

## ORIXÁS: Um Ambiente para Desenvolvimento de Sistemas Baseados em Conhecimento

*Vera Maria B. Werneck<sup>1</sup>; Monica Geraldini<sup>2</sup>; Luiz Reuter S. Torro<sup>1</sup>;  
Rafael B. Teixeira<sup>1</sup>; Rodrigo Alves da Costa<sup>1</sup>; Rogério J. da Silva<sup>1</sup>;  
Marcelo Lício de Jesus<sup>1</sup>; André A. Bezerra<sup>1</sup>; Rafael C. Moreira<sup>1</sup>*

<sup>1</sup> UERJ - Universidade Estadual do Rio de Janeiro  
Departamento de Informática e Ciência da Computação  
Rua São Francisco Xavier 524, 6º Andar, Bloco B  
Maracanã - 20 550-013 - Rio de Janeiro – Brasil

<sup>2</sup> COPPE/UFRJ - Programa de Engenharia de Sistemas e Computação  
Universidade Federal do Rio de Janeiro  
e-mail: vera@ime.uerj.br

### RESUMO

O desenvolvimento de Sistemas Baseados em Conhecimento apresenta várias dificuldades como a necessidade de aquisição do conhecimento de experiências pessoais de especialistas, a modelagem desse conhecimento e a garantia da qualidade do produto gerado. Diferentes pesquisas tem concentrado em soluções para cada um desses aspectos. No entanto, qualidade em software não se atinge de forma espontânea. É necessário um controle melhor que integre esses diferentes aspectos e conduza a melhores resultados. Este trabalho descreve o ambiente ORIXÁS, cujo objetivo é apoiar o desenvolvimento de sistemas baseados em conhecimento, de acordo com um processo de desenvolvimento bem definido de acordo com as normas ISO e através do uso sistemático de ferramentas que possibilitam a aplicação do método KADS-Estendido, de uma estratégia de aquisição de conhecimento, de procedimentos gerenciais e de controle da qualidade adequados.

**Palavras Chaves:** Ambiente de Desenvolvimento, Processo, Sistema Baseado em Conhecimento, Ferramentas

### ABSTRACT

Knowledge Based Systems construction and quality assurance have many problems that remain unsolved. Quality is not a spontaneous achievement in software development. Different technologies present different characteristics and these must be taken into account when defining software engineering that is adequate to the construction of systems applying in a particular technology. This work describes ORIXÁS, an environment for the development of Knowledge Based Systems with a well-defined development process according to ISO standards, through the systematic use of tools that allow the application of the extended KADS method, as well as suitable approach to knowledge acquisition, management and quality control procedures.

**Key-words:** Development Environment, Process, Knowledge Based Systems, Tools.

## 1. Introdução

Sistemas baseados em conhecimento (SBC) surgiram na década de 60 como uma solução para possibilitar o desenvolvimento de aplicações inteligentes para problemas complexos. No entanto, inicialmente foram construídos de forma "ad-hoc", sem uso de métodos e técnicas específicas. Embora essas sejam características dos momentos iniciais de introdução de uma nova tecnologia, este processo em geral conduz a produtos de baixa qualidade e com tempo de desenvolvimento e custos ultrapassando o inicialmente previsto.

Garantir qualidade de produto, entretanto, não é trivial. É necessário ter-se um processo de desenvolvimento sistemático e adequado, isto é, uma engenharia de software adequada à construção desses sistemas.

O conceito de Ambientes para Desenvolvimento de Software (ADS) surgiu buscando combinar técnicas, métodos e ferramentas com o objetivo de prover um meio através do qual o engenheiro de software pudesse ter um apoio automatizado ao construir produtos de software [1].

ADS é um sistema computacional que provê suporte para o desenvolvimento, reparo e melhorias em software e para o gerenciamento e controle destas atividades, contendo uma base de dados central, que atua como um repositório para todas as informações relacionadas ao projeto ao longo do seu ciclo de vida, e um conjunto de ferramentas de apoio para as várias atividades técnicas e gerenciais passíveis de automação que devem ser realizadas no projeto [2].

O desenvolvimento de SBC, no entanto, envolve a preocupação com vários aspectos: a necessidade de aquisição do conhecimento de experiências pessoais de especialistas que é reconhecida como o gargalo do desenvolvimento desses sistemas [3], a modelagem desse conhecimento através de métodos adequados e a garantia da qualidade do produto gerado através de um processo sistemático que permita controle e gerência. ORIXÁS foi, então, desenvolvido baseado nessas questões. Nesse sentido, definimos um processo de desenvolvimento particular para SBC, uma estratégia de aquisição de conhecimento para auxiliar nessa atividade e um método para modelagem do conhecimento desde o nível conceitual até ao nível físico, denominado KADS-Estendido. Todos esses aspectos são disponibilizados no ORIXÁS através de um conjunto de ferramentas integradas que apoia o engenheiro do conhecimento durante todo o desenvolvimento de SBC.

Neste artigo descrevemos o ambiente ORIXÁS, apresentando como resolvemos cada uma das dificuldades listadas: a definição do processo de desenvolvimento (seção 2), a estratégia de aquisição de conhecimento adotada (seção 3) e o método KADS-Estendido utilizado (seção 4). Na seção 5 apresentamos o ORIXÁS no que se refere ao conjunto de ferramentas que apoia todo o desenvolvimento de SBC. Na seção 6, apresentamos trabalhos relacionados e na seção 7 as conclusões deste trabalho.

## 2. Processo de Desenvolvimento do ORIXÁS

O processo de desenvolvimento do ORIXÁS foi definido a partir das normas ISO 9000-3 [4] e 9126 [5], contemplando as atividades relativas à construção e avaliação da qualidade do produto e à gerência do projeto. O processo (figura 1) foi definido, experimentado, revisto e validado durante o desenvolvimento de diferentes projetos [6], [7], [8], [9].

O processo de desenvolvimento pressupõe uma estrutura geral para o desenvolvimento integrado de sistemas, partindo do princípio que a tecnologia baseada em conhecimento deve ser desenvolvida de forma gradativa. Por esse motivo, o modelo segue um processo evolutivo

composto de três estágios de desenvolvimento: análise de viabilidade, que verifica a viabilidade de desenvolvimento e a utilidade do sistema; evolução, onde são desenvolvidas versões do sistema num processo evolutivo (o número de versões depende do projeto); e operacional, onde é desenvolvida a versão operacional do produto.

