

Definição e Construção de Ambientes de Desenvolvimento de Software Orientados a Organização

Karina Villela, Guilherme Horta Travassos, Ana Regina Cavalcanti da Rocha

Universidade Federal do Rio de Janeiro
COPPE - Programa de Engenharia de Sistemas e Computação
Caixa Postal 68511 CEP 21945-970 Rio de Janeiro, RJ – Brasil
E-mail: {kvillela,ght,darocha}@cos.ufrj.br

Resumo

Este artigo define Ambientes de Desenvolvimento de Software Orientados a Organização (ADSOrg) como sendo ambientes de desenvolvimento de software que apoiam a gerência do conhecimento relevante para os processos de desenvolvimento e manutenção de software da organização. Além disso, propõe um modelo que especifica os componentes necessários para que tais ambientes contemplem os seus requisitos. Uma estratégia para a construção de ADSOrg a partir de um meta-ambiente também foi definida, implementada e utilizada. Este artigo ainda apresenta um resumo dos resultados de uma pesquisa de campo realizada com a finalidade de orientar os futuros esforços de incorporação de conteúdo e desenvolvimento de ferramentas para ADSOrg.

Palavras-chave: Gerência de Conhecimento, Ambiente de Desenvolvimento de Software, Ontologia, Linguagem para Modelagem de Processo.

Abstract

This paper defines Enterprise Oriented Software Development Environments (EOSDE) as software development environments that support the management of relevant knowledge for organizational software development and maintenance processes. Furthermore, a model is presented that specifies the necessary components to satisfy the requirements of such environments. A strategy for the construction of EOSDE from a meta-environment was also defined, implemented and used. Finally, this paper presents the results of a survey carried out in an attempt to guide future work on incorporating content and developing tools for EOSDE.

Key-words: Knowledge Management, Software Development Environment, Ontology, Process Modeling Language.

1. Introdução

Diferentes abordagens têm sido propostas para melhorar a qualidade de processos e produtos. No entanto, pouca atenção tem sido dada à produção, armazenamento, compartilhamento e uso de conhecimentos relevantes para o desenvolvimento e a manutenção de *software*. Sempre que o conhecimento requerido não está disponível, o desenvolvedor de *software* tem que partir do zero para encontrar a solução do problema, o que tem sérias implicações em termos da qualidade da solução. Verifica-se, desta forma, a importância da administração sistemática dos conhecimentos relevantes neste contexto. Mas como integrar a administração sistemática do conhecimento com as atividades de Engenharia de Software? As áreas de pesquisa de Gerência do Conhecimento e Ambientes de Desenvolvimento de Software são fundamentais para elaborar uma resposta a esta questão.

Gerência do Conhecimento implica na administração, de forma sistemática e ativa, do capital intelectual de uma organização, utilizando tecnologia apropriada e visando fornecer benefícios estratégicos à organização. Recentemente, chegou-se ao consenso de que a Gerência do Conhecimento deve ser freqüentemente embutida em processos e que estes devem ser re-projetados para acomodar a Gerência do Conhecimento [1].

Ambientes de Desenvolvimento de Software (ADS) são sistemas computacionais que apóiam o desenvolvimento, reparo e melhoria de produtos de software bem como o controle e gerenciamento destas atividades [2]. ADS têm evoluído ao longo do tempo para fornecer apoio mais amplo e efetivo aos desenvolvedores de software, tendo surgido os Ambientes de Desenvolvimento de Software Centrados em Processo [3] e os Ambientes de Desenvolvimento de Software Orientados a Domínio (ADSOD) [4,5].

2. Objetivo

O objetivo deste trabalho de pesquisa foi propor uma solução para introduzir Gerência do Conhecimento em Ambientes de Desenvolvimento de Software, de forma que o conhecimento adquirido ao longo de vários projetos possa estar disponível para o desenvolvedor de *software* dentro do seu contexto de trabalho e que este possa contribuir diretamente para a geração de novo conhecimento a ser utilizado em outros projetos.

3. Definição da Solução

O trabalho de pesquisa iniciou-se com a definição de Ambientes de Desenvolvimento de *Software* Orientados a Organização (ADSOrg) como sendo ADS centrados em processo que apóiam a Gerência do Conhecimento ao longo dos processos de desenvolvimento e manutenção de software. O objetivo de tais ambientes é: (i) apoiar os desenvolvedores de software na execução de suas atividades, fornecendo todo o conhecimento que tenha sido capturado e acumulado pela organização por sua importância para o desenvolvimento e a manutenção de software, e (ii) apoiar o aprendizado organizacional a partir do aprendizado dos desenvolvedores nos projetos de software. O conceito de ADSOrg estende o conceito de ADSOD [4,5], uma vez que conhecimento sobre o domínio de aplicação é apenas um dos tipos de conhecimento requeridos no desenvolvimento e na manutenção de software.

ADSOrg pretendem apoiar o desenvolvimento e a manutenção de *software* tanto em organizações nas quais estas atividades são atividades de negócio (*Organizações de Software*), quanto em organizações nas quais são atividades de apoio ao negócio (*Organizações com outro Tipo de Negócio*).

Uma vez definido o conceito de ADSOrg, foram definidos os seus requisitos e um modelo com o conjunto de componentes necessários para atender a esses requisitos.

3.1. Requisitos de ADSOrg

A identificação de dois tipos de organizações que desenvolvem e mantêm software resultou na definição de dois tipos de ADSOrg, um para cada tipo de organização. Nesta subseção, primeiro são apresentados os requisitos comuns a ambos os tipos de ADSOrg e, então, são discutidos os requisitos específicos de cada um dos tipos.

3.1.1. Requisitos Comuns

Independente do tipo específico, ADSOrg devem satisfazer os seguintes requisitos [6,7]: (i) armazenar e fornecer para as equipes de projeto um modelo da organização, contemplando estrutura, processos e capital intelectual, (ii) armazenar conhecimento especializado sobre o desenvolvimento e a manutenção de software na organização e fornecer este conhecimento para as equipes de projeto quando necessário, além de (iii) apoiar a contínua evolução do conhecimento armazenado no ambiente.

A disponibilidade de um modelo da organização (requisito (i)) agiliza a localização de especialistas que podem trazer contribuições para o projeto e serve de base para levantamentos de requisitos relacionados aos processos organizacionais. A localização de

especialistas é muito importante quando o conhecimento necessário não está disponível no ADSOrg ou quando, apesar de disponível, a sua aplicação requer entendimento mais profundo.

