Um Simulador Estocástico de Processo de Software Baseado em Conhecimento

Breno Bernard Nicolau de França[1,2], Rodrigo Quites Reis[1,2]

[1]Programa de Pós-Graduação em Ciência da Computação
Universidade Federal do Pará (PPGCC-UFPA)
Belém – PA – Brasil

[2]Laboratório de Engenharia de Software - Universidade Federal do Pará
Belém – PA – Brasil

breno@webapsee.com, quites@computer.org

Abstract. *The process behavior understanding capability, besides the opportunity to previously infer the impact of process changes, make simulation a potential decision support tool to software organizations. This paper presents a software process simulation model based on the concepts of Software Process Historical Data, Case-Based Reasoning (Analogy) and Monte Carlo Simulation. The current model is implemented and integrated to an Automated Environment, which enables the extraction of previous enactment data.*

Resumo. *A capacidade de entendimento de como o processo se comporta em termos de desempenho, além da oportunidade de verificação "a priori" do impacto de mudanças no processo, tornam a simulação uma ferramenta com potencial para tomada de decisão nas organizações. Este artigo apresenta um modelo de simulação de processo de software baseado nos conceitos de Histórico de Processo de Software, Raciocínio Baseado em Casos (analogia) e Simulação de Monte Carlo. O modelo encontra-se implementado e integrado em um Ambiente Automatizado, explora os recursos já disponíveis na plataforma para extrair e executar modelos de simulação com base no histórico da organização.*

1. Introdução

Grande demanda tem surgido, tanto na Academia quanto na Indústria, por processos organizacionais definidos para apoiar o desenvolvimento de software. Isto é evidenciado em trabalhos científicos como [Jacobson, Booch e Rumbaugh 1999] e em editais de contratação de projetos de software, inclusive do Governo [TI Inside 2007], onde padrões de qualidade e maturidade em desenvolvimento de software são exigidos.

Essa demanda ocorre em função da qualidade do produto ter uma íntima relação com a qualidade do processo executado para construir este produto [Fuggetta 2000]. Por isso, iniciativas como o MPS.Br [Softex 2008] visam, assim como CMMI [CMU/SEI 2006] e normas ISO [ISO/IEC 2004], estabelecer boas práticas e padrões de qualidade e maturidade em software, certificando e avaliando organizações que desenvolvem software quanto à sua capacidade de produzir resultados planejados e com

alta qualidade. Embora exista grande esforço pela qualidade, muitos projetos de software falham ou ultrapassam custos e prazos por problemas de planejamento [Little 2006].

Os motivos comuns para a utilização da simulação de modelos de processos recaem, segundo [Kellner, Madachy e Raffo 1999], em seis grandes categorias: 1) gerência estratégica, 2) planejamento, 3) gerência operacional e de controle, 4) melhoria de processo e adoção de tecnologia, 5) compreensão e treinamento, e 6) aprendizado.

A simulação tem sido referenciada, ainda, como uma ferramenta útil para alcançar os mais altos níveis do CMMI [Raffo, Vandeville e Martin 1999] e como técnica viável para implementação das práticas do nível B do MPS.Br [Softex 2006]. Isto ocorre devido sua abordagem quantitativa que facilita a análise e controle do desempenho dos processos de desenvolvimento das organizações, áreas chave dos níveis 4 e 5 (CMMI) e B e A (MPS.Br). Já para níveis iniciais (CMMI nível 2 e MPS.Br nível E), a modelagem da simulação de processo de software ajuda no planejamento e definição dos seus processos, além das mudanças ocorridas durante a definição.

Os ambientes e modelos de simulação propostos atualmente possuem muitas simplificações, gerando resultados que não condizem com a realidade das organizações de software hoje em dia. Além disso, poucas abordagens para simulação de processos de software são independentes do processo utilizado. As limitações inseridas fazem com que muitos modelos de simulação estejam voltados para um processo de software adotado por uma organização em específico, a fim de representar com mais fidelidade a execução do processo adotado na realidade.

Dentre as propostas que possuem características mais abrangentes e, ainda assim, conseguem uma boa representação do modelo real, as abordagens híbridas apresentadas em [Martin and Raffo 2000] e em [Lakey 2003] são as que, atualmente, obtêm melhores resultados. Entretanto, exigem um grande esforço para definir o modelo de simulação para os processos de software reais.

Mesmo as propostas de modelos generalizados de simulação de processo de software [Raffo, Nayak e Wakeland 2005], onde são utilizados componentes reutilizáveis de modelos de processo para facilitar a modelagem, a incorporação de funções de distribuição de probabilidade para gerar os valores das variáveis aleatórias, que representam o comportamento da execução do processo, é realizada de maneira não integrada, ou seja, supõe-se que esta análise já tenha sido realizada.

O avanço deste trabalho está associado ao fato com a possibilidade de "previsão" do comportamento das três grandes variáveis do processo de desenvolvimento de software: **custo**, **tempo** e **qualidade do produto gerado**, além de variáveis auxiliares que ajudam na determinação dos valores das variáveis anteriores. Esta "previsão" é realizada em curto tempo através de um Simulador de Processo de software. Este reúne abordagens apresentadas na literatura, conforme os trabalhos relacionados e o estado da arte apresentados na seção 3, em uma abordagem baseada em conhecimento para a solução dos problemas de **calibragem** e **não determinismo** que envolvem os modelos de simulação atuais, além da **independência** do modelo de processo a ser simulado. Além disso, no modelo proposto, os dados históricos de

execuções anteriores obtidos a partir de um repositório estruturado de um PSEE fornecem subsídio para simulação utilizando o Método de Monte Carlo (MMC) [Goldsman 2007].

