Proposta de um Ambiente de Apoio a Estimativas em Projetos de *Software* Através Simulação

Dawilmar Guimarães Araújo[1,2]
e-mail: dawilmar@lac.inpe.br

Nilson Sant´Anna[2]
e-mail: nilson@lac.inpe.br

Germano Souza Kienbaum[2]
e-mail: germano@lac.inpe.br

Universidade de Taubaté-UNITAU[1]
R Dr Emilio Winther, 1504,31
12030-001, Centro, Taubaté, SP, Brasil.

Lab Computação e Matemática Aplicada (LAC) [2]
Instituto Nacional de Pesquisas Espaciais (INPE)
Av. dos Astronautas, 1758 CP 515
12245-970 São José dos Campos, SP, Brasil

Resumo

A atividade de estimar é considerada uma importante e difícil decisão de cunho gerencial dos ambientes de desenvolvimento de software e é referenciada como prática-chave para a maturidade e qualidade dos processos das organizações desenvolvedores de software. Diante desta posição, este trabalho vem discutir em linhas gerais uma proposta de um ambiente (arquitetura, técnica e ferramentas) que apóie as atividades gerenciais relacionadas a estimativas e aferição de estimativas para aspectos diversos como custo, esforço, prazo, etc de projetos de software. O ambiente proposto é caracterizado por meio da modelagem de processos de desenvolvimento de software para os ambientes de engenharia de software com o uso de simulação e esta última é realizada de forma combinada com um mecanismo que auxilie na definição e quantificação de elementos de tarefas. Uma implementação da proposta visa apoiar atividades gerenciais de definição de equipes de desenvolvimento.

Palavras-chave: engenharia de software, modelagem e simulação de processos de software, decomposição de elementos de tarefas, estimativas e aferição de parâmetros.

1. Introdução

Organizações desenvolvedoras de ***software*** interessam-se por melhoria de seus processos. Os esforços encontrados na literatura podem ser vistos principalmente pelo discernimento de modelos de maturidade e melhorias de processos (PBQPS, 2005)[1]. Como exemplo de modelos podem ser citados CMMI (***Capability Maturity Model Integration***) da SEI (***Software Engineer Institute***, 2005)[2], SPICE (***Software Process Improvement and Capability dEtermination***) ou ISO 15504 (***International standard Organization***).

Um ADS (Ambiente de Desenvolvimento de S***oftware***) adequado deve oferecer condições para definir, gerenciar, medir e controlar os processos de ***software***[11], e como exemplo de um modelo de ambiente que propicia estas facilidades pode se citar o Ambiente de Desenvolvimento de Engenharia de ***Software*** Centrado no Processo, do original PSEE (***Process-Centered Software Engineering Environment***).

Assim, presente nos modelos de maturidade e inseridos nestes ambientes ADS, uma das práticas chave de sucesso das gerências da Engenharia de ***Software*** muito evidenciada é a estimativa de parâmetros como custo, prazo, esforço etc. Gerentes devem cuidadosamente tratar desta decisão. Isto influencia diretamente, por exemplo na definição do processo de desenvolvimento adotado e na qualidade do produto de ***software*** a ser desenvolvido.

Por outro lado, um dos pontos chave de sucesso das gerências da Engenharia de Software é definição da equipe. Gerentes devem também tratar da decisão da escolha da equipe. Isto também influencia na qualidade do produto de software a ser desenvolvido, no processo de desenvolvimento adotado, nos resultados operacionais rotineiros da gerência, e nas dimensões dos parâmetros como custos, prazo, esforço, produtividade e qualidade**.**

Definir uma equipe é uma das etapas do gerenciamento de projetos de software que deve ser realizada ***a priori***, e é uma tarefa bastante crítica. A boa definição inclui a escolha das pessoas certas pela quantidade e qualidade (conhecimento, experiência, habilidade etc), para as tarefas certas, nos momentos certos (definidos num bom modelo de processo de desenvolvimento).

[1] http://www.pbqps.mct.gov.br

[2] http://www.sei.cmu.edu

Para esta atividade gerencial, o conhecimento que o gerente deve ter dos elementos participantes e das atividades (ou conjunto de tarefas) deve ser suficientemente adequado e bastante criterioso para o sucesso de seu trabalho.