Para atender ao requisito (ii), um ADSOrg precisa conter o conhecimento sobre o processo de software padrão da organização [8] e sobre a experiência de desenvolvimento e manutenção de software adquirida pela organização, o que pode envolver normas e diretrizes, melhores práticas com exemplos de projetos que as utilizaram, produtos intermediários, roteiros de documentos, relatos de lições aprendidas em projetos anteriores, entre outros. Sempre que pertinente, cada item de conhecimento armazenado no ambiente deve ser acompanhado da identificação de especialistas internos e materiais de referência. Conhecimento adquirido pela organização sobre desenvolvimento e manutenção de software é fundamental para as atividades de elaboração da Proposta de Fornecimento e do Plano do Projeto, pois transforma estas atividades, que são centradas no conhecimento e na experiência do gerente do projeto, em atividades centradas no conhecimento e na experiência da organização. Atividades de análise, projeto, implementação, teste e implantação podem ser apoiadas por notícias relativas às tecnologias utilizadas, melhores práticas em relação aos métodos selecionados, reutilização de produtos intermediários, entre outros.

Uma questão importante é o fato do conhecimento de uma organização estar em constante evolução. Experiências relevantes para o desenvolvimento ou manutenção de software podem ser adquiridas a cada projeto. Além disso, o modelo da organização também sofre alterações ao longo do tempo. Neste contexto, é fundamental que os ADSOrg apoiem a atualização do conhecimento organizacional fornecido pelos mesmos (requisito (iii)).

3.1.2. Requisitos de ADSOrg para Organizações de Software

Em *Organizações de Software*, a disponibilidade de conhecimento (mesmo que somente parcial) sobre os clientes e sobre os seus respectivos domínios representa uma vantagem estratégica na competição por novos projetos. Assim, os requisitos dos ADSOrg para *Organizações de Software*, além dos requisitos comuns já discutidos, são [6,7]: (i) armazenar e fornecer modelos das organizações clientes para as equipes de projeto, contemplando estrutura, processos e capital intelectual, além de (ii) armazenar conhecimento sobre os domínios de negócio dos clientes e fornecer este conhecimento ao longo dos projetos de software relacionados a cada cliente.

Conhecimento sobre a organização cliente e sobre o seu domínio de negócio é fundamental nas atividades iniciais de um projeto de software, pois estas atividades também exigem uma visão global da organização cliente e experiência de desenvolvimento de software no seu domínio de negócio. No decorrer do projeto, conhecimento sobre o domínio é muito útil. Nas atividades de análise e de projeto, por exemplo, serve como ponto de partida para o levantamento de requisitos e para a modelagem dos dados do sistema [4]. Conhecimento sobre a organização cliente também pode ser útil no decorrer de um projeto, pois o conhecimento do domínio não elimina a interação com a organização cliente e o projeto pode ter como finalidade o apoio, controle ou automatização de um ou mais processos da organização cliente.

3.1.3. Requisitos de ADSOrg para Organizações com outro Tipo de Negócio

Organizações com outro Tipo de Negócio desenvolvem e mantêm sistemas de software para uso próprio. Sendo assim, os requisitos de um ADSOrg neste contexto são os requisitos comuns já discutidos, acrescidos apenas dos requisitos: armazenar conhecimento sobre os domínios de negócio da organização e fornecer este conhecimento ao longo dos projetos de software da organização. Estes requisitos são similares ao que caracterizam um ADSOD [4].

3.2. Modelo de Componentes para ADSOrg

A figura 1 fornece uma visão macroscópica dos componentes de um ADSOrg. O *Repositório da Organização* e os *Serviços/Ferramentas de Gerência do Conhecimento* compõem a *Infra-estrutura de Gerência do Conhecimento*. Os *Serviços/Ferramentas de Gerência do Conhecimento* têm como objetivo facilitar o armazenamento de dados, conhecimentos e experiências no *Repositório da Organização*, além de apoiar a constante disseminação e atualização dos mesmos. As figuras 2a e 2b detalham o *Repositório da Organização* para os dois tipos de ADSOrg definidos.

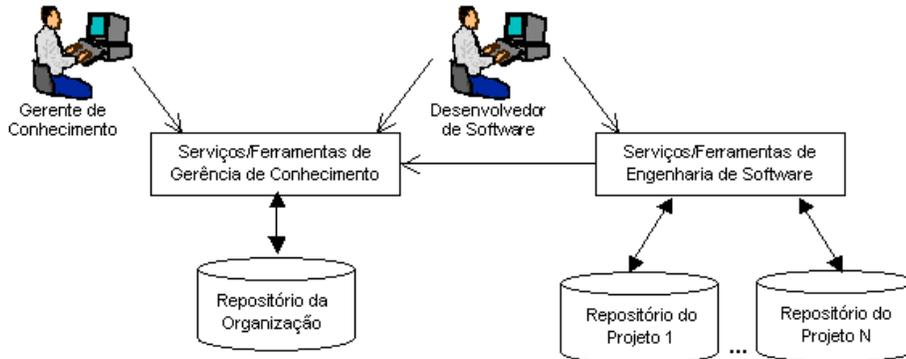


Figura 1 – Visão Macroscópica dos Componentes de um ADSOrg

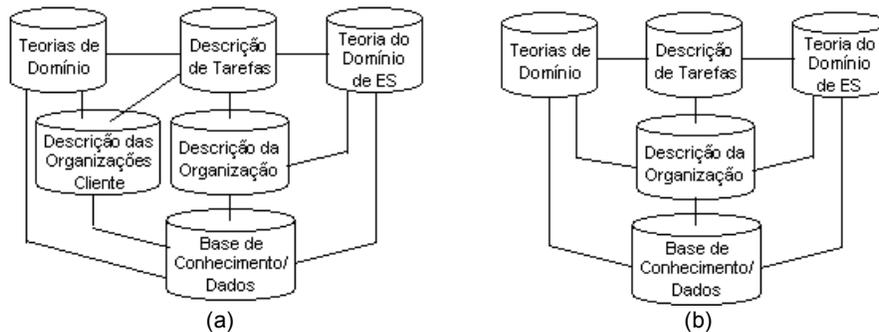


Figura 2 – Repositório da Organização: ADSOrg para Organizações de Software (a) e ADSOrg para Organizações com Outro Tipo de Negócio (b)

O componente *Descrição de Tarefas* contém a descrição de tarefas genéricas, comuns a diferentes domínios e organizações. Como definido OLIVEIRA *et al.* [5], a descrição de uma tarefa é formada por uma descrição de alto nível, uma ontologia dos conceitos associados com a tarefa, as inferências necessárias a sua solução ou a sua decomposição em sub-tarefas, e um conjunto de referências bibliográficas. O objetivo é apoiar o desenvolvedor no entendimento de um problema a partir do entendimento das tarefas que compõem o problema.

O componente *Teorias de Domínio*, inicialmente proposto para compor os ADSOD [4,5], organiza o conhecimento sobre os domínios de negócio. Uma *Teoria de Domínio* é decomposta em subteorias, que, por sua vez, podem ser decompostas em novas subteorias ou conter uma ontologia e o mapeamento com as tarefas genéricas que aplicam os conceitos da ontologia. Além de promover o entendimento do domínio, as ontologia de domínio são utilizadas para orientar o registro e a atualização do capital intelectual da organização.