Os estágios são divididos em fases. A *Definição do Domínio do Problema* está presente somente no primeiro estágio do desenvolvimento, ou seja, na Análise de Viabilidade, com o objetivo de se obter uma visão geral do problema. Nesta fase realiza-se a definição do domínio do problema, bem como das áreas do conhecimento, do escopo do projeto, e do relacionamento deste com a organização.

Em cada um desses estágios são previstas as seguintes fases: Planejamento do Projeto, Análise do Conhecimento, Projeto da Versão, Construção da Versão, Avaliação do Produto, Operação da Versão e Avaliação do Processo.

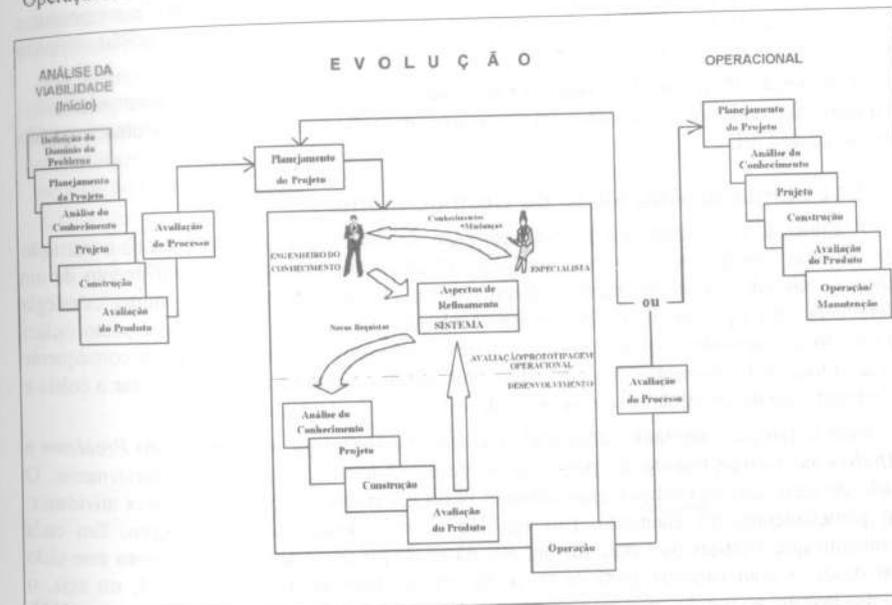


Figura 1. Modelo do Processo de Software do ambiente ORIXÁS

O *Planejamento do Projeto* envolve a produção inicial do Plano do Projeto, sendo que em cada estágio do desenvolvimento são incorporados ao Plano novos detalhes, adaptando-o quando necessário. As atividades realizadas são as de planejamento do desenvolvimento do projeto, planejamento do estágio e avaliação do Plano do Projeto. Na *Análise do Conhecimento* são determinados os aspectos chave do sistema, seus requisitos e o conhecimento contido no produto em desenvolvimento. Esta começa com a aquisição e elicitação do conhecimento desta versão, gerando o projeto conceitual do sistema, definido no documento Especificação de Requisitos. As atividades são a análise do conhecimento específico, a especificação do conhecimento e a avaliação da Especificação de Requisitos. Durante o *Projeto da Versão*, os requisitos do sistema e o modelo conceitual do sistema se transformam na Especificação de Projeto, que é o produto final. Neste momento são

definidos a representação do conhecimento a ser adotada na base de conhecimento e o seu mecanismo de inferência. Na *Construção da Versão*, os programas e base de conhecimento do sistema são produzidos com a realização das atividades de codificação, avaliação e integração de módulos, e teste de integração. Na *Avaliação do Produto*, os especialistas, usuários e outros avaliadores (técnicos) testam o sistema. O teste de aceitação do produto, ou seja, a versão construída do sistema determina se os requisitos descritos na Especificação de Requisitos encontram-se no produto, avaliando-se a base de conhecimento e o comportamento esperado do sistema. A *Operação da Versão* permite a aceitação do sistema com registro em tempo de execução dos problemas e modificações sugeridas pelo usuário. E a *Avaliação do Processo* consiste na avaliação do processo de desenvolvimento do estágio de desenvolvimento realizado. Esta fase é muito importante pois fornece ao engenheiro de software um *feedback* para o planejamento do próximo estágio.

O processo é detalhado contendo a descrição das fases, suas atividades e os produtos gerados em cada uma dessas fases e seus roteiros de documentação. Uma descrição completa do processo pode ser encontrada em [9].

A avaliação da qualidade é realizada ao longo de todo o processo de desenvolvimento, quando são realizadas avaliações intermediárias e avaliações finais de produtos, segundo proposta de [10].

### 3. Estratégia de Aquisição de Conhecimento

A aquisição do conhecimento possui um papel fundamental quando se trata da construção de sistemas inteligentes [3], sendo um fator crítico no processo de desenvolvimento de um sistema baseado em conhecimento. Assim sendo, ORIXÁS está apoiado numa estratégia detalhada para aquisição de conhecimento. A abordagem adotada privilegia o aspecto cíclico da aquisição, considerando-se a complexidade do raciocínio do especialista e a conseqüente necessidade de refinamentos sucessivos para sua captura com o objetivo de otimizar a coleta e o refinamento do conhecimento especializado.

Essa estratégia é diretamente utilizada nas fases de *Definição do Domínio do Problema* e *Análise de Conhecimento* do processo de desenvolvimento apresentado anteriormente. O ciclo de aquisição de conhecimento definido nessa estratégia engloba as seguintes atividades: (i) planejamento, (ii) elicitação, (iii) registro e (iv) revisão do conhecimento. Em cada momento que a estratégia é utilizada nas atividades do processo de desenvolvimento esse ciclo vai desde o planejamento geral desta tarefa, até a obtenção do produto final, ou seja, o conhecimento adquirido, podendo se repetir várias vezes para contínuos refinamentos [11], [12].

