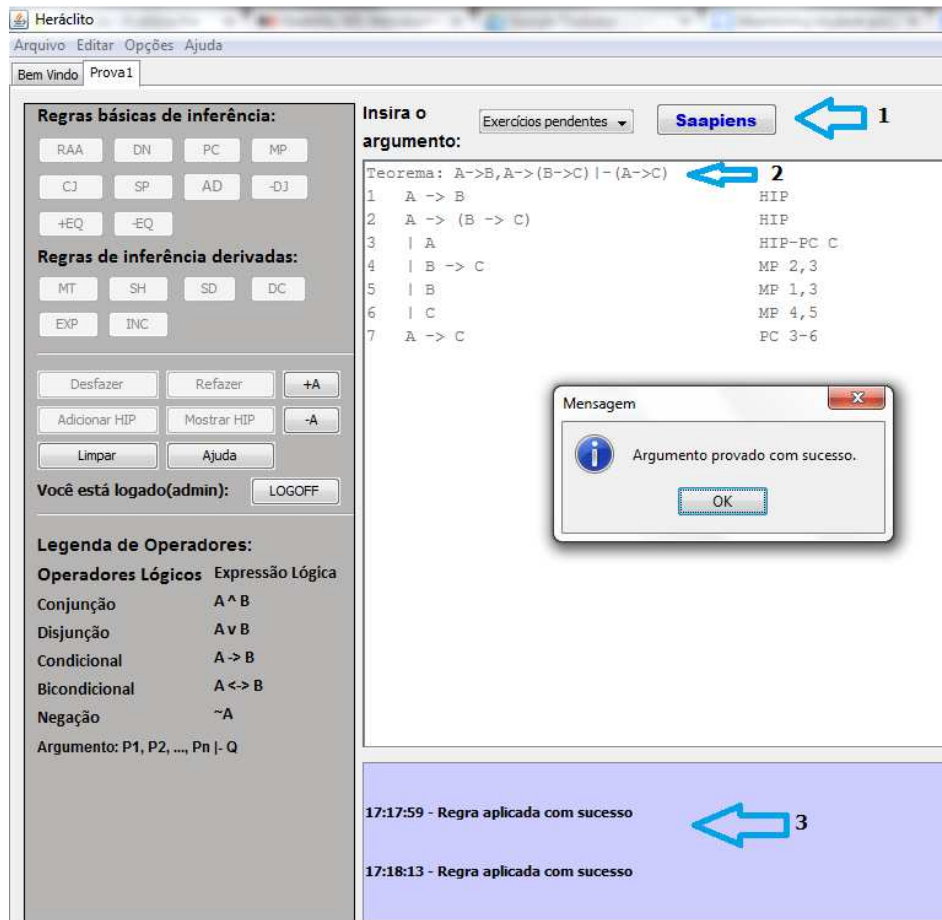


Fonte: Elaborado pelo autor

Durante o processo de resolução é realizado o monitoramento do uso da ferramenta, através de consultas regulares às bases de dados da aplicação e verificação dos Logs dos registros de atividades, com o propósito também de sanar possíveis dúvidas que pudessem ocorrer. Ao finalizar tarefas as propostas os alunos fazem a desconexão da plataforma.

Caso os recursos de tutoria do OAH tenham sido habilitado, então na própria interface do editor de provas serão apresentadas mensagens e dicas do tutor para ajudar o aluno no processo de resolução da questão. A Figura 31, que apresenta a interface do editor de provas do Heráclito, ilustra essa funcionalidade: mensagens do serviço de tutoria são apresentadas no painel inferior (3), em relação a demonstração sendo feita pelo aluno no painel (2). O botão de retorno (1) permite que o aluno retorne à tela principal do OAH a fim de consultar algum material ou selecionar um novo argumento a ser resolvido. O serviço de tutoria também pode interagir com o aluno através de caixas de mensagens, tal como indicado na Figura 31.

Figura 32 OAH sendo utilizado



Fonte: elaborado pelo autor

6.2.3 Etapa de avaliação e aperfeiçoamento

Nessa etapa, o professor irá avaliar como decorreu o processo de uso e resolução do OAH por parte dos alunos. Para que isso ocorra, inicialmente o professor irá proceder a consulta individual do aluno para analisar os registros da solução das questões do OAH proposto. No módulo de análise SAAPIENS, o professor seleciona cada um dos alunos e gera a consulta, conforme apresentado na Figura 32.

