

Codipse-Req: Uma Ferramenta para Engenharia de Requisitos em Desenvolvimento Distribuído de Software

Regiane Andrade Brito, Alexandre Marcos Lins de Vasconcelos

Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851, 50732-970, Recife – PE – Brazil

{rab3, amlv}@cin.ufpe.br

***Abstract.** This paper presents a cooperative tool built as an Eclipse plugin to support requirements engineering. This tool support work in distributed teams, providing collaboration and communication among stakeholders. This tool was developed using a new Eclipse's Project Framework named ECF (Eclipse Communications Framework) that aims to make it easy collaborative applications development.*

***Resumo.** Este artigo apresenta uma ferramenta cooperativa para engenharia de requisitos, construída na forma de um plugin para o Eclipse. A ferramenta auxilia o trabalho de equipes distribuídas, permitindo edição colaborativa e comunicação entre os stakeholders. A ferramenta foi construída com base em um novo framework do projeto Eclipse, o ECF (Eclipse Communications Framework), que pretende auxiliar a construção de aplicações colaborativas.*

1. Introdução

A crescente busca por maior competitividade tem levado as empresas a adotarem um modelo distribuído de desenvolvimento de software, onde diferentes partes do sistema são desenvolvidas em localidades distintas. Tentando realizar desenvolvimento a baixo custo, empresas têm atravessado fronteiras, formando um mercado global. Essa mudança tem causado impacto tanto em marketing e distribuição, quanto na forma de concepção, produção, projeto, teste e entrega dos sistemas para os clientes (Herbsleb e Moitra, 2001).

Esta nova área, conhecida como Desenvolvimento Distribuído de Software (DDS) (Herbsleb e Moitra, 2001) é uma área de pesquisa atual em engenharia de software (Gruhn, 2002). O desenvolvimento de software já é uma atividade complexa quando realizada localmente, com equipes distribuídas essa complexidade é ainda maior.

O DDS ainda apresenta muitas barreiras relacionadas principalmente a comunicação e coordenação da equipe distribuída (Hawryszkiewicz e Gorton, 1996). Em especial, a disciplina de engenharia de requisitos merece uma maior atenção (Lopes e Audy, 2003), por ser considerada como a maior responsável pela qualidade do sistema final (Nuseibeh e Easterbrook, 2000) e por ser uma atividade complexa pois requer muita interação entre pessoas com conhecimentos e culturas diferentes.

2. Engenharia de Requisitos em DDS

Engenharia de requisitos (ER) é o processo de descoberta de funcionalidades e restrições de um sistema, identificando *stakeholders* e suas necessidades, bem como documentando essas descobertas de uma forma que possibilite a análise, comunicação e uma implementação futura (Nuseibeh e Easterbrook, 2000).

No desenvolvimento com equipes co-localizadas, a coleta de requisitos ocorre em reuniões face a face, considerada a forma mais efetiva de comunicação (Damian et al, 2000). Esses encontros tornam mais fácil também as atividades de negociação e resolução de conflitos.

Em DDS essa atividade tende a ser dificultada por problemas inerentes da distribuição geográfica, além dos problemas já conhecidos da área como dificuldade de comunicação entre os *stakeholders* (Kotonya e Sommerville, 1998). Alguns estudos de caso relatam experiências na área de engenharia de requisitos distribuída, apresentando as lições aprendidas (Damian e Zowghi, 2003) (Prikladnicki et al, 2003).

3. Suporte a Engenharia de Requisitos Distribuída

Fornecer suporte a Engenharia de Requisitos Distribuída envolve tanto definir um processo adequado quanto uma ferramenta colaborativa para auxiliar a execução desse processo. Essa ferramenta deve ser definida através da integração com a área de CSCW (*Computer Supported Cooperative Work*) para que a mesma possua funcionalidades como: comunicação, coordenação, percepção e memória de grupo (Araújo, 2000).

Em relação ao processo de engenharia de requisitos, a figura 1 apresenta como este deve ser caracterizado, em relação ao processo tradicional. No site da ferramenta (CODIPSE, 2005) é possível fazer o *download* de um processo definido (baseado no SPEM). A apresentação desse processo não está no escopo desse trabalho.