O componente *Teoria do Domínio de ES* é similar ao componente *Teorias de Domínio*, sendo que o universo de discurso é a Engenharia de Software (ES).

O componente *Descrição da Organização* descreve a organização que utiliza o ADSOrg para desenvolver e manter software, estabelecendo, dentro do contexto da estrutura e dos

processos organizacionais, quais as tarefas genéricas que são executadas e o conhecimento de Engenharia de Software que é requerido. Se a organização desenvolve software para uso próprio (*Organização com outro Tipo de Negócio*), este componente também estabelece qual conhecimento sobre os domínios de negócio é requerido ao longo de sua estrutura e de seus processos organizacionais. O mapa das competências disponíveis na organização também faz parte do componente *Descrição da Organização* e especifica quais são as competências possuídas por cada funcionário e em que grau estas competências são possuídas. Os modelos dos processos organizacionais permitem especificar o contexto em que um item de conhecimento foi criado e o contexto de aplicação pretendido para o mesmo.

Descrição das Organizações Cliente é um componente específico dos ADSOrg para Organizações de Software (figura 2a), similar ao componente *Descrição da Organização*, mas que contém a descrição das organizações cliente da organização de software. Esta descrição também referencia as tarefas genéricas que são executadas e o conhecimento de domínio que é utilizado ao longo da estrutura e dos processos organizacionais.

Neste contexto, é fundamental dispor de uma *Ontologia de Organização* que defina um vocabulário comum para a descrição de qualquer organização.

O componente *Bases de Conhecimento/Dados* armazena o conhecimento e os dados relevantes para a organização que foram obtidos ao longo de seus vários projetos de software, onde os itens de conhecimento/dados, ao serem armazenados, devem ser associados a termos das ontologias existentes no ambiente, de forma que possam ser recuperados, também, a partir da exploração e/ou seleção dos termos das ontologias.

Na figura 1, tem-se, ainda, os *Serviços/Ferramentas de Engenharia de Software* e os *Repositórios de Projeto*. Os *Serviços/Ferramentas de Engenharia de Software* têm como objetivo apoiar as atividades dos processos de desenvolvimento e manutenção de software. As *Ferramentas de Engenharia de Software* devem ser capazes de fornecer conhecimento organizacional relevante para o projeto no momento em que ele se faz necessário para os desenvolvedores, utilizando, para isto, a *Infra-estrutura de Gerência do Conhecimento*. Exemplos são uma ferramenta que apóie o gerente na identificação dos riscos do projeto, fornecendo os riscos identificados em projetos similares da organização, e uma ferramenta de modelagem capaz de reutilizar a definição e a estrutura de organização dos conceitos de uma Teoria do Domínio na construção de um modelo de dados ou de classes. Um *Repositório de Projeto* armazena todas as informações relativas ao projeto de software.

A definição de *Teorias de Domínio* depende das organizações para as quais serão construídos os ADSOrg e a *Descrição de Tarefas* pode ocorrer na medida em que as tarefas são referenciadas por *Teorias de Domínio*. Já a *Teoria do Domínio de ES*, a *Base de Conhecimento/Dados* e os *Serviços/Ferramentas de Engenharia de Software* devem ser construídos de forma incremental e de acordo com as prioridades definidas pelos próprios desenvolvedores de software, sendo descrita, na seção 3.5, uma pesquisa realizada com o intuito de identificar os tipos de conhecimentos considerados mais importantes pelos desenvolvedores de software. Restam, como aspectos fundamentais a serem tratados no âmbito do modelo proposto, a definição de uma ontologia de organização e de uma linguagem para modelagem de processos de forma a viabilizar a *Descrição da Organização* e, quando pertinente, a *Descrição das Organizações Cliente*. Estes aspectos são abordados nas seções 3.3 e 3.4.

3.3. Ontologia de Organização

Para construção da ontologia de organização, foi utilizado um processo definido com base nos trabalhos de USCHOLD e GRUNINGER [9], FALBO [10] e OLIVEIRA [4], que consistiu

de identificação do propósito, especificação de requisitos, captura, conceituação formalização e avaliação.

O propósito estabelecido para a ontologia foi o de fornecer um vocabulário comum que possa ser utilizado para representar conhecimento útil para os desenvolvedores de software sobre as organizações envolvidas em um projeto de software. Cenários de motivação foram descritos para mostrar a utilidade da ontologia e questões gerais de competência foram formuladas.

A definição da ontologia foi baseada em pesquisa bibliográfica sobre ontologias relacionadas, em pesquisa bibliográfica na área de Administração de Empresas e em entrevistas com especialistas da área. As questões gerais de competência foram refinadas, dando origem às questões de competência, que representam a especificação de requisitos da ontologia.

Para iniciar a captura da ontologia, termos e frases potencialmente relevantes foram identificados e atribuídos a grupos de trabalho, que tiveram sua semântica definida e deram origem as sub-ontologias de Capital Intelectual, Estrutura, Artefatos, Comportamento e Estratégia Geral. Para cada uma das sub-ontologias, conceitos, relações e restrições foram descritos em linguagem natural e exemplificados. Na formalização, foi utilizada lógica de primeira ordem. A avaliação da ontologia ocorreu ao longo de todo o processo.

A sub-ontologia de Capital Intelectual estabelece o vocabulário necessário para descrever o capital intelectual de uma organização. **Pessoas** são fundamentais para o funcionamento de uma organização, atuando na execução das atividades necessárias ao cumprimento da missão da organização. As competências possuídas pelos profissionais de uma organização são de grande importância para os próprios profissionais, pois são utilizadas para estabelecer o seu papel e o seu valor na organização, e também para a organização, pois representam o seu capital intelectual. **Competências** tornam as pessoas capazes de executar atividades que envolvem algum grau de dificuldade e podem ser classificadas, de acordo com a sua natureza, em **conhecimento**, **habilidade** e **experiência**. Experiências geralmente representam a aplicação de conhecimentos na prática. Por fim, um **domínio de conhecimento** é um conjunto de conhecimentos reunidos de acordo com a homogeneidade de conteúdo.

A sub-ontologia de Estrutura estabelece o vocabulário necessário para descrever como a organização está estruturada. Uma **organização** pode ser definida como um grupo de pessoas trabalhando em conjunto para o cumprimento de uma missão. Existem diversas maneiras de estruturar uma organização de forma a estabelecer a divisão de trabalho, mas os componentes básicos são **cargos**, **unidades organizacionais** e **comissões**. **Posições** são o resultado não só da alocação de cargos a unidades organizacionais, mas, principalmente, da definição da estrutura necessária para que a unidade organizacional funcione de forma eficiente e econômica. Tanto cargos quanto posições especificam competências desejadas, atividades a serem executadas e responsabilidades a serem assumidas pelas pessoas que fazem parte da organização. Desta forma, foi introduzido o conceito de **Agente**, que é uma especificação de um perfil necessário para que a organização cumpra a sua missão através da execução de atividades. Um agente pode representar um cargo ou uma posição. O preenchimento das vagas em uma organização é representado pela alocação de pessoas às posições existentes e a consequente ocupação dos possíveis cargos, considerando, para isso, as competências possuídas pelas pessoas e as competências requeridas pelas posições e cargos. Pessoas também podem fazer parte de comissões. Por fim, **objetivos** são enunciados escritos sobre os resultados a serem alcançados num período de tempo determinado, podendo ser estabelecidos para a organização como um todo, para unidades organizacionais e para posições.

