CoMDD: uma abordagem colaborativa para auxiliar o desenvolvimento orientado a modelos

David Fernandes Neto

CoMDD: uma abordagem colaborativa para auxiliar o desenvolvimento orientado a modelos

David Fernandes Neto

Orientadora: ***Profa. Dra. Renata Pontin de Mattos Fortes***

Monografia apresentada ao Instituto de Ciências Matemáticas e de Computação — ICMC/USP, para o exame de Qualificação, como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP - São Carlos
Janeiro/2012

Resumo

O

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

API *Application Programming Interface*

CDE *Collaborative Development Environment*

1

Introdução

O intrdução: reescrita da proposta

tempo do verbo: presente

Tema (descrição geral da área): como as pessoas desenvolvem software: basicamente uma ide orientada a codigo e um svn da vida. // Geralmente temos no desenvolvimento de software um conjunto de ferramentas necessárias para que ele seja colaborativo. São ides, svns, bugtracks, emails, instant messangers, video/voice conference e por ai vai.

problema: falar dos problemas do desenvolvimento orientado a código e dos svns da vida (merge por ex) // problemas do MDD local: programas necessarios instalados, processo de check-in, check-out muito demorado, ainda gera muitos conflitos-> falar mal dos svns, gits... ainda sao ferramentas que requerem uma curva de aprendizado grande no inicio, principalmente porque são varios os tipos e cada um tem uma abordagem diferente de uso, por exemplo, o svn é diferente do git que é diferente do da microsoft, ainda tudo isso para alguem que nao é desenvolvedora e que trabalha com programação é complicado de acompanhar o projeto. O ideal é deixar o especialista do dominio o mais proximo possível do especialista da solução para evitar problemas de especificao e etc // convergencia para web, videos, musicas, armazenamento de arquivos ... e menos coisas armazenadas localmente //levantar todas as vantagens da wiki e as desvantagens do svn/git ... como as pessoas colaboram na wiki? // qual meu problema: na realidade o problema 1 é: ter que instalar softwares localmente, tendencia de tudo estar na web. // falar dos problemas de desenvolvimento de codigo... e propor mdd como solução daí falar dos problemas de mdd como é feito e propor colaboração aí falar dos problemas de mdd colaborativo (do svn, do git) e propor uma wiki // Quais os problemas de se fazer merge? Atraza em quanto tempo o projeto? // Posso começar

a destacar os problemas de reuso, ou de desenvolvimento, de uma maneira geral, problemas do svn/git, problemas do desenvolvimento de codigo ...

justificativa: assim, com o uso de mdd resolvemos vários problemas como ... e com o uso de wikis resolvemos outros problemas como ..., por isso o objetivo deste trabalho é ... //A importancia do MDD, Vantagens do mdd: por que usa-lo?! -> justifica o uso de mdd // Definição do MDD // Modelos sao colaborativos // Mas como o MDD é usado: stand alone e para compartilhamento de código as pessoas podem usar um svn. Daí vem os problemas dos svns e a justificativa das wikis. // wikis sao as ferramentas que mais promovem colaboração atualmente // no desenvolvimento de sistemas complexos, uma pessoa nao é o suficiente: colaboração para complexidade, mas nao podemos dizer que wiki atende o desenvolvimento de sistemas complexos mdd é bom para sistemas complexos: mdd para complexidade // as pessoas colaboram mais e interagem mais e permite que os stakeholders possam densenvolver ou pelo menos entender melhor o que está-se desenvolvendo (usando a dsl // modelos sao mais faceis de entender e geram mais comunicacao e colaboracao // - Falar: MDD aumenta o nivel de abstracao permitindo que o especialista possa programar e nao somento o desenvolvedor. Uma criança pode programar usando uma wiki mas nao usando eclipse e svn, mesmo que no eclipse seja modelos. A crianca é do dominio e o desenvolvedor o especialista// nivel da solução: java, c .... nivel do problema: UML, p ex