O restante do texto está organizado da seguinte forma: a seção 4 apresenta o modelo proposto e discute algumas decisões tomadas na solução, a seção 5 apresenta um exemplo contextualizado de como o modelo proposto funciona, a seção 6 apresenta as considerações finais e as direções futuras deste modelo de simulação.

2. Contextualização

Este trabalho busca contribuir com avanços no escopo da área denominada Tecnologia de Processo de Software [Derniame 1992], cujo foco está na automação de processos de software do ponto de vista da gerência, execução e engenharia de processos.

A proposta desse modelo de simulação é baseada principalmente em três conceitos: 1) Simulação de Monte Carlo, 2) Estimativa por Analogia - Raciocínio Baseado em Casos (RBC) – e, 3) Histórico de Execução de Processos de Software.

O Método de Monte Carlo consiste na formulação de um processo estocástico, o qual produz uma variável aleatória cujo valor esperado é a solução de um problema. Uma aproximação do valor esperado é então obtida a partir de uma amostra da distribuição de probabilidade resultante. A solução é considerada uma aproximação pelo fato de que sua base (dados que originam as distribuições) seja uma fonte de erro nas distribuições, pois o número de elementos na amostra é finito. Assim, a acurácia do método é diretamente proporcional ao tamanho da amostra de entrada [Bauer 1958].

A Simulação de Monte Carlo (SMC) modela um problema como um sistema, o qual é descrito como um conjunto de variáveis aleatórias regidas por alguma distribuição de probabilidade, de onde são gerados valores (pseudo) aleatórios para representar o comportamento de um sistema real. Estes valores gerados fazem parte da solução do problema a ser resolvido, seja esse valor parcial ou final. Cada número aleatório gerado é uma fonte de incerteza. Esta incerteza é acumulada até o final da execução do método (simulação) e pode ser tratada, desde que se conheça sua grandeza.

A técnica de Estimativa Baseada em Analogia [McConnel 2006], muito utilizada para estimar esforço e prazo de projetos de software, parte do princípio de criar uma estimativa com um bom grau de acurácia para um novo projeto através da comparação com projetos anteriores similares. Este problema possui todas as características de um problema a ser resolvido utilizando a técnica de RBC.

Delany *et al* [Delany, Cunningham e Wilke 1998] descrevem, em um estudo abrangente, as principais limitações para a técnica de RBC em resolver o problema de estimativa de esforço no desenvolvimento de software. Dentre elas se destaca a falta de atributos concretos para se comparar projetos de software em sua fase inicial (planejamento). Em um trabalho posterior, Delany e Cunningham (2000) apresentam uma proposta de um modelo RBC para estimativas de esforço com uma série de direcionadores de custo para caracterizar um projeto de software desde sua concepção.

Os experimentos realizados por [Shepperd, Schofield e Kitchenham 1996] e [Shepperd e Schofield 1997] com a ferramenta ANGEL utilizaram nove bases de dados

diferentes de projetos de software. Para validar a abordagem baseada em analogia, foi utilizada a abordagem *Jack Knifing*, onde se retira uma unidade (um projeto neste caso) da base de dados e tenta-se realizar a estimativa para esta unidade com base nas unidades restantes. Então, o resultado da estimativa é confrontado com os resultados alcançados desta unidade (neste caso esforço real necessário).

A coleta de métricas durante a execução do processo é um requisito muito citado para PSEEs (*Process-centered Software Engineering Environments*) [Bandinelli *et al* 1994] e para desenvolvimento de software em geral. Entretanto, este requisito nem sempre tem sido tratado com eficiência, ou seja, utilizando as métricas de forma adequada, o que tem causado uma falta de interesse em manter os esforços de coletar métricas. Este conhecimento (histórico dos processos) pode ser utilizado para ajudar na tomada de decisões para alocar pessoas e recursos no processo, além disso, utilizado para simulação de processo [Scacchi 1999].

Para a simulação de processo de software é importante dar um enfoque maior na meta-atividade de execução de processos, pois se trata do comportamento que a simulação se propõe a "imitar".

Muitos dos problemas que surgem no decorrer da execução de um processo de software podem ser detectados durante a simulação. Por exemplo, a utilização excessiva de recursos da organização, atividades realizadas seqüencialmente quando poderiam ser paralelizadas, entre outros. A simulação de processos deve, assim como a execução, considerar aspectos organizacionais, tais como: pessoas envolvidas, recursos utilizados, habilidades dos agentes, afinidade entre os agentes ao realizarem tarefas em grupo, políticas da organização, entre outros.

3. Trabalhos Relacionados

Várias abordagens vêm sendo experimentadas pela literatura, com diferentes propósitos, desde o auxílio no diagnóstico "a priori" [Mi e Scacchi 1990], passando pela análise e desempenho de processos [Martin e Raffo 2001] até no desenvolvimento de jogos interativos que auxiliam no treinamento de gerentes novatos ou estudantes [Drappa e Ludewig 2000] [Dantas, Barros e Werner 2004]. Entretanto, apenas três modelos de simulação são comentados devido restrições de espaço.

3.1 Articulator

Dentro da categoria de simuladores integrados à PSEEs, um dos trabalhos pioneiros é definido por Mi e Scacchi (1990). O *Articulator* é um ambiente com apoio à modelagem e simulação de processos que utiliza uma abordagem baseada em conhecimento.

O modelo do *Articulator* é baseado em uma rede de recursos (materiais e pessoal) que estabelecem restrições de dependência entre as tarefas. Cada tarefa é realizada por um conjunto de agentes inteligentes, cujo comportamento é baseado em estratégias pré-definidas para lidar alocação de recursos e decisões relacionadas a sua capacidade de lidar com um número elevado ou baixo de carga de trabalho.