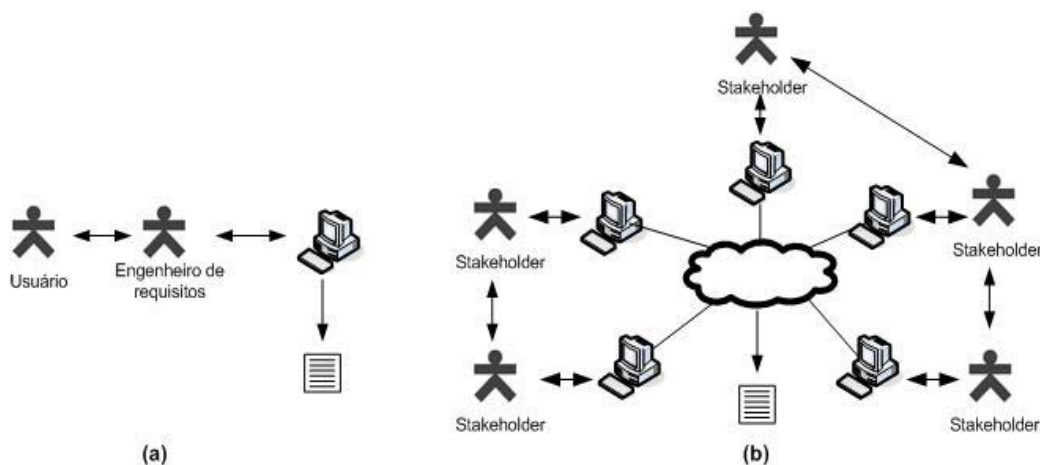


Figura 1: (a) um processo mediado por um desenvolvedor (b) um processo totalmente cooperativo, adaptado de (Ambriola e Gervasi, 1998)

A idéia de construir um ambiente colaborativo para Engenharia de Requisitos não é nova. Existem algumas propostas na literatura, como o ambiente CRETA (Togneri et al, 2002) que é baseado na web e permite a integração com um processo de engenharia de requisitos e o Teamwave instanciado para engenharia de requisitos (Herlea e Greenberg, 1998).

4. A ferramenta Codipse-Req

A ferramenta está sendo construída como parte do projeto CODIPSE (*COoperative and DIstributed Process Suport Environment*) e é disponibilizada pela licença CPL (<http://www.eclipse.org/legal/cpl-v10.html>). O *download* pode ser feito no site do projeto no Tigris (<http://codipse.tigris.org/>), através de “Documents & Files”.

O Eclipse é um projeto de software livre que fornece uma plataforma para integração de ferramentas em geral, popularizado por também ser uma IDE para aplicações Java. O desenvolvimento de *plugins* ocorre através de “pontos de extensão” definidos por outros *plugins*, incluindo os internos à plataforma Eclipse. Dessa forma, um novo *plugin* implementa “pontos de extensão” e define novos pontos para que outros *plugins* possam utilizar.

O Projeto Eclipse disponibiliza o EMF (*Eclipse Modelling Framework*). Este *framework* possibilita a integração entre modelos e código fonte, possibilitando a transição de um para o outro. Através de modelos como um XML Schema ou UML, é possível gerar automaticamente as classes e interfaces correspondentes, bem como editores (usados para criar instâncias do modelo).

Recentemente, o Eclipse iniciou o desenvolvimento de um novo *framework* denominado ECF (*Eclipse Communications Framework*). Esse projeto pretende facilitar a criação de aplicações colaborativas para a plataforma Eclipse, fornecendo uma API para mensagens síncronas e assíncronas, as quais podem ser enviadas entre pessoas, entre *plugins* ou entre os dois.

A figura 2 apresenta a arquitetura do ambiente, utilizando extensões dos *frameworks* citados acima.

