
FA_PorT: Um Framework para Sistemas Portfólio-Tutor utilizando Agentes

Fábio Nicácio de Medeiros, Flávio Mota Medeiros, Arturo Hernández Domínguez

Instituto de Computação – Universidade Federal de Alagoas (UFAL)
Campus A. C. Simões, BR 104 – Norte, Km 97 – Maceió – AL – Brasil – Tel: 32141364

{fabioniacio, flaviommedeiros}@gmail.com, arturo@tci.ufal.br

Abstract. *In the context of authentic assessment, this work proposes a framework for portfolio-tutoring systems (FA_PorT), based on ITS (Intelligent Tutoring System) and agents. An electronic portfolio represents a tool to record and maintain significative documentation about learning of learners. The framework proposed, aims to reuse of internal architecture (or skeleton) of a portfolio-tutoring system based on ITS and Agents. In this way, development time of a new portfolio-tutoring system will be reduced and facilitated. A specification of framework for portfolio-tutoring systems based on components, agents and ITS, was developed and is presented in this paper. FA_PorT is being implemented using java language.*

Keywords: *Framework, Electronic Portfolio, Intelligent Tutoring System and Agents.*

Resumo. *No contexto de aplicação de uma avaliação mais autêntica, a proposta deste trabalho é um framework para sistemas portfólio-tutor (FA_PorT), baseado em STI (Sistema Tutor Inteligente) e agentes. Um portfólio eletrônico representa uma ferramenta para registrar e manter documentação e informação significativa sobre o aprendizado dos alunos. O framework proposto, objetiva o reaproveitamento da estrutura interna (núcleo comum) de um sistema portfólio-tutor baseado em STI e Agentes. Desta forma, o tempo de desenvolvimento de um novo sistema portfólio-tutor será reduzido e facilitado. Uma especificação do framework para sistemas portfólio-tutor baseado em componentes, agentes e STI, foi realizada e é apresentada neste artigo. O FA_PorT está sendo implementado utilizando a linguagem java.*

Palavras-Chave: *Framework, Portfólio Eletrônico, Sistema Tutor Inteligente e Agentes.*

1. Introdução

Atualmente, há um incrível número de instituições que necessitam de uma nova forma para a avaliação do corpo discente [Eportconsortium 2006]. No contexto de uma avaliação autêntica para o aluno, a abordagem portfólio eletrônico permite o registro histórico do progresso, desempenho e atividades didáticas [Nascimento 2002], contendo as informações e documentações que são cada vez mais significativas sobre o trabalho e o processo de aprendizagem do aluno como a evolução de suas habilidades [Eportconsortium 2006]. Tratando-se de um elemento de apoio na avaliação do aluno, o portfólio eletrônico permite identificar, em um meio significativo, trabalhos

ou artefatos construídos e associados ao progresso e habilidades desenvolvidas, isto é possível devido as várias atividades didáticas realizadas durante um ou vários períodos letivos [Nascimento 2002].

No portfólio eletrônico vários artefatos de avaliação são usados [Nascimento 2002], [Sistêlos 1999]: projetos, seminários, exercícios, desenvolvimento de atividades didáticas em grupos, etc. Tais artefatos produzem uma grande quantidade de informação que deve ser analisada detalhadamente. A partir disto é possível ter informações precisas sobre o progresso, a performance e o acompanhamento do aluno.

O objetivo deste trabalho é a construção de um framework [Fayad et al. 1999] para sistemas portfólio-tutor baseados em STI (Sistema Tutor Inteligente) [Hatzilygeroudis and Prentzas 2004] [Wenger 1987], fornecendo uma estrutura interna (esqueletos de aplicações) para criação de sistemas portfólio-tutor, que podem ser personalizadas às necessidades específicas de uma aplicação e utilizados on-line via internet. Cada sistema portfólio-tutor construído pelo framework pode ser usado no contexto de Educação a Distância.