O *Planejamento* consiste na definição dos participantes e recursos do processo, das fontes de conhecimento e das sessões de elicitação, sendo que a cada sessão pode-se ter um novo ciclo, reorganizando totalmente o planejamento geral ou simplesmente alterando o planejamento das sessões. Essas atividades podem ser realizadas seqüencialmente ou não e podem se repetir diversas vezes, dependendo das necessidades de sedimentação do conhecimento. Durante o planejamento são elaborados o Plano Geral de Aquisição de Conhecimento e o Plano das Sessões de Elicitação segundo o roteiro proposto em [11].

Uma questão fundamental na elaboração do Plano Geral de Aquisição de Conhecimento é a seleção correta das técnicas de elicitação de conhecimento. Esta seleção pode ser feita com o auxílio de guias que são um instrumento educativo e normativo, introduzindo qualidade ao processo de desenvolvimento [13]. Os Guias de Técnicas de Elicitação de Conhecimento são

disponibilizados ao engenheiro do conhecimento durante o desenvolvimento e fornecem, no momento do planejamento da elicitação do conhecimento, uma forma de padronização sobre como e quando as técnicas de elicitação devem ser utilizadas. Nesta estratégia foram elaborados Guias com 29 diferentes técnicas de Elicitação de Conhecimento classificadas nos seguintes grupos: Entrevistas, Análise e Observação, Casos, Texto, Organização, Revisão e Grupos [11], [12].

A atividade de *Elicitação* é realizada durante as sessões de elicitação do conhecimento. Neste momento devem ser documentados os dados informativos sobre a sessão, além da própria sessão de elicitação e sua avaliação que verifica se o objetivo da sessão foi atingido e se todas as questões e dúvidas levantadas foram abordadas. Ao final da sessão de elicitação, é necessário reavaliar o Plano Geral de Aquisição de Conhecimento, com o objetivo de verificar se ainda permanece válido. Dependendo dessa verificação, ele pode ser alterado, devendo ser utilizado nas sessões subsequentes. A síntese da sessão é documentada no Formulário da Sessão de Elicitação de Conhecimento.

O *Registro* é uma atividade fundamental para a aquisição do conhecimento, pois neste momento, o conhecimento elicitado na atividade anterior é documentado de maneira que possa ser utilizado com facilidade nas próximas etapas do desenvolvimento. Para documentar o conhecimento obtido, o ambiente ORIXÁS se apoia numa documentação preestabelecida e nos modelos do método KADS-Estendido descritos na próxima seção.

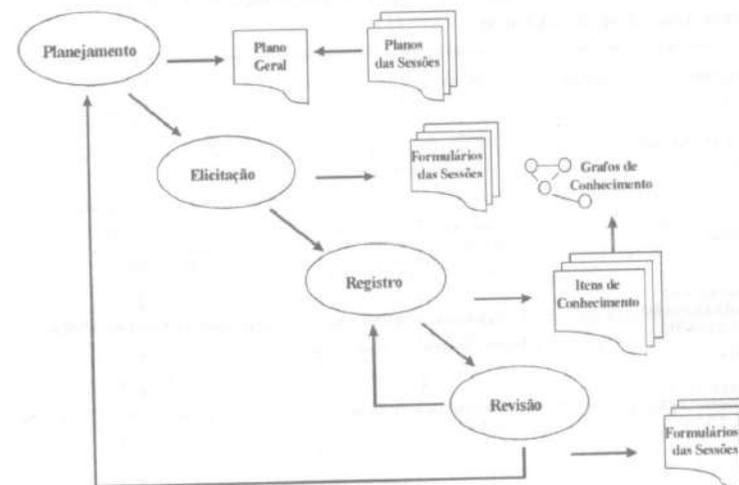


Figura 2. Ciclo de Aquisição de Conhecimento

A atividade de *Revisão*, realizada pelo engenheiro do conhecimento e especialista do domínio, se refere à análise do conhecimento registrado na estrutura do conhecimento, na modelagem ou na documentação prevista no Planejamento do Projeto. Para realização dessa tarefa podem ser aplicadas as técnicas de elicitação do grupo de Revisão. A *revisão* é documentada através do *Formulário da Sessão de Elicitação*. Em seguida à *revisão*, pode haver a necessidade de se agendar novas sessões de elicitação, com a modificação do planejamento ou de se voltar ao *registro*.

A estratégia de aquisição definida pode ser utilizada para múltiplos especialistas. Este fato pode ocasionar alguns problemas como conflitos e divergência de opiniões. Assim, com o

objetivo de identificação e eliminação dos conflitos existentes entre as visões dos diversos especialistas, aplicam-se as técnicas de elicitação do conjunto *Grupos*, que devem estar previstas no planejamento da aquisição. A aplicação das técnicas de Grupo permite a comparação das estruturas individuais e a geração de uma estrutura única, chamada de *Estrutura de Consenso*, agregando a opinião de todos os especialistas.

#### 4. Método KADS-Estendido

ORIXÁS adota o método KADS-Estendido, que é uma extensão do método KADS [14]. O KADS-Estendido foi definido a partir da experiência obtida no desenvolvimento de SBC utilizando o método KADS, que não se mostrou suficiente para a especificação e projeto de um SBC. No KADS-Estendido foi considerado ainda conceitos e experiência no desenvolvimento de sistemas e o conhecimento de outros métodos de desenvolvimento.

O KADS-Estendido foi definido e utilizado em diferentes trabalhos [6], [15], [16], [17]. A partir desse uso, detectamos alguns problemas e realizamos melhorias gerando uma 2ª versão [9].

O método KADS-Estendido prevê modelos para serem elaborados nas diferentes fases do processo de desenvolvimento descrito na seção 2. Na fase de Definição do Domínio do Problema por exemplo foi definido o Modelo do Conhecimento. Na de Análise do Conhecimento foi definido o Modelo de Especialidade, e para o Projeto da Versão foram definidos os Modelos Lógico e Físico. A Figura 3 apresenta uma visão geral do método com seus modelos, diagramas e estruturas.

