PHEASANT Implementation

Sousa, Vasco Nuno da Silva de

May 8, 2007

Acknowledgments

Contents

1 Introdução **2**

2 Análise de Ferramentas **3**
2.1 Procedimento 3
2.2 Ferramentas 3
2.2.1 AToM^3 4
2.2.2 DSL Tools 4
2.2.3 GEMS 5
2.2.4 GME 5
2.2.5 GMF 5
2.2.6 Metaedit+ 6
2.3 Conclusões 6

Chapter 1

Introdução

Em 2005, ... PHEASANT ...

Este projecto tem como objectivo a implementação de um editor plenamente funcional, que permita a utilização prática de PHEASANT. Para isso será feita uma análise das ferramentas de desenvolvimento do paradigma em que se encontra o PHEASANT, verificando o seu estado e capacidades, defenição prática da linguagem, defenição do editor, e apuramento e refinamento de todos os elementos envolvidos tendo em vista a melhor e eficás utilisação de PHEASANT.

Chapter 2

Análise de Ferramentas

Neste capitulo procedemos á análise das ferramentas disponiveis, sendo que a maioria de natureza academica, apesar da existencia, ainda que nos primeiros passos, de ferramentas comerciais.

Esta análise tem por objectivo aferir qual a ferramenta ou ferramentas mais apropriadas, tendo em vista, a implementação especifica deste projecto, assim como a capacidade de possibilitar a sua evolução e melhoramento.

2.1 Procedimento

Devido a restrições de tempo e natureza do projecto, a análise foi efectuada de uma forma horizontal, em detrimento de uma análise em profundidade de cada uma das ferramentas, baseando-se para isso em grande parte na documentação disponivel e subsequente teste das funcionalidades descritas, tendo como ponto de observação as caracteristicas pertinentes ao desenvolvimento do projecto e respectiva utilização.

2.2 Ferramentas

Da quantidade de ferramentas disponiveis, quer de uso comercial, quer as de desenvolvimento academico, foram estudadas as que se aprentaram acessiveis para tal e apresentavam um nivel de maturidade minimo, antes de qualquer tipo de teste.

De destacar que todas as ferramentas analisadas definem, a utilisação da linguagem de dominio especifico de uma forma visual, verificando no momento a validade de todos os elementos e as suas ligações, não permitindo estados intremedios invalidos, e tendo em conta que essa utilisação é dependente do editor em que se define a linguagem. De notar ainda que apesar de todas as ferramentas guardarem as defenições em XML á excepção do AToM^3, a defenição do XML usado é difrente entre elas, sendo mesmo proprietaria no caso do DSL Tools.

2.2.1 AToM^3

Desenvolvido no Modelling, Simulation and Design Lab (MSDL) na School of Computer Science of McGill University e com a colaboração do Prof. Juan de Lara of the School of Computer Science, Universidad Autónoma de Madrid (UAM), Espanha, o AToM^3 é uma ferramenta de modelação multi-paradigma, de licença livre, fortemente centrada na transformação de modelos.

Como caracteristicas do AToM^3 temos o desenvolvimento da linguagem de dominio especifico de forma esquematica de forma analoga á sua propria utilização, utilizando uma parametrisação de diagrama de classes, num modelo unico que define a linguagem, a sua inreacção e aspecto visual, a inclusão de um pequeno editor vectorial para os simblos utilizados na linguagem desenvolvida, e a capacidade de durante desenvolvimento de soluções permitir o uso de varias linguagens, desde que estas não sejam mutuamente incompativeis, gerando ligações e elementos genericos entre estas. O facto desta ferramente ser desenvolvida em Phython permite ainda a inclusão de scripts, ou alteração dos componentes do proprio editor, alterando directamente o seu codigo.

Por outro lado a sua utilização é limitativa, quer pela interface pouco intuitiva, de onde se destaca a necessidade de utilizar atalhos de teclado e o rato em simultaneo para introduzir elementos ou relações, não tendo disponivel nenhuma funcionalidade de undo e redo, quer a unica forma de costumizar a sua utilização seja por uso de scripts em Phython, mantendo-se a interfasse final limitada a uma barra de ferramentas. Existe ainda uma distancia entre os varios documentos de referencia base, sendo feitas assumções de conhecimento sem existir documentação intremedia que preencha essa distancia, sendo o caso na forma de defenir ligações entre elementos.

2.2.2 DSL Tools

Sendo disponibilizada como uma das ferramentas do SDK para Visual Studio 2005, esta é a abordagem da MicroSoft ao paradigma de Dominos Especificos.

Fortemente ligado ao C#, e com um esquema de defenição das linguagens encadeado, que por um lado mantem tudo arrumado, por outro impede a arruamção dos elementos de uma forma familiar e a gosto, por outro lado a defenição dos aspectos visuais da linguagem criada numa area especifica do mesmo espaço de defenição funcional da linguagem, contraria quando em grande quantidade a legibilidade gerada pela arrumação dos elementos.