A sub-ontologia de Artefatos agrupa os conceitos e relações que definem os artefatos em termos de sua natureza e composição. **Artefato** é qualquer elemento produzido pelo

homem e não por causas naturais, podendo exercer diferentes papéis em uma organização, tais como o de insumo ou produto de uma atividade. Artefatos podem ser classificados, de acordo com a sua natureza, em *bens*, *documentos* e *componentes*. Bens podem ser classificados em *bens de usufruto* e *bens de produção*, sendo que bens de produção podem ainda ser classificados em *hardware*, *software* e *equipamentos*. Um componente pode ser um *componente de hardware*, um *componente de software* ou uma *peça*.

A sub-ontologia de Comportamento está baseada na ontologia definida por FALBO [10]. Uma *atividade* é uma ação de transformação que pode requerer competências para a sua execução e, ao ser executada, fazer uso de bens de produção (recursos), consumir matérias-primas e artefatos de entrada (insumos), além de produzir artefatos de saída (produtos). Uma *matéria-prima* é uma matéria bruta ou pouco elaborada utilizada como insumo em uma atividade, no sentido de ser um objeto de transformação desta atividade. Uma atividade pode ser classificada, quanto ao papel que desempenha em relação à missão da organização, em *atividade de negócio* e *atividade de apoio ao negócio* e, quanto a sua natureza, em *atividade operacional*, *atividade de gerência* e *atividade de controle da qualidade*. Um *processo* é um conjunto de atividades estruturadas e destinadas a resultar em um artefato ou serviço de valor para a organização ou para um determinado cliente ou mercado. Uma *combinação de atividade* é um agrupamento de atividades que permite representar a decomposição de uma atividade em um conjunto de atividades. *Procedimentos* são condutas bem estabelecidas e ordenadas para a execução de atividades [10], sendo classificados em *métodos*, *técnicas* e *diretrizes*. Tanto métodos quanto técnicas podem ser classificados de acordo com o tipo de atividade que podem apoiar. Diretrizes podem ser classificadas em *roteiros* e *normas*. Um *procedimento* pode ser automatizado por *ferramentas de software* e um método, como procedimento sistemático, descreve uma ou mais atividades em termos de outras atividades. Uma organização possui o seu comportamento definido pelo conjunto de processos executados na organização e estes podem estar em conformidade com normas. *Projetos* são empreendimentos que são conduzidos pela organização de acordo com um determinado esquema, que define o processo para execução do projeto e as equipes de projeto.

A sub-ontologia de Estratégia Geral estabelece o vocabulário necessário para descrever os aspectos que definem como a organização interage com o ambiente. Uma organização atua em um domínio de conhecimento, o que significa que ela possui capital intelectual relativo ao domínio e executa atividades que requerem conhecimentos que pertencem a este domínio. Um *serviço* é um produto não tangível, abstrato, oferecido pela organização para satisfazer a necessidade ou desejo de um determinado cliente ou mercado, em contraste com o artefato, que é um produto tangível. Artefatos e serviços são negociados entre organizações, que assumem o papel de fornecedoras e clientes.

3.4. Linguagem para Modelagem de Processos

A definição de uma linguagem gráfica para modelagem de processos organizacionais com fins descritivos foi motivada pelo fato das linguagens existentes não contemplarem a representação dos conhecimentos requeridos e produzidos ao longo dos processos ou possuírem um conjunto excessivo de objetos com traçados, muitas vezes, semelhantes ou desnecessariamente complexos. A linguagem definida é composta de elementos gráficos que podem ser do tipo área, objeto ou ligação, onde uma ligação estabelece uma relação entre dois objetos e uma área agrupa objetos, definindo um contexto para os mesmos. Objetos ainda permitem adornos, utilizados para representar explicitamente características dos objetos. Nas tabelas 1 a 4, cada elemento da linguagem é apresentado.

Tabela 1 – Notação e Definição dos Objetos

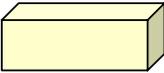
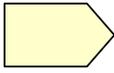
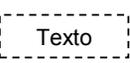
Notação	Definição
	<i>Nome:</i> Processo <i>Descrição:</i> Objeto referente ao conceito de mesmo nome definido na ontologia de organização <i>Atributos Especiais:</i> Origem (Interno, Externo)
	<i>Nome:</i> Evento <i>Descrição:</i> Objeto que representa um acontecimento no ambiente que provoca o início ou fim de um processo. A notação é proveniente do produto comercial de <i>workflow ARIS ToolSet</i>
	<i>Nome:</i> Ator <i>Descrição:</i> Objeto que representa um pessoa, agente ou unidade organizacional, que são conceitos da ontologia de organização. A notação foi utilizada por KRUCHTEN [11]
	<i>Nome:</i> Atividade <i>Descrição:</i> Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação foi utilizada por KRUCHTEN [11] <i>Atributos Especiais:</i> Origem (Interna, Externa) e Granularidade (Elementar ou Composta)
	<i>Nome:</i> Estado Inicial <i>Descrição:</i> Objeto proveniente dos diagramas de estado que indica onde é iniciado o fluxo de atividades que define um processo ou uma atividade composta
	<i>Nome:</i> Estado Final <i>Descrição:</i> Objeto proveniente dos diagramas de estado que indica onde é encerrado o fluxo de atividades que define um processo ou uma atividade composta
	<i>Nome:</i> Conhecimento Explícito <i>Descrição:</i> Objeto que representa um conhecimento que pode ser expresso em palavras e números e ser facilmente transmitido e compartilhado. A notação foi proposta por ALLWEYER [12]
	<i>Nome:</i> Conhecimento Implícito <i>Descrição:</i> Objeto que representa um conhecimento que é altamente pessoal e difícil de formalizar, o que o torna também difícil de ser compartilhado. A notação foi proposta por ALLWEYER [12]
	<i>Nome:</i> Comunicação <i>Descrição:</i> Objeto que representa a comunicação de dados ou informações a partir da, ou para a, execução de uma atividade. A comunicação pode ser verbal ou escrita e exemplos são e-mail e fax
	<i>Nome:</i> Repositório <i>Descrição:</i> Objeto que representa um meio magnético para o armazenamento de dados e informações. A notação é proveniente do produto comercial de <i>workflow ARIS ToolSet</i>
	<i>Nome:</i> Arquivo <i>Descrição:</i> Objeto que representa um local físico para armazenamento de documentos e comunicações escritas
	<i>Nome:</i> Documento <i>Descrição:</i> Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação é proveniente do produto comercial de <i>workflow ARIS ToolSet</i>
	<i>Nome:</i> Componente de Hardware <i>Descrição:</i> Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação é baseada na notação de componente da UML
	<i>Nome:</i> Componente de Software <i>Descrição:</i> Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação é baseada na notação de componente da UML
	<i>Nome:</i> Peça <i>Descrição:</i> Objeto referente ao conceito de mesmo nome definido na ontologia de organização
	<i>Nome:</i> Matéria-Prima <i>Descrição:</i> Objeto referente ao conceito de mesmo nome definido na ontologia de organização
	<i>Nome:</i> Bem <i>Descrição:</i> Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação pode ser substituída por uma mais significativa para o modelo específico <i>Atributos Especiais:</i> Tipo (Usufruto, Software, Hardware e Equipamento de Produção)
	<i>Nome:</i> Nota Explicativa <i>Descrição:</i> Objeto que permite que notas explicativas sejam adicionadas ao modelo <i>Atributos Especiais:</i> Texto

