

Um Ambiente Integrado de Apoio ao Aprendizado de Programação

Wellington Ricardo Pinheiro

Orientadora: Prof^a. Dr^a. Leliane Nunes de Barros

Co-orientador: Prof. Dr. Fabio Kon

Departamento de Ciência da Computação
Instituto de Matemática e Estatística - Universidade São Paulo

wrp@ime.usp.br

3 de setembro de 2007

Conteúdo

- 1 Parte I - Introdução
- 2 Parte II - Ambiente de Apoio ao Aprendizado de Programação
- 3 Parte III - Arquitetura do AAAP

Terminologia

Aprendizado Eletrônico (*e-learning*, AE)

É uma forma de proporcionar o aprendizado, treinamento ou programas educacionais através de meios eletrônicos.

Terminologia

Aprendizado Eletrônico (*e-learning*, AE)

É uma forma de proporcionar o aprendizado, treinamento ou programas educacionais através de meios eletrônicos.

Sistemas de Aprendizado Eletrônico (SAE)

São sistemas que disponibilizam um ambiente colaborativo, através de um conjunto de ferramentas como: fórum, *chat*, mensagem instantânea, e-mail, entre outras.

Pilares do Aprendizado Eletrônico



- Aspectos institucionais: referem-se à incorporação do Aprendizado Eletrônico na instituição
- Aspectos tecnológicos: como são tratadas questões de conectividade, segurança, facilidade de manutenção e personalização, entre outras
- Aspectos pedagógicos: qual a melhor forma de proporcionar o aprendizado utilizando a tecnologia disponível

Classificações

Sistemas de Aprendizado Eletrônico são classificados em:

- *Learning Management Systems* (LMS): permitem a *disponibilização de conteúdo* de aprendizagem e uso de ferramentas colaborativas
- *Learning Content Management Systems* (LCMS): permitem a *criação do conteúdo*, para uso posterior em sistemas LMSs

Exemplos: ATutor, WebCT (agora Blackboard), WebAula, TelEduc, Moodle, Blackboard, Sakai, Tidia-Ae.

Sistemas do tipo LMS podem ser extensões de sistemas do tipo LCMS, e vice-versa.

Objetivos dos SAEs

Alguns dos principais objetivos dos SAEs são:

- Permitir a colaboração entre alunos e professores
 - através de ferramentas como fórum, *chat*, mensagem, etc.
- Prover um ambiente adequado para a disseminação do conhecimento
- Facilitar a administração das disciplinas
- Controlar a evolução do aluno em atividades de aprendizado
- Aplicação e gerenciamento de atividades *on-line*

Objetivos dos SAEs

Qual o suporte dado pelos SAEs para a parte prática de certas disciplinas?

Objetivos dos SAEs

Qual o suporte dado pelos SAEs para a parte prática de certas disciplinas?

Em geral, estão fora do escopo dos SAEs:

- Tratar disciplinas de forma particularizada
- Oferecer suporte à parte prática (alguns SAEs podem oferecer suporte limitado)

O professor é responsável por decidir quais ferramentas específicas deverão ser utilizadas.

Áreas e Ferramentas

Ferramentas de suporte para o ensino/aprendizagem em diferentes áreas do conhecimento

Área	Ferramenta	Funcionalidades
Arquitetura	CAD	Construção de plantas e ambientes, simulações e visualizações em diferentes perspectivas.
Matemática	Mapple, Mathematica, Matblab, etc.	Criação de gráficos, descrição de fórmulas e resolução automática de problemas.
Engenharia	CASEs	Simulações, validações e resolução de problemas.

Justificativa

A área de programação também não possui um ambiente de desenvolvimento que seja integrado com os SAEs.

Justificativa

A área de programação também não possui um ambiente de desenvolvimento que seja integrado com os SAEs.

Quem seria beneficiado por essa integração?

Justificativa

A área de programação também não possui um ambiente de desenvolvimento que seja integrado com os SAEs.

Quem seria beneficiado por essa integração?

- Professores
- Alunos novatos
- Alunos experientes
- Monitores

Justificativa

Quais os benefícios dessa integração?

