

Mestrado em Informática

UCE30-EL, Engenharia de Linguagens

Projecto Integrado

Ano Lectivo de 2009/10

1 Objectivos

Com este projecto integrado pretende-se sedimentar os conhecimentos introduzidos nas aulas teóricas dos 4 módulos desta UCE: Engenharia Gramatical, Processamento Estruturado de Documentos, Análise e Transformação de Software e Scripting no Processamento de Linguagem Natural.

2 Resultados da aprendizagem

No final do projecto, os alunos devem ter adquirido um conjunto de competências tecnológicas, específicas e genéricas detalhadas na rubrica "Resultados de Aprendizagem" da página oficial da disciplina, em

<http://wiki.di.uminho.pt/twiki/bin/view/Mestrado/EL>.

Nomeadamente, capacidade para:

- desenvolver especificações da sintaxe/semântica de linguagens com gramáticas de atributos, para construir entre outras Linguagens de Domínio Específico (DSLs) concretas.
- gerar programas usando ferramentas automáticas baseadas em gramáticas de atributos, para produzir com eficiência processadores eficientes.
- representar, armazenar e manipular Conhecimento com eficiência.
- escrever scripts para automatização de uma variedade de tarefas e transformações.
- resolver problemas usando transformações via Expressões Regulares.
- usar sistemas guiados por regras de produção (condição-reacção), compreendendo as vantagens e o funcionamento dessa aproximação à programação .
- construir e usar corpora para dele extrair informação diversa.
- construir front-ends poderosos para a análise de linguagens de programação.
- desenvolver software como sendo uma tarefa de transformar de forma sistemática e eficiente programas, utilizando métricas e outras técnicas para otimizar programas (e.g. cálculo parcial, detecção de código morto, etc), fazer o seu slicing (para facilitar a análise e compreensão), e melhorar a sua estrutura (e.g. refactoring).
- definir testes para software e testar automaticamente programas em diferentes linguagens de programação.
- criar representações visuais adequadas à compreensão clara do conhecimento complexo detido.
- especificar uma linguagem de anotação para um conjunto de requisitos e saber usá-la, ou usar outra definida por outrem, recorrendo às ferramentas adequadas.

- implementar transformações de documentos para diversos fins: extracção de conhecimento, publicação na Web, intercâmbio de informação, ...
- utilizar soluções de armazenamento para documentos anotados.
- implementar um projecto de publicação electrónica recorrendo a normas internacionais abertas: XML, XSLFO e XSL.

3 Organização e Funcionamento

O projecto será desenvolvido em grupos de 2, fora das aulas, em ambiente Linux (este ano será o único ambiente de trabalho oficial).

A única aula presencial de 1h/semana será usada para:

- Apresentação do projecto (no todo e em cada uma das suas fases), discussão do enunciado e identificação dos requisitos, alinhamento de estratégias e planeamento da resolução.
- Formação teórica básica em temas tecnológicos transversais aos 4 módulos e essenciais ao desenvolvimento do projecto, com proposta de temas a investigar.
- Gestão dos projectos; Apresentações Intercalares (avaliação); e Seminários.

Ao longo e no fim do ano lectivo, cada grupo apresentará à equipe docente o trabalho realizado e os resultados obtidos em vários momentos de controlo (segundo o calendário abaixo), devendo entregar um relatório técnico de desenvolvimento devidamente estruturado e fundamentado, criado desde a primeira aula com uma ferramenta de *Literate Programming* como, p.ex., o NuWeb.

3.1 Calendarização

O projecto deve ser executado ao longo de todo o ano estando a **entrega final** agendada para 2010/06/30.

Para controlo da situação e avaliação intermédia, haverá 3 apresentações intercalares do projecto, que coincidem com o meio do 1º semestre, o início do 2º semestre e o meio do 2º semestre. Mais concretamente nos dias:

1. 2009/12/07,
2. 2010/03/01 e
3. 2010/05/03.

4 Enunciado

O que se pretende neste projecto é desenvolver um Ambiente baseado na Web para Análise e Transformação de Software Aberto, que doravante designaremos por FOSSAT (*Free Open Source Software Analysys and Transformation*)

O objectivo principal é criar um ambiente de inspecção, medida e adaptação de código que se aplique a pacotes de software aberto disponíveis em ambientes típicos como SourceForge, o Google Code ou o Codeplex.

O sistema FOSSAT deve permitir indicar o caminho para o pacote e, depois de o ir buscar, disponibilizará a seguinte funcionalidade:

- analisar o pacote e criar um Manifesto (num formato a definir) com toda a meta informação sobre as componentes do pacote, analisando o tipo de ficheiros, as makefiles, os ReadMe, etc.
- disponibilizar uma série de métricas sobre o pacote ou sobre uma das componentes identificadas.
- analisar um dado componente.
- transformar um dado componente, conforme especificado à parte.

- criar um base de conhecimento (na realidade uma base de dados documental), onde se acumula toda a informação sobre os pacotes analisados e que permite, entre outras coisas, algum tipo de comparação entre os pacotes (o comparador recebe a função de comparação, que pode ser mais ou menos sintáctica/semântica).

5 Desenvolvimento

Para desenvolver metodicamente este projecto, segundo as boas práticas do desenvolvimento rigoroso de SW orientado por testes, definem-se abaixo as principais etapas do processo. Seguem-se as tarefas específicas que serão metas em cada semana.

5.1 Etapas do Desenvolvimento

Para concretizar o trabalho preconizam-se as seguintes etapas:

1. Analisar o problema
2. Conceber o sistema desenhando a sua arquitectura com base na Web.
3. Modelar os dados do problema e as operações requeridas.
4. Conceber a Interface que disponibilize as operações pedidas e apresente os resultados desejados
5. Criar uma bateria de testes que garanta a satisfação dos requisitos.
6. Conceber e Implementar a Interface do sistema com os utilizadores, usando .
7. Desenvolver um protótipo do sistema.
8. Testar o protótipo, repetindo todos os passos anteriores até que todos os testes sejam satisfeitos.
9. Concluir a documentação do sistema que deve ter sido elaborada, usando *literate programming*, ao longo de todas as etapas precedentes.
10. Preparar um artigo científico e uma apresentação (com demo) sobre o sistema desenvolvido.

5.2 Tarefas a Realizar

Para organizar a resolução deste projecto, propõe-se que se realizem as seguintes tarefas específicas¹ (algumas das quais poderão prosseguir em paralelo) dentro das várias etapas acima identificadas:

1. Procurar (usando os recursos de pesquisa disponíveis na Web) **sistemas equivalentes** ao FOSSAT.
2. Procurar informação sobre o **manifesto** (definição, exemplos, standards, etc.).
3. Pesquisar o que há sobre **métricas de software** (escolher um conjunto de 5 básico, comum a todos os grupos, e mais umas 10 distintas) .
4. etc....

¹Esta lista não está concluída! Novas tarefas irão sendo identificadas e acrescentadas à medida que estas se concluíam e o projecto vá avançando.