

Odyssey-LE: Uma Infra-Estrutura de Reutilização para o Domínio de Processamento Legislativo

*Cláudia Werner¹, Regina Braga¹, Mônica Zopelari^{1,2}
Márcio Barros¹, Leonardo Murta¹*

{werner, regina, zopelari, marcio, murta}@cos.ufrj.br
<http://www.cos.ufrj.br/~odyssey>

¹COPPE/UFRJ - Programa de Engenharia de Sistemas
Universidade Federal do Rio de Janeiro
Caixa Postal 68511 – CEP. 21945-970
Rio de Janeiro – Brasil

²Câmara Municipal do Rio de Janeiro

Resumo

Este trabalho apresenta uma infra-estrutura de apoio ao desenvolvimento de software baseado em reutilização, denominada Odyssey-LE, que provê modelos e componentes reutilizáveis para o domínio de processamento legislativo. Esta infra-estrutura está sendo construída baseada nos resultados obtidos no contexto do Projeto Odyssey, em desenvolvimento na COPPE/UFRJ, que oferece suporte ao Desenvolvimento Baseado em Componentes (DBC) em domínios específicos. Frameworks orientados a objetos, arquiteturas de software, técnicas de inteligência artificial, engenharia de domínio, hipermídia, e mediadores são algumas das tecnologias adotadas pelo Odyssey.

Tópicos de Interesse: Projeto de Pesquisa em andamento; Construção colaborativa do conhecimento usando redes e WWW

1. Introdução

O Desenvolvimento Baseado em Componentes (DBC) está cada vez mais ganhando adeptos e é uma das propostas promissoras para a melhoria da produção de software nos próximos anos. O DBC tem como objetivo a definição de componentes interoperáveis, com interfaces bem definidas, evidenciando os tipos de relacionamentos permitidos por estes componentes. Desta forma, a complexidade no desenvolvimento de aplicações é reduzida, assim como os custos, através da reutilização de componentes exaustivamente testados.

Para que a reutilização de componentes no desenvolvimento de novas aplicações em um dado domínio seja eficaz, é necessário prover uma infra-estrutura de apoio à reutilização, que privilegie o DBC em todos os seus aspectos.

Este artigo apresenta um protótipo da infra-estrutura **Odyssey-LE**, que está sendo construída baseada nos resultados obtidos no contexto do Projeto Odyssey [BRA99a] [BRA00a] [WER99] [WER00] em desenvolvimento na **COPPE/UFRJ**, desde 1998. Este projeto tem como objetivo construir uma infra-estrutura de reutilização baseada em modelos de domínio. Esta infra-estrutura provê suporte ao DBC e algumas das tecnologias exploradas são: *frameworks* orientados a objetos, arquiteturas de software, engenharia de domínio, técnicas de inteligência artificial, hipermídia e mediadores (técnica utilizada para o armazenamento e recuperação de dados distribuídos e heterogêneos). A linguagem de programação utilizada é **Java**.

O objetivo deste trabalho é disponibilizar uma infra-estrutura de apoio ao desenvolvimento de aplicações no domínio de processamento legislativo, baseado na reutilização de componentes de software previamente construídos. Tal iniciativa partiu de uma parceria entre o grupo de reutilização de software da **COPPE/UFRJ** e a equipe de informática da **Câmara Municipal do Rio de Janeiro** e visa oferecer a possibilidade de melhoria na produção do software legislativo, com custo reduzido, nas diversas Casas

Legislativas, na medida em que as soluções encontradas podem ser disponibilizadas a nível nacional. Como o processamento legislativo é comum a todas as Casas Legislativas, resguardadas as particularidades de cada uma, este foi o domínio escolhido para realização deste trabalho.

O artigo está organizado da seguinte forma: inicialmente, descrevemos a arquitetura geral da infra-estrutura e os processos de engenharia de domínio e aplicação apoiados pela mesma. Nas seções seguintes, apresentamos o editor de diagramas, um *framework* para documentação de componentes, o navegador inteligente e os aspectos referentes à persistência de componentes.

2. Arquitetura do Odyssey-LE

O uso de técnicas de reutilização nas fases iniciais do desenvolvimento de aplicações baseado em componentes facilita a reutilização de componentes em fases mais avançadas do desenvolvimento (ex. implementação). Uma infra-estrutura de reutilização baseada em modelos de domínio é capaz de apoiar uma reutilização efetiva durante todo o desenvolvimento de software, provendo métodos, ferramentas e procedimentos adequados para a especificação de modelos e aplicações do domínio.