Critérios como tempo e habilidade em projetos, experiências, indicadores de respostas à qualidade e produtividade etc fazem parte do acervo de itens que vão participar da tomada de decisão.

De uma maneira geral pode-se dizer que os grandes desafios da Engenharia de ***Software*** estão, entre outros, nos seguintes pontos:

a) Existe uma preocupação e interesse pela melhoria dos processos de ***software***, uma vez que estes estão relacionados com a qualidade do produto de ***software***;

b) Existe uma grande dificuldade de se estabelecer métricas e estimativas e aferi-las para aspectos diversos como custo, prazo, risco, esforço e qualidade nos ADS;

c) Os ADS são dinâmicos, o que leva os envolvidos a estabelecer diversos modelos de processos exigindo diferentes técnicas, ferramentas para o sucesso de sua implementação; e

d) Os ADS dependem fortemente de recursos humano. A boa escolha da equipe e sua integração ao processo antecipam riscos, incerteza e aumenta a produtividade com qualidade e sucesso das gerências.

1.1 Objetivos

A partir deste contexto, buscou-se pensar numa tecnologia que fosse capaz de apoiar as atividades relacionadas a realização de estimativas, uma contribuição para a definição e padronização de processos de ***software***, aumento da confiabilidade dos resultados das estimativas por meio de aferição de parâmetros como custo, prazo, esforço etc.

Assim, desenvolver uma metodologia de apoio à Engenharia de ***Software***, mais especificamente aos aspectos relacionados à modelagem de processos de ***software*** para que possam ser aplicadas adequadamente nas ações de elaborações de processos e de estimativas de parâmetros pertinentes aos ambientes ADS como esforço, custo, prazo, risco etc.

Obter um modelo de aferição de estimativas dos parâmetros já citados e dos aspectos relacionados às tarefas (e a sua decomposição e caracterização) por meio do uso da técnica de simulação combinada com requisitos de tarefas.

2. Processos de *software*

O processo de ***software*** é entendido como um conjunto de atividades, métodos, práticas e transformações, usadas para desenvolver e manter o ***software*** e produtos associados [4].

A capacitação em desenvolvimento de ***software*** pressupõe a existência de um processo de ***software*** definido e a aplicação de um modelo de melhoria de processos. Neste sentido, a capacitação em desenvolvimento de ***software*** reflete o grau de institucionalização de uma infra-estrutura e da cultura relacionada aos métodos, práticas e procedimentos de desenvolvimento de ***software*** de uma organização, ou seja, dos processos de ***software*** e a sua maturidade.

As práticas descritas pelos modelos de maturidade, também identificadas como ***práticas-chave,*** tanto conduzem à melhoria dos processos como servem de referência para a avaliação da maturidade dos mesmos, o que permite projetar melhoramentos.

Já por outro lado, um processo definido tem documentação que detalha o que é feito, quando, por quem, o que é utilizado e o que é produzido. A maturidade do processo indica até que ponto um processo é modelado, bem como seus demais aspectos ligados a sua gestão, métricas, controle e eficácia são realizados.

2.1 Simulação em processos de *software*

Simulação de processos de ***software*** podem ser visto de forma geral em três clássicos modelos: evento-discreto, contínuo e híbrido.

Para o primeiro, simulação evento-discreto, entende-se que os eventos de interesse acontecem pontualmente no tempo e não importa o que aconteça entre eles. Assim por exemplo, iniciar uma codificação, terminar o código, iniciar uma interpretação ou análise de requisitos e muitos outros podem ser caracterizados facilmente no tempo.

Já o segundo, da simulação contínua, busca capturar aspectos dinâmicos que o modelo anterior não o faz. As atividades dos processos de ***software*** são desenvolvidas por agentes humanos e consomem um certo tempo que por ventura ocasionam modificação em suas respostas quanto à produtividade, qualidade dos produtos que geram etc. Fatores como stress podem diminuir estas respostas, ou motivação pode aumentar. Modelos de simulação contínua visam a captura e entendimento destes fenômenos.

Os chamados modelos híbridos aparecem quando existem variáveis (contínua e discreta) dos dois tipos de modelos envolvidos no estudo e que são considerados conjuntamente. Também quando a partir de qualquer modificação que se faz em um ou outro modelo e que alterem a estrutura conceitual do modelo original.