A cada ciclo da simulação, também chamado de estado, uma série de ações são executadas pelos agentes inteligentes. Estas ações exigem que recursos estejam

disponíveis para serem realizadas e um conjunto de ações configuram um estado. Existe um mecanismo de resolução de conflitos para resolver conflitos na utilização dos recursos. Neste simulador também existe o mecanismo de busca que realiza consultas otimizadas no repositório de processos modelados para recuperar informações durante a simulação e um mecanismo para aquisição de conhecimento que recupera informações sobre uma base estruturada de conhecimento adquirido com o passar das simulações.

3.2 Illium

A ferramenta de simulação *Illium* [Barros, Werner e Travassos 2000b] permite a construção e avaliação de modelos dinâmicos de projetos de software. O modelo é representado textualmente e o formalismo usado na sua definição segue a técnica de dinâmica de sistemas. Além disso, utiliza também o paradigma de gerência de projetos de software baseada em cenários [Barros, Werner e Travassos 2000a], estendendo o modelo de dinâmica de sistemas originalmente proposto por Abdel-Hamid e Madnick (1991) para acomodar a utilização deste paradigma.

Três formas de conduzir a simulação são adotadas pela ferramenta *Illium*: 1) a simulação baseada em tempo, onde a cada ciclo da simulação ocorre um registro das dos valores correntes das variáveis do processo; 2) a simulação estocástica utilizando o Método de Monte Carlo e 3) a variação de parâmetros da simulação, que analisa o impacto no comportamento da execução do processo a partir de modificações realizadas sobre os valores dos parâmetros da simulação.

3.3 Simulação Híbrida

O modelo apresentado em [Martin e Raffo 2000] e [Martin e Raffo 2001] é descrito de forma híbrida, isto é, composto por partes contínuas (dinâmica de sistemas) e partes discretas (evento-dicreta). O processo de software é modelado de forma discreta como conjunto de atividades conectadas em seqüência e o ambiente do projeto e suas variáveis transversais são modelados de maneira contínua de acordo com o modelo proposto por [Abdel-Hamid e Madnick 1991].

As atividades do processo são modeladas de acordo com três tipos genéricos: desenvolvimento, inspeção e re-trabalho. As atividades de desenvolvimento são as que produzem ou modificam atributos artefatos, tais como número de erros inseridos, tamanho, entre outros. As atividades de inspeção servem para detectar erros e as de re-trabalho para corrigir os erros injetados em atividades passadas, ou ainda, injetar novos erros por conta de correções mal sucedidas.

O esforço de cada atividade é calculado através de uma distribuição de porcentagens entre as fases do processo de um total que, por sua vez, é calculado através do modelo COCOMO [Boehm 1981]. Quanto às durações das atividades, é utilizado um fator de produtividade associado a cada pessoa envolvida no processo e o número de pessoas alocadas à atividade. Questões de qualidade são representadas por taxas contínuas, para mais detalhes sobre as equações, consultar [Martin e Raffo 2000].

Este modelo abrange ainda questões de paralelismo de atividades, porém com poucas alternativas para modelagem da dependência temporal entre as atividades se comparado ao modelo de conexões disponível atualmente em ambientes de

desenvolvimento de software centrados no processo (tal como apresentado em [Lima Reis 2003]).

A produção de artefatos nas atividades é representada por filas que traçam o progresso dos itens individualmente. Por sua vez, os recursos são apresentados de forma bastante simplificada, enfatizando apenas o seu papel na habilitação ou não a execução de atividades, em função da sua disponibilidade.

Quanto aos aspectos do ambiente do projeto, as três principais variáveis (em mais alto nível) modeladas são: o **nível de pessoal** em termos de disponibilidade e utilização, com isso, é possível identificar casos de super e sub-alocação de pessoal; a **taxa de produtividade**, utilizada no cálculo de duração das atividades; e as **taxas de erros**, para geração, detecção e correção de artefatos.

3.4 Diferencial da proposta

Os principais diferenciais do simulador aqui apresentado em relação aos outros trabalhos da literatura são: a **extração** do modelo preditivo para realizar simulações (estocásticas) e a integração a um Ambiente de Desenvolvimento de Software Centrado em Processo.

A extração a partir da análise de histórico de execuções passadas incorpora as incertezas inerentes ao processo de desenvolvimento de software por meio das funções distribuição de probabilidade e utilização da Simulação de Monte Carlo. A integração com o ambiente WebAPSEE [Lima *et al*. 2006] faz com que todos os dados necessários (incluindo métricas) a respeito dos modelos de processo, projetos e da organização estejam disponíveis para utilização no modelo de simulação.

4. O Modelo Proposto

As sub-seções seguintes apresentam: 1) a correspondência do modelo de simulação aqui apresentado com os modelos de maturidade de desenvolvimento de software, 2) uma visão geral do modelo e, 3) a forma como este utiliza o conhecimento armazenado no repositório de histórico de execuções de processo.

4.1 Evolução Recente

O modelo deste trabalho é uma evolução do modelo apresentado em [França 2007], entretanto, com algumas diferenças. O modelo COCOMO II [Bohem *et al* 2000] não é mais utilizado para cálculo de estimativas de esforço e duração das atividades. Ao invés deste modelo analítico, o modelo desta versão é estocástico no que diz respeito à determinação da duração de atividades e determinação de valores como produtividade, taxa de erros inseridos, entre outros. Esta característica é desejável considerando a dinâmica e variabilidade do processo de software.

4.2. Correspondência com Modelos de Maturidade

Os modelos de maturidade em desenvolvimento de software como CMMI, ISO e MPS.Br possuem, em geral, áreas-chave destinadas à definição dos processos organizacionais, implementação de planos de medição, bem como áreas destinadas à

gerência quantitativa de seus projetos e, em maior grau de maturidade, a melhoria contínua dos processos padrões da organização [Raffo, Vandeville e Martin 1999].