Figura 33 Seleção de Aluno - Acompanhamento

Selecione um aluno para gerar o relatório

Alunos						
	Matricula	Nome	ID do curso	ID da turma	ID da disciplina	Sem/Ano
<input type="radio"/>	1212121	Aluno Teste 03	Sistemas de Informacao	Lógica segunda a noite	132132	2/14
<input checked="" type="radio"/>	1234567	Aluno Teste 02	Sistemas de Informacao	Lógica segunda a noite	10020	2/14
<input type="radio"/>	1313131	Aluno Teste 03	Sistemas de Informacao	Lógica segunda a noite	121212	2/14
<input type="radio"/>	2323232	Aluno Teste 04	Sistemas de Informacao	Lógica segunda a noite	10020	2/14
<input type="radio"/>	2345678	Aluno Teste 01	Sistemas de Informacao	Lógica segunda a noite	60550	1/14
<input type="radio"/>	3434343	Aluno Teste 05	Sistemas de Informacao	Lógica segunda a noite	10020	2/14

Relatório

Fonte: Elaborado pelo autor

Na consulta individual, o professor consegue visualizar informações importantes que servem para verificar o desempenho do aluno na resolução de cada questão do OAH. Na Figura 36, atributos representando o sumário dos registros de interação do aluno podem ser visualizados. É possível visualizar no item (1) a identificação do Aluno 2, com um resumo da sua solução, onde são exibidos, o tempo médio de solução do aluno e o tempo total dispendido. No quadro das questões o professor pode visualizar as questões que foram resolvidas e não resolvidas item(3), indicado pelo status “c” concluída e “n” não concluída. Nas questões 7 e 8 item (2) o aluno resolveu as questões, porém com um tempo elevado e acima da média normal pré-definida para as mesmas (conforme Figura 33). É possível verificar o nº de passos na sua solução, bem como o nº de tentativas, inclusive se houve solicitação de Ajuda ou Dicas do Sistema Tutor. Na questão 9 e 10 item(3), o professor percebe que o aluno tentou várias vezes, porém não obteve êxito na sua solução.

Figura 34 Consulta Individual do aluno

Relatório										
Nome:	Aluno Teste 02 1		Turma:	Lógica segunda a noite						
Semestre:	2/14		Curso:	Sistemas de Informacao						
Tempo total:	00:90:20		Tempo médio:	00:11:02						
Menor tempo:	00:00:00		Maior tempo:	00:29:08						
Completo:	80.0%		Parcial:	0.0%		Não realizado:	20.0%			
Questão	Cls	Resolução	Tempo	Início	Término	Nº passos	Nº T OK	Nº T Err	Ajuda	Dica
1	0	c	00:02:08	31/07/2014	31/07/2014	6	1	0	0	0
2	0	c	00:05:08	31/07/2014	31/07/2014	7	1	0	0	0
3	0	c	00:08:08	31/07/2014	31/07/2014	8	1	0	0	0
4	0	c	00:05:08	31/07/2014	31/07/2014	6	1	0	0	0
5	0	c	00:09:08	31/07/2014	31/07/2014	7	1	0	0	0
6	0	c	00:09:08	31/07/2014	31/07/2014	9	2	1	0	0
7	0	c	00:29:08	31/07/2014	31/07/2014	10	1	2	1	1
8	0	c	00:22:08	31/07/2014	31/07/2014	14	2	1	1	1
9	0	n	00:00:00	31/07/2014	31/07/2014	0	0	8	1	1
10	0	n	00:00:09	31/07/2014	31/07/2014	0	0	9	1	1

Fonte: Elaborado pelo autor

Foi possível também ao professor, consultar cada questão consolidada do OAH, e até poder avaliar o grau de dificuldade apresentada pela mesma esta de acordo com o que é apresentado pelos alunos na sua resolução. Na Figura 34, é apresentada a consulta que professor faz à cada uma das questões que compõe o OAH. Nessa consulta ele avalia, o nº de vezes que a questão foi utilizada em um OAH e onde (exercício ou avaliação), assim como o nº de vezes que ela foi respondida tanto correta, quanto incorretamente, o seu tempo médio de solução, assim como verificar a otimização da resolução com base no nº de “passos”, nº mínimo e máximo.