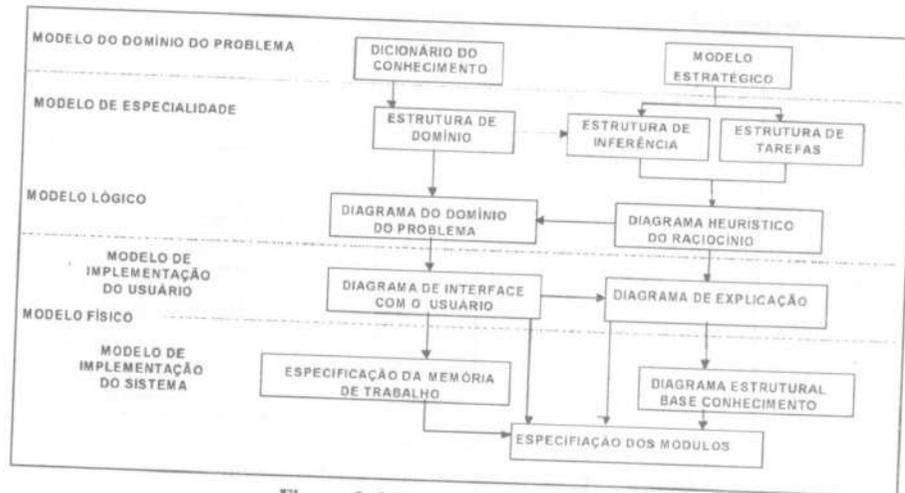


Figura 3. Método KADS-Estendido

O *Modelo do Conhecimento* define o problema em termos de um dicionário de conceitos e permite que o engenheiro do conhecimento crie grafos específicos de conhecimento através das diferentes primitivas em termos de conhecimento fatorial, inferencial e estratégico. Este modelo consiste numa visão do conhecimento de forma geral e específica sem tratamento que vise o desenvolvimento de sistemas. No *Modelo do Conhecimento* é guardado todo o conhecimento adquirido que será utilizado na base de conhecimento, fornecendo uma linguagem livre de implementações e com recursos de diferentes grafos de conhecimento. Este modelo é o produto do registro do conhecimento realizado no ciclo de aquisição.

O *Modelo de Especialidade* tem por base o modelo de especialidade do método KADS com alterações que permitem obter uma melhor visualização do processo de raciocínio e consistência entre as estruturas das diversas camadas do conhecimento. O *Modelo de Especialidade* do KADS-Estendido é composto de três camadas com as seguintes estruturas: Estrutura de Domínio, Estrutura de Inferência e Estrutura de Tarefas. A *Estrutura de Domínio* do KADS-Estendido utiliza, apenas, a representação gráfica da Linguagem de Definição do Domínio (DDL-"Domain Description Language") [14]. A *Estrutura de Inferência* representa o processo de raciocínio do sistema e é utilizado com base numa biblioteca de tarefas. A *Estrutura de Tarefas* representa de forma procedural ou através de uma árvore de tarefas, como as primitivas definidas na estrutura de inferência.

No *Modelo Lógico* tem-se a maior contribuição do KADS-Estendido. A experiência prática verificou ser necessário se ter uma especificação do sistema com uma visão mais próxima do desenvolvedor. No método KADS existem várias propostas para essa passagem, fundamentadas em linguagens formais [14]. A experiência, bem sucedida, de emprego de métodos semi-formais/gráficos e o perfil das equipes de desenvolvimento, normalmente disponíveis em empresas, não habituadas ao uso de métodos formais, fez com que se optasse por continuar usando uma linguagem gráfica no Modelo Lógico. Este é composto por dois diagramas: Diagrama Heurístico do Raciocínio e Diagrama do Domínio do Problema, construídos a partir do Modelo de Especialidade do KADS-Estendido.

O *Diagrama Heurístico do Raciocínio* tem como principal objetivo definir a estrutura de investigação do processo de solução do problema que originará a estrutura da base de conhecimento. Este diagrama é definido a partir do Modelo de Especialidade. O *Diagrama do Domínio do Problema* é, na realidade, uma nova visão da Estrutura de Domínio, definida segundo a Linguagem de Definição do Domínio. Este diagrama é gerado a partir do Diagrama Heurístico do Raciocínio através de um algoritmo. A representação gráfica dos dois diagramas evidencia sua compatibilidade. Assim, tem-se garantida a consistência entre o Diagrama do Domínio do Problema e o Diagrama Heurístico do Raciocínio o que é fundamental para a fase de projeto, pois a partir deles será definido o Modelo Físico. O Diagrama Heurístico do Raciocínio está baseado nos conceitos de classe da Linguagem de Definição do Domínio (DDL-"Domain Description Language") [14].

A modelagem física é composta de dois modelos: Modelo de Implementação do Usuário e Modelo de Implementação do Sistema. Esses modelos estão baseados numa representação semelhante às usadas no modelo lógico do KADS-Estendido.

O *Modelo de Implementação do Usuário* é composto do Diagrama de Interface com o Usuário e do Diagrama de Explicação do Raciocínio. O *Diagrama de Interface com o Usuário* é construído a partir do Diagrama do Domínio do Problema, identificando-se explicitamente a hierarquia dos comandos e a interação do sistema com o usuário. Este diagrama fornece uma visão geral das consultas, dos comandos disponíveis em cada tela, das informações fornecidas pelo usuário e das respostas do sistema. O *Diagrama de Explicação do Raciocínio* é construído a partir do Diagrama Heurístico do Raciocínio e do Diagrama de Interface com o Usuário. Define as informações de explicação e o momento em que o sistema baseado em conhecimento fornece essas explicações a seus usuários. Deve ser apresentado junto com o Diagrama de Interface com o Usuário.

O *Modelo de Implementação do Sistema* possui o Diagrama Estrutural da Base de Conhecimento, a Especificação da Memória de Trabalho e a Especificação dos Módulos. O *Diagrama Estrutural da Base de Conhecimento* é construído a partir das classes heurísticas

do Diagrama Heurístico do Raciocínio, sendo definidas também as estruturas das regras do processo de raciocínio. Este diagrama contém a forma interna de representar o conhecimento no sistema e o formato das regras da base de conhecimento. A *Especificação da Memória de Trabalho* define o conteúdo dos dados que serão armazenados no sistema, dependendo das facilidades oferecidas pelo ambiente de programação utilizado. A *Especificação dos Módulos* define cada módulo do sistema numa representação próxima da linguagem de programação a ser utilizada na etapa de construção.