2. Sistemas Portfólio-Tutor

Nas referências sobre trabalhos de sistemas portfólios eletrônicos e sistemas tutores inteligentes, existem vários tipos de sistemas desenvolvidos como TecLec [Costa et al. 1998], eFolios Project [Eportconsortium 2006], Connecticut College e-Portfolio [Eportconsortium 2006], Iwebfolio [Eportconsortium 2006], Folio-ePortaro [Eportconsortium 2006], o POETA [Sistêlos 1999], o Portfólio-Tutor [Nascimento 2002] e o Portfólio Eletrônico para Web [Silva 2002]. Poucos sistemas foram baseados no reuso da estrutura (núcleo comum) e em componentes para a construção de um novo sistema portfólio eletrônico, por exemplo o framework Sakai [Ellis and Coppola 2006]. Em relação aos sistemas portfólios eletrônicos e tutores inteligentes existentes, o framework proposto neste artigo tem como principais contribuições o reuso da arquitetura (núcleo comum) do sistema portfólio eletrônico baseado em componentes e agentes associados aos princípios de STI [Hatzilygeroudis and Prentzas 2004] [Wenger 1987], e a redução de esforços diminuindo o tempo de desenvolvimento de um novo sistema portfólio eletrônico acoplado a uma camada tutor, adotando os princípios de STI.

3. Especificação do Framework para sistemas Portfólio-Tutor baseado em Componentes, STI e Agentes

A especificação do FA_PorT é representada pela identificação dos componentes, arquitetura do framework e construção de aplicações.

3.1. Identificação dos Componentes

O framework FA_PorT foi criado a partir da análise de três sistemas concretos, dois portfólios eletrônicos e um portfólio-tutor.

1. O sistema POETA (Portfólio Eletrônico Temporal e Ativo) [Sistêlos 1999] é baseado nos princípios de banco de dados temporal e ativo, considerando a recuperação dos dados de forma pró-ativa (administrada por eventos temporais) a partir de uma camada de armazenamento para poder gerar relatórios (individu-

- ais ou em grupos), gráficos e mensagens sobre a performance e o progresso na aprendizagem;
2. Um portfólio eletrônico para web [Silva 2002] baseado também nos princípios de banco de dados temporal e ativo, foi desenvolvido para realizar um registro de artefatos e informações sobre o progresso e a performance de alunos associados a disciplina de programação orientada a objetos;
 3. O sistema Portfólio-Tutor [Nascimento 2002] é representado por duas camadas acopladas, baseado nos princípios do portfólio eletrônico (camada portfólio) e sistema tutor inteligente (camada tutor) (Figura 1).

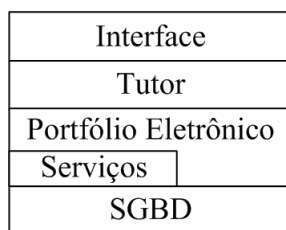


Figura 1. Arquitetura de um sistema Portfólio-Tutor

Durante a análise do domínio da camada portfólio eletrônico, vários componentes foram identificados: interface do usuário, atividades, disciplina, curso e conexão com o banco de dados.

A especificação da camada tutor do FA_PorT foi baseada na arquitetura de um sistema tutor inteligente em que são considerados como principais módulos [Hatzilygeroudis and Prentzas 2004], [Viccari and Giraffa 2003]: o modelo do domínio, o modelo do aluno, as estratégias pedagógicas, a interface e o controle.

Uma modelagem orientada a agentes foi utilizada para especificar as camadas tutor e portfólio. Essa modelagem foi necessária para mostrar flexibilidade, entidades pró-ativas, comportamento autônomo para solução de problemas e estrutura organizacional complexa. A metodologia orientada a agentes para especificar as camadas tutor e portfólio utilizada foi a GAIA, proposta por Wooldrige [Wooldrige et al. 2000]. Na metodologia GAIA, um sistema é representado como uma organização computacional que consiste na identificação e interação de papéis. Foram especificados, nas fases de análise e projeto os seguintes modelos: modelo de papéis (fase de análise), modelo de interação (fase de análise), modelo de agentes (fase de projeto) e modelo de comunicação (fase de projeto). Desta maneira, os vários agentes associados as camadas tutor e portfólio são identificados e especificados nas Figuras 2 e 3.