problema do desenvolvimento tradicional -> justifica mdd

uso de mdd com svn -> problema dos svns -> justifica wikis

objetivo: incentivar o uso de mdd e de wikis para desenvolvimento de código e tentar evidenciar que o uso de modelos em uma wiki é possível de ser usado para gerar codigo entre duas pessoas, da mesma forma que uma ide com um svn o faz. Assim, a hipotese deste trabalho é ... // queremos promover o uso de mdd colaborativo, algo que a princípio nao se conhece seu uso, entao entre os objetivos deste trabalho está em promover mdd colaborativo e a partir de uma abordagem simples de seu uso. // nosso objetivo: promover o uso de mdd e de wikis a partir de evidencias de que elas juntas podem trazer produtividade e integração ao inves do modelo tradicional de desenvolvimento // incentivar tbm o uso de mdd //Meu objetivo é incentivar o uso de MDD e de ferramentas colaborativas mais simples, como é a wiki e por isso acredito que teríamos mais produtividade e a aproximação maior de todos os envolvidos no processo, e com isso softwares mais próximos dos que o cliente precisa.

- Hipotese: 1- Código-fonte é mais complicado de ser desenvolvido por mais de uma pessoa, logo é também menos reutilizável. Só o especialista consegue fazer e nao o cara do dominio. -> falar mal do desenvolvimento tradicional

2- Modelos são mais colaborativos, simples, genéricos e reusáveis (nao falar em reuso pra nao criar uma expectativa errada pro leitor) e permitem que não especialistas possam desenvolver/acompanhar; alem de poder-se checar por conformidade os modelos

3- Artefatos online (wikis, gdocs) são mais colaborativos que controles de versao, bug tracks, ou outras ferramentas de gestao de configuração por terem interfaces mais amigáveis -> Assim a Web aumenta ainda mais colaboração

4- LOGO: MDD na Wiki traz mais colaboração, cooperação e produtividade (porque o comdd é mto mais acessível e pela sua simplicidade as pessoas podem colaborar mais facilmente), no desenvolvimento de software -> isso que tenho q avaliar , do que traria com eclipse e svn (ideal de avaliar)

hipótese 2: Para desenvolver um sistema colaborativamente usando uma DSL seria possível usar uma ao invés do processo tradicional de desenvolvimento (IDE+svn)?

hipótese 3: as pessoas usam a wiki para uma série de fatores, como discussoes, documentações. Principalmente em virtude da sua simplicidade e agilidade (de acordo com o dicionario, simplicidade é ..., agilidade é ...), mas será que elas poderiam serem usadas para produzir codigo fonte? Ou seja, sera que as wikis podem servir como uma ide e ainda assim manter suas caracteristicas-/beneficios de agilidade/simplicidade?

Escopo/Limitações: nao é foco deste trabalho comparar o desenvolvimento de codigo com o de mdd uma vez que ha uma vasta literatura sobre isso. Este trabalho parte do principio de que o mdd é vantajoso para o desenvolvimento mais que o desenvolvimento tradicional (segundo ....)

resultados esperados: espera-se que com este trabalho o uso de dsls seja cada vez mais empregado no desenvolvimento de sistemas e de que as wikis evoluam como uma ide para que um dia tenhamos o uso de wikis associadas a dsls para sistemas grandes e complexos.

metodo: para tentarmos evidenciar que nossa hipótese faremos uma avaliação. Será pedido para duas equipes diferentes a mesma tarefa, sendo que uma delas usara o eclipse+svn e a outra uma wiki apenas. Ambas usarão dsls e geração de código, mas nao terao os recursos tradicionais de uma ide como import, highlight sugestao ... Para isso será implementado a dsl no eclipse e numa wiki usando o xtext/xpand e o antlr respectivamente. // Foi desenvolvido um sistema (implementação do comdd) para se realizar um estudo de caso a fim de evidenciar que a hipótese do trabalho é verdadeira ou não

fechar com uma descrição sucinta dos demais capítulos.

esse trabalho sugere o uso de mdd ao inves de linguagens como java, c e etc e o uso de wikis para colaboração ao inves de svns, git.... porque wikis sao extremamente mais simples de entender, mais dinamicas e promovem mais interação–> nao sei se isso é objetivo ou proposta ou hipotese ou sei la o que

2

Trabalhos Relacionados

Artigos

Programas semelhantes já em uso: sugestao do michetti, aquele que o chiquito mandou na lista, ...