Figura 35 Consulta questão consolidada

Visualizar Questão	
ID da Questão:	11
Argumento a resolver:	(C->D)->C -(C->D)->D
Categoria:	1
Nível:	Básico
Ranking de Corretas:	37
Ranking de Erradas:	6
Número de Utilizações:	43
Uso em Avaliação:	0
Uso em Exercício:	2
Tempo Médio de Solução:	5.05
Número Mínimo de Passos:	7
Número Máximo de Passos:	8
<input type="button" value="Excluir"/>	
<input type="button" value="Editar"/>	

Unisinos - Universidade do Vale do Rio dos Sinos - PIPCA

Fonte: Elaborado pelo Autor

Com base nas informações analisadas e coletadas é possível gerar a tabela contendo um resumo e visualizar de maneira sintética todo o processo da sua resolução. Na figura 35 é apresentado o resumo do OAH contendo as informações dos alunos referentes às questões não respondidas (legenda “N”) ou com as questões respondidas corretamente no tempo previsto ou abaixo (células em branco) ou com o tempo excedido (legenda “T”).

Figura 36 - Resumo da resolução do OAH

SAAPÍENS - Heráclito

OBAA
PARCERIA DE ENTÃO DAOS DE
QUESTOS DE APRENDIZAGEM

Resumo do O A H

Resumo -> Respostas Incorretas x Corretas com tempos Excedidos
1..10/10

Nrquestao	Cls	Aluno 1	Aluno 2	Aluno 3	Aluno 4	Aluno 5
1	Fundamental					
2	Fundamental					
3	Basica					
4	Basica					
5	Basica					
6	Intermed					T
7	Intermed	T	T	T	T	T
8	Intermed.		T		N	
9	Avançada		N	T	T	T
10	Avançada		N		N	T

Legenda T=Tempo excedido N= Não Resolvida

Novo Relatorio

Fonte : Elaborado pelo autor

Em relação à avaliação das questões propostas no OAH, o professor percebeu que as questões classificadas como fundamentais e básicas, tiveram sua conclusão na totalidade e dentro dos parâmetros estabelecidos inicialmente. Nas questões classificadas como intermediárias, que exigiam um pouco mais de raciocínio, ou que possuíam um nº maior de provas, apenas um aluno (o nº 4) não conseguiu resolver uma das questões, na demais todos resolveram. Uma observação constatada foi que uma das questões, classificadas como intermediária teve seu tempo de solução excedido, indicando que a mesma pode estar com seu tempo referencial subdimensionado e que a mesma sofrerá um ajuste no seu tempo na base dedados. Nas questões de nível avançado, considerada as mais difíceis, pois além de possuírem um maior número de passos na sua solução usam um número maior de regras de inferência. Verificou-se que apenas três alunos conseguiram resolvê-las, os outros alunos (nº2 e nº4) apenas o segundo conseguiu uma delas, porém com o tempo acima da média.