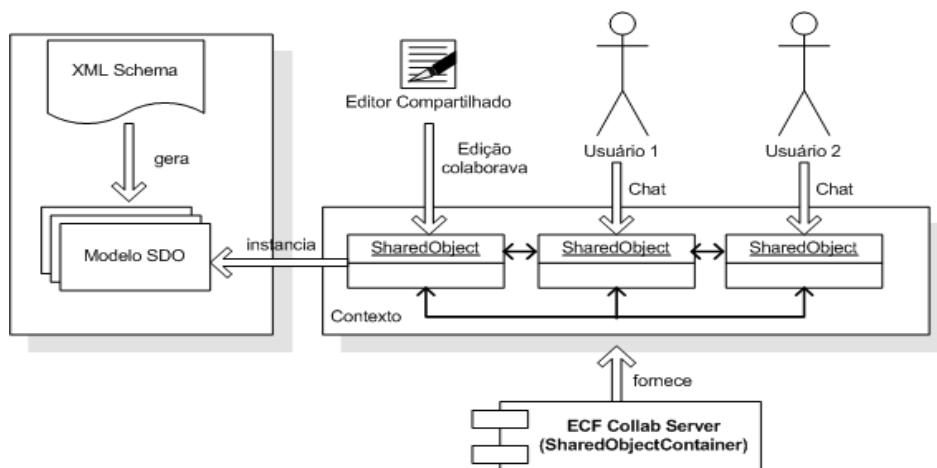


Figura 2 – Arquitetura da ferramenta

O Modelo SDO (*Service Data Objects*) é gerado utilizando o EMF, a partir de um XML Schema para representação de um documento de requisitos (EclipseSrs, 2005). O SDO é um padrão para representação de dados, o qual possui implementação no EMF. Esse padrão foi usado porque o ECF possui suporte a ele. Um editor compartilhado foi construído para que uma instância desse modelo pudesse ser criada de forma síncrona por um engenheiro de requisitos e editado pelos demais *stakeholders*.

A aplicação de *chat* foi construída com base em um exemplo disponibilizado pelo ECF e adaptações foram feitas para que este incluísse o papel do participante no processo. Os conceitos principais no ECF são: *SharedObject* e *SharedObjectContainer*. *SharedObject* representa objetos compartilhados que podem comunicar-se entre si (o modelo e os usuários são representados dessa forma). O *SharedObjectContainer* define um contexto (um projeto, por exemplo) para que esses objetos possam trocar mensagens. Além disso, ele fornece o protocolo de comunicação que deve ser utilizado (o ECF Collab Server utiliza TCP).

5. Exemplo de utilização

Como exemplo de utilização da aplicação vamos considerar uma empresa de desenvolvimento de software com equipes distribuídas, o que levou a necessidade de utilizar um ambiente desse tipo. A tela da figura 3 apresenta a funcionalidade de *login*. Para isso, o usuário deve escolher um servidor, um nome e qual seu papel.

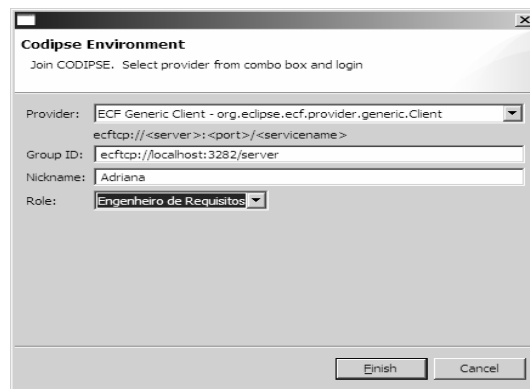


Figura 3 – Tela de login

Após isso, os participantes podem editar um modelo pré-existente como o mostrado na figura 4. Nesse exemplo, foram criadas seções para requisitos não-funcionais, casos de uso e requisitos funcionais. Sempre que o modelo é salvo as alterações são propagadas e alterações não salvas serão perdidas.