## 5. O Ambiente do ORIXÁS

ORIXÁS é um sistema computacional que dá apoio ao desenvolvimento de sistemas baseados em conhecimento, através do gerenciamento das atividades definidas no processo de desenvolvimento descrito na seção 2, utilizando a estratégia de aquisição e método de modelagem apresentados anteriormente. Este ambiente está sendo construído em ambiente Windows/95 na linguagem Delphi 4.0 com o objetivo de fornecer suporte a equipes de desenvolvimento de SBC na plataforma de microcomputadores.

Em relação a ADS, ORIXÁS foi desenvolvido baseado numa infra-estrutura básica de ADS [1], [18]. A especificação de cada ferramenta e funcionalidade do ambiente tem gerado especializações e mudanças nesse modelo, sendo atualmente modelado usando UML [19].

Dessa forma definimos como organizar o processo de desenvolvimento no ADS e integrar as ferramentas. A Tabela 1 fornece uma visão geral do ORIXÁS com suas fases, seus produtos finais, os métodos que foram definidos para serem utilizados nas fases deste processo e as ferramentas.

FASES	PRODUTO	MÉTODOS/ TÉCNICAS	FERRAMENTAS
DEFINIÇÃO DO DOMÍNIO DO PROBLEMA	Especificação do Domínio do Problema	Estratégia de Aquisição KADS-Estendido (Modelo do Conhecimento)	OXOSSI - Ferramenta de Aquisição do Conhecimento
PLANEJAMENTO DO PROJETO	Plano do Projeto	-	Planejamento do Processo Búzios-Ferramenta de Documentação
ANÁLISE DO CONHECIMENTO	Especificação de Requisitos	Estratégia de Aquisição KADS-Estendido (Modelo Especialidade)	OXOSSI - Ferramenta de Aquisição do Conhecimento K-Est - Ferramenta KADS-Estendido
PROJETO DA VERSÃO	Especificação de Projeto	KADS-Estendido (Modelo Lógico e Modelo Físico)	K-Est - Ferramenta KADS-Estendido
CONSTRUÇÃO DA VERSÃO	Versão Construída	-	Linguagem de Programação
AValiação DO PRODUTO	Versão Aceita	Método Rocha-estendido	Ferramenta de Avaliação
OPERAÇÃO DA VERSÃO	Versão em Uso	-	-
AValiação DO PROCESSO	Processo Avaliado	CMM	Planilha de KPA2

Tabela 1 - Processo de Desenvolvimento do ORIXÁS segundo suas Fases, seus Produtos, Métodos e Ferramentas

em questão, o que é feito através das seguintes atividades: planejamento do processo de desenvolvimento e planejamento da documentação. Desta forma a cada atividade do processo são definidos os documentos que serão produzidos. A chamada às diversas ferramentas definidas para ORIXÁS é feita através do item *Construção* do menu janela principal.

O objetivo principal das ferramentas no ORIXÁS é auxiliar no processo de produção, avaliação e gerência do software. Assim sendo, apoiam desde a elicitação do conhecimento passando pela análise do conhecimento, projeto e construção até o refinamento da versão do sistema. Isto pressupõe que estas ferramentas fornecem suporte à produção de documentos e de programas e à gerência das diferentes versões dos produtos que serão produzidos ao longo do processo.

Atualmente, ORIXÁS tem a ferramenta de aquisição do conhecimento, ferramenta K-Est que apoia o uso do método KADS-Estendido e ferramenta Búzios de documentação. Além dessas ferramentas, a infra-estrutura do ORIXÁS permite a definição de processos e a geração do Plano do Processo de Desenvolvimento. Na próxima versão do ORIXÁS estão previstas uma ferramenta de avaliação da qualidade para auxiliar o controle da qualidade ao longo de todo o processo e uma ferramenta de acompanhamento e controle das atividades do desenvolvimento de um SBC.

### 5.1 OXOSSI: Ferramenta de Aquisição do Conhecimento

A ferramenta de aquisição de conhecimento, em concordância com as questões mencionadas na estratégia de aquisição, possui como objetivo apoiar o ciclo de aquisição, durante as fases de Definição de Domínio e Análise de Conhecimento do processo de desenvolvimento do ambiente ORIXÁS [11], [12].

A ferramenta permite a elaboração dos Planos Gerais de Aquisição para todos os estágios do processo de desenvolvimento e dos Planos das Sessões de Elicitação, para cada sessão de elicitação de conhecimento realizada. Ela visa disponibilizar uma estrutura para o registro do resultado e avaliação da sessão de elicitação e para o registro da representação desse resultado, além de apresentar o conteúdo do conhecimento adquirido e registrado de forma a facilitar a revisão. Além disso permite, também, o registro do conhecimento através do editor de conhecimento.

A figura 5 apresenta a tela principal da ferramenta, onde são elaborados os Planos Gerais de Aquisição, os Planos das Sessões de Elicitação e as chamadas para os outros módulos da ferramenta (registro da sessão, editor do conhecimento, guias de técnicas de elicitação, glossário de termos e bibliografia).

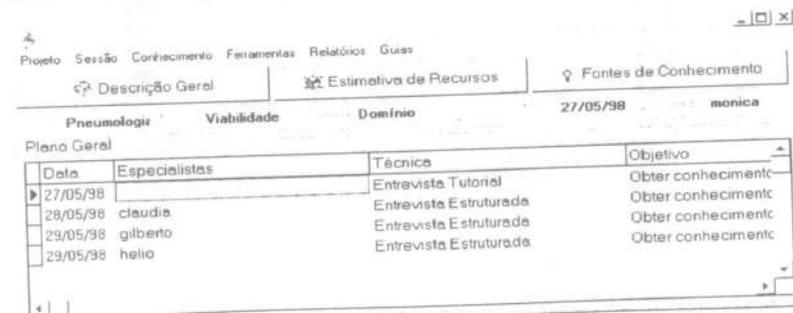


Figura 5. Janela Principal da Ferramenta de Aquisição de Conhecimento

O detalhamento do planejamento para cada sessão é realizado na janela de Planejamento da Sessão com acesso a partir da opção Planejamento da opção Sessão na barra de menu da janela principal (figura 5). Assim é possível definir o escopo das questões a serem levantadas na sessão de elicitação, quais recursos devem ser utilizados para auxiliar o engenheiro na condução da sessão e quais fontes de conhecimento serão examinadas. É importante ressaltar a necessidade de consultas aos Guias de Técnicas de Elicitação no momento do planejamento.