Independentemente da escolha do modelo de simulação, deve os desenvolvedores possuir as duas principais habilidades:

i) de escrever modelos e re-escrever em ambientes de simulação (conhecer o processo de simulação, a ferramenta utilizada e tudo que os cerca); e

ii) de conhecer processos de desenvolvimento de ***software*** (objeto de investigação).

3. Estimativas e aferição das estimativas em processo de *software*

Segundo a SEI, o trabalho de estimativas poderá ser realizado considerando por exemplo, por dados históricos, através de análise estatística, por estimativas de especialistas, por métodos e técnicas orientadas por parâmetros dos projetos através da aceitação de projetos similares etc.

Estimativas podem ser vistas em macro e micro estimativas. Cabendo a primeira uma previsão do todo. Para esta estimativa conhece-se avaliação de especialistas (subjetivas, sujeitas a modificações), matemáticas (usando dados históricos), comparação e analogia (projetos com atributos simulares e os principais atributos análogos). Já na visão micro da estimativa acrescenta-se um esforço associado com cada componente ou atividade do processo separadamente. Neste caso, faz-se uma decomposição em menores partes do projeto.

Para melhor compreender estas relações entre alguns modelos de estimativas existentes, uma classificação é apresentada de forma resumida na **Tabela 1**. Nesta classificação observa-se que estes modelos podem ser segmentados e baseados em níveis de abstração, de acordo com o grau de validação do modelo. Seus agrupamentos estão apresentados numa versão preliminar, em um outro trabalho mais completo um estudo mais sistemático é realizado.

Tabela 1. Tipos de modelos de estimação [04]

Níveis de abstração	
alto	Equações lineares (simples) Modelos determinísticos Ex. Defeitos = αKSLOC +βEffort
↓	Relacionamento Multi-variável (complexo) Ex. Modelos de regressão e estatísticas Ex. Monte Carlo.
↓	Modelos de processos (estático) Ex. Dados históricos das atividades de cada ciclo de vida. Ex. Comparação entre os processos e variáveis.
↓	Modelos de processos (dinâmicos) Modelagem dinâmica de sistemas. Ex. Representação continua de interação de variáveis.
baixo	Modelos de processos com simulação Ex. Evento-Discreto, contínua e híbrida. Ex. Evento-discreto e combinado. Ex. Comparação produto x processo capturado.

Entendendo a **Tabela 1**, tem-se que os altos níveis de abstração (alto) produzem mais semelhança entre os modelos. Existem equações simples que relatam parâmetros de saídas para algumas variáveis independentes (estimadas). Esta abordagem busca ver processo de ***software*** como uma caixa preta.

Em um nível mais detalhado (baixo) observa-se a utilização de modelos de processos baseados em simulação, onde se localiza esta proposta, a de usar a simulação para estimativas [12].

O propósito deste trabalho é apresentar uma proposta de pesquisa em andamento que visa combinar aspectos de prazo, custo, esforço etc com simulação de processos.

3.1. Obtenção e analise de variáveis candidatas

Com base em dados e informações confiáveis podem ser estabelecidas as estimativas. Obter dados é caro, gasta tempo e exige cuidado.

Um planejamento relativo à coleta, análise e tudo que envolve o uso de dados devem ser feitos para garantir a validade dos dados.

Para se obter um dado eficientemente, alguns princípios devem ser seguidos (não necessariamente nesta ordem de importância): i) ter objetivos e planejamento antes da tomada de dados; ii) escolher os dados a serem obtidos baseado no modelo do processo que está sendo analisado; iii) dados do próprio processo devem ser cuidadosamente definidos e gerenciados, e iv) ter um plano de coleta de dados que deve subsidiar a gerência.

O objetivo da obtenção de dados e análise é usar tão bem quanto possível, os dados de forma objetiva, absoluta, e explícita. Para ser mais significativa uma medida deve ser robusta, compreensível, explicitar propriedades do processo, propor melhorias estratégicas, ser natural quanto ao resultado do processo, ser simples, previsível e tratável.

A análise de dados apóia o desenvolvimento a manutenção de ***software***. Por exemplo: a análise de dados de erros pode ajudar a identificar áreas onde crescem a probabilidade de ter erros. Análise de dados de testes podem ser usados para encontrar relações entre erros detectados durante o processo de desenvolvimento e erros identificados na entrega de produtos, a efetividade de cada tipo de teste para busca de erros, custo de busca de defeitos etc.