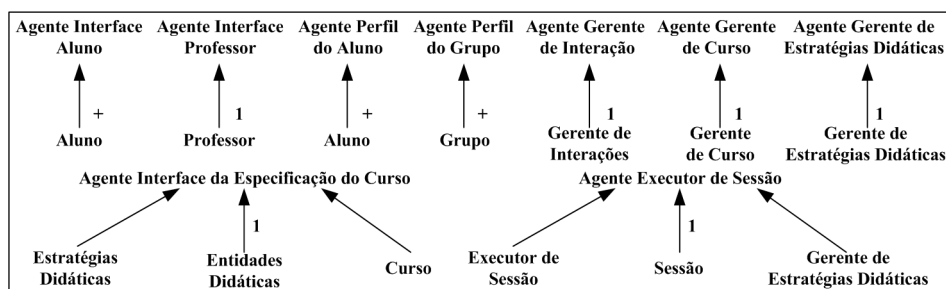


Figura 2. Modelo de Agentes

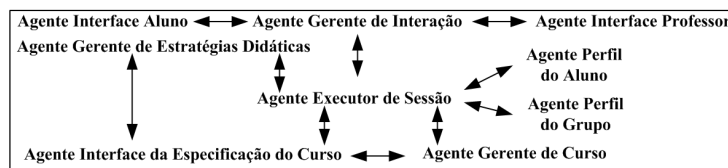


Figura 3. Modelo de Comunicação

3.2. Arquitetura do Framework

Apresenta-se a arquitetura do Framework (Figura 4) e as camadas associadas a cada aplicação construída a partir do mesmo, em que os elementos das camadas (Interface, Agentes, Tutor, Portfólio Eletrônico e Serviços) de uma nova aplicação portfólio-tutor são representadas através de um conjunto de componentes organizados da seguinte maneira:

- Componente_Cli - Representa o componente *i* da camada de Interface;
- Componente_CAj - Representa o componente *j* da camada de Agentes;
- Componente_CTk - Representa o componente *k* da camada de Tutor;
- Componente_CPl - Representa o componente *l* da camada de Portfólio Eletrônico;
- Componente_CSm - Representa o componente *m* da camada de Serviços.

Cada componente possui sua interface. A classe Framework possui atributos representando as interfaces dos componentes do framework e métodos (o método template que representa o controle e os pontos de adaptação de código). As aplicações são construídas considerando as camadas de um sistema portfólio-tutor (Interface, Agentes, Tutor, Portfólio e Serviços).

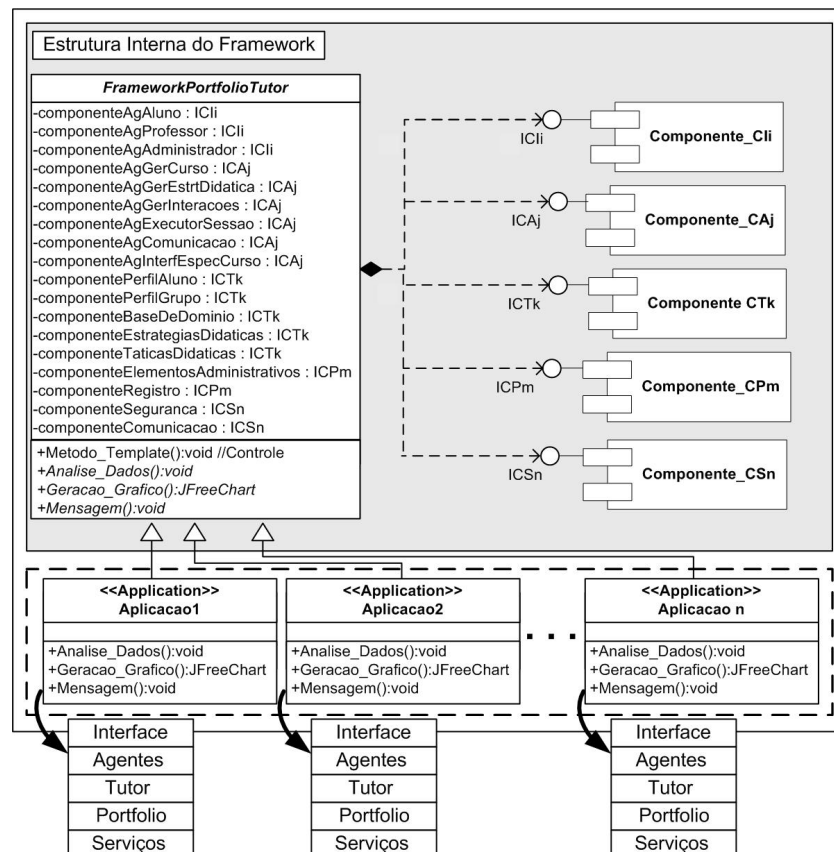


Figura 4. Arquitetura do framework FA_PorT para sistemas Portfólio-Tutor

• Especificação de Componentes

Para cada componente interno do framework foi especificado a descrição da funcionalidade, a interface do componente, as operações da interface e um diagrama de classes do componente. Como exemplo de especificação de um componente do framework, o agente gerente de estratégia didática está ilustrado na Figura 5.