- Professor e aluno utilizam um ambiente de programação compatível
- Facilidade de instalação e configuração do ambiente (alunos novatos)
- Agilidade em certas tarefas do aluno:
 - obtenção e submissão de atividades de programação, uso de testes pré-definidos, auxílio no aprendizado de programação e acesso à certas documentações disponibilizadas em repositórios.
- Agilidade em certas tarefas do professor e/ou monitor:
 - uso de corretores automáticos de programas, uso de repositórios de testes para fazer a aceitação de atividades, possibilidade de indicar quais ferramentas o aluno deverá utilizar para o desenvolvimento de atividades.

Ambiente de Apoio ao Aprendizado de Programação (AAAP)

Características Gerais

O ambiente de ensino de programação terá como características:

- Servir como um ambiente comum, entre alunos e professores, para o desenvolvimento de programas
- Permitir uma integração com outras ferramentas dos SAEs
- Facilitar tarefas como: disponibilização e entrega de atividades
- Permitir o uso de ferramentas adequadas para o desenvolvimento de atividades
- Disponibilizar ferramentas de auxílio para correção de atividades
- Disponibilizar para o aluno uma ferramenta para auxiliá-lo no aprendizado de programação

Pré-Requisitos

Os seguintes itens são pré-requisitos no ambiente:

- Ser funcional e prático
- Possibilidade de extensão e integração (com os SAEs)
- Ser personalizável

Pré-Requisitos

Os seguintes itens são pré-requisitos no ambiente:

- Ser funcional e prático
- Possibilidade de extensão e integração (com os SAEs)
- Ser personalizável

Esses pré-requisitos são importantes pois causam impacto direto na aceitação do ambiente.

Proposta do AAAP

Criar o ambiente de apoio ao aprendizado de programação, constituído de dois outros ambientes:

- Ambiente do Usuário: onde estarão as componentes responsáveis pelas atividades de programação e atividades relacionadas
- Ambiente do SAE: conterá as componentes responsáveis pela manipulação de informações contidas no SAE, que irão suportar as funcionalidades requeridas pelo Ambiente do Usuário

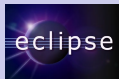
Soluções

Ambiente do Usuário

Soluções

Ambiente do Usuário

Utilizar o Eclipse como plataforma de execução das componentes do ambiente.

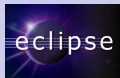


Ambiente do SAE

Soluções

Ambiente do Usuário

Utilizar o Eclipse como plataforma de execução das componentes do ambiente.



Ambiente do SAE

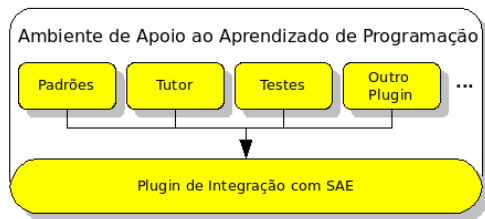
...aqui a questão é mais complicada...

Ferramentas no Ambiente do Usuário

As seguintes ferramentas serão construídas:

- Integração com o SAE
- Testes
- Padrões de Programação¹
- Tutor Inteligente

As ferramentas no ambiente do usuário serão construídas como *plugins* do Eclipse.



¹Padrões de Projeto, Arquiteturais e Elementares.

Ferramentas no Ambiente do SAE

O ambiente do SAE também deve conter algumas ferramentas com a finalidade de:

- Recuperar e devolver informações requisitadas pelo ambiente do usuário
- Complementar as funcionalidades do ambiente do usuário
- Configurar e executar os corretores automáticos de programas

Ferramenta de Integração com o SAE

Funcionalidades Principais:

- Visualizar a lista de atividades de programação do aluno
- Obter e instalar atividades de programação
- Prover um ambiente adequado para o desenvolvimento da atividade de programação
- Submeter as atividades de programação
- Servir de interface para acesso ao SAE, para outras ferramentas do ambiente

Ferramenta de Testes

Funcionalidades Principais:

- Manter repositório de testes para atividades
- Permitir criação de testes de unidade tanto pelo professor quanto para o aluno
- Permitir criação de testes de aceitação pelo professor
- Geração de relatórios de execução de testes
- Servir como base para um corretor automático de programas

Ferramenta de Padrões

Funcionalidades Principais:

- Busca de padrões de programação em repositórios
- Acesso à documentação dos padrões de programação
- Obtenção de exemplos (em várias linguagens)
- Inferência de padrões utilizados no código do desenvolvedor

Ferramenta de Correção Automática de Programas

Funcionalidades Principais:

- Permitir que o corretor seja executado no momento que o professor desejar
- Utilizar o corretor para fazer a aceitação de atividades (durante a submissão)
- Definir os testes que serão utilizados para execução no corretor (uso de repositórios de testes)
- Usar o resultado da execução do corretor para compor a nota

Ferramenta de Tutor

ProPAT: Tutor de Padrões Elementares.

- Plugin de Tutor para o Eclipse
- Utiliza casos de testes para verificação
- Utiliza Diagnóstico Baseado em Modelos (modelos corretos)
- Funciona para subconjunto de instruções da linguagem C

Ferramenta de Tutor

ProPAT: Tutor de Padrões Elementares.

- Plugin de Tutor para o Eclipse
- Utiliza casos de testes para verificação
- Utiliza Diagnóstico Baseado em Modelos (modelos corretos)
- Funciona para subconjunto de instruções da linguagem C

O objetivo do trabalho será ampliar o ProPAT para:

- Funcionar também com a linguagem Java
- Incorporar outras instruções da linguagem C
- Utilizar modelo de falhas típicas de programação

Considerações Arquiteturais

Características Arquiteturais dos SAEs.

Os SAEs utilizam, basicamente, dois modelos arquiteturais:

- Baseado em Módulos: as ferramentas são módulos inteiros acoplados em um núcleo do sistema
- Baseado em Serviços: as ferramentas disponibilizam e utilizam serviços (*Service Oriented Architecture* - SOA) de outras ferramentas

Características Arquiteturais dos SAEs.

Os SAEs utilizam, basicamente, dois modelos arquiteturais:

- Baseado em Módulos: as ferramentas são módulos inteiros acoplados em um núcleo do sistema
- Baseado em Serviços: as ferramentas disponibilizam e utilizam serviços (*Service Oriented Architecture* - SOA) de outras ferramentas

Os SAEs que utilizam o modelo SOA são: Tidia-Ae e o Sakai.

Para o AAAP, optamos por utilizar um SAE baseado em serviços.

OSIDs

Open Service Interface Definitions (OSIDs)

Contratos entre fornecedores e consumidores de serviços.



OSIDs

Os objetivos das OSIDs são:

- Desenvolvida pela *Open Knowledge Initiative* (OKI)
- Permitir independência das ferramentas em relação ao sistema de Aprendizado Eletrônico
- Composta de especificação, definição e implementação de interfaces (*binding*)
- A OKI é responsável pelas especificações

Exemplos de OSIDs

Entre as OSIDs disponíveis, estão:

- *Agent*: representação de indivíduos ou processos que chamam certos tipos de serviços
- *Assessment*: responsável por tarefas como: administração, organização, avaliação e armazenamento de provas e atividades
- *Scheduling*: gerenciamento de eventos em calendários compartilhados
- *User Messaging*: permite a comunicação e notificação entre usuários

A lista completa de OSIDs disponíveis pode ser encontrada em:

<http://plectrudis.mit.edu/okicomunity/staticpages/index.php/theOSIDs>

Interface Java para a OSID Agent

```
package org.osid.agent;

public interface Agent extends java.io.Serializable {
    public String getDisplayName() throws org.osid.agent.AgentException;

    public org.osid.shared.Id getId() throws org.osid.agent.AgentException;

    public org.osid.shared.Type getType() throws org.osid.agent.AgentException;

    org.osid.shared.Properties getPropertiesByType(
        org.osid.shared.Type propertiesType)
        throws org.osid.agent.AgentException;

    org.osid.shared.TypeIterator getPropertyTypes()
        throws org.osid.agent.AgentException;

    org.osid.shared.PropertiesIterator getProperties()
        throws org.osid.agent.AgentException;
}
```

