



Universidade Federal de Pernambuco  
Centro de Informática  
Pós-Graduação em Ciência da Computação

## **Protótipo de Validação da *OntoPRIME***

### **Documento de Arquitetura**

Por  
Antônio Campelo  
Cristine Gusmão  
Leonardo Amorim  
Marcelo Guedes  
Monique Monteiro

Profa. Dra. Patrícia Tedesco  
Orientadora

Prof. Dr. Hermano Perrelli de Moura  
Co-orientador

**Recife, setembro de 2004**

## INFORMAÇÃO DO DOCUMENTO DE ARQUITETURA

<b>Título do documento</b>	Documento de Arquitetura para o protótipo de validação de prova de conceito da OntoPRIME: Ontologia de Riscos para Ambientes de Desenvolvimento de Software Multiprojetos		
<b>Autor(a)</b>	Monique Monteiro		
<b>Comentários</b>			
<b>Nome do arquivo</b>	OntoPRIME_DocumentoArquitetura.doc		
<b>HISTÓRICO DE REVISÕES</b>			
<b>Data</b>	<b>Revisão</b>	<b>Descrição</b>	<b>Autor</b>
07/09/04	1.0	Documento inicial.	Monique Monteiro

## SUMÁRIO

LISTA DE FIGURAS.....	4
1. OBJETIVOS .....	5
2. DESCRIÇÃO GERAL DA ARQUITETURA .....	5
3. ORGANIZAÇÃO DO SISTEMA .....	6
3.1. Camada de Apresentação .....	7
3.2. Camada de Negócio .....	7
3.3. Camada de Inteligência .....	7
4. PACOTES DO SISTEMA.....	8
4.1. Pacotes a nível de camadas .....	8
4.2. Pacotes a nível de implementação.....	9

## **LISTA DE FIGURAS**

Figura 1: Arquitetura em Camadas.....	5
Figura 2: Organização do Sistema.....	6
Figura 3: Pacotes a nível de camadas .....	9
Figura 4: Pacotes a nível de implementação .....	10

## 1. OBJETIVOS

Este documento tem por objetivo apresentar uma descrição geral da arquitetura do protótipo desenvolvido para validar o mecanismo de identificação de riscos para um projeto da versão **alfa** da OntoPRIME - Ontologia de Riscos para Ambientes de Desenvolvimento de Software Multiprojetos. Foi dada especial atenção à descrição desta arquitetura uma vez que ela deverá ser utilizada **não apenas** para este protótipo, mas também servirá como um modelo para aplicações futuras que façam uso da OntoPRIME. A seguir, será dada a descrição geral do padrão arquitetural utilizado e da distribuição dos componentes dentro da arquitetura descrita.

## 2. DESCRIÇÃO GERAL DA ARQUITETURA

O padrão arquitetural utilizado no desenvolvimento do protótipo é uma versão do **Padrão Arquitetural em Camadas**, caracterizado pela divisão dos diferentes níveis do sistema em diferentes camadas. Em particular, a versão utilizada foi adaptada para atender às necessidades de um sistema inteligente típico. Uma arquitetura em camadas como esta em geral possui as camadas mostradas na **Figura 1**:



**Figura 1 - Arquitetura em Camadas**

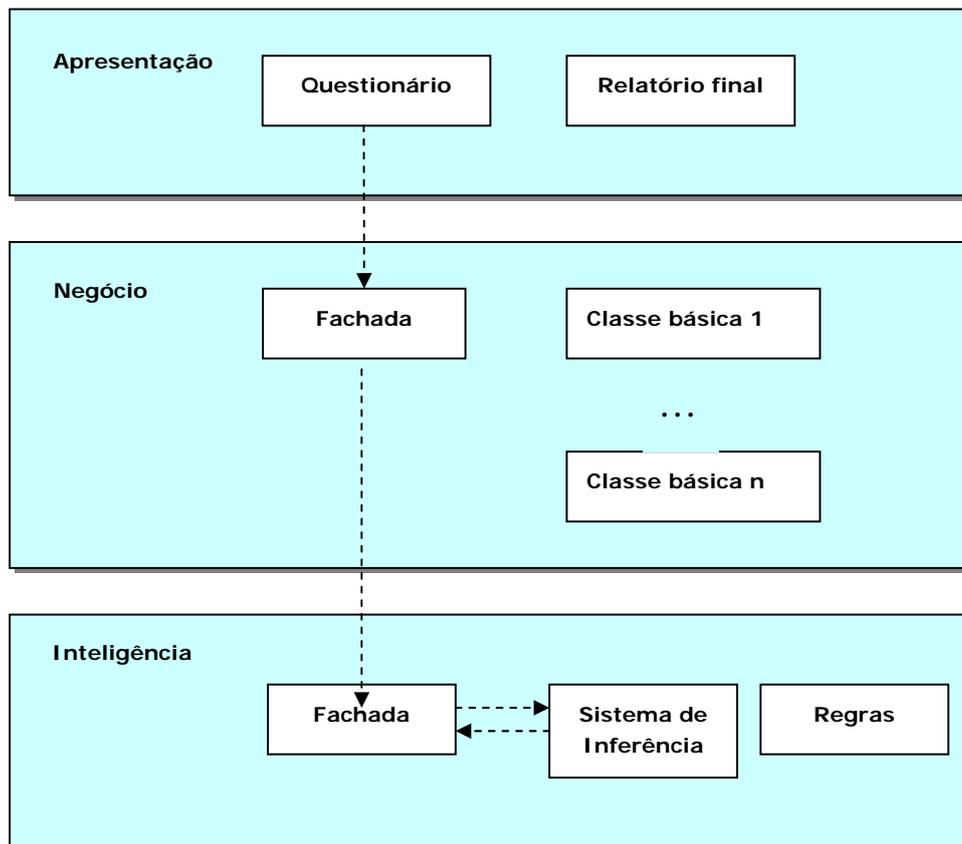
- **Apresentação:** Esta camada é responsável pela interação do usuário com o sistema, por meio da qual são realizadas as operações de entrada e saída. Pode ser implementada graficamente, por meio de uma *GUI – Graphical User Interface* ou Interface Gráfica do Usuário.
- **Negócio:** Contém as entidades que modelam e implementam os conceitos utilizados

no domínio da aplicação.

- **Camada de Inteligência:** Responsável por implementar a inteligência do sistema, fornecendo uma abstração sobre qual sistema dedutivo é utilizado e sobre como ele é implementado. A aplicação em questão foi projetada de modo a permitir completa independência do sistema dedutivo utilizado.

### 3. ORGANIZAÇÃO DO SISTEMA

A **Figura 2** mostra a distribuição dos elementos do sistema entre as diferentes camadas da arquitetura citadas na seção anterior:



**Figura 2 - Organização do Sistema**

### 3.1. Camada de Apresentação

Esta camada possui os elementos de interface com o usuário, neste caso, um *wizard* contendo perguntas sobre as características do projeto a ser analisado, baseado no questionário proposto pelo Paradigma de Gerenciamento de Riscos do SEI. A interface foi construída utilizando componentes do *toolkit* SWING [ZUKOWSKI, 1999] da plataforma JAVA [GOSLING, 2000], linguagem utilizada na implementação de todo o sistema. Esta interface comunica-se com o sistema através da fachada da camada de negócio.

### 3.2. Camada de Negócio

Camada responsável por implementar a lógica de negócio da aplicação. Contém os seguintes elementos:

- **Fachada:** Classe que fornece uma interface para os serviços oferecidos pela aplicação. Comunica-se diretamente com a fachada da camada de inteligência.
- **Classes básicas:** Representam os conceitos utilizados no sistema (ie.: risco, projeto, características do projeto), bem como as classes ou conceitos presentes na ontologia desenvolvida.