Para documentar a realização da sessão de elicitação, o engenheiro seleciona Resultado a partir da opção Sessão, na barra de menu da janela principal (figura 5), entrando então no módulo de Resultado da Sessão para preenchimento dos dados. Neste momento o engenheiro do conhecimento informa os participantes, o resumo do conhecimento obtido, o material produzido (fitas de vídeo, arquivos eletrônicos, etc.), registra a avaliação e os motivos que acarretaram sucesso ou fracasso ou quaisquer outras observações.

O registro do conhecimento é realizado através de um editor de conhecimento, onde o engenheiro pode criar Itens de Conhecimento ou Grafos de Conhecimento. Ao criar um novo item de conhecimento o engenheiro deve informar se ele está baseado no conhecimento de um único especialista, ou de múltiplos especialistas, fornecendo também sua definição. A partir da seleção de um determinado item de conhecimento, o engenheiro pode verificar, em quais sessões de elicitação aquele item foi identificado. O engenheiro do conhecimento pode optar por criar o item ao elaborar os grafos de conhecimento. Nos grafos, os itens de conhecimento são nós, interligados através de relacionamentos, definidos de acordo com o tipo de conhecimento e, conseqüentemente, com o tipo de grafo a ser elaborado. São três os tipos de grafo, denominados Modelo Factual, Modelo Inferencial e Modelo Estratégico. Para cada grafo, o engenheiro deve especificar o nome, o autor e a data em que foi criado.

A ferramenta permite também a inclusão de bibliografia e impressão dos planos e formulários.

5.2. K-Est: Ferramenta de Apoio ao Método KADS-Estendido

A ferramenta K-Est tem como objetivo apoiar a modelagem de SBC com o método KADS-Estendido através da definição de seus modelos, diagramas, estruturas e especificações. A Figura 6 apresenta uma visão geral dos componentes da ferramenta. Esta ferramenta contempla somente os Modelos de Especialidade, Lógico e Físico. O Modelo do Conhecimento pode ser especificado na ferramenta de aquisição do conhecimento.

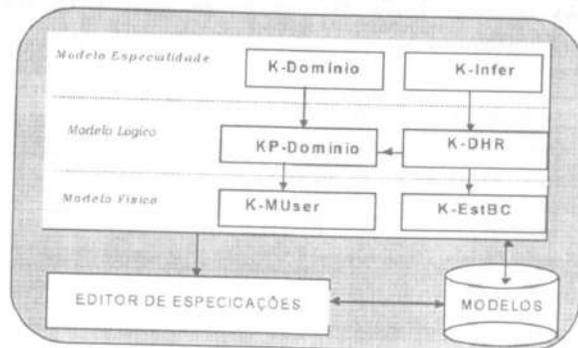


Figura 6: Componentes da Ferramenta K-Est

O Modelo de Especialidade é composto das seguintes estruturas: Estrutura de Domínio, Estrutura de Inferência e Estrutura de Tarefas. O módulo K-Domínio é um editor gráfico para apoiar elaboração da Estrutura de Domínio (Figura 7). O engenheiro do conhecimento pode definir os elementos de uma Estrutura de Domínio a partir da barra de ferramentas (Figura 7).

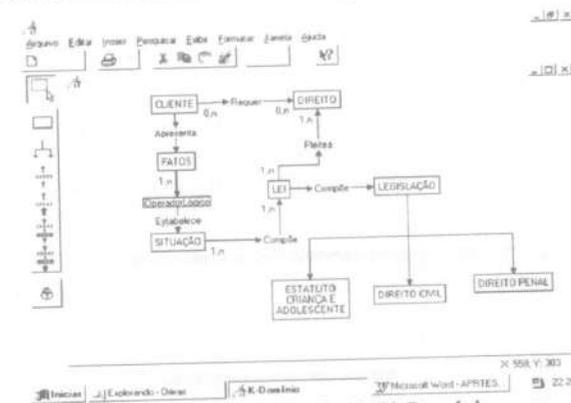


Figura 7: Janela do K-Domínio

O módulo K-Infer permite a definição das estruturas de inferência e de tarefas do modelo de especialidade. Estas podem ser definidas a partir de uma biblioteca de modelos ou como novas. Este módulo também é um editor gráfico que comporta a definição dos dois tipos de diagramas. A partir de uma Estrutura de Inferência o engenheiro do conhecimento pode gerar a Estrutura de Tarefas correspondente.

O Modelo Lógico é composto dos diagramas: Diagrama Heurístico do Raciocínio e Diagrama do Domínio do Problema. O Diagrama Heurístico do Raciocínio é gerado através de um algoritmo a partir da Estrutura de Inferência do Modelo de Especialidade. O módulo K-DHR permite que o Diagrama Heurístico do Raciocínio seja gerado em 8 passos, partindo de uma Estrutura de Inferência. Alguns desses passos são totalmente automatizados. Nos passos semi-automatizados, o K-DHR disponibiliza o ícones para melhorar a visualização ou acrescentar detalhes permitidos pelo método (Figura 8).

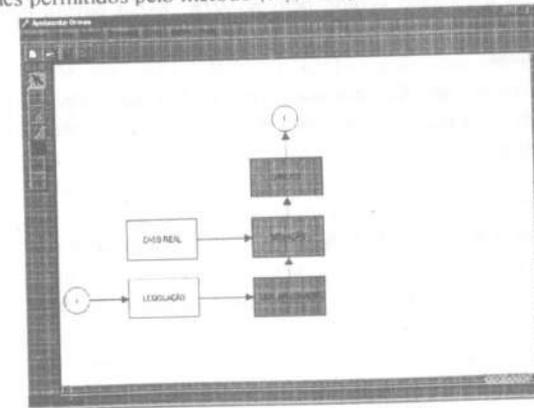


Figura 8: Janela do K-DHR

O Diagrama do Domínio do Problema é, na realidade, uma nova visão da Estrutura de Domínio, sendo gerado pelo engenheiro do conhecimento, a partir do Diagrama Heurístico do Raciocínio através de um algoritmo. O módulo *KP-Domínio* permite que o engenheiro do conhecimento a partir de uma Estrutura de Domínio, gere o Diagrama do Domínio do Problema. O *KP-Domínio* possui um assistente que apoia a construção do Diagrama do Domínio do Problema através da apresentação dos passos para obtenção do diagrama disponibilizando os ícones a serem usados em cada passo (Figura 9).