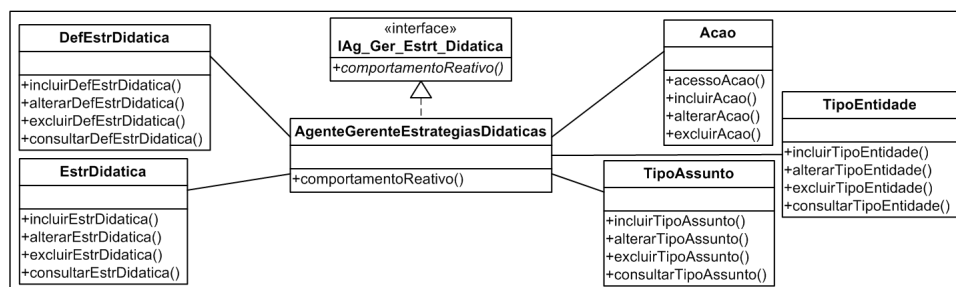


Figura 5. Diagrama de classes do componente agente gerente de estratégias didáticas do framework FA_PorT

3.2.1. Funcionamento de um novo Sistema Portfólio-Tutor

Um novo sistema portfólio-tutor é construído a partir do uso do framework FA_PorT. Cada novo sistema portfólio-tutor (ou aplicação) terá um funcionamento pré-definido, este é especificado no framework através do método *template*¹. Este método no contexto do FA_PorT representa o controle ou funcionalidades de cada sistema portfólio-tutor.

• Sessão de ensino on-line

Uma sessão de ensino on-line associada a um novo sistema portfólio-tutor é especificada para a aprendizagem de um grupo virtual específico de alunos e é iniciada quando a camada tutor inicia uma estratégia [Silva 2000]. Esta é especificada e definida passo a passo, utilizando as seguintes táticas de ensino [Silva 2000]: tática de reuso, tática de chat, tática de regras - *if condição then ação*, tática de mudança de estratégia, tática de envio de informações (associadas ao módulo de domínio) recuperadas da camada portfólio.

As principais funcionalidades da camada portfólio eletrônico no sistema portfólio-tutor são [Nascimento 2002] analisar o progresso do aluno; identificar e armazenar os trabalhos elaborados por cada aluno; registrar os resultados (avaliação e comentários do professor); fornecer suporte para o monitoramento da performance dos alunos e dos grupos; gerar mensagens de forma automática através de e-mail para informar (e lembrar), por exemplo de datas de atividades que serão realizadas e o prazo vencerá muito em breve; gerar, de forma pró-ativa, gráficos e relatórios associados a performance e ao progresso do grupo de alunos (durante as atividades); e, finalmente, prover funcionalidades administrativas.

• Comportamento do aluno

O “modelo do aluno” proposto no sistema portfólio-tutor é representado por: perfil do grupo, zona de comportamento, perfil individual e histórico de navegação. O

¹É baseado no padrão de projeto *template method* [Gamma et al. 1995].

perfil do grupo corresponde ao nível de conhecimento do aluno (básico, intermediário ou avançado) de acordo com os conceitos que serão aprendidos. Enquanto que em uma sessão de ensino, dependendo de seu desempenho, os alunos irão fazer parte das zonas de comportamento. Estas são consideradas como zona crítica inferior, zona intermediária inferior, zona normal, zona intermediária superior e zona crítica superior. O critério para fazer parte de uma zona de comportamento é o desempenho registrado do aluno [Nascimento 2002].

O perfil individual do aluno permite acompanhar a aprendizagem do aluno e registrar todos os conceitos aprendidos. Assim como o histórico de navegação que registra todos os recursos didáticos utilizados pelo aluno.