Os parâmetros de custo, prazo, esforço são talvez os mais trabalhados na literatura, isto evidencia o quanto se qualifica o processo de ***software*** a partir deles. Verifica-se também que são estreitamente dependentes do recurso humano dos ambientes ADS.

Para estes parâmetros, uma vez que estes são dependentes, cabem estudos comparativos entre os resultados do ADS ao correspondente recurso humano utilizado.

Por exemplo, estimativa de custo de ***software*** é feito medindo o tempo de esforço requerido para completar um produto de ***software***. Esforço é usualmente representado pelo número de homens-mês (HM).

Estimativa de custo é importante porque ele oferece um parâmetro fundamental (essencial) para o planejamento e

gerenciamento de projetos de ***software***. Sem uma estimativa acurada deste parâmetro, gerentes de projetos não teriam base para determinar quanto tempo e esforço cada fase do ***software*** e atividades seriam gastos, e eles não teriam efetivo jeito de monitorar, controlar os projetos. Estariam sem parâmetros.

Existem alguns métodos de estimativas, como por exemplo modelos algoritmos COCOMO (***Constructive Cost Model***), opiniões de especialistas (usando técnicas de Inteligência Artificial), analogias ***top-down***, ***bottton-up*** etc.

Alguns outros métodos híbridos são propostos para estimar outros diferentes aspectos. Como exemplo o aspecto da interpretabilidade dos Diagramas de Estados da UML (***Unified Modeling Language***) realizada por Piattini[04]. Neste caso é apresentado uma metodologia específica que combina o uso Lógica F***uzzy*** com modelo matemático de predição que busca o tempo médio gasto por uma pessoa para interpretar um Diagrama de Estados da UML.

Como acontece com outros métodos, cada um tem suas limitações, vantagens e desvantagens, e nenhuma das alternativas citadas é melhor do que outra considerando todos os aspectos globalmente e nem os considerando isoladamente.

3.2 Simulação para aferição das estimativas

Diante do exposto da seção anterior, o uso combinado de técnicas visa contribui para diminuir ou eliminar deficiência (limitações) das técnicas que busca realizar estimativas mais acuradas, sendo passivo e aceitável dentre diversas outras combinações com a mesma intenção, que é a de demonstrar as vantagens de usar modelos combinados ou híbridos para se realizar estimativas, verificou-se mais uma vez esta oportunidade, a de usar a simulação para estimativas e aferição destas.

Existem vários enfoques de modelos de simulação aplicáveis ao estudo dos diferentes aspectos do processo de ***software***. Entre eles, apresentado por Kellner [03], onde classifica estas áreas como planejamento, gerenciamento estratégico, treinamento e aprendizado, adoção de tecnologia e melhoria de processo. Desta ultima classificação Raffo [05] mostra uma aplicação para conquistar altos níveis de CMM.

Assim a pesquisa realizada pelos autores deste texto compreenderá como usar simulação e conduzir experimentos, a partir de um modelo de processo simulado, adotando passos sólidos para as etapas de obtenção e manutenção de curvas de distribuição de alguns parâmetros como custo, prazos etc, tudo isto para apoiar a estimativa e aferição das estimativas dos ambientes de ***software***.

Na **Figura 1** é apresentado o esquema inicial de como estará relacionado o simulador no contexto. Os elementos que estão nas fronteiras do simulador fornecerá o modelo proveniente do ***workflow*** do processo, das tarefas (requisitos da tarefa), e dados de entrada do processo (provenientes dos geradores de entidades e tempos de atividades iniciais). Como saída a partir de um conjunto de replicações controladas (interativas) os gerentes poderão aferir seus resultados de estimativas.