Para definição do processo padrão para organização é necessário o conhecimento quanto aos recursos disponíveis na empresa, sejam estes materiais ou pessoais, artefatos de software produzidos e consumidos durante a execução do processo e ferramentas a serem utilizadas a fim de produzir estes artefatos. A alocação destes componentes do processo configuram um problema de tomada de decisão que pode ser resolvido com auxílio de simulação. Além disso, problemas de paralelismo entre atividades para otimização de recursos e tempo são facilmente visualizados com a utilização de abordagens discretas de simulação [Martin e Raffo 2000].

Nos níveis mais altos do CMMI e MPS.Br, por exemplo, existe a necessidade de gerenciar de forma quantitativa os projetos executados, permitindo uma avaliação contínua do desempenho dos processos da organização [Softex 2006]. Isto é, a gerência de projetos está integrada com o processo de medição a fim de utilizar as métricas coletadas durante a execução. Com isso, é possível estabelecer limites de controle para o processo e medidas para monitorar o desempenho do mesmo, permitindo que ações preventivas e corretivas possam ser tomadas de maneira pró-ativa, ou seja, antes que grandes problemas ocorram de fato.

Com a coleta de métricas alinhada aos objetivos de monitoração do processo, é possível obter dados para utilizar em modelos de simulação (como o proposto nesse trabalho) a fim de melhorar a tomada de decisão na organização, prever impacto de modificações no processo e justificar iniciativas de melhoria de processo [Raffo 1999].

4.3. Visão Geral do Modelo

Segundo a classificação apresentada por Goldsman (2007), a abordagem de simulação utilizada é classificada como **dinâmica** - pela natureza dinâmica do sistema modelado (processo de software), **estocástica** - pela utilização do MMC e não ser determinística, e **contínua** - devido ao tempo avançar em pequenos intervalos de tempo constantes.

A abordagem aqui proposta para simulação é realizada, unindo os três conceitos apresentados na seção 2, em três momentos:

1. No início da simulação é utilizada a técnica de analogia para determinar atividades passadas semelhantes para cada atividade do processo a ser simulada [França 2007]. Esta etapa é um *setup* da simulação, onde o modelo preditivo será extraído da base histórica (seção 4.4) para simular o processo dado como entrada (processo instanciado que se deseja simular);
2. A partir das atividades semelhantes recuperadas, é montada uma Função Distribuição de Probabilidade (FDP) para representar o comportamento de cada variável (duração, número de erros detectados, entre outras) para cada uma das atividades passadas. Estas variáveis determinam a dinâmica da execução do processo;
3. Ao final, é executada a simulação utilizando o MMC com base nas FDP originadas. Esta execução faz uso do Mecanismo de Execução do ambiente WebAPSEE [Lima Reis 2003], onde os eventos de execução

(início/termino de uma atividade) são disparados de acordo com os valores gerados pelas FDPs que representam a duração de cada atividade.

A estrutura da base histórica (repositório de processos executados) utilizada nesse modelo é a da base de processos executados no ambiente WebAPSEE.

A simulação de Monte Carlo funciona, nesse modelo, da seguinte forma: o modelo do sistema (processo) utilizado é o próprio modelo de processo, definido no ambiente WebAPSEE. Cada atividade é um processo (termo da simulação orientada processo), que requer um tempo para ser executado, como um caixa de atendimento em um serviço bancário. Os recursos requeridos e utilizados podem ser materiais e humanos e possuem um custo por unidade associado. Os artefatos produzidos e consumidos em cada atividade possuem atributos como tamanho e complexidade, por exemplo.

O modelo possui diversas variáveis aleatórias para representar os valores da execução do processo, tais como: duração de atividades, produtividades dos agentes na realização de tarefas e disponibilidade de recursos. Estas variáveis aleatórias têm seu valor determinado de acordo com a distribuição de probabilidade que rege seu comportamento. Por exemplo, uma atividade de revisão de um documento de arquitetura tem a duração estimada (em dias) de acordo com uma distribuição normal com média 5 e desvio 1,3. As distribuições são determinadas de acordo com o histórico de execução de atividades similares à atividade em questão (ver seção 4.3). A cada ciclo (intervalos pequenos e constantes) do relógio da simulação, as variáveis são atualizadas com os valores gerados pelas distribuições.

Com a simulação iniciada, é possível realizar diversas execuções/observações consecutivas e analisar um padrão ou tendências no comportamento da execução e no desempenho do processo. Essa tendência pode ser observada em alta granularidade, por exemplo, o aumento do custo do projeto em uma fase do projeto, ou em menor granularidade, por exemplo, na observação de uma atividade importante que represente um gargalo no processo em termos de duração, causando atrasos no projeto. Convém observar que este trabalho é um instrumento para apresentar estes indicadores, enquanto que o processo de resolução está fora do escopo da solução vislumbrada.

4.4. Utilização do Conhecimento Organizacional

Esta seção apresenta de que forma o mecanismo RBC realiza o tratamento das informações contidas no histórico de processos para gerar o modelo estocástico de simulação, ou seja, a base para a Simulação de Monte Carlo.

O esquema do funcionamento integrado do mecanismo RBC com a análise dos casos similares para gerar as distribuições de probabilidade é apresentado na Figura 1.