Ferramentas de MDD: eclipse, MPS, ...

Ferramentas que auxiliam o trabalho colaborativo (sharepoint)...

3

Revisão Bibliográfica

3.1 Model Driven Development

Pq MDD? Vantagens do uso de Modelos vantagens do mdd

3.2 Domain Specific Languages

3.2.1 Templates

3.3 Wiki

o que é colaboração e quais seus benefícios?

Colaboracao e MDD: SVN

Colaboração na wiki

4

Desenvolvimento

Definição do CoMDD Ferramenta de Modelagem Transformações Suporte a Design Rational: pode ser os comments das páginas? Suporte a dois tipos de comentarios, o da pagina principal e o das alterações Suporte a Controle de Versões: histórico Controle de Acesso merge: plugin criar um bate-papo na wiki highlight na wiki deficiencia: nao compila como uma ide, entao subentende-se que o codigo que ela gera esta sintaticamente correto, mas ela pode retornar erros como integrar testes? é possível alterar a propria gramatica, bem como as transformações ainda pela propria wiki, mas isso seria um script Falar de como é feito o desenvolvimento tradicional colaborativo -> Arquitetura do desenvolvimento tradicional e comparar com a nossa abordagem Arquitetura do CoMDD Vantagens do CoMDD: sao as vantagens da wiki associadas as do mdd DDS Produtividade Comunicação Pair Programming??

ESTUDO DE CASO: CONCLUSAO DO ESTUDO, COLOCAR AQUI o anexo com as perguntas

Arquitetura da implementação/solução.

Método: Colocar a ordem do desenvolvimento dos procedimentos para vibializar a execução na wiki. Mostrar os passos do desenvolvimento.

Escopo do trabalho: Este trabalho foca principalmente em sistemas de pequeno porte, no caso foi testado para uma classe e duas pessoas, mas acredita-se que para algo em torno disso a abordagem funcione também.

Colocar os modelos transformados -> Detalhes no anexo

falar da edição de subartefatos: numa wiki é muito simples voce criar uma pagina dentro de outra [[]]. por exemplo, acho que outras wikis sustentam melhor o merge, ver a wikipedia

Limitações do trabalho: avaliar em que casos pode-se usar uma wiki no lugar de um svn nao é escopo deste trabalho

5

Conclusão

Será que o CoMDD traz inovação qnd comparado ao uso do eclipse+svn? Quais suas contribuições????? Que beneficios em termos de colaboração o comdd traz qnd comparado a abordagens eclipse+svn? Que outros benefícios posso trazer?

As minhas contribuições estao totalmente relacionadas as contribuições do mdd e da wiki, por ex: por que as pessoas nao usam o drupal ou o joomla ou fazem um site para documentação? Da mesma forma. Nao estamos propondo o fim dos svns ou (mas sim das ferramentas locais, pois tudo pode ser web hoje, vide o c9 - ide na nuvem), mas estamos dizendo: olha, com uma wiki e uma dsl vc pode fazer mta coisa e colaborar mto mais. // O principal diferencial deste resultado em relação a outros trabalhos existentes é o aspecto colaborativo, permitindo a edição de modelos e a transformação de software por diferentes desenvolvedores sem a necessidade de ferramentas/ ambientes individuais instalados nas estações de trabalho dos desenvolvedores.