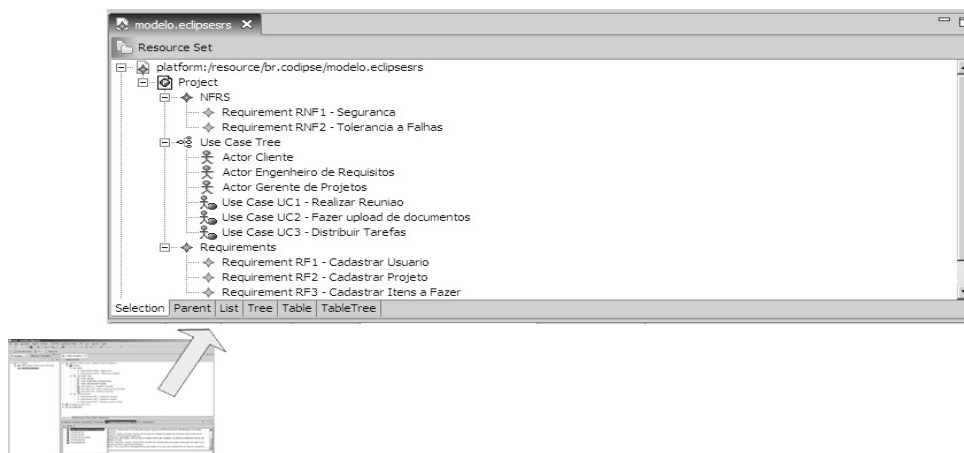


Figura 4 – Edição colaborativa

Além disso, durante essa sessão, os participantes podem trocar mensagens instantâneas, bem como visualizar o papel dos demais envolvidos, como mostrado na figura 5.

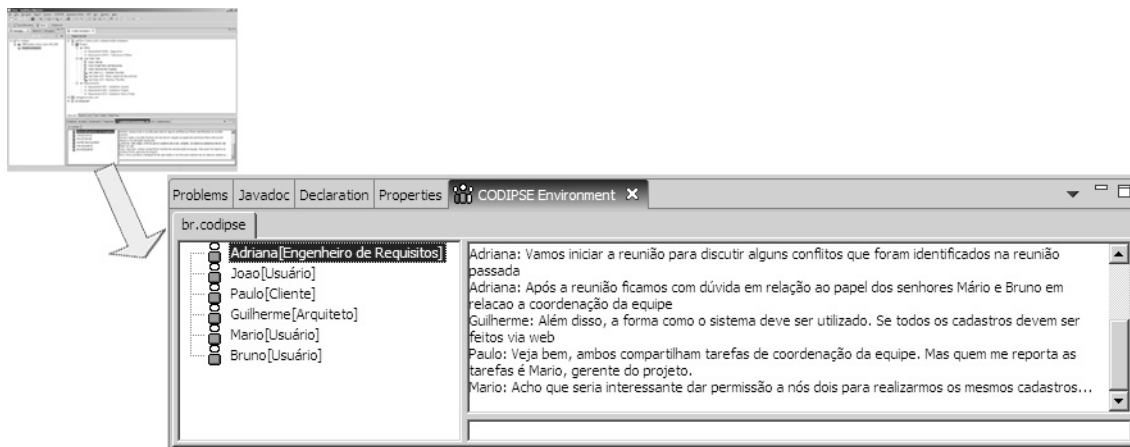


Figura 5 – Chat entre os participantes

A maior dificuldade na implementação foi o entendimento do *framework* ECF e da arquitetura de *plugins* do eclipse. No entanto, a quantidade de código criado foi relativamente pequena, visto a reutilização do código disponibilizado como exemplo pelo ECF e pela utilização do XML Schema já definido. Após isso, o trabalho foi concentrado na criação do editor do modelo, pois houve a adaptação para que o mesmo pudesse ser feito de forma colaborativa e utilizando o padrão SDO.

6. Considerações finais e Trabalhos Futuros

Esse artigo apresentou a construção de uma ferramenta cooperativa para engenharia de requisitos, através da utilização dos *frameworks* ECF e EMF disponibilizados pelo projeto Eclipse. A utilização de tecnologias de colaboração tem sido apontada como um dos principais fatores para auxiliar a engenharia de requisitos distribuída (Lloyd et al, 2002) (Damian e Zowghi, 2003) (Prikladnicki et al, 2003).