Os modelos ao ser alterada a linguagem base, tentam ler tanto quanto ainda faça sentido na nova defenição, descartando tudo o resto e emitindo mensagens de erro nesse sentido.

O DSL Tools funciona ainda numa base de templates que convertem os esquema em codigo antes da compilação, passo que é necessario a cada teste

da linguagem, uma vez que o desenvolvimento desta processa-se da mesma forma que a sua utilisação.

2.2.3 GEMS

Apresentando uma interfasse semelhante ao GME, sendo do mesmo grupo de laboratorios de desenvolvimento, e correndo como plug-in do eclipse como o GMF, e tendo como objectivo criar de alguma forma uma ponte entre varias ferramentas de meta-modelação para Dominios Especificos, o GEMS é ...

Apesar de apresentar-se como um movimento benefico para o desenvolvimento do paradigma de desenvolvimento especifico e visual, o GEMS aprenta-se mais pobre e limitado do que o GME e ainda sem a capacidade do GMF dentro do ambiente eclipse.

... reutilização e exportação ...

2.2.4 GME

Desenvolvido no Institute for Software Integrated Sistems, na Universidade de Vanderbilt, Nashville Tennessee, US.

O GME é a mais simples das ferramentas analizadas, quer pela interface simples como pela documentação exaustiva, de resto a mais bem estruturada do conjunto, o GME é tambem a mais bem perparada para desenvolvimento em grandes escalas, em que apesar de não ser possivel personalizar o ambiente de desenvolvimento para o utilizador final das linguagens criadas, a capacidade de criar sub-esquemas e a disponibilisação de uma shell capaz de interpretar varias linguagens incluindo pearl, permitem grande flexibilidade e rapidez de desenvolvimento.

Com o refinamento da defenição da linguagem, o GMF permite manter varias verções registada, tentado ler os esquemas que utilisão essa mesma linguagem com a versão mais recente capaz de lê-los integralmente.

... exportação ...

... referencias cross-hierarquie

2.2.5 GMF

Inicialmente desenvolvido pela IBM foi posteriormente ofercido ao projecto eclipse, tornando-se um projecto open-source desenvolvido por uma comunidade propria.

Sendo a ferramenta com a pior documentação do conjunto, é tambem a que se apresenta com maior potencial e flexibilidade. Sendo das poucas ferramentas analizadas em que a defenição da linguagem de dominio especifico não é feita de forma esquematica, mas antes feita de forma hierarquizada dos seus componentes e atributos, torna-se necessario o conhecimento de como é feita a defenição da linguagem dentro do ambiente de desenvolvimento e a respectiva nomenclatura, tornando a curva de aprendisagem da ferramenta

bastante acentuada. Por outro lado este acesso tão proximo da defenição permite controlar todos os aspectos da linguagem conferindo-lhe um potencial que não está presente nas outras ferramentas.

O GMF é tambem a unica ferramenta que premite, de facto, personalizar o ambiante de utilização da linguagem desenvolvida, tendo em vista um utilizador final distinto do desenvolvimento da propria linguagem, sendo ainda permitindo em ambiente eclipse abstarir-se do mesmo e criar uma plataforma de alguma forma autonuma.

... reutilização e exportação ...

2.2.6 Metaedit+

Sendo a terceira ferramenta comercial do conjunto, a unica esplicitamente paga e a mais antiga, o Metaedit+ tem uma forma particular de meta modelar a linguagem especifica, em que as sua defenição é feita por meio de caixas de propriedades, onde se definem, as propriedades dos nós entidade, relações e restrições entre os nós. Não sendo directo o processo de criação, torna a defenição de relações restritas entre vários nós fácil, simplesmente defenindo os pontos de ligação da relação e quais as suas restrições.

É tambem juntamente com o AToM^3 a unica ferramenta de modelação com um editor vectorial para criação de formas personalizadas para os nós.

Ao alterar-se a definição da linguagem base, os esquemas lêm todos os elementos presentes, desagregando os nós das ligações que passaram a ser invalidas, mas mantedo todos no esquema para edição pelo utilizador.

... exportação ...

2.3 Conclusões

No momento desta analise, o desenvolvimento de tecnicas e ferramentas sobre o paradigma de dominios especificos, encontra-se nos primeiros estagios, e ainda não é possivel encontrar um ambiente de desenvolvimento que providencie uma solução absoluta e que seja capaz de lidar com todas as nuances e requesitos deste tipo de desenvolvimento.

Assim foi necessario selecionar a ferramenta que mais se ajustasse ao desenvolvimento deste projecto. Neste caso surgem dois candidatos, o GME pela sua simplicidade e no entanto permitir soluções de grande escala, e a possibilidade de editar por linha de comandos, util para uma transição de utilizadores abituados a programar as suas soluções. O outro candidato é o GMF por permitir o desenvolvimento em qualquer tipo de plataforma, ser possivel gerar um cliente em que o utilizador final da linguagem abstrai-se de estar a utilizar o eclipse, e o nivel de configurabilidade e especificação das linguagens em GMF, mentem uma porta aberta a futuros desenvolvimentos, sem comprometer tudo e que foi feito até então.