

ABCTool - Uma Ferramenta para Cooperação Baseada na Arquitetura do Sistema

Cynthia Maria Silva de Barros
Mestranda do PPGEE-PUC-Minas*
cmsbarros@zipmail.com.br

Carlos Alberto Marques Pietrobon
Professor-Orientador do PPGEE_PUC-Minas*
Professor da PUC-Minas e UFOP
capietro@pucminas.br, capietro@iceb.ufop.br

Resumo

O projeto de um sistema de software pode se tornar um trabalho bastante complexo em se tratando da construção de grandes sistemas. Assim, cada vez mais os projetistas têm reconhecido a importância do design da arquitetura no projeto do sistema com o propósito de facilitar o trabalho de desenvolvimento. Outro aspecto que também tem sido observado é o fato de que estes sistemas são desenvolvidos por grupos de pessoas que trabalham paralelamente e cooperativamente. A dificuldade de comunicação entre a equipe gera problemas na hora da integração das partes que cada um desenvolveu. Esse artigo propõe uma ferramenta de apoio ao desenvolvimento cooperativo de software baseado na arquitetura do sistema sendo desenvolvido.

Palavras-Chave: Arquiteturas de Software, Estilos Arquiteturais, Projeto de Sistemas, Engenharia de Software, Trabalho Cooperativo Suportado por Computador.

1. Introdução

O processo de design de software de sistemas complexos consiste em, inicialmente, fragmentá-los em componentes menores (subsistemas), mais específicos e mais facilmente tratáveis e inteligíveis. O conjunto destes componentes e a forma como interagem entre si formam a arquitetura do sistema. A descrição da arquitetura do sistema tem se tornado fundamental pelo fato dela mesma se constituir uma base ou um vocabulário específico em torno do qual a equipe pode conversar e aprender sobre o sistema. Ela também permite uma estruturação da própria equipe de desenvolvimento.

Tipicamente, cada componente é atribuído a uma ou mais pessoas encarregadas de desenvolvê-lo. Estas pessoas, normalmente, são profissionais diferentes, cada qual possuindo interesses e visões também diferentes. Em um determinado momento, as diversas partes devem ser integradas para montar o produto. Desta integração surgem conflitos que dificultam a atividade de desenvolvimento. Um exemplo seria a integração de componentes cujas interfaces não correspondem. A questão é como evitar ou minimizar os problemas e a perda de tempo em recursos para resolvê-los. Este é um problema complexo cuja solução não pode ser obtida focando apenas o desenho do software. É necessário abordar outros aspectos e atividades que, hoje em dia, são tratados genericamente pelo nome de trabalho cooperativo suportado por computador. Um aspecto importante seria: Como fazer com que todos os envolvidos trabalhem juntos em direção a um só objetivo?

Buscando soluções para estas questões, resolvemos combinar os conceitos de arquitetura de software e trabalho cooperativo suportado por computador. Escolhemos usar esta combinação por diversas razões. Arquitetura adiciona estrutura ao processo de desenvolvimento de software. Por exemplo, arquitetura pode ser utilizada para dividir o sistema em subsistemas e, conseqüentemente, isto influencia a divisão e o controle das

* PPGEE – Programa de Pós-Graduação em Engenharia Elétrica

tarefas. Arquitetura de software também modela outras informações que podem ser exploradas com diversos propósitos tais como a gerência de configuração de software e a coordenação de equipes.

Assim, vimos propor um modelo de ferramenta de apoio ao desenvolvimento cooperativo de software, onde a coordenação e a cooperação entre as equipes e/ou profissionais envolvidos nesse processo está baseada na arquitetura do sistema alvo do desenvolvimento.

2. Conceitos Relacionados

Arquitetura de Software

Para construir sistemas com rapidez e eficiência, refazendo o mínimo de processos possíveis é necessário entender corretamente o problema e a proposta de solução. A forma da estrutura de alto nível do sistema, incluindo sua subdivisão em subsistemas, sua concorrência intrínseca, a alocação de subsistemas ao hardware e software, o gerenciamento de dados, a coordenação de recursos globais, a implementação do controle de software, as condições externas e as prioridades de balanceamento, é chamada *Arquitetura do Sistema* [6]. A arquitetura pode ser entendida como um modelo para visualização e controle do sistema.

No desenho da arquitetura pode-se usar *Padrões Arquiteturais* que são modelos já conhecidos e catalogados para esse fim. Entre os padrões arquiteturais mais comuns, temos ([3,4]): Tubos e Filtros (*Pipelines*), Camada (*Layered Systems*), Cliente-servidor (*client-server*), Três camadas ou Multi-camadas (*Three-Tier and Multi-Tier Systems*), Sistemas que utilizam agentes e/ou federações e agregações, Repositórios, Orientado a Objetos, Programa Principal / Sub-rotinas.