Tabela 2 – Notação e Definição dos Adornos

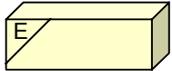
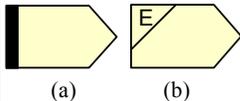
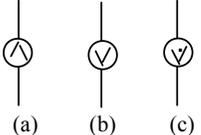
Notação com Adornos	Definição dos Adornos
	<i>Descrição:</i> Adorno que indica que o processo é externo, ou seja, que é executado por outra organização
	<i>Descrição:</i> (a) Adorno que indica que a atividade é composta, o que significa que ela pode ser decomposta em sub-atividades; (b) Adorno que indica que a atividade é externa, ou seja, que é executada por outra organização
	<i>Descrição:</i> (a) Adorno que indica a operação lógica E; (b) Adorno que indica a operação lógica OU; (c) Adorno que indica a operação lógica OU Exclusivo
	<i>Descrição:</i> Adorno que indica que foi especificado um caminho para acesso ao conhecimento disponível em meio magnético. Este adorno só deve ser utilizado se a visualização do modelo for apoiada por uma ferramenta de software que permita o acesso ao conhecimento <i>Atributos Especiais:</i> Localização do Arquivo

Tabela 3 – Notação e Definição das Áreas

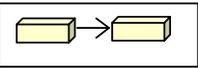
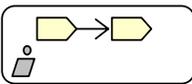
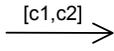
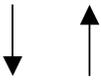
Notação	Definição
	<i>Nome:</i> Grupo de Processos <i>Descrição:</i> Área que agrupa processos relacionados
	<i>Nome:</i> Área de Ator <i>Descrição:</i> Área que agrupa atividades executadas por um ator ou grupo de atores. O ator ou o grupo de atores também precisa estar contido na área

Tabela 4 – Definição e Notação das Ligações

Notação	Definição
	<i>Nome:</i> Fluxo de Controle <i>Descrição:</i> Ligação que indica a passagem de controle do objeto origem para o objeto destino. O c1 e o c2 indicados na notação são rótulos de condições <i>Atributo Especial:</i> Condição, formada por rótulo e descrição
	<i>Nome:</i> Fluxo de Entrada/Saída <i>Descrição:</i> Ligação que estabelece um insumo ou um produto de uma atividade. Quando o objeto de origem ou destino é um armazenador (repositório ou arquivo), a notação pode incluir os rótulos das informações trafegadas, existindo, então, um atributo especial <i>Atributo Especial:</i> Informação, formada por rótulo e descrição
	<i>Nome:</i> Ligação Não Direcionada <i>Descrição:</i> Ligação utilizada para conectar bens de produção utilizados como recursos para execução das atividades e para conectar eventos que atuam sobre processos, provocando o seu início ou fim. No segundo caso, um atributo especial é definido <i>Atributo Especial:</i> Papel do Evento (Iniciador, Terminador)
	<i>Nome:</i> Ligação para Nota Explicativa <i>Descrição:</i> Ligação que estabelece que uma nota explicativa é referente a um elemento do modelo

3.5. Pesquisa sobre a Importância dos Tipos de Conhecimento

O processo utilizado para realização da pesquisa foi adaptado do processo de experimentação proposto por WOHLIN *et al.* [13].

O objetivo estabelecido para a pesquisa, segundo o modelo GQM [14], foi: **Analisar** os ambientes organizacionais para desenvolvimento e manutenção de software **com o propósito de** caracterizar os conhecimentos mencionados na literatura técnica de Engenharia de Software

com respeito à importância percebida em termos da necessidade de tê-los, de alguma forma, disponíveis ao longo do desenvolvimento e da manutenção de software
do ponto de vista dos programadores, analistas, gerentes de projeto e diretores
no contexto dos projetos de software conduzidos no ano de 2002 em empresas sediadas em Salvador.

A população alvo, representada pela força de trabalho envolvida com o desenvolvimento e a manutenção de software na cidade de Salvador, foi estimada em 472 pessoas a partir dos dados da Relação Anual de Informações Sociais (RAIS) - Ano Base 2001. A amostra da população foi selecionada a partir das 27 empresas do estado da Bahia que responderam às pesquisas sobre Qualidade e Produtividade no Setor de Software Brasileiro nos anos de 1999 e 2001 [15,16] ou que participaram do censo, realizado em 2001, das empresas associadas a SOFTEX [17]. Destas empresas, 6 já haviam sido fechadas ou não puderam ser localizadas e 3 não aceitaram participar da pesquisa. Sendo assim, o tamanho inicial da amostra foi de 99 programadores, 203 analistas, 61 gerentes e 30 diretores, totalizando 393 pessoas.

O planejamento da pesquisa prosseguiu com a formulação de hipóteses, a definição do projeto da pesquisa, a preparação da instrumentação (que consistiu de questionários em papel e disponíveis na Web), a identificação das ameaças à validade da pesquisa [13] e a adoção de procedimentos para tratamento dessas ameaças.

A taxa de não resposta obtida com a operação da pesquisa nas empresas foi de 55,56% para programadores, 62,56% para analistas, 44,26% para gerentes de projeto e 66,67% para diretores. Foram obtidos 153 questionários válidos, sendo 41 de programadores, 70 de analistas, 32 de gerentes e 10 de diretores.

Grupos de empresas, projetos e participantes foram estabelecidos para permitir o teste das hipóteses, que foram formuladas de forma a investigar a independência da importância atribuída ao tipo de conhecimento em termos do tamanho da empresa, tamanho do projeto, e da formação e experiência dos profissionais. Os participantes da pesquisa, agrupados em programadores, analistas, gerentes e diretores, tiveram pesos atribuídos a sua opinião de acordo com a sua formação e tempo de experiência. Para testar as hipóteses, foi selecionado o teste não paramétrico do Qui-quadrado de Pearson.