A infra-estrutura Odyssey-LE pode ser vista como um arcabouço onde modelos conceituais do domínio, modelos de projeto (arquiteturas de software específicas do domínio) e modelos implementacionais (*frameworks*) são especificados para o domínio de processamento legislativo, disponibilizando componentes reutilizáveis neste domínio. Durante o processo de desenvolvimento de aplicações estes componentes são refinados e adaptados ao contexto de um projeto específico.

A Figura 1 apresenta uma visão diagramática dos principais elementos do Odyssey-LE. Os modelos de domínio constantes da infra-estrutura são especificados e posteriormente evoluídos segundo as atividades de um processo para construção de componentes reutilizáveis, denominado Odyssey-ED [BRA99b]. Estes componentes são reutilizados durante o desenvolvimento de aplicações através de um processo denominado Odyssey-EA [MIL00] – detalhes sobre estes processos são apresentados na próxima seção. As atividades destes processos são suportadas por um conjunto de ferramentas, dentre elas, ferramentas para definição dos modelos, ferramentas para armazenamento e recuperação dos modelos, e ferramentas para navegação e reutilização dos modelos.

Os principais usuários da infra-estrutura são o *engenheiro do domínio*, o *especialista do domínio* e o *engenheiro de software* responsável pelo desenvolvimento de aplicações no domínio. O engenheiro do domínio e o especialista utilizam a infra-estrutura, principalmente, para especificar e evoluir os conceitos do domínio. O engenheiro de software utiliza a infra-estrutura para obter conhecimento sobre o domínio da aplicação e reutilizar este conhecimento na especificação de sua aplicação.

3. A Engenharia de Domínio e o Desenvolvimento de Aplicações

Um processo de Engenharia de Domínio (ED) é composto basicamente das seguintes etapas:

1. Análise do domínio, que determina os requisitos comuns em um domínio com o objetivo de identificar as oportunidades de reutilização.
2. Projeto do domínio: Utiliza os resultados da análise do domínio para identificar e generalizar soluções para os requisitos comuns através da especificação de uma arquitetura de software do domínio. As oportunidades de reutilização identificadas na análise do domínio são refinadas de forma a especificar as restrições do projeto.
3. Implementação do domínio: Transforma as oportunidades de reutilização e soluções do projeto para um modelo de implementação.

Os produtos gerados nestas etapas são os modelos conceituais do domínio, modelos de projeto e modelos de implementação, que são os componentes reutilizados no processo de desenvolvimento de aplicações no domínio. Todos os modelos gerados têm como

premissa básica facilitar o entendimento por parte do usuário dos principais conceitos e funcionalidades do domínio. Além disso, todos os modelos especificados possuem ligações semânticas uns com os outros, as quais denominamos de rastro (*trace*). Portanto, se o desenvolvedor estiver examinando um determinado modelo, é possível “rastrear” outros modelos que descrevem o mesmo assunto. Assim, o entendimento do domínio como um todo e de seus componentes em particular fica facilitado, além de auxiliar na identificação de que modelos serão reutilizados nas diversas fases do desenvolvimento de uma determinada aplicação.