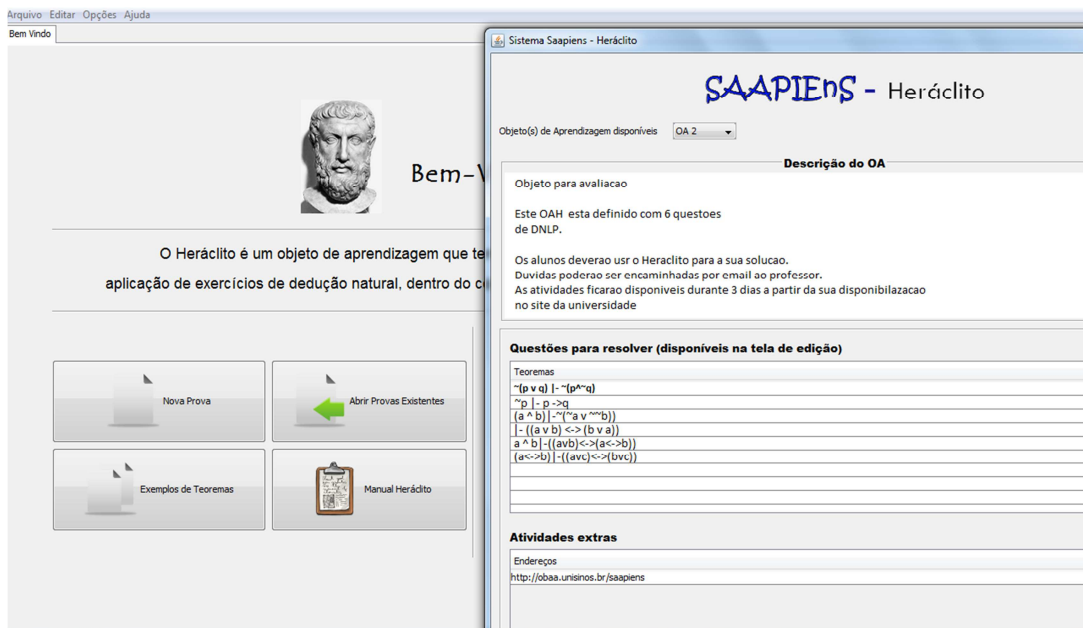
Por meio das consultas do módulo de análise, o professor pôde perceber que a maioria dos alunos teve facilidade no uso do OAH e na resolução dos exercícios. Nas questões não finalizadas, verificou-se que, conforme o tipo da questão e da sua resolução inferiu-se uma dificuldade maior no uso de determinadas regras de inferências, percebido através do tempo utilizado para resolver, o numero de tentativas erradas e por fim a desistência na sua solução.

6.1.4. Etapa de ação pedagógica

O acompanhamento do uso do OAH foi importante, pois forneceu informações ao professor para mapear um diagnóstico e desenvolver as ações pedagógicas necessárias visando melhorar seu processo de ensino-aprendizagem. Entre as possíveis ações, observou-se que alguns conteúdos específicos poderiam ser mais bem trabalhados em sala de aula.

Porém, outra ação importante está diretamente relacionada ao SAAPiEnS, envolvendo a criação de um novo OAH com conteúdos especialmente desenvolvidos aos alunos que apresentaram deficiência na resolução dos exercícios. Nesse OAH as questões foram elaboradas num grau crescente de dificuldade e com materiais de apoio abrangendo especificamente as dificuldades observadas no processo de acompanhamento, incluindo uma série de exemplos de resolução de argumentos (exemplos de provas) concluídos (ver Figura 35). Esse OAH também foi disponibilizado aos alunos com um prazo maior para sua.

Figura 37 Novo OAH gerado



Fonte :Elaborado pelo autor

6.3 Análise das avaliações

Os testes funcionais mostram que o SAAPiEnS está apto para atuar na criação e acompanhamento de objetos de aprendizagem para o ensino de dedução natural na Lógica. O cenário hipotético ilustrou como o uso das ferramentas do SAAPiEnS possibilitou ao professor construir um objeto de aprendizagem de uma forma simples e fácil e com recursos que permitiram acompanhar e avaliar o desempenho alcançado pelos alunos de uma maneira relevante. Isso ajudou na seleção das ações corretivas, como também permite prever situações futuras agindo pro-ativamente. Na medida em que a base das informações referente ao uso dos

OAHs vai sendo persistida, novas análises podem ser realizadas e novas conclusões inferidas. Será possível ao professor, estabelecer critérios no desenvolvimento de planos de ensino, adequado à realidade ao contexto dos seus alunos. As facilidades oferecidas pelo uso da ferramenta possibilitarão também o desenvolvimento dos objetos de aprendizagem customizados especificamente para um determinado fim, visando atingir determinado grupo de alunos, podendo assim contribuir de maneira qualitativa no progresso realizado pelos mesmos, no decorrer do semestre letivo.

7 CONSIDERAÇÕES FINAIS

Este capítulo tem por objetivo, apresentar as considerações finais e contribuições sobre o ferramenta desenvolvida denominada SAAPIEnS. Uma ferramenta de autoria de objetos de aprendizagem e acompanhamento pedagógico no ensino da Dedução Natural na Lógica de Predicados, que teve como um dos seus objetivos maiores, o de integrar-se à infraestrutura existente da MILOS (GLUZ e VICARI, 2010) como parte do Sistema de Autoria e do Sistema de Apoio Pedagógico provido pela plataforma. Além disso, serão apresentados os resultados alcançados e na seção trabalhos de futuros as sugestões a serem desenvolvidas visando a melhoria deste trabalho.