O resultado do teste das hipóteses, apesar de estatisticamente significativo, não tem, sozinho, importância prática para o estabelecimento de uma estratégia incremental para incorporação de conteúdo e desenvolvimento de serviços e ferramentas para uso nos ADSOrg. Desta forma, os tipos de conhecimento mencionados no questionário da pesquisa foram ordenados de acordo com a importância atribuída por cada grupo de participantes (exemplos: programadores de micro-empresa, analistas de projetos pequenos, gerentes com pouca experiência, diretores). Em seguida, os tipos de conhecimento foram ordenados novamente, mas de acordo com a sua ocorrência entre os 25 mais importantes e os 25 menos importantes identificados por cada grupo. O raciocínio utilizado considerou que um tipo de conhecimento é mais importante quanto maior for o número de ocorrências do mesmo entre os 25 mais importantes e menor for o número de ocorrências entre os 25 menos importantes. Uma parte dos resultados obtidos é apresentada na tabela 5, que contém os 50 tipos de conhecimento melhor classificados, de um total de 75, sendo utilizados os seguintes acrônimos: DM, para conhecimento sobre o domínio de aplicação, ORG para conhecimento sobre a organização que desenvolve e mantém software, MP para melhores práticas da organização no desenvolvimento e na manutenção de software, DH para dados históricos da organização relacionados ao desenvolvimento e à manutenção de software, OC para outros conhecimentos relacionados ao desenvolvimento e à manutenção de software, PT para conhecimento sobre parceiros técnicos, CL para conhecimento sobre clientes e LT para indicação de referências da literatura técnica.

Tabela 5 – Classificação Geral dos Tipos de Conhecimentos quanto à Importância atribuída nos Ambientes para Desenvolvimento e Manutenção de Software

	Grupo	Tipo de Conhecimento
1 ^o	MP	Codificação
2 ^o	MP	Testes
3 ^o	MP	Documentação de Software
4 ^o	MP	Gerência de Qualidade
5 ^o	OC	Processo de Software da Organização
5 ^o	MP	Processo de Solução de Problemas
7 ^o	MP	Análise e Especificação de Requisitos de Software
8 ^o	MP	Projeto (<i>design</i>) de Software
9 ^o	DM	Domínio da Aplicação
10 ^o	MP	Gerência de Projeto
11 ^o	MP	Engenharia de Sistemas
12 ^o	OC	Roteiros de Documentos com Exemplos de Uso
13 ^o	MP	Avaliação e Melhoria de Processo de Software
14 ^o	OC	Itens de Software Úteis para Desenvolvimentos Futuros
15 ^o	MP	Modelagem de Processo
16 ^o	MP	Manutenção de Software
17 ^o	OC	Avaliação e Áreas para Melhoria do Processo de Software da Organização
18 ^o	ORG	Distribuição de Competências entre os Profissionais
19 ^o	LT	Relatos sobre as MP da Indústria de Software
20 ^o	ORG	Objetivos e Metas Organizacionais
20 ^o	MP	Treinamento
22 ^o	DH	Gerência de Projeto
23 ^o	MP	Operação de Sistema e Suporte a Usuários
23 ^o	OC	Tipos de Software desenvolvidos na Organização
25 ^o	MP	Reutilização de Itens de Software
26 ^o	LT	Indicação de Referências sobre Inovações Tecnológicas em ES
26 ^o	LT	Relatos sobre as LA da Indústria de Software
28 ^o	ORG	Áreas Críticas para o Alcance dos Objetivos Organizacionais
29 ^o	DH	Métricas de Produto
30 ^o	LT	Indicação de Referências sobre Conhecimento Teórico em ES
31 ^o	ORG	Restrições, Deficiências e Potenciais
32 ^o	DH	Gerência de Qualidade
33 ^o	DH	Métricas de Processo
34 ^o	ORG	Processos Organizacionais
35 ^o	ORG	Missão da Organização
36 ^o	MP	Gerência de Configuração
37 ^o	MP	Gerência de Riscos
38 ^o	ORG	Diretrizes e Normas
39 ^o	OC	Respostas para as Perguntas mais Frequentes entre os Desenvolvedores
40 ^o	ORG	Estrutura Organizacional
41 ^o	CL	Processos Organizacionais
42 ^o	MP	Processo de Fornecimento de Produto de Software
43 ^o	CL	Estrutura Organizacional
44 ^o	CL	Alocação de Profissionais à Estrutura Organizacional
45 ^o	PT	Restrições, Deficiências e Potenciais
46 ^o	ORG	Relações Informais existentes entre os Profissionais
47 ^o	CL	Diretrizes e Normas
48 ^o	CL	Missão da Organização
49 ^o	ORG	Contratos/Acordos com Fornecedores, Parceiros Técnicos e Clientes
50 ^o	DH	Gerência de Riscos

4. Construção da Solução

A Estação TABA foi definida e construída na COPPE/UFRJ para auxiliar na definição, implementação e execução de Ambientes de Desenvolvimento de *Software*, tendo sido estendida ao longo dos últimos dez anos [4,18,19]. Para a construção de ADSOrg, a infra-estrutura fornecida pela Estação TABA foi novamente estendida e utilizada. O

Meta-ambiente da Estação TABA já possuía funcionalidades para auxiliar o engenheiro de software a definir e instanciar o ADS mais adequado a um projeto específico. A estratégia para construção de ADSOrg estabeleceu um passo anterior à instanciação, que é a configuração do Meta-ambiente para uma organização, de forma a prover o ambiente ideal para o acúmulo e gerência do conhecimento organizacional relevante para as atividades de desenvolvimento e manutenção de software (Ambiente Configurado), a partir do qual os ADSOrg Instanciados mais adequados aos projetos específicos da organização podem ser criados para disponibilizar conhecimentos, auxiliar a execução das atividades e apoiar o aprendizado organizacional a partir dos projetos. Um Ambiente Configurado e os respectivos ADSOrg Instanciados contemplam, juntos, os requisitos e componentes propostos para um ADSOrg, representando uma solução para implementação do conceito.

Verificou-se, então, a necessidade de definição de dois processos para implementação da estratégia: um para orientar a configuração do Meta-ambiente para uma organização, definido como parte deste trabalho e resumido na tabela 6, e outro para orientar a instanciação de ADSOrg a partir de um Ambiente Configurado, o qual foi definido por BERGER [20].