Trabalho Cooperativo Suportado Por Computador - TCSC

Trabalho Cooperativo Suportado por Computador – TCSC (Computer-Supported Cooperative Work – CSCW) é tido como um termo genérico que combina o entendimento das maneiras como as pessoas trabalham em grupo com as tecnologias disponíveis (groupware) de software, hardware, serviços e técnicas associadas [5]. É uma área de pesquisa que tem o objetivo de integrar o trabalho de várias pessoas envolvidas em uma meta comum, dentro de um universo cooperativo, através do compartilhamento de recursos de uma forma eficiente, estabelecendo-se assim, comportamentos dinâmicos sociais e tecnológicos. Essa atividade colaborativa não é fácil de ser implantada, necessitando que toda uma filosofia de trabalho cooperativo seja difundida para que os atores envolvidos alcancem seus objetivos [1]. Ferramentas que suportem o processo também são necessárias.

A tecnologia (hardware/software) gerada pela pesquisa em TCSC e que habilita o trabalho em grupo é denominada groupware. TCSC engloba os seguintes requisitos [5]: Coordenação; Compartilhamento de informações e do espaço de trabalho; Suporte a aspectos individuais e de grupo; Controle dinâmico do trabalho; Memória organizacional do grupo; Dinâmica de grupo; Suporte à comunicação; Geração de idéias; Estabelecimento de políticas de acesso, Reconhecimento e disseminação de mudanças.

O Processo Cooperativo de Desenvolvimento de Software Considerando Arquitetura

Quando na fase de projeto, a primeira atividade é a de design da arquitetura onde o passo inicial consiste em dividir o sistema em subsistemas. Subsistemas nada mais são que, em aplicações pequenas, os próprios componentes, e em aplicações maiores, o conjunto de

componentes que compartilham propriedades comuns, tais como funcionalidade similar, mesma localização física, execução no mesmo tipo de hardware, etc. Para cada subsistema devem ser definidas as interfaces que ele fornece.

Devemos observar que a divisão do sistema em subsistemas não se faz apenas por uma questão de dividir ou agrupar funcionalidades. Segundo Wilson de Pádua [2], um desenho bem feito divide adequadamente o produto em componentes cuja implementação pode ser dividida eficazmente entre os membros de uma equipe de desenvolvedores, permitindo o desenvolvimento paralelo desses componentes, diminuindo os prazos de implementação.

Se por um lado, subsistemas e/ou componentes podem e devem ser desenvolvidos paralelamente, muitos são os problemas decorrentes disso. A comunicação entre os membros de uma equipe, ainda hoje, constitui um dos gargalos do desenvolvimento. Uma prática comum no desenvolvimento é o fato de componentes e subsistemas serem construídos de maneira particular, onde a equipe ou o profissional trabalha em relativo isolamento dos outros grupos. Cada parte constrói seus próprios simuladores para representar outros subsistemas com os quais seu código precisa interagir. No final da implementação, na hora da integração das diversas partes, muitas vezes, a falta de detalhes essenciais relativas à especificação da interface tais como tipos de mensagens, tipos de retorno, ou seja, elementos que definem a *assinatura* da interface, leva a problemas que impedem desde a compilação do sistema até a execução correta do mesmo.

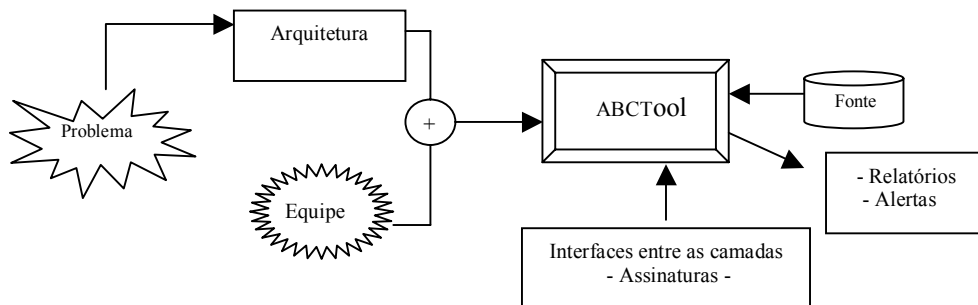
3. ABCTool – Architecture Based Coordination Tool

A ABCTool é uma ferramenta que tem como objetivo coordenar equipes de desenvolvimento de software trabalhando sobre um mesmo projeto, e cuja distribuição de atividades é feita baseada nos componentes de uma arquitetura. Ela permite que as equipes trabalhem paralelamente, implementando os subsistemas da arquitetura e, periodicamente, sincronizem o resultado de seus trabalhos corrigindo conflitos nas interfaces entre esses componentes arquiteturais.

A ferramenta tem como entrada a solução arquitetural escolhida para o sistema em desenvolvimento, as pessoas ou grupos responsáveis pela construção do software e a definição das interfaces entre os diversos componentes da arquitetura. Como saída, ela gera um relatório indicando conflitos de comunicação entre dois ou mais elementos arquiteturais (Figura 1).

A avaliação é realizada quando os componentes desenvolvidos pelos membros ou equipes são armazenados em uma base central de projeto, que armazena todo o software individualmente desenvolvido mas não integrado (este está na base de produção). Baseada nas informações de entrada citadas anteriormente, ABCTool confere o código implementado com as definições gerando um relatório dos problemas detectados e as pessoas envolvidas responsáveis. O processo de coordenação é assíncrono, ou seja, somente quando requisitado é que ocorre a avaliação da existência de conflitos. Outro aspecto importante é que a avaliação ocorre antes que seja feita a integração de toda produção individual de forma a gerar a base de produção atualizada.