Interface PHP para a OSID Agent

```

<?php

class Agent {
    function getDisplayName () {
        die ("Method ".__FUNCTION__."() declared in interface ".__CLASS__."
" has not been overloaded in a child class.");
    }
    function &getId () {
        die ("Method".__FUNCTION__."() declared in interface ".__CLASS__."
" has not been overloaded in a child class.");
    }
    function &getType () {
        die ("Method".__FUNCTION__."() declared in interface ".__CLASS__."
" has not been overloaded in a child class.");
    }
    function &getPropertiesByType ( &$propertiesType ) {
        die ("Method".__FUNCTION__."() declared in interface ".__CLASS__."
" has not been overloaded in a child class.");
    }
    function &getPropertyTypes () {
        die ("Method".__FUNCTION__."() declared in interface ".__CLASS__."
" has not been overloaded in a child class.");
    }
    function &getProperties () {
        die ("Method".__FUNCTION__."() declared in interface ".__CLASS__."
" has not been overloaded in a child class.");
    }
}
?>

```

SAEs baseados em Serviços

Tidia-Ae

- Projeto financiado pela FAPESP
- Tem como objetivo o ensino à distância utilizando redes de altas velocidades
- Desenvolvido por várias instituições de ensino do estado de São Paulo
- Tem planejado o uso das OSIDs a médio prazo (± 6 meses?)

SAEs baseados em Serviços

Tidia-Ae

- Projeto financiado pela FAPESP
- Tem como objetivo o ensino à distância utilizando redes de altas velocidades
- Desenvolvido por várias instituições de ensino do estado de São Paulo
- Tem planejado o uso das OSIDs a médio prazo (± 6 meses?)

Sakai

- É uma iniciativa internacional (teve início no MIT)
- Possui desenvolvedores no mundo inteiro
- É um produto mais maduro que o Tidia-Ae
- Ainda usa poucas OSIDs
- Parte do comitê do Sakai é responsável pela definição das OSIDs!

Soluções

Voltando à questão...

Ambiente do Usuário

Utilizar o Eclipse como plataforma de execução das componentes do ambiente.



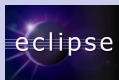
Ambiente do SAE

Soluções

Voltando à questão...

Ambiente do Usuário

Utilizar o Eclipse como plataforma de execução das componentes do ambiente.



Ambiente do SAE

Utilizar o Sakai para a construção de nossas ferramentas, tomando cuidado de usar as OSIDs sempre que possível.

Características Arquiteturais

Quanto à distribuição dos ambientes do AAP:

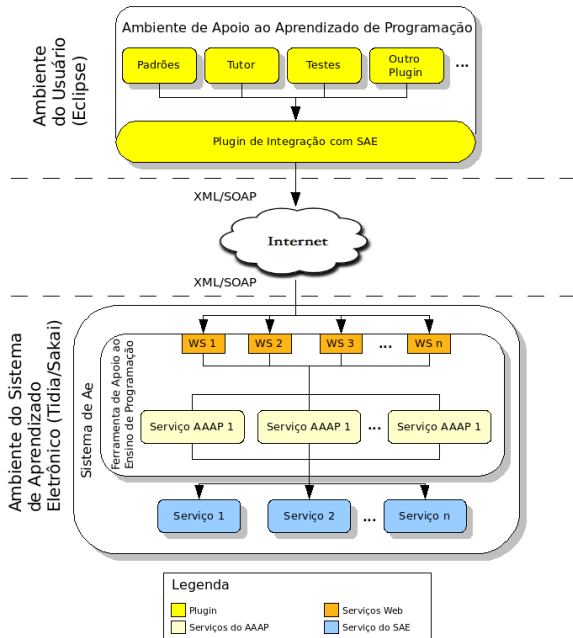
- O ambiente do usuário e ambiente do SAE poderão ser executados em máquinas distintas.