Tabela 6 – Processo de Configuração do Meta-ambiente para uma Organização

<p>Atividade 1: Contextualizar configuração, de forma a fornecer os subsídios necessários à elaboração da proposta de configuração.</p> <p>Sub-atividades:</p> <ul style="list-style-type: none"> • <u>Realizar entrevistas</u> • <u>Caracterizar a organização</u> • <u>Identificar cultura organizacional na área de software</u> • <u>Identificar objetivos para a configuração do ambiente</u>
<p>Atividade 2: Elaborar proposta para configuração do ambiente para a organização, o que implica em especificar o conteúdo e as ferramentas que serão disponibilizadas no Ambiente Configurado e em planejar a configuração.</p> <p>Sub-atividades:</p> <ul style="list-style-type: none"> • <u>Identificar as Teorias de Domínio</u> a serem disponibilizadas • <u>Identificar os processos</u> a serem definidos e/ou que precisam ser revistos • <u>Identificar as ferramentas</u> a serem disponibilizadas • <u>Definir equipes e responsabilidades</u> • <u>Definir cronograma</u> • <u>Definir custos</u> • <u>Enviar a proposta</u> para os Representantes da Organização
<p>Atividade 3: Registrar parecer sobre a proposta, indicando a sua aprovação, a necessidade de modificações e ajustes, ou a não aprovação da proposta.</p>
<p>Atividade 4: Definir o processo padrão da organização, que contempla os processos de desenvolvimento e de manutenção.</p> <p>Sub-atividades:</p> <ul style="list-style-type: none"> • <u>Caracterizar o processo padrão</u>, estabelecendo (se desejado) um processo padrão a ser utilizado como base • <u>Incluir as atividades do processo selecionado como base</u> na definição do novo processo (quando pertinente) • <u>Incluir atividades ISO/IEC 12207</u> • <u>Incluir atividades próprias do desenvolvimento ou da manutenção de determinados tipos de software</u> • <u>Incluir atividades próprias da organização</u> para a qual o ambiente está sendo configurado • <u>Incluir atividades orientadas a domínio</u> • <u>Estabelecer as atividades obrigatórias</u> do processo padrão • <u>Adaptar os roteiros de documentos</u> para a organização
<p>Atividade 5: Definir processos especializados para os paradigmas de desenvolvimento utilizados na organização e, opcionalmente, para diferentes tipos de software.</p> <p>Sub-atividades:</p> <ul style="list-style-type: none"> • <u>Especializar atividades do processo padrão</u>, de forma a torná-las mais adequadas ao paradigma e métodos • <u>Incluir atividades próprias do desenvolvimento ou da manutenção de determinados tipos de software</u> • <u>Estabelecer as atividades obrigatórias</u> do processo especializado • <u>Especializar roteiros de documentos</u> definidos para a organização (quando necessário).

Tabela 6 – Processo de Configuração do Meta-ambiente para uma Organização (cont.)

<p>Atividade 6: Definir teoria do domínio para um determinado domínio de conhecimento.</p> <p>Sub-atividades:</p> <ul style="list-style-type: none"> • <u>Revisar o propósito</u> estabelecido para a teoria do domínio durante a elaboração da proposta de configuração • <u>Especificar os requisitos</u> da teoria do domínio • <u>Capturar a teoria do Domínio</u> • <u>Conceituar a teoria do Domínio</u> • <u>Formalizar a teoria do Domínio</u> • <u>Identificar as tarefas</u> que são executadas no contexto definido por cada subteoria do domínio • <u>Avaliar a teoria do Domínio</u> • <u>Implementar a teoria do Domínio</u> no Meta-ambiente TABA
<p>Atividade 7: Descrever tarefa que tenha sido identificada como pertinente ao contexto de uma subteoria do domínio.</p> <p>Sub-atividades:</p> <ul style="list-style-type: none"> • <u>Descrever a tarefa em linguagem natural</u> • <u>Definir a estratégia para solução da tarefa</u> • <u>Capturar conceitos e relações da tarefa</u> • <u>Descrever o fluxo de controle da tarefa</u> • <u>Formalizar a Descrição da Tarefa</u> • <u>Avaliar a Descrição da Tarefa</u> • <u>Implementar a Descrição da Tarefa</u> no Meta-ambiente TABA
<p>Atividade 8: Incluir a referência a novas ferramentas na descrição dos processos padrão e/ou especializados, de forma que possam ser integradas na criação do Ambiente Configurado</p>
<p>Atividade 9: Criar Ambiente Configurado para a organização.</p> <p>Sub-atividades:</p> <ul style="list-style-type: none"> • <u>Definir o Ambiente Configurado</u>, selecionando o processo padrão e as Teorias de Domínio a serem fornecidas • <u>Gerar o Ambiente Configurado</u> (código executável) • <u>Testar o Ambiente Configurado</u>

Dois ferramentas foram desenvolvidas como parte deste trabalho: a ferramenta *Config* (figura 3), que apóia o processo de configuração de ambientes para as organizações, e a ferramenta *ProcKnow*, que apóia a descrição dos processos organizacionais, fornecendo este conhecimento para orientar os desenvolvedores na execução de suas atividades (figura 4).

The screenshot shows the 'Configuração de Ambiente' application window. The 'Realizar Entrevistas' tab is active. The 'Organização' dropdown is set to 'Unidade de Cardiologia e Cirurgia Cardiovascular / Fundação Bahiana de Cardiologia'. The 'Entrevistas' table contains the following data:

Identificação	Data/Hora	Situação
Entrevista com Patrocinador	12/08/2003	Agendada
Entrevista com o Responsável Técnico	15/08/2003	Agendada

The 'Participantes' table is empty. The 'Roteiro' field contains '-Objetivos da Configuração'. The 'Resultado' and 'Aspectos a Esclarecer' fields are empty.