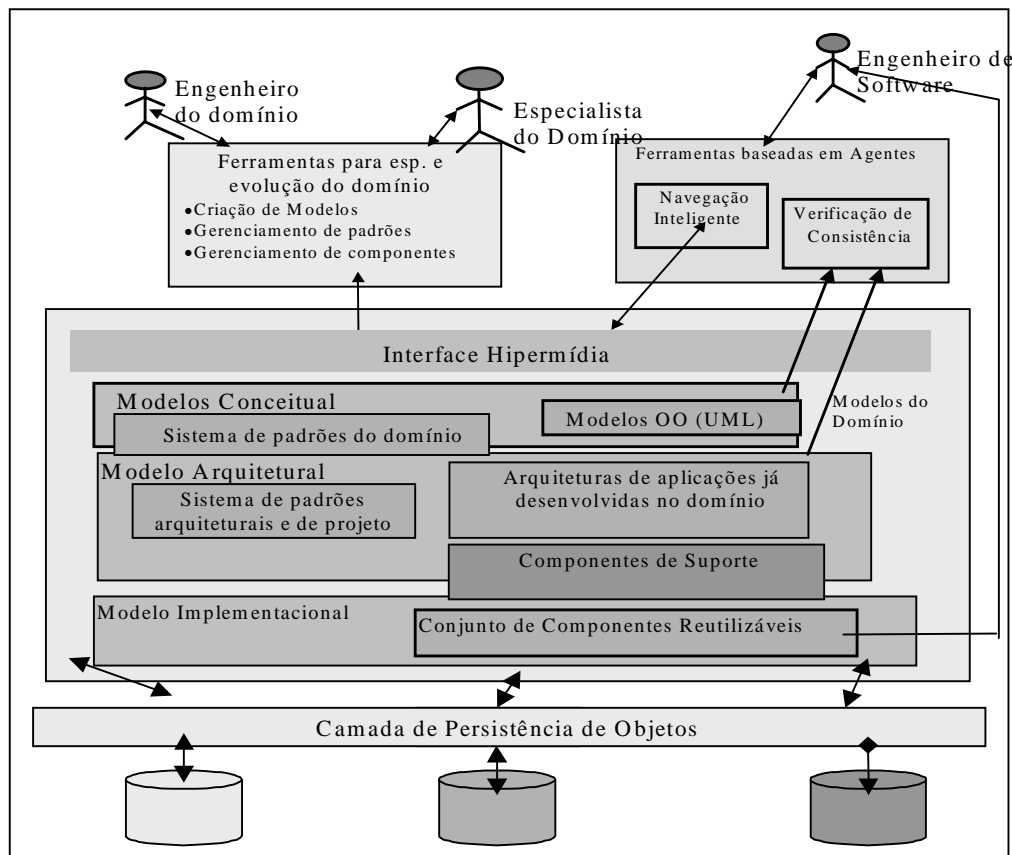


Figura 1 - Visão diagramática dos principais elementos da infra-estrutura Odyssey-LE

Um processo de Engenharia de Aplicações (EA), no contexto do desenvolvimento baseado em reutilização, se dá através de fases, tal como num processo de desenvolvimento convencional (i.e., planejamento, análise, projeto e implementação). Porém, suas atividades são orientadas à reutilização, ou seja, o foco é no aproveitamento e adaptação dos componentes do domínio, sempre que possível de forma automática. O conceito de “rastreamento” entre os modelos que compõem o domínio é muito importante na atividade de escolha de componentes reutilizáveis durante o desenvolvimento de uma aplicação.

Na fase de análise, a escolha de um determinado contexto de domínio, automaticamente, determina um corte no número de componentes disponíveis para seleção, deixando só os ligados ao escopo definido. Assim, o desenvolvedor pode escolher as funcionalidades do domínio que julga necessárias a sua aplicação (Figura 2) [MIL00]. As atividades posteriores do desenvolvimento também são influenciadas por estas escolhas, na medida em que elas relacionam, através do “rastreamento”, os *use-cases* e as classes associadas, e delimitam o escopo de informações do domínio que serão utilizadas na aplicação. Na fase de implementação, é possível gerar código, automaticamente, a partir

dos atributos e serviços das classes definidos no modelo da aplicação, na linguagem de programação escolhida (atualmente, Java, C++ ou Delphi).