- **O controle na aplicação portfólio-tutor**

O controle associado a cada sistema portfólio-tutor deve levar em conta o comportamento pró-ativo da camada portfólio e a realização de sessões de ensino gerenciadas pela camada tutor. A seguir, é representado o algoritmo associado a um sistema portfólio-tutor.

```
// Algoritmo associado ao método template ou funcionalidade associada ao sistema
// portfólio-tutor em uma sessão de ensino

// sessão de ensino on-line:

// recuperando a estratégia de ensino da camada portfolio
recuperar_estrategia_ensino(estratégia);

// interpretando a estratégia de ensino
tatica_corrente <- obter_primeira_tatica(estratégia)

// executar a tática corrente i da estratégia
for tática i and (tatica_corrente <> null) in estratégia {

    // reconhecer e executar a tatica_corrente i da estratégia
    case of tipo de tatica_corrente {
        u: recuperar o recurso didático necessário da camada portfólio;
            execute tática de reuso;
            update (perfil do grupo, zona de comportamento, perfil individual);
        c: execute tática de chat;
        r: execute tática de regras;
            update (perfil do grupo, zona de comportamento, perfil individual);
        g: execute tática de mudança de estratégia;
        s: recuperar o recurso didático necessário da camada portfólio;
            execute tática de envio de informação;
    }
    tatica_corrente <- proxima_tatica (estratégia, tatica_corrente);
}

// Funcionalidade pró-ativa da camada portfólio: a especificação do comportamento
// pró-ativo da camada do portfólio é baseada na seguinte representação:
// when event if (condition) then to execute action(s)
// descrição: quando um evento ocorrer então uma condição é verificada, se a
// condição for verdadeira a ação é executada.

when data corrente = data atividade x,
    for cada aluno em G // G representa um grupo de alunos
        if ( trabalho do aluno i associado a atividade x faltar)
            then
                preparar e enviar mensagem de email para o aluno i (mensagem, aluno)
                // lembrete para o envio do trabalho que falta

when problema de performance ou progresso é detectado em um grupo de alunos
    executar analise_dados(problema, grupo_alunos)
    gerar_relatorio ( problema, relatorio )
    e enviar_relatorio ( relatorio, professor ) com gráficos para o professor
    por email
    estabelecer_comunicacao_sincrona ( professor, grupo_alunos )
```

```

when registro da performance histórica é requerida pelo professor sobre o aluno i ou
um grupo y
  gerar_relatorio ( performance_historica, relatorio )
  e enviar_relatorio ( relatorio, professor ) com gráficos para o professor
  por email

```

3.2.2. Pontos de Adaptação

Um framework é uma arquitetura de software reutilizável. Uma aplicação criada a partir do framework pode ser “customizada” através de hotspots. Hotspots representam pontos de adaptação de código que precisam ser redefinidos em uma aplicação [Fayad et al. 1999].

Os pontos de adaptação de código e a especificação (utilizando a notação UML-F²) da classe interna do framework FA_PorT são mostradas na Figura 6. Os pontos de adaptação considerados neste trabalho são: a análise dos dados, a geração de gráficos e a mensagem.

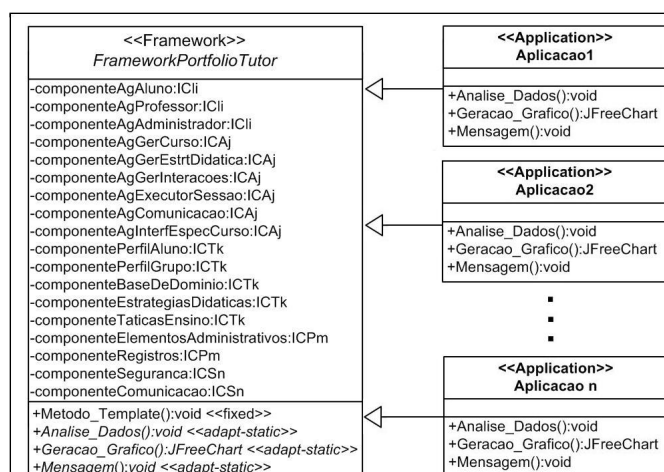


Figura 6. Diagrama de classes do framework FA_PorT utilizando a notação UML-F

3.3. Construção de Aplicações Portfólio-Tutor

Uma aplicação (sistema portfólio-tutor) é criada através de uma instância de uma classe Aplicacao1 ou Aplicacao2 que herda da classe FrameworkPortfolioTutor. Cada novo sistema portfólio-tutor possui funcionalidades default que podem ser “customizadas”, isto é possível através da redefinição de alguns métodos como a análise dos dados, a geração de gráficos e a mensagem.