4. O ambiente interativo de estimação

Algumas características já identificadas para este ambiente conduzem a decisão da escolha e da definição da estrutura do ambiente.e da ferramenta mais adequada, bem como alguns elementos necessários (e caracterizados) que complementará este ambiente, com por exemplo a

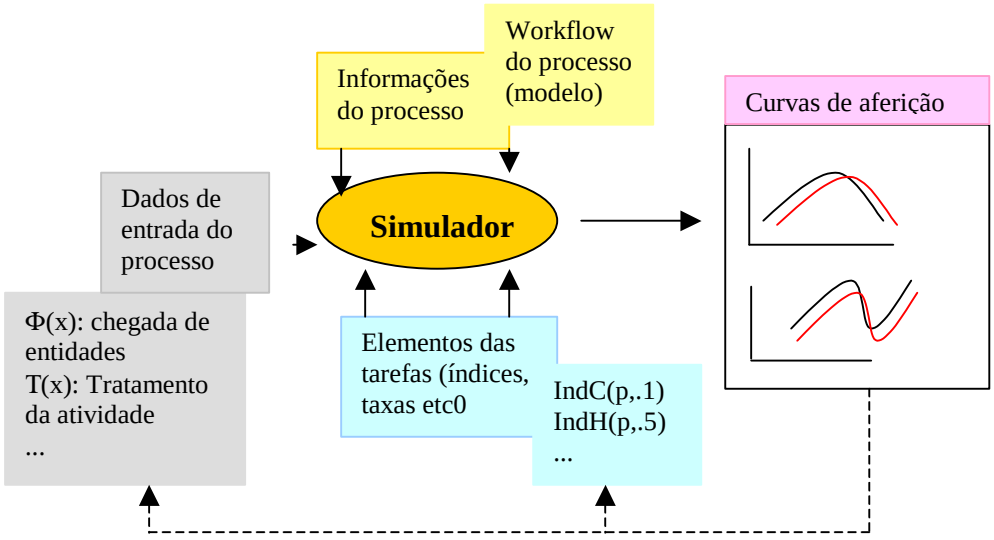

Figura 1. Componentes que participam da simulação combinada para obtenção de curvas de estimativas e aferição

linguagem de interface, base de dados, definição da ferramenta de simulação etc.

A definição mais eficiente das atividades conforme requisitos das tarefas identificadas no processo de software a ser definido, e a alocação de executores que para Engenharia de ***Software*** são atores das atividade para as tarefas de acordo com perfis e atributos, deverão ser evidenciados e nortearão a construção do modelo do processo.

Aspectos relacionados às tarefas fazem parte do modelo de processo, e introduzirão modificações nos elementos do modelo do processo. A partir da simulação resultados da introdução realizada pode ser conhecido por meio das curvas de estimativas e aferição.

Obviamente este processo é dinâmico, sendo assim melhorias contínuas são identificadas e esperadas. Usa-se da própria simulação para escolha entre alternativas que levará em diante a definição do processo aceito naquele instante. Diante disto os resultados das respostas de simulação passaram a ser as novas entradas em uma nova rodada de simulação.

Em uma outra perspectiva, a abordagem estará vinculada ao ambiente clássico proposto em três camadas sendo: camada do usuário, camada núcleo e camada de dados conforme **Figura 3**.

Camada do usuário: um editor gráfico de modelo, onde de preferência sejam estereotipados conforme notação de uma linguagem de modelagem de processos, por exemplo SPEM (***Software Process Engineer MetaModel***), com sub-modelos estendidos e o da linguagem do ambiente do simulador em uma única interface.

A interface gráfica de comunicação com usuário terá seu desenvolvimento focado na relação das características de notação, conforme requisitos do processo de ***software***.

Uma linguagem de modelagem a ser utilizada para a modelagem do processo de ***software*** deve representar os elementos mais genéricos em um nível de abstração tal que permita o gerente do processo uma visão global e sustentável aos interesses da modelagem. Está em estudos a definição desta linguagem de modelagem de processos, a princípio acreditando que adotar uma linguagem já conhecida, sólida na comunidade como a SPEM parece ser bastante producente.

Camada núcleo: constituído a partir de um modelo de ***workflow*** de processo, ***linkado*** a um núcleo de simulação combinado com a capacidade de lidar com as questões chave da decomposição e caracterização das tarefas de forma interativa; e

Camada de dados: serão armazenados os elementos do modelo para manipulação do programa da camada de núcleo que permitirá criação, manutenção, reuso de modelos e partes destes.

Um esquema de classes (exemplo) destes elementos que serão trabalhados e persistidos na base de dados é simplificado na **Figura 2.**

Parte desta proposta está sendo desenvolvida em um outro trabalho e estará vinculada às pesquisas do grupo de Simulação e Gestão do LAC[3].