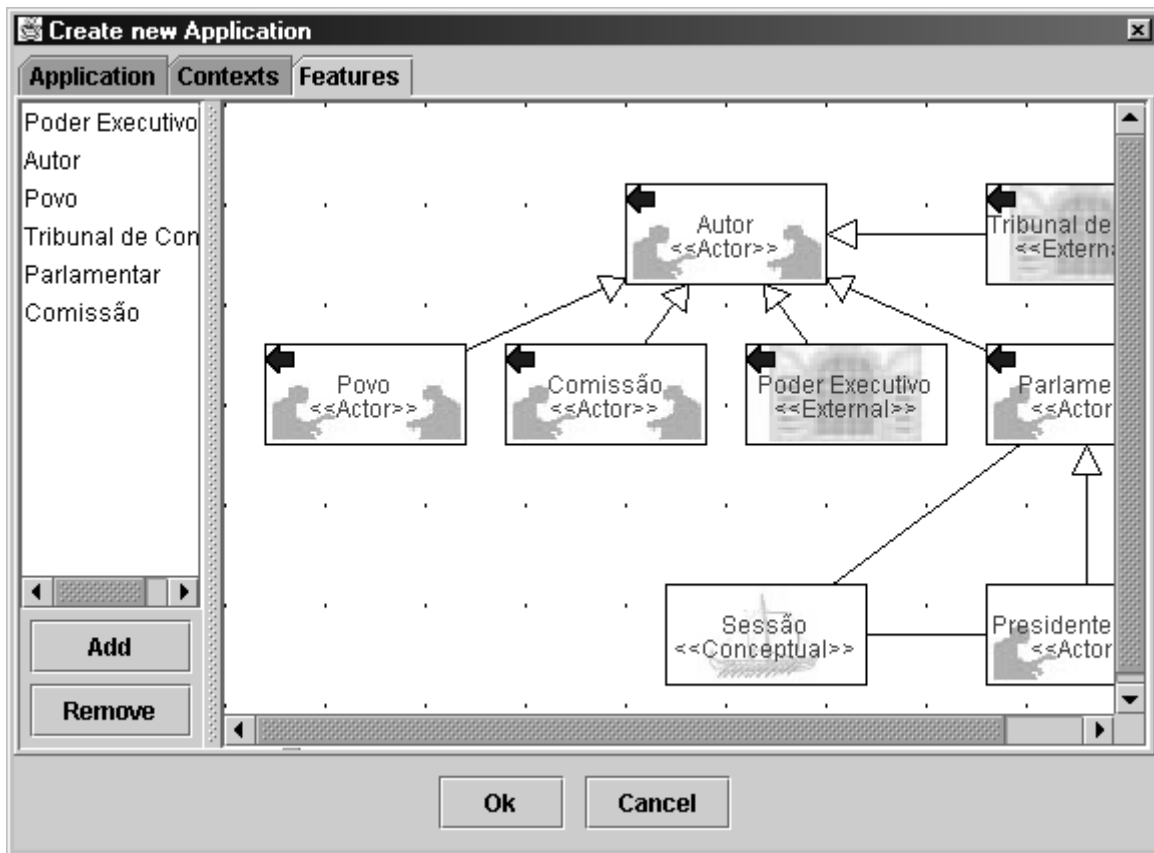


Figura 2 – Exemplo de escolha de componentes

4. Editor de Diagramas

O editor de diagramas tem como principal objetivo o suporte à criação dos modelos definidos pelo Odyssey-ED [BRA99a] [BRA99b]. A notação utilizada é a UML [FOW97], com algumas construções adicionais para suportar o desenvolvimento de componentes reutilizáveis. Os principais modelos gerados são: modelo de características (*features*) - visões conceitual e funcional -, modelos de "use case" do domínio, modelos de classes, modelos de colaboração e demais modelos constantes da notação UML. A infra-estrutura construída provê suporte automatizado para o conceito de "rastreadabilidade" entre os diversos modelos, previamente definido.

A figura 3 mostra uma sessão típica de utilização do editor de diagramas construído.

5. Documentação de Componentes

Para dar apoio à criação e disponibilização de uma documentação padrão de componentes, o Odyssey-LE apresenta um *framework* de documentação, denominado *FrameDoc*. Este *framework* permite a criação de *templates* de documentação (figura 4a), que são posteriormente preenchidos (figura 4b) para cada tipo de componente (ex. classe, caso de uso, diagrama de estados, etc.). Os tipos de campos fornecidos para a construção do *template* são: texto, memo, *checkbox*, *radiobutton*, som, imagem, vídeo e HTML.

Algumas das características do *FrameDoc* são: a exportação da documentação adotando o padrão HTML, utilização de hiperligações no campo HTML para outros documentos ou páginas na Internet, criação de padrões de documentação (representados pelos *templates*), geração de documentação modular, utilização de multimídia na

documentação, geração da documentação implícita ao componente (ex. diagramas UML) e visualização da documentação a ser gerada.