Nota-se que a ABCTool não suporta nenhum processo de seleção de arquitetura e gerência de configuração, o que pode ser considerado uma proposta de trabalhos futuros de ampliação da ferramenta.

Figura 1: Visão da Ferramenta ABCTool

4. Metodologia - Estado Atual - Trabalhos Relacionados

Para o desenvolvimento deste trabalho, primeiramente, fizemos um levantamento e estudo da bibliografia disponível. Num segundo momento, elaboramos um questionário aplicado em um grupo de profissionais ligados a Engenharia de Software (incluindo desenvolvedores e usuários) para levantamentos de cenários que identificassem as principais necessidades ou requisitos do trabalho cooperativo no desenvolvimento de software no que tange a desenvolvimento em equipes de elementos arquiteturais. Depois da tabulação e análise dos dados, passamos a modelagem e implementação da ferramenta ABCTool na qual suportamos a gerência de um sistema baseado em uma arquitetura três camadas. Para validar a ferramenta está sendo construído um protótipo de uma aplicação que segue o modelo de três camadas, sobre o qual simularemos as situações conflitantes. Assim, a dissertação encontra-se na fase de implementação. Um protótipo das interfaces já foi desenvolvido em Java. A idéia é que esta ferramenta seja utilizada e acessada via um navegador na web.

Nota-se que ferramentas e métodos têm sido desenvolvidos para trabalhar com a arquitetura do sistema [3,4,7,8,9,10]. Muitos deles são voltados para a seleção da arquitetura a partir de padrões; outros para a descrição dos elementos arquiteturais através de Linguagens de Descrição de Arquiteturas – ADL’s; outros ainda são voltados para a construção de frameworks para famílias de sistemas. Existem também, ferramentas tais como páginas da web, sistemas de controle de versões, e-mails, etc, que apóiam o trabalho cooperativo, ajudando uma equipe a se comunicar. No entanto, todas elas apóiam partes isoladas do desenvolvimento de software, ou seja, apóiam atividades relacionadas à arquitetura ou atividades relacionadas à comunicação entre os membros do grupo. A grande contribuição deste trabalho está em agregar essas funcionalidades.

Os resultados esperados consistem em produzir uma ferramenta de fácil utilização, que contribua para gerenciar o trabalho de uma equipe, minimizando os esforços decorrentes da detecção e correção de conflitos das interfaces dos componentes arquiteturais e, assim, otimizar o tempo de desenvolvimento vinculando as especificações com o que foi efetivamente desenvolvido. Esperamos também integrar, de uma maneira mais efetiva, as equipes de desenvolvimento.

5. Referências Bibliográficas

[1] BORGES, M.,R. S.; ARAÚJO, R., M.; DIAS, M.,S. – “Suporte por Computador ao Desenvolvimento Cooperativo de Software: Classificação e Propostas”; Relatório Técnico; UFRJ, 1997.

- [2] PÁDUA, W. P. F. – **“Engenharia de Software – Fundamentos, Métodos e Padrões”**; Livros Técnicos e Científicos Editora S.A. -LTC, RJ 2001.
- [3] SILVA, M. C. – **“Identificação de Estilos de Arquiteturas: Um Processo Dirigido por Conhecimento”**, Tese de Mestrado – PUC–Rio -Setembro de 2000.
- [4] BACHMANN, F.; BASS, L.; CHASTEK, G.; DONOHOE, P.; PERUZZI, F. – **“The Architecture Based Design Method”**, Technical Report – CMU/SEI-2000-TR-001 ESC-TR-2000-001 – Pittsburgh, PA, 2000.
- [5] SOUZA, C. S.; LEITE, J. C.; PRATES, R.O.; BARBOSA, S. D. J. - **“Projeto de Interfaces de Usuário: Perspectivas Cognitivas e Semióticas”**; Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação, vol. II, 1999, p. 425-476.
- [6] LAU, Y. T. – **“The Art Of Objects - Object-Oriented Design and Architecture”** – MA, 2001.
- [7] LUCENA, C. J. P; BARBOSA, A. C. P. – **“Integração de Frameworks de Software – Monografias em Ciência da Computação”** – N°02/00 – Departamento de Informática – ISSN 0103-9741 – PUC–RIO.
- [8] GARLAN, D. – **“Software Architecture”** – Wiley Encyclopedia of Software Engineering, J. Marciniak (Ed.), John Wiley & Sons, 2001.
- [9] GARLAN, D.; SHAW, M. – **“Software Architecture – Perspectives on an Emerging Discipline”** – An Alan R. Apt Book – New Jersey, 1996.
- [10] GARLAN, D.; KOMPANEK, A. – **“Reconciling the Needs of Architecture Description with Object-Modeling Notations”**, Artigo apresentado nos Procedimentos da Terceira Conferência Internacional da Linguagem de Modelagem Unificada - <<UML>> 2000, York, UK, Outubro 2000.