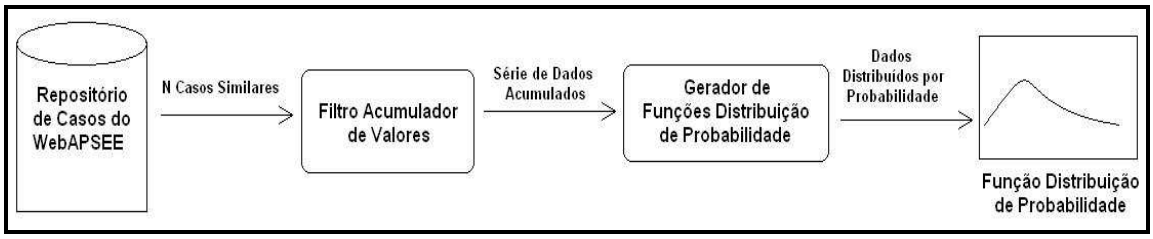

Figura 1 – Funcionamento integrado RBC e Análise de Dados.

Para cada atividade de um processo, uma coleção de atividades similares anteriores é gerada a partir da base histórica (repositório WebAPSEE). Estas atividades são, então, filtradas pelo modelo RBC através de um filtro acumulador de valores, que para cada atributo da atividade relevante para o comportamento da simulação (duração, custo, qualidade dos produtos, entre outros) gera uma série dados (valores) acumulados.

A partir destas séries são geradas as Funções Distribuição de Probabilidade para cada atributo das atividades do processo. Estas distribuições são responsáveis pela geração de números aleatórios atribuídos às variáveis do modelo de simulação, determinando a dinâmica para execução de cada atividade durante a simulação.

Outras variáveis podem ter seu comportamento dado de acordo com uma distribuição baseada em dados históricos, tais como: produtividade, aprendizado durante o processo, grau de habilidade e afinidade dos desenvolvedores, taxa de disponibilidade de recursos, tempo médio entre falha/defeito de recursos materiais, entre outras. Isso diferencia este modelo dos modelos baseados em Dinâmica de Sistemas, onde as taxas que representam essas variáveis são fixas, dadas por rígidas equações diferenciais.

A idéia de utilizar RBC na inicialização da Simulação de Monte Carlo parte do princípio de que esta última técnica é utilizada há muitos anos com sucesso em contextos industriais que possuem produção e serviço em larga escala, por exemplo, manufatura, linhas de montagem e logística [Goldsman 2007]. Nestes casos, os processos são bem definidos e as atividades dos processos são sempre similares, chegando a serem repetitivas (dependendo do caso). Assim, a técnica RBC na base histórica funciona como um filtro que resulta somente atividades similares como em um regime de Linha de Produção de Software [Clements 2005], ou seja, somente atividades que possuem um histórico e que já foram executadas anteriormente, seja em um contexto igual ou relativamente similar.

Apesar da similaridade estar presente no modelo, isto não significa que todos os processos na base histórica devam ser originados obrigatoriamente de um processo padrão. Esta similaridade é medida em nível de atividade (considerando o contexto do projeto), e não em nível de processo. O princípio é que pode haver atividades similares inclusive em processos diferentes.

Um sistema RBC é constituído de 4 partes: 1) Representação dos Casos, 2) Função de Similaridade, 3) Método de Recuperação e 4) Método de Adaptação.

Para este trabalho foi utilizada a representação de casos de maneira **orientada a objetos** (assim como em [França 2007]), visto que o sistema e seu modelo de dados já apresentam as informações estruturadas neste paradigma. Entretanto, o modelo proposto

possui informações de contexto do projeto (identificação, descrição e duração). Quanto à função de similaridade, é apresentada de maneira similar à versão anterior: um somatório ponderado do grau de similaridade entre os atributos/relacionamentos multiplicados pelos pesos. O método de recuperação continua o mesmo, a **recuperação em dois níveis** [Von Wangenheim 2003] [França 2007].

A grande inovação do modelo RBC é que este é utilizado como um pré-processamento para a simulação, ou seja, atua na extração do modelo preditivo e não mais como fator determinante da duração de atividades durante a simulação.

O método de Adaptação, onde concentra grande parte das modificações do modelo proposto, é chamado de **Composicional**. Na adaptação composicional, o problema atual (atividade a ser simulada) terá sua solução baseada na solução utilizada pelos casos mais similares contidos na base de casos. Segundo Von Wangenheim (2003), neste método "os novos componentes de solução adaptados de vários casos anteriores são combinados para produzir uma nova solução composta". Esta combinação não, necessariamente, precisa ser estrutural[1], mas para compor novos valores aos atributos conhecidos com base em valores de vários casos anteriores. A adaptação composicional da solução proposta ocorre da seguinte forma.

Sendo a solução composta pelo **custo** e **tempo** necessários para executar a atividade. O custo é a soma dos custos dos recursos requeridos (consumíveis) adicionada do custo por hora de cada um dos agentes envolvidos (multiplicado pela quantidade de horas - **tempo**). O tempo é determinado a partir do conjunto de durações das **n** atividades mais similares recuperadas da base de casos. Por exemplo, na Tabela 1 é apresentada (primeira coluna) uma série de atividades semelhantes a uma qualquer **Atividade A** e as respectivas durações de cada uma quando foi executada (primeira coluna). É este conjunto de dados (de durações da segunda coluna) que serve de base para adaptar um conjunto de casos anteriores para uma solução de um caso novo.

Tabela 1. Atividades similares e suas respectivas durações.

Atividade Similar	Duração
Atividade B	7
Atividade C	5
Atividade D	8
Atividade E	5
Atividade F	6

A solução parte da idéia de que problemas similares possuem soluções similares, ou seja, o tempo necessário para executar a nova atividade será similar ao tempo referente às atividades similares.

[1] Em termos de estrutura de dados.

5. Exemplo de Funcionamento do Modelo

Nesta seção um exemplo simples (focado em apenas uma atividade) é apresentado para ilustrar o funcionamento de como são extraídas as informações do modelo de maneira geral. Parte de um modelo de processo é apresentada na Figura 2 na notação do WebAPSEE. Esta visualização é a mesma utilizada quando da modelagem e execução da simulação, entretanto, é possível acrescentar parâmetros na inicialização (restrições de custo, duração e qualidade) e gerar relatórios das saídas da simulação.