### 3.3. Camada de Inteligência

Camada responsável pela implementação da inteligência do sistema, projetada de forma que seja possível abstrair nas camadas superiores qual mecanismo dedutivo é utilizado (ex.: lógica de primeira ordem, lógica *fuzzy* [CHEN, 1995], etc.). Contém os seguintes elementos:

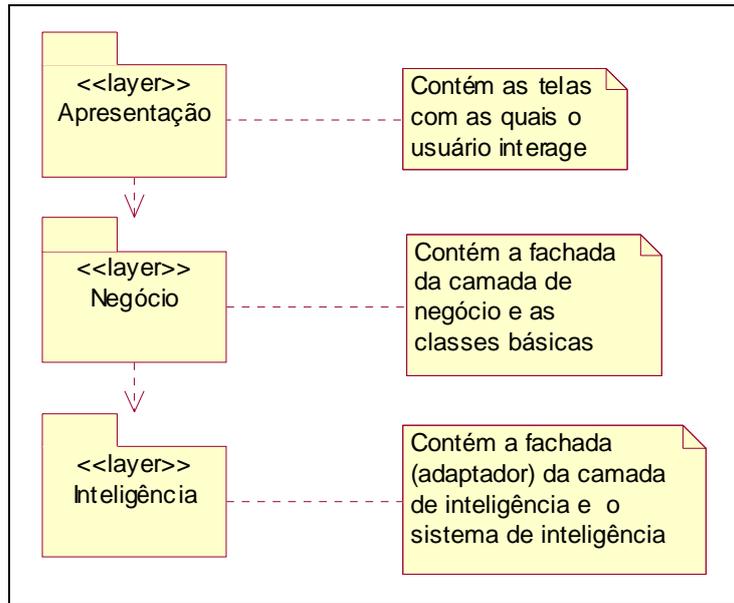
- **Fachada:** Interface para os serviços oferecidos pela camada de inteligência. Sua implementação instancia o sistema de inferência e atua como um **adaptador** para o sistema de inteligência utilizado, fazendo a comunicação entre este e a camada imediatamente superior.
- **Sistema de Inferência:** Módulo que encapsula o mecanismo utilizado para implementar a inteligência do sistema. Em particular, o sistema de inferência utilizado nesta aplicação é o *JEOPS* [FIGUEIRA, 2001], que utiliza o formalismo de **lógica de primeira ordem** para representar as regras e axiomas da ontologia, uma vez que este foi o formalismo utilizado para representar tais regras e axiomas na construção da ontologia.

## 4. PACOTES DO SISTEMA

Os pacotes são agrupamentos lógicos das classes do sistema. A divisão do sistema em pacotes pode ser representada por duas formas: a nível de camada ou a nível de implementação. Do ponto de vista conceitual ou de análise, costuma-se utilizar a primeira representação, de modo que a segunda é utilizada em um contexto mais concreto, durante a fase de projeto. Nas próximas seções, serão mostradas as duas representações da arquitetura utilizando **notação UML** [RUMBAUGH, 2004].

### 4.1. Pacotes a nível de camadas

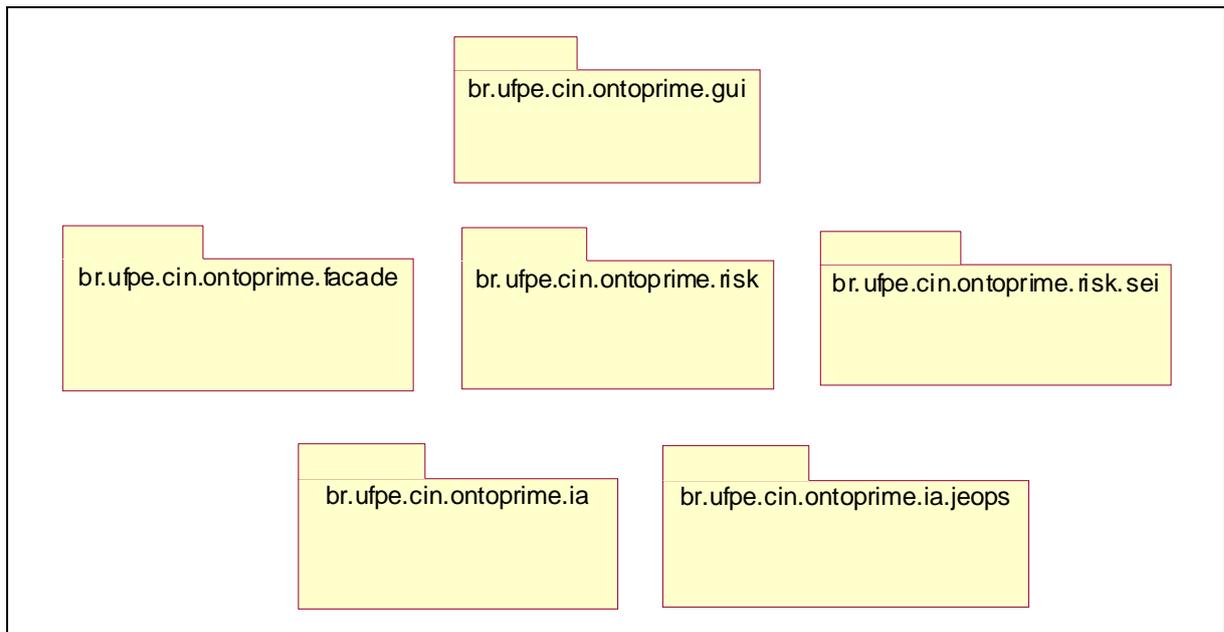
Na **Figura 3** é exibida uma divisão em que cada pacote mostrado corresponde basicamente a uma camada da arquitetura. Para cada pacote desta divisão é descrito o conjunto de classes e componentes que dele fazem parte. Esta é uma visão abstrata, com foco na arquitetura e mais próxima do modelo de análise. A **Figura 3** pode ser considerada como uma versão simplificada da **Figura 2**, porém utilizando notação de pacotes.