eu tenho que dizer que o estudo de caso foi o suficiente para provar ... (ser bem especifico), mas que para (aqui posso ser mais abrangente) é necessário melhorar a ferramenta (deixar ela mais proxima de uma ide) e partir para problemas mais reais/complexos.

vejam, as wikis nunca foram pensadas para esse fim e por isso podem melhorar para este propósito. usar uma wiki para densenvolvimento que nao seja mdd talvez nao seja tao interessante.

que beneficios o comdd traz quando comparado ao eclipse e svn? nao instalar programas vc ganha agilidade, maior participação, mais pessoas tem acesso a edição de maneira facilitada, ou seja, quando vc facilita o acesso e a edicao vc permite que mais pessoas possam colaborar. É só pensar no contrario para ver que é verdadeiro, pois se o desenvolvedor precisar do dobro de tempo para instalar um svn/ide e os mesmos forem 3 vezes mais dificeis de serem usados temos com isso menos pessoas nao desenvolvedoras participando. Talvez para o desenvolvedor nao importe

o quanto seja dificil usar um svn e o quanto isso va demandar tempo no desenvovlimento de software (na realidade isso importa pq vai atingir diretamente na produtividade), mas para o nao desenvolvedor isso importa muito -> posso colocar uma pergunta assim no questionario dos videos.

conclusão: 100% dos usuarios, quando questionados se para a dada tarefa que abordagem usariam, responderam o CoMDD.

implementações futuras: Edição das transformações e da gramática Retorno de erros (validação do modelo) Edição de subartefatos e que compoem um artefato que ira gerar o codigo Ed em tempo real Como testar o código gerado ou o modelo?

Como validar modelos? Hightlight, auto sugestão e linkagem das palavras -> questoes tecnologicas de implementação Definir a gramática e as transformações-> selado! Retorno de erro-> aparentemente nao é complicado

O CoMDD é uma abordagem que prega mdd colaboartivamente. No caso foi usado uma wiki, mais especificamente a xwiki, mas poderia ser qualquer outra wiki ou plataforma web de simples uso e que promovesse velocidade. Com as atuais tecnologias a wiki é que melhor se encaixa nesse perfil. Em relação ao domínio também, espera-se que para qualquer domínio o CoMDD possa servir, entao no caso a dsl foi para robos autonomos moveis, mas também poderia ser para qualquer outro domínio // O comdd nao exige uma wiki necessariamanete, mas ela é tecnologia atual mais proxima da necessaria. Entretanto, muito ainda deve ser desenvolvido em cima das wikis para que elas possam de fato serem usadas como ferramentas de desenvolvimento de codigo, ao inves de apenas documentação ou afins.-> considerações finais

Este trabalho está longe de criar um ambiente de desenvolvimento tão robusto e avançado quanto hoje são as ides, bug trackers, svns... até porque essas ferramentas já tem anos de maturidade. Este trabalho objetiva sim mostrar que o alinhamente de duas tecnologias aparentemente nao interligáveis é capaz de desempenhar papel semelhante e motivar novos estudos nesse caminho. O ideal seria transformar a wiki numa ide de fato, criar um merge de modelos, permitir o desenvolvimento simultaneo e a edição em tempo real que hoje tem o google. Mas a ideia de que quando voce edita um arquivo esta editando diretamente no servidor e portanto a cópia final agiliza o trabalho evitando conflitos.

é interessante para projetos menores, que tem a interedependencia entre suas classes nao muito grande, com poucas classes e que estas dependem pouco entre si, mas que necessitam fortemente da colaboração de várias pessoas, e se torna mais interessante quando essa pessoas nao sao desenvolvedoras. // ainda eh util para produção de algoritmos, mas que nao foi testado

6

Apêndice A

Antlr: conceitos e explicação da nossa gramática

7

Apêndice B

CoMDD: como implantar uma dsl em uma wiki?

8

Apêndice C

Eclipse: xtext/xpand