A modelagem física é composta de dois modelos: Modelo de Implementação do Usuário e Modelo de Implementação do Sistema. O *Modelo de Implementação do Usuário* é composto do Diagrama de Interface com o Usuário e do Diagrama de Explicação do Raciocínio. O módulo *K-MUser* deverá gerar o Modelo de Implementação do Usuário automaticamente a partir do modelo lógico, permitindo a entrada de alguns detalhes pois este deve ser um guia para a prototipação da interface do SBC.

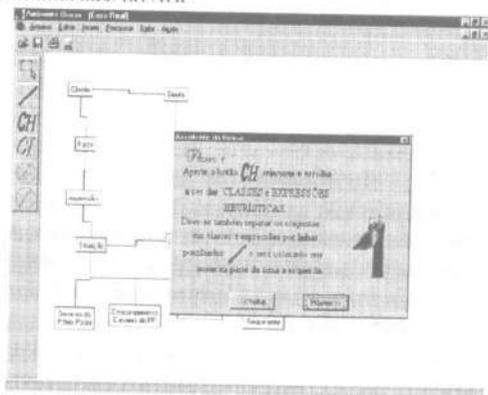


Figura 9: Janela do KP-Domínio

O *Modelo de Implementação do Sistema* possui o Diagrama Estrutural da Base de Conhecimento, a Especificação da Memória de Trabalho e a Especificação dos Módulos. O módulo *K-EstBC* deverá gerar o Diagrama Estrutural da Base de Conhecimento a partir do Diagrama Heurístico do Raciocínio gerado pelo *K-DHR*. *K-Est* deverá possuir também um *Editor de Especificações* para poder editar a Especificação da Memória de Trabalho e a Especificação dos Módulos.

Atualmente, o *Modelo de Especialidade* e o *Modelo Lógico* estão totalmente implementados e integrados a partir de uma base de dados global e o *Modelo Físico* encontra-se em fase de especificação.

### 5.3 Ferramenta Búzios

A ferramenta Búzios tem como objetivo apoiar a documentação a nível do planejamento dos documentos e sua produção.

O coordenador do projeto, ao desenvolver um SBC, definirá um *Plano de Documentação* para o projeto e detalhará os documentos necessários. A partir das atividades do processo são identificados os documentos necessários, os *patterns* de documentação adequados e os recursos para produção e distribuição dos documentos. Na definição detalhada dos documentos são definidos os itens adequados a cada um dos documentos e as guias de preenchimento de cada item. O planejador da documentação executa esta tarefa adaptando

os *formatos* genéricos definidos no ambiente e representados nos *patterns* de documentos.

A partir da seleção de um documento, todos os tópicos previamente descritos estão disponíveis na área de exibição, para que possam ser convenientemente preenchidos. No preenchimento do documento, o autor pode acessar as informações sobre aquele documento contidas no *pattern* de documento correspondente.

Um documento é um elemento importante em um ambiente pois define se uma atividade foi concluída e, também, envolve as pessoas da equipe na sua produção. Assim, o *Editor de Documentos* pode informar ao ambiente, através dos meta-dados associados ao documento, a situação do documento, as pessoas envolvidas na sua autoria e, ainda, uma lista de distribuição para que o mesmo seja enviado automaticamente a seu público alvo.

Esta ferramenta encontra-se em fase de construção sendo totalmente especificada.

## 6. Trabalhos relacionados

O apoio ao desenvolvimento de SBC tem sido reconhecido por diferentes comunidades científicas na área de Inteligência Artificial. Algumas propostas importantes nesse sentido são: KACTUS/CommonKADS [21], METHONDOLOGY [22], PROTEGE II [23] e GUIDA [24]. Essas propostas dão ênfase à aquisição do conhecimento e propõem um processo de desenvolvimento sendo que as três primeiras utilizam uma construção baseada em ontologias. A proposta de [24] não propõe nenhum apoio de ferramentas e as outras têm propostas de ferramentas isoladas que apoiam parte do processo de desenvolvimento de SBC.

O ORIXÁS no entanto tem um enfoque baseado na construção de ferramentas integradas em um processo de desenvolvimento, com ênfase tanto na construção como gerência e qualidade.

Em relação a ADS existem outros esforços que têm buscado atender às características de engenharia de software específicas como a configuração de interface aberta para permitir a integração de ferramentas externas [25], o apoio ao desenvolvimento orientado a objetos [26], [27], o apoio à reutilização de software [28], [29] e o apoio à colaboração e à cooperação [30]. No entanto, eles buscam soluções genéricas e ORIXÁS se baseia em soluções específicas para SBC de forma a prover melhor produtividade para os engenheiros de conhecimento que lidam com esta tecnologia não tradicional.

## 7. Conclusão

O desenvolvimento de SBC não é trivial e necessita de uma atenção especial na sua engenharia de software para apoiar a definição de todos os requisitos de um sistema que trabalha com conhecimento. Assim, procuramos destacar neste trabalho a importância de se ter um ambiente para o desenvolvimento de SBC com suas ferramentas integradas a um processo que assegura a realização das atividades, apoiado a uma estratégia de aquisição de conhecimento e num método de desenvolvimento próprio.

ORIXÁS propõe uma infra-estrutura que privilegia aspecto da aquisição e modelagem do conhecimento e da integração de ferramentas, num ambiente de desenvolvimento. Futuramente, ORIXÁS deve ser utilizado no desenvolvimento de SBC a serem desenvolvidos pelo Departamento de Informática e Ciência da Computação da UERJ em conjunto com o Hospital Universitário Pedro Ernesto (UERJ/RJ).