É representado, no exemplo abaixo, a especificação do método template, dos três pontos de adaptação de código e a construção de duas aplicações portfólio-tutor gerados a partir do framework FA_PorT.

```

// Classe framework que deve ser reutilizada
public abstract class FrameworkPortfolioTutor{
    // Declaração dos atributos de todos os componentes.
    public FrameworkPortfolioTutor(){
        // Chamada ao controle de fluxo do framework

```

²Uma extensão da UML (Unified Modelling Language) específica para a modelagem de frameworks orientados a objetos [Fontoura et al. 2000].

```

        template();
    }
    private void template(){
        // Código associado ao método template ou fluxo de controle do framework
        // FA_PorT. Ver seção 3.2.1
        ...
    }
    // Pontos de Adaptação de Código
    public void analise_dados () { ... }
    public Grafico geracao_grafico () { ... }
    public void mensagem () { ... }
}

// Construindo aplicações ou sistemas portfólio-tutor derivadas da classe
// FrameworkPortfolioTutor.
class Aplicacao1 extends FrameworkPortfolioTutor {
    // Os pontos de adaptação de código não serão redefinidos.
}

class Aplicacao2 extends FrameworkPortfolioTutor {
    public Grafico geracao_grafico () {
        // Código associado na redefinição do método geração de gráfico que é um novo
        // formato de gráfico.
        ...
    }
}

// instanciamento da classe Aplicacao1
pt1 = new Aplicacao1 ();
// pt1 representa uma nova aplicação ou sistema Portfólio-Tutor sem redefinir os
// pontos de adaptação de código.

// instanciamento da classe Aplicacao2
pt2 = new Aplicacao2 ();
// pt2 representa uma nova aplicação ou sistema Portfólio-Tutor que possui
// funcionalidades específicas e a classe Portfolio_Tutor2 está redefinindo o método
// geração de gráfico.

```

4. Implementação do framework FA_PorT

O framework FA_PorT está sendo implementado, utilizando a linguagem java e para o módulo dos agentes está sendo utilizado o framework JADE³. A arquitetura do framework (núcleo comum) e vários componentes associados já foram implementados na primeira versão. As implementações da camada portfólio são os componentes de elementos administrativos e os registros. Para a camada tutor são os componentes de estratégias didáticas, o domínio, o perfil do aluno e o perfil do grupo. Em relação aos testes e a validação do framework, duas aplicações foram implementadas ⁴.

4.1. Aplicação 1

A aplicação 1 objetiva auxiliar o professor da disciplina de Engenharia de Software. Esta trabalha com a implementação default do framework FA_PorT, ou seja, é um sistema portfólio-tutor que possui funcionalidades específicas predefinidas sem precisar da redefinição dos pontos de adaptação de código. A tela da geração default de gráficos associada a essa aplicação, construída, através do framework FA_PorT é mostrada na Figura 7.

³JADE (Java Agent DEvelopment Framework) pode ser encontrado no site <http://jade.tilab.com/>.

⁴O professor de duas disciplinas foi consultado para a definição de recursos e sessões de ensino on line, essas aplicações foram construídas objetivando seu uso nas disciplinas de Engenharia de Software e Análise de Sistemas II do curso de ciências da computação do Instituto de Computação da UFAL.

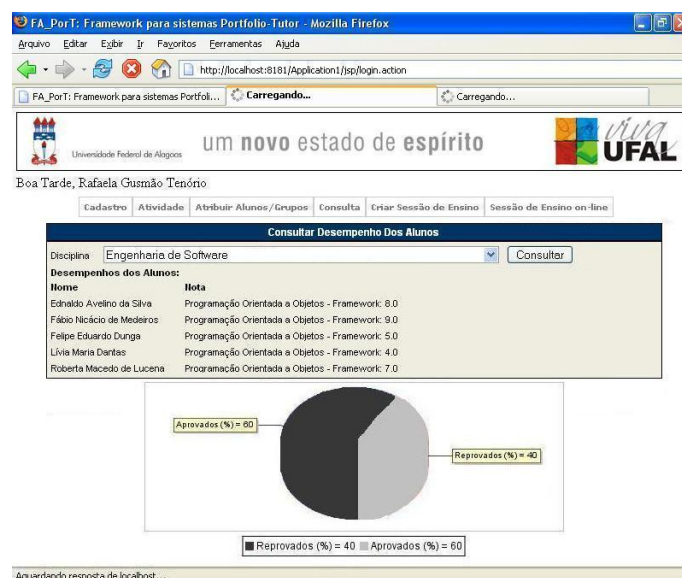


Figura 7. Interface da aplicação 1 (derivada do framework), tela do desempenho dos alunos

4.2. Aplicação 2

Esta segunda aplicação objetiva auxiliar o professor da disciplina Análise de Sistemas II. Ela representa um novo sistema portfólio-tutor que possui uma funcionalidade redefinida, que é o método geração de gráfico. A tela da geração redefinida de gráfico associada a essa aplicação construída através do framework FA_PorT é mostrada na Figura 8.