7.1 Conclusão

Essa dissertação teve durante o seu desenvolvimento, o propósito da criação uma ferramenta de autoria de objetos de aprendizagem voltados à um domínio específico, no caso a Dedução Natural na Lógica e também, uma ferramenta de acompanhamento no uso desses objetos, além de responder a uma questão de pesquisa proposta para esse trabalho.

Para realização dessa dissertação, um conjunto de objetivos foi definido, conforme descritos no Capítulo 2. Para atender esses objetivos, primeiro foi realizado uma pesquisa sobre o estado da arte apresentada no Capítulo 4. Os estudos serviram de base tecnológica, para a definição do modelo e construção do protótipo inicial da ferramenta de autoria e acompanha. Foram pesquisadas ferramentas de autoria disponíveis, para que fossem avaliadas e servissem de subsídios para identificar as contribuições do SAAPIENS, assim como verificar a existência do problema de pesquisa a ser investigado.

O protótipo inicial foi desenvolvido com base nessas especificações, visando atender não só requisitos tecnológicos estudados, assim como os requisitos pedagógicos. Inicialmente, foram implementadas funcionalidades na ferramenta de autoria e num segundo momento, as funcionalidades do sistema de acompanhamento.

Com a integração à infraestrutura da plataforma da MILOS, todos os OAs Heráclito criados, podem utilizar os serviços dos agentes pedagógicos disponíveis na plataforma e manter a compatibilidade com a proposta OBAA. Nesse caso os OAs Heráclito suportam um serviço tutor de ensino de demonstrações formais com recursos de acompanhamento pedagógico.

A validação funcional da ferramenta foi realizada em laboratório, por meio de experimentos, afim de avaliar os requisitos funcionais e demais recursos oferecidos na criação dos OAHs.

O processo de avaliação da usabilidade e do potencial educacional do SAAPIEnS ocorreu através de um cenário hipotético de uso onde se exemplifica como um professor poderia criar um OAH, disponibilizar este OAH aos alunos, acompanhar o desempenho dos alunos durante o processo de resolução dos exercícios do OAH e empreender ações pedagógicas, incluindo revisão e criação de um novo OAH para resolver problemas detectados com a ajuda da ferramenta de acompanhamento.

Todas as etapas desde o processo do seu planejamento, criação de um OAH, até a sua disponibilização foram cobertas no cenário. O processo de acompanhamento dos alunos na sua resolução, através dos recursos de análises dos dados e do processo de interação também foi coberto. Propor ações pedagógicas e desenvolver estratégias específicas visando auxiliar o aluno, na aquisição dos conhecimentos necessários no domínio dos conteúdos

desenvolvimentos também foi contemplado pelo cenário, tendo assim atendido plenamente os seus requisitos.

Portanto, conclui-se que o suporte do SAAPIENS a um processo integrado de autoria e acompanhamento de objetos de aprendizagem pode oferecer aos professores a possibilidade de melhorar seu processo de ensino em ambiente tecnológico. Isso pode levar ao desenvolvimento de novas competências pelos professores, tornando seu trabalho mais criativo e compensador. Também pode auxiliar no sucesso dos alunos na apropriação de novos saberes. Um resultado mais geral é que o trabalho contribui com mais evidências que o uso das tecnologias em contextos educacionais contribui de maneira substancial no processo de ensino-aprendizagem.

7.2 Contribuições

Durante as pesquisas empreendidas para o desenvolvimento da presente dissertação, foram elaborados dois artigos científicos submetidos a eventos nas áreas de Sistemas Multiagente (WESAAC) e da Informática na Educação (SBIE). O artigo submetido ao WESAAC (Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações) de 2014 foi aceito e já está publicado. Este artigo apresenta a ferramenta de autoria e o processo de acompanhamento com o uso da tecnologia multiagentes:

RODRIGUES, A M; GLUZ, Carlos J. SAAPIENS-Heráclito: A Tool for Authoring Learning Objects and Educational Support in Education in Natural Deduction in Propositional Logical 8º Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações (WESAAC) 2014 (ISSN 2177-2096)

RODRIGUES, A M; CAMPEZZATO M.G.; GLUZ, Carlos J. SAAPIENS-Heráclito: Uma ferramenta de Autoria de Objetos de Aprendizagem e Apoio Pedagógico na Dedução Natural na Lógica Proposicional. (O artigo foi publicado nos anais do SBIE (Simpósio Brasileiro de Informática na Educação) de 2014.