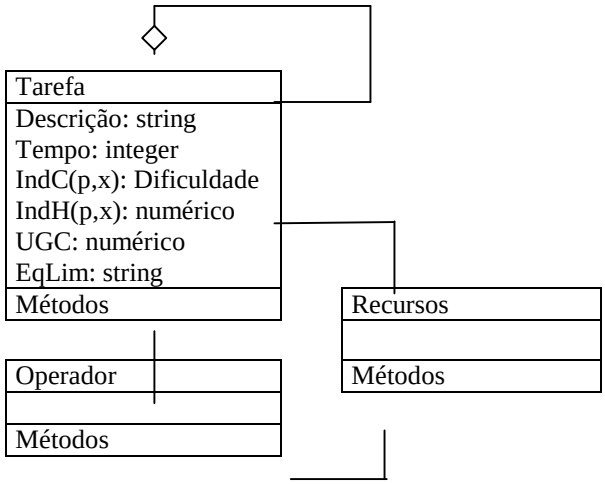

Figura 2. Diagrama de classes persistentes para modelagem das tarefas

5. Considerações finais

Processos de desenvolvimento de ***software*** são muito dinâmicos, e influenciado pela pressão da competitividade para entrega produtos de ***software*** com qualidade em tempo e custo razoável. Implementar mudanças nos processos acabam sendo difíceis, caras e podem introduzir erros.

Neste sentido a Simulação vem prover de forma viável e barata aplicação para avaliar e reduzir este risco através de analises quantitativa de modificação de processos propostos em termos de performance do processo em diversos cenários experimentais.

Um bom trabalho de modelagem e simulação também dá suporte aos processos de tomadas de decisões nas organizações. Esta estrutura mais fortemente centrada a processos do que produto. Casos de negócios podem ser desenvolvidos para prover melhoramento dos processos.

A definição de curva de distribuição de tarefas foi visto como uma grande oportunidade deste trabalho, e com o uso de simulação para apoiar esta intenção para a visão micro-estimativa de processo espera-se obter uma maior e mais segura forma de lidar com os aspectos relacionados com o processo de ***software*** e os aspectos dos ambientes de software como prazo, custo, esforço etc e a definição de equipes.

Outros benefícios desta proposta só serão conseguidos a partir da conclusão do desenvolvimento de um ambiente de simulação integrado capaz de lidar conjuntamente com os aspectos complementares citados às duas áreas de estudo.

[3] NEMESIS/LAC: Núcleo de Estudos em Modelagem e Simulação de Sistema/Laboratório de Matemática e Computação Aplicada.