O modelo da figura 2 representa parte de um processo-exemplo de elicitação e análise de requisitos. A **atividade-problema** foco para este é exemplo é "Especificação de Requisitos", cujo estado está "*Ready*", ou seja, pronta para executar.

Para esta atividade (e seus atributos) são recuperados da base de casos (repositório de processo do WebAPSEE) os quinze (15) casos mais similares, considerando um limiar de similaridade, que pode ser determinado ou como entrada do usuário - especificando o nível de acurácia dos resultados ou por um limite médio dos valores de similaridades calculados entre esta atividade e suas similares, por exemplo. Dadas as quinze atividades similares (conjunto de **atividades-caso**) com seus respectivos graus de semelhança aferido, por exemplo, acima de 65% de semelhança, a duração estimada da **atividade-problema** será gerada randomicamente a partir de uma FDP extraída das quinze **atividades-caso** similares, com o grau de acurácia em torno de 65%.

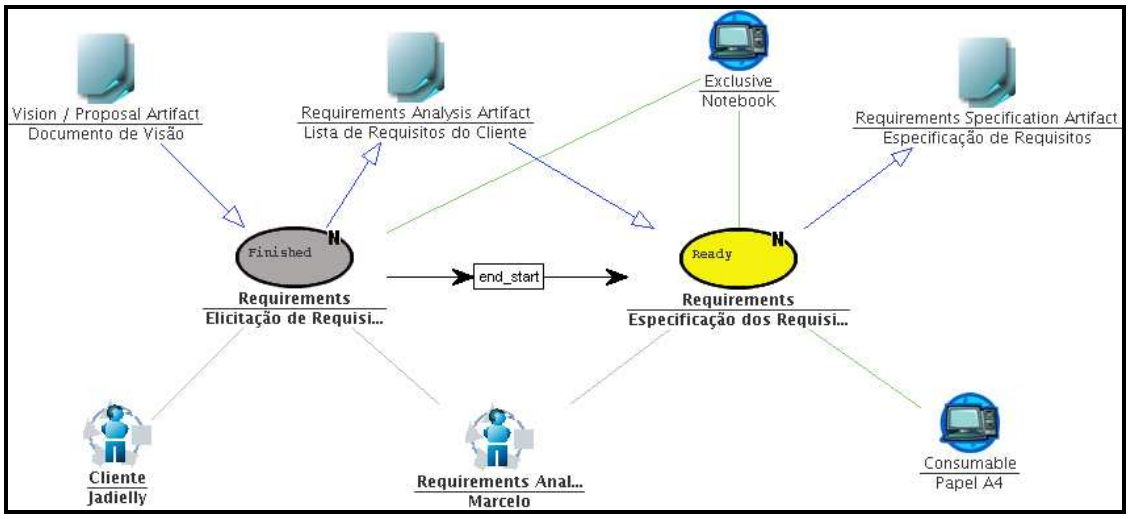

Figura 2. Modelo de Processo Exemplo.

Convém ressaltar que o limiar de similaridade (neste exemplo, 65%) é função do grau de acurácia geral desejado para a simulação (definido pelo usuário). A similaridade entre cada atividade é dada por uma soma ponderada da similaridade local entre os atributos destas atividade. Esta similaridade local é determinada pela comparação entre os atributos, na qual o método varia dependendo do tipo do atributo (texto, número, hierarquia de objetos, entre outros).

Durante a determinação da distribuição de probabilidade da duração (em dias), por exemplo, não é correto que uma atividade mais similar (90%) tenha mesma

probabilidade de ocorrência na distribuição que uma atividade menos similar (75%), ou seja, dizer que a duração da **atividade-caso** pode ser aproximadamente igual à duração da atividade menos similar com mesma probabilidade da mais similar. O que ocorre é que quanto mais se executa uma atividade com os mesmos recursos, produzindo os mesmos artefatos (com tamanhos equivalentes) e realizadas por pessoas de mesmo papel/cargo a tendência é que os valores das durações se aproximem. Assim, é preciso estabelecer pesos maiores para as atividades mais similares e a forma de fazer isso é inserir estes pesos na distribuição através do aumento da probabilidade de ocorrência.

A série de dados utilizada para gerar distribuições é não-ordenada e com possibilidade de dados replicados. Para considerar que, por exemplo, o caso mais similar tenha maior probabilidade, o seu valor é replicado na série em **X** vezes, onde **X** é um fator determinado em função dos graus de similaridade relativos.

Uma vez gerada a distribuição, é possível gerar os valores aleatoriamente para as variáveis da simulação. Neste exemplo, é apresentada a duração em dias para a atividade de especificação de requisitos (figura 3).

Segundo a distribuição apresentada na figura 3, a probabilidade de ocorrência de valores no intervalo fechado de 10 a 14 é maior que os outros valores. Para esta distribuição, o valor da media é de 10.7, com desvio de 2.97 e extremos 5 e 15. Por exemplo, ao gerar vinte números aleatoriamente a amostra resultante é mostrada no Quadro 1 a seguir, comprovando a maioria (treze em vinte) de ocorrência de valores entre 10 e 14.

11.0 11.0 7.0 9.0 7.0 7.0 14.0 10.0 12.0 5.0
12.0 5.0 11.0 13.0 11.0 5.0 10.0 13.0 12.0 12.0

Quadro 1. Amostra aleatória gerada a partir da distribuição.

A análise a ser feita é que se, em execuções passadas, a duração das atividades mais similares ocorreu em sua maioria em um dado intervalo (como no exemplo: entre 10 e 14), é mais provável que em uma nova execução a duração esteja neste intervalo. Entretanto, nada impede que a nova duração esteja fora deste intervalo, por isso a característica estocástica da Simulação de Monte Carlo é desejada. Assim, com a abordagem estocástica é possível representar as incertezas inerentes à natureza dessas variáveis através dos geradores de números aleatórios.