SCHWARTZ, G; RODRIGUES, A M; GLUZ, Carlos J. Modelo de Categorização de Mediação Baseado em Registros Padronizados de Interações e Atos da Fala. O artigo publicado nos anais do SBIE (Simpósio Brasileiro de Informática na Educação) de 2014.

7.3 Trabalhos futuros

Durante o desenvolvimento da ferramenta de autoria, percebeu-se a necessidade de criar novos recursos ao SAAPIENS, visando complementar o processo de avaliação e acompanhamento do desempenho dos alunos. Um dos recursos possíveis é o uso de ferramentas de mineração de dados educacionais através de análises previsoras, algoritmos de *clusterização* e de classificação incorporando-as à plataforma do SAAPIENS. Outra evolução possível é o suporte para a criação de OAH adaptativos baseados no perfil dos alunos, tanto individualmente, quanto em grupo. A integração do sistema de acompanhamento do SAAPIENS com o Moodle, que é um LMS amplamente empregado, também é uma extensão futura importante para o SAAPIENS.

Referências

- ADOBE - **Actionscript Technology Center**. Disponível em: <<http://www.adobe.com/devnet/actionscript.html/>>. Acesso em: 20 mai. 2014.
- BAKER, R. S. J. D.; YACEF, Kalina. The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining*, v. 1, n. 1, p. 3-17, 2009.
- BALL, Simon; TENNEY, Julian. *Xerte—a user-friendly tool for creating Accessible Learning Objects*. Springer Berlin Heidelberg, 2008.
- BELLIFEMINE, Fabio et. al. *JADE Programmer's Guide*. Tilab, 2010. Disponível em <http://jade.tilab.com/doc/programmersguide.pdf> Acesso em 12 mai. 2013.
- BORDINI, Rafael Heitor; VIEIRA, Renata. Linguagens de Programação Orientadas a Agentes: uma introdução baseada em AgentSpeak (L). **Revista de informática teórica e aplicada. Porto Alegre. Vol. 10, n. 1 (2003), p. 7-38**, 2003.
- BORDINI, Rafael Heitor; HÜBNER, Jomi Fred; VIEIRA, Renata. Introdução ao desenvolvimento de sistemas multiagentes com Jason. **XII Escola de Informática da SBC**, v. 2, p. 51-89, 2004.
- BORDINI, Rafael H.; HÜBNER, Jomi Fred; WOOLDRIDGE, Michael. **Programming multi-agent systems in AgentSpeak using Jason**. John Wiley & Sons, 2007.
- BRUEGGE, Bernd; DUTOIT, Allen H. *Object-Oriented Software Engineering Using UML, Patterns and Java-(Required)*. Prentice Hall, 2004.
- CARROLL, John M. Five reasons for scenario-based design. **Interacting with computers**, v. 13, n. 1, p. 43-60, 2000.
- CASTRO, J. F. B. de; ALENCAR, F. M. R.; SILVA, C. T. L. L. **Engenharia de Software Orientada a Agentes**. In: BREITMAN, K; ANIDO, R. (Eds). *Atualizações em Informática*. Rio de Janeiro: Capítulo 5, PUC-RIO, 2006.
- CASTRO, J; KOLP, M.; MYLOPOULOS, J. **Towards requirements-driven information systems engineering: The tropos project**. *Information Systems*, Reino Unido, v. 27, n.6, p. 365-389, set. 2002 .
- CETIC: Pesquisa sobre o uso das tecnologias de informação e comunicação no Brasil [livro eletrônico] : TIC Educação 2012 = **Survey on the use of information and communication technologies in Brazil : ICT Education 2012** – São Paulo : Comitê Gestor da Internet no Brasil, 2013. <http://www.cetic.br/publicações>. Acesso em: 10 jul.2014.
- FLORES, C.; SEIXAS, L.; GLUZ, J. C. ; VICARI, R. A Model of Pedagogical Negotiation. *Lecture Notes in Computer Science*, Heidelberg, v. 3808, n. 2005, p. 488-499, 2005.
- FIPA. Disponível em: <<http://www.fipa.org/>>. Acesso em: 12 jul. 2013.
- GADOTTI, Moacir. *Perspectivas atuais da educação. São Paulo em perspectiva*, v. 14, n. 2, p. 03-11, 2000.