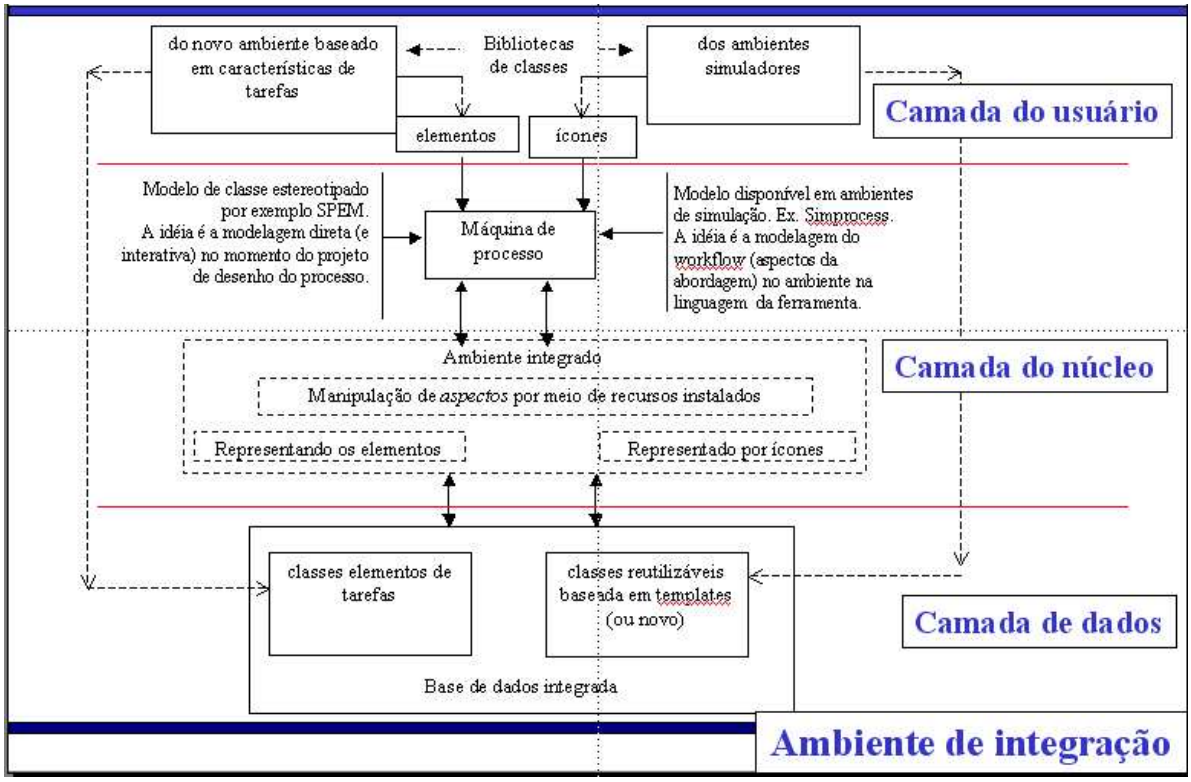

Figura 3. Arquitetura do ambiente

6. Bibliografia

[01] Araújo D G-Trzesniak P; ***Caracterização metrológica de indicadores.*** Comunicação interna, EFEI, 1999.

[02] Freitas Filho, P J; ***Introdução a modelagem e simulação de sistemas.*** Visual Books, São Paulo, 1999.

[03] Hauston, D X; Ferreira, S; Collofello, J S; Montgomery, D C; Mackulak, G T; Shunk, D L; ***Behavioral characterization: finding and using the influential factors in software process simulation models.*** The Journal of Systems and ***Software***, **59**, 259-270, Mar, 2001.

[03] Ioana, R A; Collofello, J A; Lakey, P B; ***Software process simulation for reliability management.*** The Journal of Systems and ***Software***, **46**, 173-182, Nov, 1999.

[04] Kellner, M I; Raffo, D M and Madachy, R J; ***Software process simulation modeling: Why? What? How?*** The Journal of systems and ***software***, **46**, (2-3), 91-105, apr, 15, 1999.

[05] Piattini, M at al; ***Construcción de un modelo de predicción para el entendimiento de los diagramas de estados en UML***. Grupo ALARCOS. D Departamento de

Informática, Universidad de Castilla-La Mancha. Ciudad Real (España), 2002.

[06] Raffo, D M and Harrison, W; ***Combining Process Feedback with Discrete Event Simulation Models to Support Software Project Management.*** International ***Software*** Process Simulation Modeling Workshop (ProSim 2004), Edinburgh, Scotland, 24-25 May, 2004.

[07] Raffo, D M; Harrison, W and Vandeville, J; ***Software Process decision support: making process tradeoffs using a hybrid metrics, modeling and utility framework.*** Proceedings of the 14th international conference on ***Software*** engineering and knowledge engineering. ACM International Conference Proceeding Series. pgs: 803 – 809. July, 2002.

[08] Raffo, D M; Vandeville, J V and Martin, R H; ***Software Process Simulation to Achieve Higher CMM Levels***. The Journal of systems and ***software***, **46**, (2-3), 91-105, apr, 15, 1999.

[09] Ruiz, M; ***Modelado y Simulação para la Mejora de los Procesos Software.*** Tesis Doctoral. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla, 2003.

[10] Salviano, C F; ***Tutorial: Introdução aos modelos CMM, ISO/IEC 15504 (SPICE) e CMMI***. V Simpósio Internacional de Melhoria de Processo de ***Software***, Recife-PE, Nov-2003.

[11] SANT´ANNA, N: ***Um ambiente integrado para o apoio ao desenvolvimento e gestão de projetos de software para sistemas de controle de satélites***. Tese de Doutorado. (INPE-TDI: 23567812-3), São José dos Campos-SP. 2000.

[12] Silva, F A D; ***Um modelo de simulação de processo de software baseado em conhecimentos para o ambiente PROFSOFT,*** Dissertação de Mestrado, UFRGS, Porto Alegre-RS. 2001.