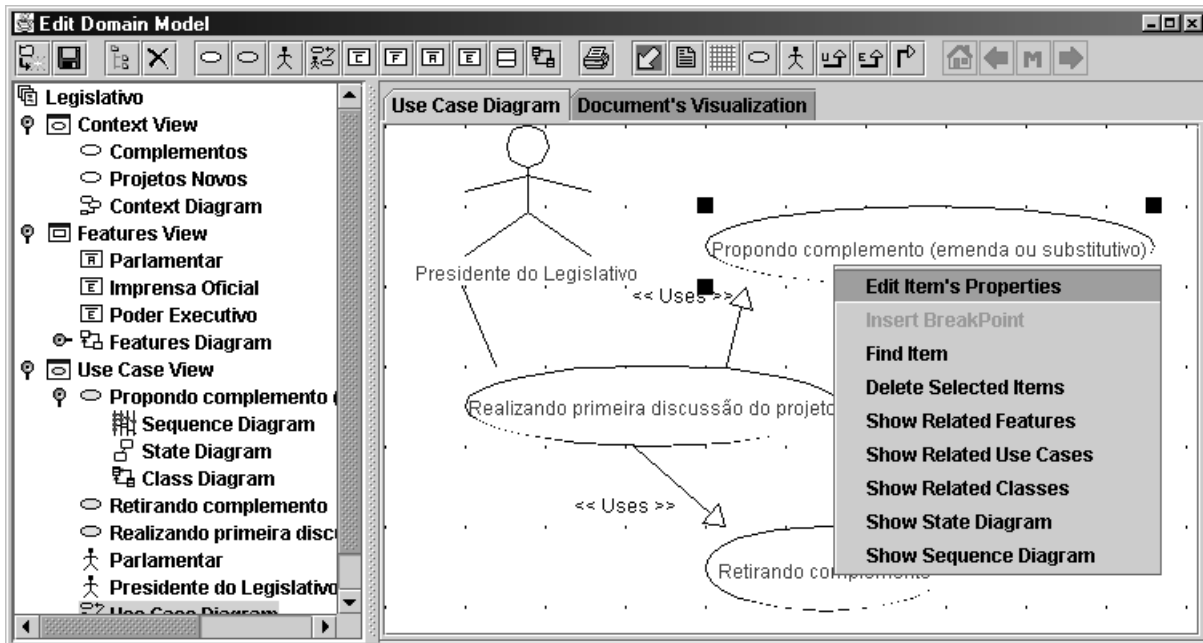


Figura 3 – Editor de Diagramas Odyssey-LE

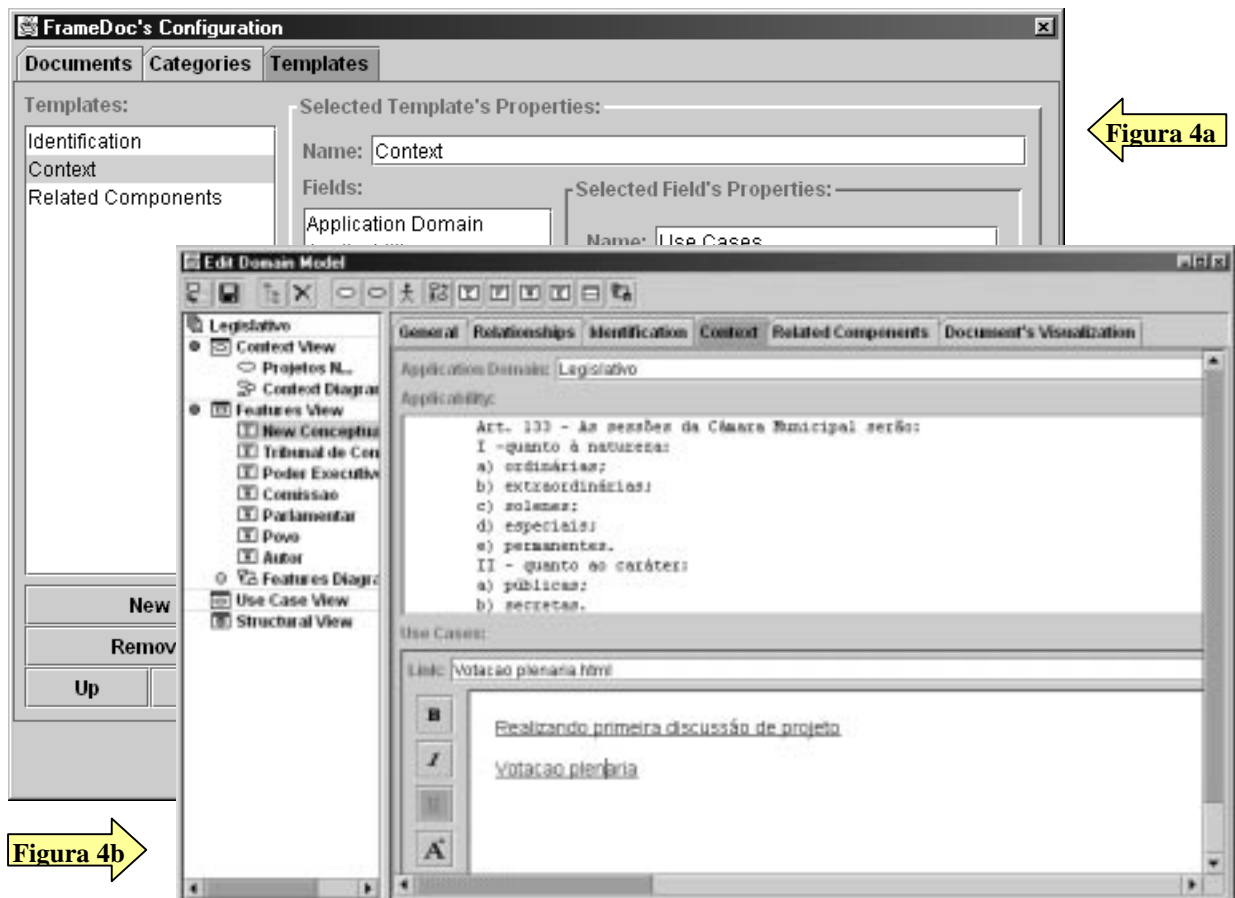


Figura 4 - Janelas de configuração (a) e edição (b) de documentos

6. Navegação Inteligente

No contexto da infra-estrutura Odyssey-LE, a busca por informações do domínio é um requisito importante. No entanto, esta tarefa nem sempre é trivial de ser realizada, dado o volume e diversidade das informações do domínio que podem ser coletadas. Além disso, a forma de busca e acesso a estas informações pode ser complexa, do ponto de vista do usuário. Com o objetivo de minimizar estes problemas, o Odyssey-LE provê agentes que têm como objetivo recuperar e filtrar informações do domínio, disponibilizando-as ao usuário de forma adequada. Os agentes baseiam-se em *ontologias do domínio* e num modelo evolutivo do usuário que mantém seus principais interesses. Ontologias de domínio são um conjunto de termos do domínio, incluindo uma descrição dos mesmos e exemplos de uso, e os relacionamentos entre estes termos representados de maneira formal.

O desenvolvedor, ao se cadastrar como usuário Odyssey-LE, responde a um questionário indicando seu nível de conhecimento sobre os termos do domínio, quais termos são considerados relevantes e quais são seus interesses por certas categorias do domínio. Este cadastramento relaciona o usuário a um *estereótipo* (i.e., classificação de usuários com perfis similares). Técnicas de aprendizagem de máquina são, então, usadas para a identificação de padrões comportamentais do usuário e evolução de seu perfil. Baseado neste perfil, é possível realizar:

- Apresentação adaptativa: permite que o conteúdo dos nós hipermídia seja adaptado às expectativas e requisitos do usuário, a partir de uma base de conhecimento (i.e., regras *if-then*) e de um algoritmo de inferência, que é alimentado por informações contidas no modelo do usuário;
- Navegação inteligente: caminhos de navegação mais adequados ao usuário são inferidos através de um algoritmo de predição, que utiliza técnicas de recuperação de informação (ex. contagem do número das ocorrências de palavras-chave em itens de domínio relacionados) e padrões de caminhos de navegação seguidos repetidas vezes pelo usuário, ou por usuários do mesmo estereótipo.

Detalhes sobre a navegação inteligente são encontrados em [BRA00b].

7. Persistência de Componentes

A infra-estrutura Odyssey-LE disponibiliza três formas de persistência das suas informações: Serialização, GOA++ e MOR. A figura 5 exibe o relacionamento entre a infra-estrutura e suas formas de armazenamento.

A Serialização é um mecanismo da linguagem Java que permite o armazenamento de objetos e todos os demais objetos relacionados (recursivamente) em um arquivo do sistema operacional. Todas as informações dos modelos de domínio e aplicação são armazenadas em um único arquivo. Essa abordagem, apesar de mais simples, não é recomendada para o armazenamento de uma grande quantidade de informação.

