



**FUNDAÇÃO EDSON QUEIROZ  
UNIVERSIDADE DE FORTALEZA**

**Uma Metodologia de Apoio à Modelagem Unificada de  
Sistemas Interativos: Uma aplicação da Pesquisa  
Operacional**

**Dissertação de Mestrado**

**Por**

**Otoni Cardoso do Vale Júnior**

**Orientadora: Maria Elizabeth Sucupira Furtado**

**Co-orientador: Plácido Rogério Pinheiro**

**Fortaleza, CE - Brasil**

**2003**

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



**FUNDAÇÃO EDSON QUEIROZ  
UNIVERSIDADE DE FORTALEZA**

# **Uma Metodologia de Apoio à Modelagem Unificada de Sistemas Interativos: Uma aplicação da Pesquisa Operacional**

**Otoni Cardoso do Vale Júnior**

Dissertação submetida ao Corpo Docente do Mestrado em Informática Aplicada da Universidade de Fortaleza como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciência da Computação.

Orientadora: Profa. Orientadora Maria Elizabeth Sucupira Furtado, D.Sc. (UNIFOR)

Aprovada por:

---

Profa. Maria Elizabeth Sucupira Furtado, D.Sc.  
(Presidente da Banca)

---

Prof. Bernardo Lula Júnior, D.Sc.

---

Prof. Plácido Rogério Pinheiro, D.Sc.

Fortaleza, CE - Brasil

2003

**Dedico esta dissertação a Deus Pai,  
Filho e Espírito Santo, a minha família,  
namorada, e aos meus amigos.**

## **Agradecimentos**

A Deus que pelo intermédio de Jesus Cristo e do Espírito Santo me proporcionou as condições necessárias para a realização deste trabalho. Bendito e louvado seja o seu santo nome.

Aos meus pais pelo suporte moral e financeiro que permitiram que eu me dedicasse exclusivamente a este trabalho, além da dedicação e sacrifícios incomensuráveis para que eu e meus irmãos tivéssemos uma excelente orientação e qualidade de vida.

A professora Elizabeth pela minha adoção e por me conduzir com firmeza e suavidade por caminhos sempre novos.

Ao professor Plácido por conselhos importantes e por ter me apoiado em momentos difíceis.

Ao professor Bernardo por aceitar o convite para participar da banca de defesa e contribuir com suas valiosas sugestões e críticas para o enriquecimento deste trabalho.

Ao professor Clécio por ter me auxiliado a ingressar no mestrado e ter me mostrado antecipadamente, através do seu entusiasmo, o poder transformador que os trabalhos de pesquisa podem exercer no ânimo daqueles que os realizam.

Aos meus irmãos e a minha namorada Karina respectivamente pelos incentivos e equilíbrio necessários.

Aos meus amigos, em especial ao Régis Patrick, Rogério Araújo e José Wilson, por me fazerem entender o significado e o valor da amizade.

A todos, em especial do MIA, que colaboraram direta ou indiretamente para a realização deste trabalho.

## Resumo

A modelagem de sistemas interativos é uma atividade de desenvolvimento que deve ser realizada para se obter sistemas de qualidade. Durante esta atividade, é importante a aplicação de métodos e/ou modelos, cada qual responsável pela definição de procedimentos e notações que devem ser seguidos e/ou usados pelo desenvolvedor do sistema. Há na literatura, ferramentas CASE, que se por um lado, agilizam o trabalho do desenvolvedor, por outro, lhe exigem conhecimentos específicos sobre os métodos e/ou modelos, que elas implementam. Além disto, elas oferecem pouco suporte à reutilização da modelagem de um sistema interativo.

Neste trabalho, foram identificadas algumas providências para amenizar os problemas inerentes à sobrecarga de conhecimentos requeridos ao desenvolvedor, dentre as quais a unificação de métodos e de modelos. A unificação de métodos e modelos consiste na construção de um método capaz de englobar as principais vantagens oferecidas pelos demais, e de um grupo bem delimitado de modelos que proporcionem a identificação de diversos aspectos do sistema, numa perspectiva rica e integrada de visões complementares. A vantagem disto seria haver ferramentas mais flexíveis que suportassem qualquer método e/ou modelo, não impondo restrições de utilização ao desenvolvedor.

Esta unificação é possível através da realização dos processos de construção e aplicação de conceitos de modelagem, descritos na metodologia META. Conceitos de modelagem são informações inerentes a métodos e modelos organizados em diferentes níveis de abstração e interligados através de seus relacionamentos (como: herança e composição). Esta organização hierárquica facilita o processo de construção de um conceito através do reuso de especificações de conceitos já existentes. Neste trabalho foram pré-definidos vários conceitos de modelagem dentro de uma especificação gráfica e formal, os quais foram obtidos a partir de características semelhantes encontradas em métodos e/ou modelos. Para ilustrar o processo de aplicação, são validados os conceitos apresentados neste trabalho, com a modelagem de um módulo de aprendizagem de Pesquisa Operacional, tendo em vista as dificuldades prementes desta área de aplicação identificadas durante o estudo comparativo entre os sistemas de aprendizagem.