Figura 3 – Tela Inicial da Config: Atividade de Realizar Entrevistas

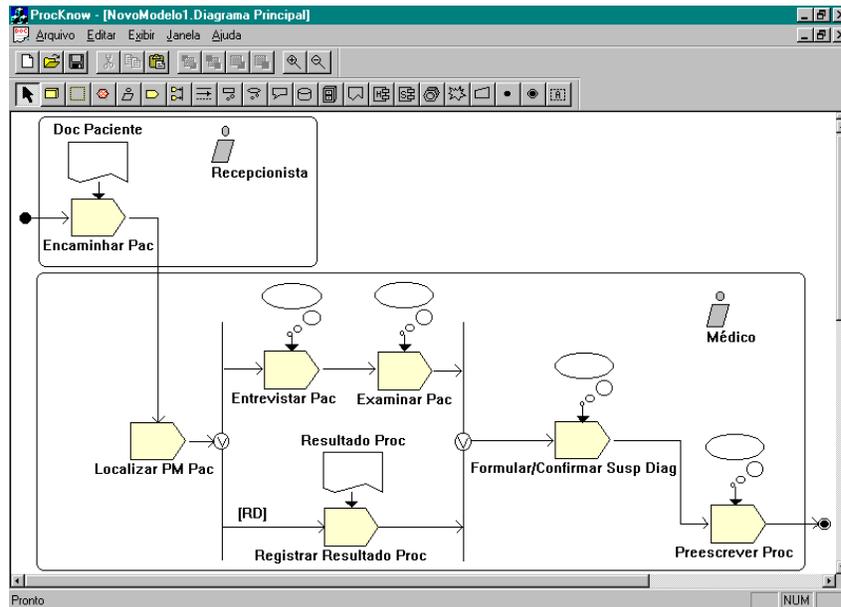


Figura 4 – Tela da Procknow: Diagrama Maximizado de um Processo

A ferramenta *Config* possui uma barra na lateral esquerda com a representação explícita do processo de configuração e dois ícones na lateral direita superior que têm, respectivamente, a finalidade de fornecer o conhecimento disponível no ambiente para auxiliar a execução da atividade e chamar o módulo da ferramenta *Acknowledge* [21] que permite o registro de conhecimento obtido na execução da atividade. Outras ferramentas foram desenvolvidas no contexto do projeto “Ambientes de Desenvolvimento de Software Orientados a Organização”, sendo importante destacar, para o contexto deste trabalho, a ferramenta *Sapiens* [22], que permite a descrição e visualização de estruturas organizacionais, englobando profissionais alocados e competências requeridas e possuídas ao longo dessas estruturas.

5. Conclusões

ADSOrg foram definidos para apoiar a Gerência do Conhecimento ao longo dos processos de software. Vários tipos de conhecimento são importantes neste contexto e, muitas vezes, problemas ocorrem por que as equipes de projeto não dispõem de conhecimento suficiente sobre o domínio de aplicação, sobre a própria organização em que trabalham e/ou sobre as práticas de Engenharia de Software que apresentaram melhores resultados na organização. Em organizações em que o desenvolvimento e a manutenção de software são atividades de negócio, conhecer o cliente é também essencial.

A Estação TABA teve a sua infra-estrutura estendida para permitir a configuração de ambientes organizacionais (Ambiente Configurado) capazes de instanciar ADSOrg para projetos específicos (ADSOrg Instanciados) e novas ferramentas foram desenvolvidas.

Para verificar a viabilidade da solução proposta para a construção de ADSOrg, a Estação TABA foi configurada para a Fundação Bahiana de Cardiologia (FBC), organização que desenvolve diferentes tipos de *software* médico. Uma importante perspectiva futura desta tese é a avaliação das experiências de uso dos Ambientes Configurados e ADSOrg Instanciados na indústria, o que já está sendo iniciado em 8 empresas de Salvador, através de um projeto conjunto da ASSEPRO-BA, COPPE/UFRJ e UNIFACS.

Outra perspectiva futura é a aplicação da pesquisa em outras cidades, de forma a reforçar ou refutar os resultados obtidos sobre a importância atribuída aos tipos de conhecimento.

6. Referências Bibliográficas

- [1] O'LEARY, D. E., STUDER, R., "Knowledge Management: An Interdisciplinary Approach", *IEEE Intelligent Systems*, v. 16, n. 1, pp. 24-25, Jan/Feb. 2001.
- [2] MOURA, L., *Taxonomia de Ambientes de Desenvolvimento de Software*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1992.
- [3] OSTERWEIL, L. J., "A Process-Object Centered View of Software Environment Architecture". In: *Advanced Programming Environments*, v. 244, *Lecture Notes in Computer Science*, Springer-Verlag, pp. 156-174, 1986.
- [4] OLIVEIRA, K., *Modelo para Construção de Ambientes de Desenvolvimento de Software Orientados a Domínio*, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1999.
- [5] OLIVEIRA, K., ZLOT, F., ROCHA, A. R., *et al.*, "Domain-oriented software development environment", *Journal of Systems and Software* (Article in Press), 2004. In: <http://authors.elsevier.com/sd/article/S0164121203002334>.
- [6] VILLELA, K., TRAVASSOS, G. H., ROCHA, A. R., "Toward Enterprise Oriented Software Development Environments". In: *Workshop de Teses e Dissertações em Engenharia de Software - XIV SBES*, pp. 379-384, João Pessoa, Brasil, Out. 2000.
- [7] VILLELA, K., ZLOT, F., SANTOS, G., *et al.*, Knowledge Management in Software Development Environments. In: 14th ICSSEA, Paris, France, 2001.
- [8] EMAM, K. E., DROUIN, J., MELO W., *SPICE – The Theory and Practice of Software Process Improvement and Capability Determination*, IEEE Computer Society Press, 1998.
- [9] USCHOLD, M., GRUNINGER, M., "Ontologies: principles, methods and applications", *The Knowledge Engineering Review*, v. 11, n. 2, pp 93-136, 1996.
- [10] FALBO, R., *Integração de Conhecimento em um Ambiente de Desenvolvimento de Software*, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1998.
- [11] KRUCHTEN, P., *The Rational Unified Process: An Introduction, Object Technology Series*, Addison-Wesley, 2000.
- [12] ALLWEYER, T., "A Framework for Re-designing and Managing Knowledge Processes", ARI White Paper, IDS Scheer AG, Saarbrücken, Germany, 1999. In: <http://www.changeware.net/doc/km.pdf> (acessado em 02/03/2004).
- [13] WOHLIN, C., RUNESON, P., HÖST, M., *et al.*, *Experimentation in Software Engineering: An Introduction, The Kluwer International Series in Software Engineering*, Norwell, USA, Kluwer Academic Publishers, 2000.
- [14] BASILI, V., CALDIERA, G., ROMBACH, H., "Goal Question Metric Paradigm". In: *Encyclopedia of Software Engineering*, v. 1, John Wiley & Sons, pp. 528-532, 1994.
- [15] SEPIN/MCT, *Qualidade e Produtividade no Setor de Software Brasileiro 1999*, Secretaria de Política de Informática/Ministério de Ciência e Tecnologia, Brasília, 2000.
- [16] SEPIN/MCT, *Qualidade e Produtividade no Setor de Software Brasileiro 2001*, Secretaria de Política de Informática/Ministério de Ciência e Tecnologia, Brasília, 2002.
- [17] SEPIN/MCT, *Levantamento do Universo de Associadas SOfTEX*, Pesquisa Censo SW, Secretaria de Política de Informática/Ministério de Ciência e Tecnologia, Brasília, 2001.
- [18] ROCHA, A. R., SOUZA, J. M., AGUIAR, T. C., "TABA: A Heuristic Workstation for Software Development". In: *Proceedings of COMPEURO 90*, pp. 126-129, Tel Aviv, Israel, May 1990.
- [19] TRAVASSOS, G. H., *O Modelo de Integração de Ferramentas da Estação TABA*, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1994.
- [20] BERGER, P., *Instanciação de Processos de Software em Ambientes Configurados na Estação TABA*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2003.
- [21] MONTONI, M., *Aquisição de Conhecimento: Uma Aplicação no Processo de Software*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2003.
- [22] SANTOS, G., *Representação da Distribuição do Conhecimento, Habilidades e Experiências através da Estrutura Organizacional*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2003.