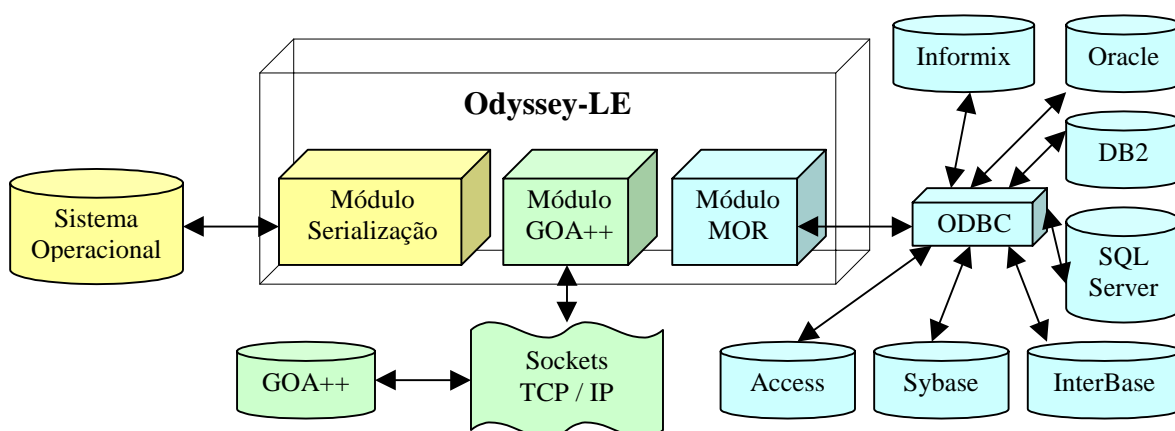


Figura 5 – Formas de armazenamento no Odyssey-LE

Uma outra opção para o armazenamento no Odyssey-LE é o GOA++ (Gerente de Objetos Armazenados) [MAT00]. O GOA++ é um gerenciador de objetos em desenvolvimento pela linha de Banco de Dados da COPPE/UFRJ. A utilização do GOA++ representa uma boa opção, na medida em que mantém o paradigma utilizado pelo Odyssey-LE (Orientação a Objetos).

Por questões de flexibilidade, foi ainda disponibilizado o MOR (Mapeador Objeto Relacional), que é uma interface genérica de armazenamento, de forma que os dados possam ser armazenados em qualquer SGBD com interface para ODBC (vide figura 5). Desta forma, o Odyssey-LE provê duas opções interessantes: uma que permite o armazenamento direto de objetos Odyssey-LE no GOA++; e outra que permite o armazenamento em bases relacionais, provendo um mapeamento entre o modelo de objetos e o modelo relacional, através do mecanismo de *reflection* de Java (um *framework* desenvolvido pela SUN, que permite a introspecção e instanciação de objetos em tempo de execução).

A utilização do MOR é semelhante a utilização de Serialização, todavia, o MOR utiliza a ponte JDBC-ODBC para acessar fontes de dados ODBC e armazenar as informações em bancos de dados, ao invés de utilizar arquivos do sistema operacional. Desta forma, o MOR pode utilizar qualquer banco de dados que tenha driver ODBC disponível, como por exemplo: Oracle, DB2, Informix, InterBase, SQL Server, Sybase, Access, etc.

Forma de Armazenamento	Nível de Complexidade de Utilização	Nível de Tecnologias Relacionadas	Nível de Confiabilidade	Propósito Principal
Serialização	Baixo	Baixo	Baixo	Testes e pequenos modelos. É indicado para o uso em ambientes que não têm acesso a um SGBD via ODBC.
GOA++	Médio	Alto	Médio	Pesquisa e desenvolvimento de tecnologias para armazenamento de objetos (protótipo acadêmico).
MOR	Médio	Médio	Alto	Utilização em produção com grandes modelos. É indicado para o uso em ambientes que têm acesso a um SGBD via ODBC

Tabela 1 – Comparação entre as formas de armazenamento no Odyssey-LE

É possível, ainda, transferir para o MOR modelos armazenados via Serialização ou GOA++. Isto pode ser necessário caso parte do modelo tenha sido desenvolvido em ambientes sem acesso a ODBC.

A tabela 1 mostra uma comparação entre as formas de armazenamento, utilizando como parâmetros os níveis de simplicidade de sua utilização, de tecnologias relacionadas e de confiabilidade. Os propósitos principais de cada tipo de armazenamento são também apresentados.