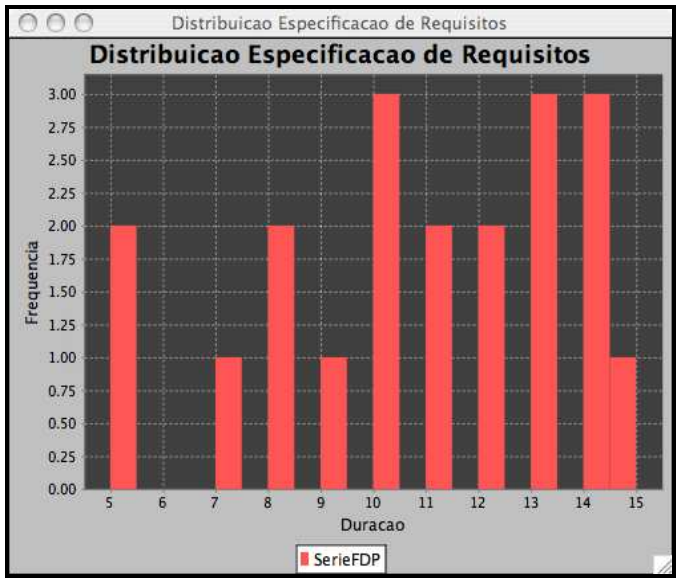

Figura 3. Distribuição de Duração para a Atividade de Especificação de Requisitos.

6. Considerações Finais e Trabalhos Futuros

A utilização da abordagem baseada em conhecimento na definição do simulador integrado ao ambiente WebAPSEE possui pontos fortes no que diz respeito á calibragem do modelo com os dados da organização. Essa integração dispensa métodos que consomem muito tempo para calibrar um modelo preditivo com os dados organizacionais, pois o modelo é extraído diretamente destes dados.

Como o modelo de simulação é gerado a partir de bases históricas detalhadas, isto é, com uma granularidade refinada do processo, as simplificações comumente encontradas nos modelos de simulação atuais não se fazem necessárias. Isto agrega possibilidade de incorporar mais informações ao modelo, tornando-o cada vez mais próximo da realidade no que diz respeito a sua acurácia. Entretanto, trabalhos adicionais precisam ser realizados para comprovar este benefício, como comentado a seguir.

As abordagens mais recentes apresentam resultados advindos de modelos híbridos de simulação [Martin e Raffo 2000]. Mostrando que para atingir um bom grau de completude na representação de processos reais é necessária a utilização de diferentes abordagens integradas, pois um processo é composta de variáveis discretas e contínuas. Pos isso, as distribuições geradas para Simulação de Monte Carlo podem ser tanto discretas quanto contínuas. Por exemplo, enquanto uma variável como tempo pode ser definida por uma distribuição contínua como uma **Normal** (Ou **Gaussiana**), uma variável discreta que represente o numero de defeitos encontrados em uma atividade de revisão como uma distribuição uniforme.

A definição e implementação deste modelo é recente e não foi possível a realização de testes reais para uma validação da proposta. Como trabalho futuro, pretende-se planejar simulações para auxiliar na melhoria, estudos comparativos e análise de processos reais. Além da implementação recente, a indisponibilidade de uma base histórica real de projetos executados inviabilizou a validação deste modelo diante

das expectativas quanto à sua acurácia. Entretanto, a técnica a ser adotada para a validação é a técnica *Jack Knifing* [Shepperd e Schofield 1997], retirando um projeto da base de projetos e simulando-o com os dados históricos dos projetos restantes. Após a simulação, compara-se os resultados da simulação com os resultados reais do projeto.

Referências

Abdel-Hamid, T.; e Madnick, S. (1991) Software Project Dynamics: An Integrated Approach, Facsimile Edition, Prentice-Hall.

Bandinelli, S.; Di Nitto, E.; Fugetta, A.; Lavazza, L. (1994) Coupled vs Decoupled User Interaction Environments in PSEEs. In *9th International Software Process Workshop*, Reston.

Barros, M.O.; Werner, C. M. L.; Travassos, G.H. (2000a) Applying System Dynamics to Scenario Based Software Risk Management, In *International System Dynamics Conference*, Bergen, Norway.

Barros, M.O.; Werner, C. M. L.; Travassos, G.H. (2000b) "*Illium*: Uma ferramenta de Simulação de Modelos Dinâmicos de Projetos de Software", In *XIV Simpósio Brasileiro de Engenharia de Software*, Caderno de Ferramentas, p. 355-358, João Pessoa, PB, Brasil.

Bauer, W. F. (1958) "The Monte Carlo Method". In: Journal of the Society for Industrial and Applied Mathematics, volume 6, number 4, pages 438-451, December.

Boehm, B. (1981) Software Engineering Economics. Prentice Hall, 1st edition.

Boehm, B.; Abts, C.; Brown, A. W.; Chulani, S.; Clark, B. K.; Horowitz, E.; Madachy, R.; Reifer, D.; Steece, B. (2000) Software cost estimation with Cocomo II. Prentice Hall, 1st edition.

Clements, P. C. (2005) "Project management in a software product line organization". In: Software, IEEE. volume 22, issue 5, pages 54 – 62.

CMU/SEI. (2006) "CMMI® for Development, Version 1.2". Carnegie Mellon University, Software Engineering Institute, Pittsburgh. Disponível em: http://www.sei.cmu.edu/publications/documents/06.reports/06tr008.html .