Figura 8. Interface da aplicação 2 (derivada do framework), tela do desempenho dos alunos

5. Conclusão

No contexto de instrumentos ou ferramentas para uma avaliação mais autêntica, a utilização de sistemas de tipo portfólio eletrônico é crescente [Eportconsortium 2006]. Estes tem sido desenvolvidos, porém poucos sistemas foram implementados focalizando o reaproveitamento da estrutura de sistemas de tipo portfólio eletrônico acoplado a um

Sistema Tutor Inteligente. O FA_PorT favorece o desenvolvimento de novos portfólios eletrônicos baseados em componentes e agentes, reduzindo de forma considerável o tempo de desenvolvimento de um novo sistema portfólio-tutor. A primeira versão do framework proposto foi implementado utilizando a linguagem de programação orientada a objetos java. O framework está sendo testado e validado a partir de duas aplicações já construídas.

Referências

- Costa, I. T., Fagundes, L. C., and Nevado, R. (1998). Projeto teclec - modelo de uma nova metodologia em ead incorporando os recursos da telemática. in: *Informática na educação - teoria e prática. Porto Alegre: UFRGS*, 1(1):83–100.
- Ellis, J. and Coppola, C. (2006). Sakai framework. <http://www.sakaiproject.org/>.
- Eportconsortium (2006). Electronic portfolio white paper. <http://eportconsortium.org/>.
- Fayad, M. E., Schmidt, D. C., and Johnson, E. R. (1999). Building application frameworks object-oriented foundations of framework design. *John Wiley Sons*.
- Fontoura, M., Pree, W., and Rumpe, B. (2000). Uml-f: A modeling language for object-oriented frameworks. *14th European Conference on Object Oriented Programming (ECOOP 2000), Springer, 63-82, Cannes, France*.
- Gamma, E., Helm, R., Jonhson, R., and Vlissides, J. (1995). Design patterns: elements of reusable object-oriented software. *Addison-Wesley*.
- Hatzilygeroudis, I. and Prentzas, J. (2004). Knowledge representation requirements for intelligent tutoring systems. *7th International Conference, ITS (Intelligent Tutoring Systems), Maceió, Alagoas, Brasil, Proceedings. Springer-Verlag*.
- Nascimento, D. M. C. (2002). Um sistema tutor acoplado a um portfolio eletrônico no contexto da educação a distância - portfólio-tutor. *Dissertação de Mestrado, Universidade Federal da Paraíba – COPIN/DSC, Campina Grande, Paraíba, Brasil*.
- Silva, A. S. (2000). Tuta - um tutor baseado em agentes no contexto do ensino à distância. *Dissertação de Mestrado, Universidade Federal da Paraíba – COPIN/DSC, Campina Grande, Paraíba, Brasil*.
- Silva, E. P. (2002). Desenvolvimento de um portfolio eletrônico para web baseado nos princípios de banco de dados ativo. *Trabalho de Conclusão de Curso, Universidade Federal de Alagoas - UFAL – Departamento de Tecnologia da Informação - TCI, Maceió, Alagoas, Brasil*.
- Sistêlos, A. J. C. M. (1999). Um ambiente computacional de apoio ao método de avaliação autêntica: Projeto poeta (portfólio eletrônico temporal e ativo). *Dissertação de Mestrado, Universidade Federal da Paraíba – COPIN/DSC, Campina Grande, Paraíba, Brasil*.
- Vicari, R. M. and Giraffa, L. M. M. (2003). *Sociedades Artificiais. Cap. Fundamentos dos Sistemas Tutores Inteligentes*. Bookman, Artmed Editora.
- Wenger, E. (1987). Artificial intelligent and tutoring systems: Computacional and cognitive approaches to the communication of knowledge. *Morgan Kaufmann Publishers, Inc. California, USA*.
- Wooldrige, M., Jennings, N. R., and Kinny, D. (2000). The gaia methodology for agent-oriented analysis and design. *Kluwer Academic Publishers, Boston*.