8. Perspectivas Futuras

Além das funcionalidades aqui apresentadas, diversos outros trabalhos estão sendo desenvolvidos no contexto do Projeto Odyssey, em geral, e com relação à infra-estrutura Odyssey-LE, em particular. Estes trabalhos estão relacionados com aspectos como gerência e acompanhamento de processos, suporte à criação e adaptação de arquiteturas específicas de domínio, verificadores de consistência e crítica de modelos, extensões ao atual apoio dado ao DBC, no que diz respeito às técnicas de gerência e acesso a componentes reutilizáveis heterogêneos e distribuídos, apoio à navegação dinâmica na Internet, e linguagem descritiva para conectores de componentes, dentre outros.

No que se refere à infra-estrutura Odyssey-LE, em particular, existe a atual perspectiva de implantá-la no contexto do Programa Interlegis do Prodasen, visando, efetivamente, a sua utilização com abrangência nacional, onde as equipes de analistas das Casas Legislativas elaborariam uma biblioteca de componentes reutilizáveis, através desta infra-estrutura, que serviria como centro de referência para os ambientes de desenvolvimento de sistemas no âmbito legislativo. Num primeiro momento, o esforço seria coordenado no sentido de se atingir um nível de qualidade e maturidade desejado para a biblioteca de componentes, sendo esta, então, distribuída para as demais Casas Legislativas.

Desta forma, se configura uma situação onde diversas equipes de analistas distribuídos geograficamente, de acordo com as respectivas Casas Legislativas de origem, poderiam trabalhar, cooperativamente, para elaborar e disseminar modelos dos componentes e padrões para a arquitetura tecnológica. Seria uma importante contribuição, para a melhoria do processo de trabalho destas equipes, a introdução da tecnologia de trabalho cooperativo apoiado por computador (*groupware*), mantendo o enfoque participativo e cooperativo da realização das atividades. Neste sentido, está prevista a incorporação de ferramentas de *groupware* à Infra-estrutura Odyssey-LE.

Agradecimentos

Ao CNPq pelo apoio financeiro e a todos aqueles que participam deste projeto de pesquisa, por seu entusiasmo e dedicação.

Referências Bibliográficas

- [BRA99a] Braga, R.; Werner, C.; Mattoso, M. "Odyssey: A Reuse Environment based on Domain Models", IEEE Symposium on Application-Specific Systems and Software Engineering Technology (ASSET'99), pp.50-57, Texas, Mar 1999.
- [BRA99b] Braga, R.; Werner, C. "Odyssey-DE: um Processo para Desenvolvimento de Componentes Reutilizáveis, X CITS, Curitiba, Mai 1999.
- [BRA00a] Braga, R.; Werner, C.; Mattoso, M. "A Reuse Infrastructure based on Domain Models", Special Issue of the Journal of Computing and Information, Winnipeg, Canadá, Vol.3, ICCI98, março 2000, ISSN 1201-8511, pp.227-234.
- [BRA00b] Braga, R.; Costa, M.; Werner, C.; Mattoso, M. "A Multi-Agent System for Domain Information Discovery and Filtering", XIV Simpósio Brasileiro de Engenharia de Software, João Pessoa, Out 2000 (aceito p/ publicação).
- [FOW97] Fowler, M. "UML Distilled: Applying The Standard Object Modeling Language", Addison-Wesley, 1997.
- [MAT00] Mattoso, M.; Werner, C.; Braga, R.; Pinheiro, R.; Murta, L.; Almeida, V.; Costa, M.; Bezerra, E. Soares, J.; Ruberg, N.; "Persistência de Componentes num Ambiente de Reuso", XIV Simpósio Brasileiro de Engenharia de Software, Caderno de Ferramentas, João Pessoa, Out 2000 (aceito p/ publicação)
- [MIL00] Miler Jr., N.; Werner, C.; Braga, R. "O uso de Modelos de Features na Engenharia de Aplicações", IDEAS'00, Cancun, México, Abr 2000, pp.85-96.
- [WER99] Werner, C.; Mattoso, M.; Braga, R.; M.Barros, L.Murta, A.Dantas, "Odyssey: Infra-estrutura de Reutilização baseada em Modelos de Domínio", XIII Simpósio Brasileiro de Engenharia de Software, Caderno de Ferramentas, Florianópolis, Out 1999, pp.17-20.
- [WER00] Werner, C.; Braga, R.; Mattoso, M.; Murta, L.; Costa, M.; Pinheiro, R.; Oliveira, A.; "Infra-estrutura Odyssey: estágio atual", XIV Simpósio Brasileiro de Engenharia de Software, Caderno de Ferramentas, João Pessoa, Out 2000 (aceito p/ publicação).