GLUZ, J. C.; VICARI, R. M. MILOS: Infraestrutura de Agentes para Suporte a Objetos de Aprendizagem OBAA. Anais do XXI SBIE, 2010.

GLUZ, J. C. INTRODUÇÃO A INFRAESTRUTURA MILOS. Pós-Graduação em Computação Aplicada (PIPCA), Universidade do Vale do Rio dos Sinos (UNISINOS).

GLUZ, J.C.; PY, M. Lógica para Computação. Editora Unisinos, 2010.

GRUBER, T. WhatisanOntology? <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>(Acessado em 23 de junho de 2013)

GRUBER, Thomas R. et al. A translation approach to portable ontology specifications. **Knowledge acquisition**, v. 5, n. 2, p. 199-220, 1993.

GUARINO, N. (1998), “Formal Ontology and Information Systems”.Disponível em:<http://www.loa-cnr.it/Papers/FOIS98.pdf> Acesso em: 23 jun.2013

HEIKKI, Mannila, “Data mining: machine learning, statistics, and databases”, IEEE, 1996.

IEEE Learning Technology Standards Committee (LTSC). Draft Standard for Learning Object Metadata (IEEE 1484.12.1-2002). Julho de 2002. Disponível em:http://ltsc.ieee.org/doc/wg12/LOM_1484_12_1_v1_Final_Draft.pdf. Acesso em: 02 jul. 2013.

Jade - Java Agent Development Framework.Disponível em: <<http://jade.tilab.com/>>. Acessoem: 28 jun. 2013.

JENNINGS, N.; JENNINGS, N. R.; WOOLDRIDGE, M. J. **Agent technology: foundations, applications, and markets**. [S.l.]: Springer, 1998

MOORE, A. W. **Statistical Data Mining Tutorials**. Disponível em: <<http://www.autonlab.org/tutorials/>>. Acesso em: 9 jul. 2013.

MEC ,Ministerioda Educação e Cultura ,.Disponível em :http://portal.mec.gov.br/index.php?option=com_content&view=article&id=20101&Itemid=872 Acesso em : 28 jul.2014.

MOSSMAN, Marcel; GOMES, Lucas dos S.; GLUZ, João Carlos. Objetos de Aprendizagem Móveis para Ensino de Dedução Natural na Lógica Proposicional. In: **Anais do Simpósio Brasileiro de Informática na Educação**. 2012.

MYSQL, A. B. ‘MySQLDocumentation. Disponível por WWW em <http://www.mysql.com/doc>, 2014.

NIELSEN, J.; NORMAN,D.**Usability 101: Introduction to Usability**. Disponível em: <<http://www.nngroup.com/articles>>. Acesso em: 10 jun.2014.

ORACLE (2014a).Glassfish application server. <http://glassfish.java.net/>. Online; Acesso em 5 mar.2013.

ORACLE (2014b). Java EE at a glance.

<http://www.oracle.com/technetwork/java/javae/index.html>. Online; Acesso em 5 mar.2014.

ORACLE (2014c). Java Server Faces Technology. Disponível em:

<<http://docs.oracle.com/javae/7/tutorial/doc/jsf-intro.htm>> Acesso em 5 mar. 2014.

ORACLE(2014d). Java Persistence API. Disponível em:

<http://www.oracle.com/technetwork/java/javae/tech/persistence-jsp-140049.html>; Acesso em: 5 mar.2014.

OWL Web Ontology 2 Language Document Overview, <http://www.w3.org/TR/2009/REC-owl2-overview-20091027/> Acessado em 26/06/2013

OWL Web Ontology Language Overview. Disponível em: <<http://www.w3.org/TR/owl-features/>>. Acesso em: 16 jul. 2013.

PARACK, Suhem; ZAHID, Zain; MERCHANT, Fatima. Application of data mining in educational databases for predicting academic trends and patterns. In: Technology Enhanced Education (ICTEE), 2012 IEEE International Conference on. IEEE, 2012. p. 1-4.

PASSERINO, L.; GLUZ, J.; VICARI, R. MEDIATEC Mediação Tecnológica em Espaços Virtuais para Apoio ao Professor Online. *RENTE. Revista Novas Tecnologias na Educação*, v. 5, p. 1, 2007.