## Agradecimentos

Os autores agradecem a Káthia Oliveira seus comentários e o apoio da UERJ e FAPERJ.

## Referências Bibliográficas

- [1] TRAVASSOS, G.H. "O Modelo de Integração de Ferramentas da Estação TABA", Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1994.
- [2] BROWN, A., EARL, A., MCDERMID, J., "Software Engineering Environments: Automated Support for Software Engineering", England, McGraw-Hill Book Company, 1992.
- [3] VELDE, W.V., SCHREIBER, B.G., "The Future of Knowledge Acquisition: a European Perspective", IEEE Expert, July, 1996.
- [4] ISO/IEC 9126/NBR 13596, "Tecnologia de Informação – Avaliação de Produto de Software – Características de Qualidade e Diretrizes para o seu Uso", Associação Brasileira de Normas Técnicas, 1996.
- [5] ISO 9000-3/NBR 13596, "Normas de Gestão da Qualidade e Garantia da Qualidade: NBR ISO 9000-3", ABNT – Associação Brasileira de Normas Técnica, 1993.
- [6] RABELO et al., "An Expert System for Diagnosis of Acute Myocardial Infarction", Artificial Intelligence in Medicine, No 10, 1995, (75-92).
- [7] WERNECK, V.M.B., "Um Ambiente de Desenvolvimento de Sistemas Baseados em Conhecimento", Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1995.
- [8] WERNECK, V.M.B. et al., "A Software Development Process for Expert Systems", Proceedings of 10th International Symposium on Methodologies for Intelligent Systems North Caroline, USA, 1997, (209-220).
- [9] WERNECK, V.M.B., "Ambiente ORIXÁS: Processo de Desenvolvimento para Sistemas Baseados em Conhecimento", Cadernos do IME, UERJ, 2000.
- [10] OLIVEIRA, K. M., "Avaliação da Qualidade de Sistemas Especialistas", Tese de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1995.
- [11] MOREIRA, M.G., "Uma Ferramenta para Aquisição de Conhecimento", Tese de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1998.
- [12] MOREIRA, M.G. e WERNECK, V.M.B.; "OXOSSI: A Knowledge Acquisition Tool for Knowledge Based System"; Proceedings of Pacific Rim Knowledge Acquisition Workshop (PKAW), PRICA98, Singapore, November 1998, (64-73).
- [13] COELHO, A. C. B., "Guias de Qualidade para a Construção de Frameworks Orientados a Objetos", Tese de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1997.
- [14] SCHREIBER, G., Wielinga, B. e Breuker, J.(ed.): "KADS: A Principled Approach to Knowledge-Based System Development"; Academic Press; 1993.
- [15] GLASZMANN, J.H, SANTOS, N.; ROCHA, A.R.C.; WERNECK, V.M., "Modelagem de um Sistemas Baseados em Conhecimento: o Ambiente Educacional Expert Piano", Workshop em Engenharia de Software e Sistemas Baseados em Conhecimento, Vitória, ES, Setembro 1996, (77-81).
- [16] ORICHIO Ivana. C. e ARAÚJO, Patricia F. F., "SEM-Stress: Um Sistema Especialistas para Diagnóstico e Tratamento Alternativos de Stress", Projeto Final de Curso, Departamento de Informática e Ciência da Computação, UERJ, 1998.

- [17] CLUNIE, G., "Escola: Ambiente de Aprendizagem Baseado em Hipertecnologias", Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2000.
- [18] ARAÚJO, M.A.P., "Automatização do Processo de Desenvolvimento de Software nos Ambientes Instanciados pe Estação TABA", Tese de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1998.
- [19] RUMBAUGH, J., JACOBSON, I. e BOOCH, G., "The Unified Modeling Language Reference Manual", Addison-Wesley, 1999.
- [20] NIELEBOCK, D. G., WERNECK, V.M.B e WERNER, C.M.L., "Uma Ferramenta para Planejamento e Produção de Documentos de Software", ISDM-98 International Standard on Document Management, Curitiba, 1998.
- [21] SCHREIBER, G., WIELINGA, B., JANSWEIJER, W., "The Kactus View on the 'O' Word", In: Workshop on Basic Ontological Issues in Knowledge Sharing/IJCAI95, Montreal, Canada, August 1995.
- [22] GÓMEZ-PÉREZ, A., FERNANDEZ, M., VICENTE, A. J., "Toward a Method to Conceptualize Domain (ologies)", Proceedings of Workshop on Ontological engineering/ECAI96, Budapest, August, 1996.
- [23] MUSEN, M. A., "Modern Architectures for Intelligent Systems: Reusable Ontologies and Problem Solving Methods". JAMIA, Journal of the American Medical Informatics Association, 1998, (46-52).
- [24] GUIDA, G. e TASSO, C. "Design and Developments of Knowledge-Based Systems: from Life cycle to Methodology", John Wiley & Sons, 1994.
- [25] BANDINELLI, S., FUGGETTA, A., LAVAZZA, L., LOI, M. e PICCO, G., "Modeling and Improving an Industrial Software Process", IEEE Transactions on Software Engineering, v. 21, n. 5, May 1995, (440-453).
- [26] BERNAS, P., "The Pals-X SDE". In: Proceedings of Software Engineering Environments, Noordwijkerhout, The Netherlands, April, 1995.
- [27] GROTH, B., HERRMANN, S., JAHNICHEN, S., KOCH, W., P., "Project Integration Reference Object Library (PIROL): An Object-Oriented Multiple-View SEE". Proceedings of Software Engineering Environments, 126-134, Noordwijkerhout, The Netherlands, April, 1995.
- [28] RANDALL, R., ETT, W., "Using Software Process to Integrate Software Engineering Environments". In: Proceedings of Software Technology Conference, Salt Lake City, USA, April, 1995.
- [29] CONRADI, R. e KARLSSON, E-A., "The REBOOT Approach to Software Reuse", Journals of Systems and Software (special issue on software reuse), V. 30, N. 3, 1995, (201-212).
- [30] LONCHAMP, J., "CPCE: A kernel for Building Flexible Collaborative Process-Centered Environments". Proceedings of Software Engineering Environments, The Netherlands, April, 1995.