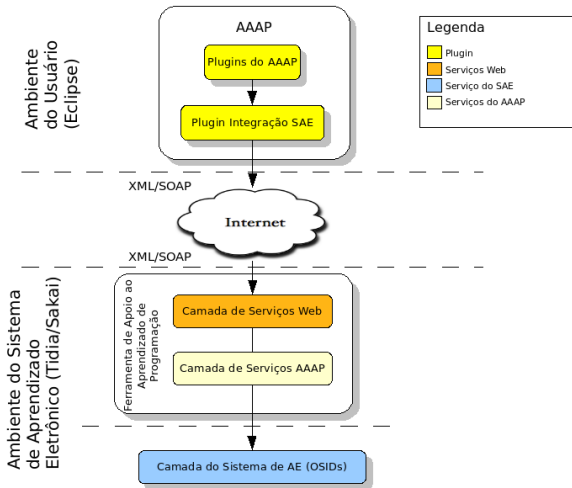


Características Arquiteturais

A arquitetura do AAAP e das ferramentas que serão desenvolvidas deverão ter as características:

- Construção de ferramentas baseadas em serviços (SOA)
- Fortemente baseadas nas OSIDs
- Uso do *plugin* de integração com o SAE para garantir isolamento entre os ambientes
- Uso de Serviços Web (*Web Services*) para integração entre ambiente do usuário e ambiente do SAE
- Criação de mecanismos para isolar o acesso à APIs proprietárias do SAE
- Interface do usuário na Web, para tarefas não disponibilizadas no ambiente do usuário





Ferramentas Relacionadas

- Ferramentas para Testes:
 - JUnit
 - JExercise
- Ferramentas para Correção Automática:
 - Web-Cat
- Ferramenta para Padrões:
 - WebOfPatterns
- Ferramenta de Tutor:
 - ProPAT

Resumo

- Sobre os Sistemas de Aprendizado Eletrônico:
 - Não faz parte de seus escopos o suporte à parte prática das disciplinas
 - Não dão suporte específico por disciplinas

Resumo

- Sobre os Sistemas de Aprendizado Eletrônico:
 - Não faz parte de seus escopos o suporte à parte prática das disciplinas
 - Não dão suporte específico por disciplinas
- Sobre a parte prática das disciplinas:
 - É interessante usar uma ferramenta/ambiente que suporte, de forma adequada, essa parte prática das disciplinas
 - É interessante que essa ferramenta/ambiente possa interagir com os Sistemas de Aprendizado Eletrônico

Resumo

- Sobre os Sistemas de Aprendizado Eletrônico:
 - Não faz parte de seus escopos o suporte à parte prática das disciplinas
 - Não dão suporte específico por disciplinas
- Sobre a parte prática das disciplinas:
 - É interessante usar uma ferramenta/ambiente que suporte, de forma adequada, essa parte prática das disciplinas
 - É interessante que essa ferramenta/ambiente possa interagir com os Sistemas de Aprendizado Eletrônico
- Sobre as questões tecnológicas:
 - É importante acompanhar evolução ocorrida nos SAEs
 - A arquitetura de serviços, baseada em OSIDs, parece ser uma escolha correta
 - Os SAEs atuais não implementam completamente as OSIDs
 - O Tidia-Ae é a plataforma final, porém, no estágio atual, ele ainda não pode ser utilizado como plataforma para desenvolvimento
 - Deverão ser utilizados mecanismos de isolamento para o acesso às APIs do Sakai

Referências

Sakai: <http://sakaiproject.org>

Tidia: <http://tidia-ae.incubadora.fapesp.br/portal>

OKI e OSIDs: <http://www.okiproject.org>

Eclipse: <http://www.eclipse.org>

JUnit: <http://www.junit.org/index.htm>

Web-Cat: <http://web-cat.cs.vt.edu/grader>

JExercise:

<http://www.cs.mcgill.ca/~martin/etx2006/papers/9.pdf>

WebOfPatterns: <http://www-ist.massey.ac.nz/wop>

ProPAT: <http://www.ime.usp.br/~articuno/eclipse>