**Figura 3 - Pacote a nível de camadas**

## 4.2. Pacotes a nível de implementação

A **Figura 4** exibe a real divisão do sistema em pacotes, de modo que cada pacote equivale a um diretório no sistema de arquivos ou a um *pacote* de acordo com o conceito correspondente na linguagem JAVA. Esta divisão é mais concreta do que a divisão a nível de camadas e aproxima-se mais do modelo de projeto.



**Figura 4 - Pacotes a nível de implementação**

A seguir segue uma descrição de cada um dos pacotes exibidos :

- *br.ufpe.cin.ontoprime.gui*: contém as classes referentes à interface gráfica.
- *br.ufpe.cin.ontoprime.facade*: contém a classe de fachada para a camada de negócio.
- *br.ufpe.cin.ontoprime.risk*: contém as classes que representam os conceitos mais comuns do domínio da aplicação, como **Risco** e **Projeto**, por exemplo. Essas classes são portanto independentes da taxonomia/ontologia utilizada.
- *br.ufpe.cin.ontoprime.risk.sei*: contém as classes da taxonomia proposta pelo Paradigma de Gerenciamento de Risco do SEI - *Software Engineering Institute*. Na implementação atual da OntoPRIME, essas classes são as mesmas utilizadas pela ontologia desenvolvida.
- *br.ufpe.cin.ontoprime.risk.ia*: contém as classes que implementam a inteligência da aplicação.
- *br.ufpe.cin.ontoprime.risk.ia.jeops*: contém as classes que interagem diretamente com a ferramenta JEOPS, como a classe que implementa a fachada da camada de inteligência e

as classes que implementam as regras.

## REFERÊNCIAS BIBLIOGRÁFICAS

[ZUKOWSKI, 1999] ZUKOWSKI, J. e STANCHFIELD, S.. *Fundamentals of JFC/Swing*. <http://java.sun.com/developer/onlineTraining/GUI/Swing1/>. Visitado em 09/09/04.

[GOSLING, 2000] GOSLING, J., JOY, B., STEELE, G. e BRACHA, G. *The Java Language Specification*. Sun Microsystems, 2000.

[CHEN, 1995] CHEN, C. H. *Fuzzy Logic and Neural Network Handbook*. McGraw-Hill Publishing Company, 1995.

[FIGUEIRA, 2001] FIGUEIRA, C. S. *JEOPS – The Java Embedded Object Production System*. <http://www.cin.ufpe.br/~jeops/>. Visitado em 09/09/04.

[RUMBAUGH, 2004] RUMBAUGH, J., JACOBSON, IVAR, BOOCH, G.. *The Unified Modeling Language Reference Manual*. 2a. Edição. Addison-Wesley Object Technology Series, 2004.