Dantas, A. R.; Barros, M. O.; Werner, C. M. L. (2004) A Simulation-Based Game for Project Management Experiential Learning. In *International Conference On Software Engineering and Knowledge Engineering*, Banff, Canada.

Delany, S. J.; Cunningham, P.; e Wilke, W. (1998) The Limits of CBR in Software Project Estimation. In *6th German Workshop on Case-Based Reasoning*, Berlin, Germany.

Delany, S. J.; Cunningham, P. (2000) "The Application of Case-Based Reasoning to Early Software Project Cost Estimation and Risk Assessment". University of Dublin, Trinity College, Department of Computer Science. Technical Report. 2000. Disponível em: http://www.cs.tcd.ie/publications/tech-reports/tr-index.00.php . Acesso em: jan. 2008.

Derniame, J. C. (1992) Software Process Technology. In *Second European Workshop*, EWSPT '92, Trondheim, Norway. Springer-Verlag, September.

Drappa, A.; Ludewig, J. (2000) Simulation in Software Engineering Training. In *22nd International Conference On Software Engineering*, ICSE '00. Limerick, Ireland: IEEE Computer Society. p. 199.

França, B. B. N. de. (2007) Proposta de um Modelo de Simulação de Processo de Software para o Ambiente WebAPSEE. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Faculdade de Computação, UFPA, Belém. Disponível em: http://labes.ufpa.br/breno/files/simulation/TCCBrenoFinal.pdf .

Fuggetta, A. (2000) Software Process: A Roadmap. In *22th International Conference On Sosftware Engineering*, Ireland. Ireland, ACM Press.

Goldsman, D. (2007) Introduction to Simulation. In *Winter Simulation Conference* (WSC'07). Washington DC, USA.

ISO/IEC. (2004) ISO/IEC 15504-3: Information Technology – Process Assessment, - Part 3: Guidance on performing an assessment**.** Geneve.

Jacobson, I.; Booch, G.; Rumbaugh, J. (1999) The Unified Software Development Process. Massachusetts: Addison Wesley Longman, Inc.

Kellner, M. I.; Madachy, R. J; Raffo, D. M. (1999) "Software Process Simulation Modeling: Why? What? How?" In The Journal of Systems and Software, number 46, pages 91-105.

Lakey, P. B. (2003) A Hybrid Software Process Simulation Model for Project Management. In *Process Simulation Modeling Workshop* (ProSim'03). Portland, Oregon, USA.

Lima, A. M.; França, B. B. N. de; Costa, A.; Sales, E. O.; Lima Reis, C. A.; Reis, R. Q. (2006) Gerência Flexível de Processos de Software com o Ambiente WebAPSEE. In *19º Simpósio Brasileiro de Engenharia de Software*, Florianópolis. Sessão de Ferramentas.

Lima Reis, C. A. (2003) Uma Abordagem Flexível para Execução de Processos de Software Evolutivos. Tese de Doutorado – PPGC, Universidade Federal do Rio Grande do Sul, Porto Alegre.

Little, T. (2006) "Schedule Estimation and Uncertainty Surrounding the Cone of Uncertainty". In IEEE Software. May-June. Volume 23, Issue 3. pages 48- 54**.**

Martin, R. H. and Raffo, D. (2000) "A Model of the Software Development Process Using Both Continuous and Discrete Models". In Software Process: Improvement and Practice, volume 5, number 2-3, pages 147-157. John Wiley & Sons, Ltd.

Martin, R. H. and Raffo, D. (2001) "Application of a hybrid process simulation model to a software development project". In The Journal of Systems and Software, volume 59, pages 237-246.

McConnel, S. (2006) Software Estimation: Demystifying the Black Art (Best Practices). Microsoft Press.

Mi, P.; Scacchi, W. (1990) “A Knowledge-Based Environment for Modeling and Simulating Software Engineering Process”. In Knowledge and Data Engineering, IEEE Transactions, volume 2, pages 283-289.

Raffo, D.; Nayak, U.; Wakeland, W. (2005) Implementing Generalized Process Simulation Models. In *6th Process Simulation Modeling Workshop* (ProSim’05). St. Louis, Missouri, USA.

Raffo, D. (1999) Getting the Benefits from Software Process Simulation. In *International Conference on Software Engineering and Knowledge Engineering* (SEKE’99), Kaiserslautern, Germany, June.

Raffo, D. M.; Vandeville, J. V.; Martin, R. H. (1999) “Software Process Simulation to achieve higher CMM levels”. In The Journal of Systems and Software, number 46, pages 163-172.

Scacchi, W. (1999) “Experience with software process simulation and modeling”. In Journal of Systems and Software, number 46, pages 183-192.

Shepperd, M.; Schofield, C. e Kitchenham, B. (1996) Effort Estimation Using Analogy. In *18th International Conference on Software Engineering* (ICSE '96), Berlin, Germany. pages 170-178.

Shepperd, M.; Schofield, C. (1997) “Estimating Software Project Effort Using Analogies”. In IEEE Transactions on Software Engineering, volume 23, number 12, pages 736-743.

Softex. (2006) MPS.BR – Melhoria de Processo do Software Brasileiro, Guia Implementação, v.1.0, Softex, Brasil.

Softex. (2008) MPS.BR – Melhoria de Processo do Software Brasileiro. Disponível em: http://www.softex.org.br/mpsbr .

TI Inside. (2007) “TCU recomenda exigência de certificado nacional de software em licitações”. Secretaria de Comunicação Social - Presidência da República, Outubro. Disponível em: http://www.presidencia.gov.br/estrutura_presidencia/Subsecretaria/noticias/clipping/noticias/assunto10/nt14ago2c/ . Acesso em: dez. 2007.

Von Wangenheim, C. G.; Von Wangenheim, A. (2003) Raciocínio Baseado em Casos. Editora Manole Ltda.