Embora a ferramenta apresentada ainda não possua todas as funcionalidades desejáveis para um contexto no qual ela se propõe, os resultados obtidos e as análises dos *frameworks* disponibilizados pelo Eclipse indicam que é possível a fácil construção de soluções mais elaboradas.

Como trabalho futuro está prevista a integração de gerência de conhecimento baseada no processo de software definido para equipes distribuídas disponibilizado no site da ferramenta (CODIPSE, 2005). O ambiente deve prover suporte para a geração de todos os artefatos e interações previstos nesse processo e incluir funcionalidades para coordenação e memória de grupo. Após a conclusão do ambiente, experimentos serão realizados dentro de uma disciplina de graduação, onde as equipes utilizarão a ferramenta e a qualidade dos artefatos gerados será analisada.

O ECF é um projeto de código aberto e estimula a participação de novos desenvolvedores. Como forma de contribuir para o ECF, as funcionalidades serão primeiro implementadas como exemplos (pois dessa forma podem ser reutilizadas) e em seguida serão acopladas na ferramenta.

7. Referências

- AMBRIOLA, V. e GERVASI, V. "The Case for Cooperative Requirement Writing". ECOOP Workshop on Automating the OO Software Development Process, 1998
- ARAÚJO, R. M. "Ampliando a cultura de processos de software – um enfoque baseado em groupware e workflow". Tese de doutorado. UFRJ - COOPE, 2000
- CODIPSE. "COoperative and DIstributed Process Support Environment". Disponível pela internet em: <http://codipse.tigris.org>. Último acesso em abril/ 2005
- DAMIAN, D., ALBERIN, A., e SHAW, M. L. G. "Using Different Communication Media in Requirements Negotiation". IEEE Software, 2000
- DAMIAN, D. e ZOWGHI, D. "Requirements Engineering challenges in multi-site software development organizations". Requirements Engineering Journal, 8, pp.149-160, 2003
- ECLIPSESRS. "XML based SRS (Software Requirements Specification)". Disponível pela internet em: <http://eclipsesrs.sourceforge.net/>. Último acesso em maio/ 2005
- GRUHN, V. "Process-Centered Software Engineering Environments, A Brief History and Future Challenges". Annals of Software Engineering 14, 363-382, Kluwer Academic Publishers, 2002
- HAWRYSZKIEWYCZ, I. T. e GORTON, I. "Distributing the software process". IEE Software, 1996
- HERBSLEB, J. D. e MOITRA, D. "Global Software Development". IEEE Software, 2001
- HERLEA, D. E. e GREENBERG, S. "Using a Groupware Space for Distributed Requirements Engineering". Proceedings of WET ICE, Stanford, California, USA, 1998
- KOTONYA, G. e SOMMERVILLE, I. "Requirements Engineering: Processes and Techniques". John Wiley & Sons Ltda, ISBN: 0-471-97208-8, 1998
- LLOYD, W. J., ROSSON, M. B., e ARTHUR, J. D. "Effectiveness of Elicitation Techniques in Distributed Requirements Engineering". Proceedings of the IEEE Joint International Conference on Requirements Engineering, 2002
- LOPES, L. e AUDY, J. L. N. "Em Busca de um Modelo de Referência para Engenharia de Requisitos em Ambientes de Desenvolvimento Distribuído". Anais do VI Workshop em Engenharia de Requisitos, 2003
- NUSEIBEH, B. e EASTERBROOK, S. "Requirements Engineering: A Roadmap". The Future of Software Engineering, Anthony Finkelstein, pp.37-46, ACM Press, 2000
- PRIKLADNICKI, R., AUDY, J. L. N., e EVARISTO, R. "Requirements Management in Global Software Development: Preliminary Findings from a Case Study in a SW-CMM context". International Workshop on Global Software Development, 2003
- TOGNERI, D., FALBO, R., e MENEZES, C. "Supporting Cooperative Requirements Engineering with an Automated Tool". 2002