PENTEADO, Fabiane; GLUZ, João Carlos. Sistema Heráclito: Suporte a Objetos de Aprendizagem Interativos e Dialéticos Voltados ao Ensino de Dedução Natural na Lógica Proposicional. In: **Anais do Simpósio Brasileiro de Informática na Educação**. 2011.

PENTEADO, Fabiane Flores. **Agente pedagógico para mediação do processo de ensino-aprendizagem da dedução natural na lógica**. 2013. 127 fDissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, RS, 2013

RAO, Anand S. et al. BDI agentes: da teoria à prática. In: **ICMAS** . 1995. p.312-319.

ROMERO, Cristobal; VENTURA, Sebastian. Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, v. 33, n. 1, p. 135-146, 2007.

ROMERO, C.; ESPEJO, P. G.; ZAFRA, A.; ROMERO, J. R.; VENTURA, S. Web usage mining for predicting final marks of students that use Moodle courses. **Computer Applications in Engineering Education**, [S.l.], Mar 2013. v. 21, n. 1, p. 135–146. ISSN 10613773. . Acesso em: 7 jul. 2013.

ROMERO-ZALDIVAR, Vicente-Arturo et al. Monitoring student progress using virtual appliances: A case study. **Computers & Education**, v. 58, n. 4, p. 1058-1067, 2012.

RUSSELL, Stuart Jonathan ; NORVIG , P; et al. **Artificial intelligence: a modern approach**. EnglewoodCliffs: Prentice hall, 1995.

SANTOS, A Recursos Educacionais Abertos no Brasil : [livro eletrônico] : o estado da arte, desafios e perspectivas para o desenvolvimento e inovação / Andreia Inamorato dos Santos ; [tradução DB Comunicação]. — São Paulo : Comitê Gestor da Internet no Brasil, 2013.

STONEBRAKER, Michael; Kemnitz, Greg. A próxima geração de sistema de gestão de dados POSTGRES. **Communications da ACM**, V. 34, n. 10, p.78-92, 1991.

TAN, Pan-Ning. STEINBACH, Micheal ; KUMAR, Vipin. Introdução ao Data Mining - Mineração de Dados. Ed. Ciência Moderna, 2009

VICCARI, R.; GLUZ, J.; PASSERINO, L.; et al. The OBAA Proposal for Learning Objects Supported by Agents. Procs. Of MASEIE Workshop – AAMAS 2010, Toronto, Canada, 2010.

VICCARI, R.; GLUZ, J.; SANTOS, E.; et al. Projeto OBAA – Rel. Técnico RTOBAA01 – Proposta de Padrão para Metadados de Objetos de Aprendizagem Multiplataforma.

UFRGS/CINTED, 2009. Disponível

em: <<http://www.portalobaa.org/obaac/padraoobaa/relatoriostecnicos/RTOBAA01.pdf/view>>.

UML. Documents Associated With UML Version 2.1.2. Disponível

em: <<http://www.omg.org/spec/UML/2.1.2/>>. Acesso em: jun. 2013

WEISS, G. Multiagent systems: A modern approach to distributed artificial intelligence. London: MIT Press, 2000.

Wiley, D. A. (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley (Ed.), *The Instructional Use of Learning Objects: Online Version*. Retrieved MONTH DAY, YEAR, from the World Wide Web: <http://reusability.org/read/chapters/wiley.doc>

W3C. World Wide Web Consortium (W3C). Disponível em: <<http://www.w3.org/>>.

Acesso em: 09 jun. 2014

WOOLDRIDGE, M. An Introduction to Multiagent Systems. John Wiley & Sons, 2009.

YU, E. '**Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering**' Proceedings of the 3rd IEEE Int. Symp. on Requirements Engineering (RE'97) Jan. 6-8, 1997, Washington D.C., USA. pp. 226-235 (acessado em 09/07/13)

YU, E ; MAO, X: **Organizational and Social Concepts in Agent-Oriented Software Engineering**. in: AOSE-2004, New York City, New York. (acessado em 9/07/13)

YU, E. Agent orientation as a modelling paradigm. **Wirtschaftsinformatik**, [S.l.], 2001. v. 43, n. 2, p. 123–132. . Acesso em: 15 jul. 2013.