## Abstract

The modeling of interactive systems is a developing activity that must be performed to construct systems with quality. During this activity, the application of methods and models is very important, each one responsible for the definition of procedures and notations that must be followed and/or used by the system developer. In the literature does exist CASE tools, that by one hand, provide work agility to the developer, but by the other hand, require specific knowledge about methods and/or models, that are implemented by them. Furthermore, they offer little support to the modeling reuse of an interactive system.

A considerable variety of measures are identified in this work to reduce the problems implications about the overload of knowledge required to the developer, like the unification of processes and models belong others. The unification of processes and models consist of construct a well defined group of process and models, that provide respetively the most important advantages provided by the existing process, and the idenfication of the several system aspects in a integrated and detailed point of view. The advantage of this will be the providing of more flexible tools that support any method and/or model, without imposing restrictions of use to the developer.

This unification is possible by performing the processes of construction and aplication of modeling concepts, described in the metodologie called META. Modeling concepts are informations intrinsic to methods and models organized in different levels of abstraction and interconnected by relationships (like: inheritance and composition). This hierarchical organization facilitate the process of construct a concept by the reuse of specifications of the existing concepts. In this work, were pre-defined some modeling concepts in a graphical and formal specification, obtained from similar characteristics found in methods and/or models. To ilustrate the process of application, the concepts presented in this work are validated, with the modeling of an module of Operations Research learning, visualizing the pressing difficulties from this application area indentified during the comparative study between learning systems.

## Sumário

### Introdução

I – Objetivos.....	11
II – Hipóteses .....	13
III – Solução.....	13
IV – Metodologia de trabalho.....	14
V – Estrutura do trabalho.....	15

### Capítulo 1 - Processo de Modelagem de Sistemas Interativos

1.1 Fases dos ciclos de vida de desenvolvimento de sistemas interativos .....	16
1.1.1 Levantamento das necessidades.....	16
1.1.2 Análise e projeto.....	17
1.1.3 Implementação.....	17
1.1.4 Integração e testes.....	17
1.2 Modelos.....	18
1.2.1 Modelos conceituais .....	18
1.2.2 Modelos de arquitetura.....	22
1.2.3 Arquitetura.....	25

### Capítulo 2 - Métodos e Sistemas de Aprendizagem

2.1 Métodos de Aprendizagem .....	30
2.1.1 Aprendizagem baseada em problemas.....	30
2.1.2 Aprendizagem através de perguntas e respostas.....	31
2.1.3 Aprendizagem através de imagens.....	31
2.1.4 Aprendizagem construtivista.....	31
2.1.5 Aprendizagem colaborativa .....	31
2.2 Sistemas de aprendizagem e estudos ergonômicos.....	32



## **Capítulo 3 - Metodologia de apoio à modelagem e concepção de Sistemas Interativos**

3.1 A problemática na especificação de sistemas interativos.....	39
3.2 Objetivos e soluções para facilitar a especificação de sistemas interativos.....	40
3.3 Problemática na aprendizagem de Pesquisa Operacional.....	41
3.4 Objetivos e solução para facilitar a aprendizagem de Pesquisa Operacional.....	41
3.5 Ilustração das soluções elaboradas.....	41
3.6 A metodologia META.....	44
3.6.1 O processo de construção e aplicação de recursos de software.....	44
3.6.2 Conceitos de modelagem.....	45
3.6.3 Fases da metodologia META.....	46
3.6.3.1 Fase inicial.....	47
3.6.3.2 Fase de definição.....	47
3.6.3.3 Fase de instanciação.....	49
3.6.3.4 Fase de evolução.....	50
3.6.4 Vantagens da proposta.....	51
3.6.5 Utilização da metodologia META: Integrando os níveis.....	51

## **Capítulo 4 - Conceitos de modelagem utilizados no processo de desenvolvimento de sistemas**

4.1 Definição de conceitos de modelagem.....	55
4.2 Formas de representação de conceitos de modelagem.....	56
4.3 Conceitos de modelagem construídos.....	58
4.3.1 Meta-conceitos construídos e representados.....	59
4.3.1.1 Meta-conceito de Modelo de Arquitetura.....	60
4.3.1.2 Meta-conceitos de Padrão de Arquitetura.....	61
4.3.1.3 Meta-conceitos de Estrutura de Modelos Conceituais.....	63
4.3.1.4 Meta-conceitos de método.....	64
4.3.2 Conceitos.....	65
4.3.3 Instâncias de conceitos.....	67

## **Capítulo 5 - Desenvolvimento de um sistema de aprendizagem de Pesquisa Operacional a partir da evolução de um método de desenvolvimento de sistemas**

5.1 Construção e instanciação do método MAPEAR a partir da evolução do método MACIA.....	71
5.1.1 Definição do método MAPEAR .....	72
5.1.2 Instanciação do método .....	74
5.1.2.1 Levantamento das necessidades.....	74
5.1.2.2 Análise e projeto.....	75
5.1.2.3 Implementação e prototipagem.....	86

## **Capítulo 6 – Conclusão**

6.1 Revisão geral.....	93
6.2 Contribuições.....	94
6.3 Trabalhos futuros.....	95

<b>Anexo A</b> .....	97
----------------------	----

<b>Anexo B</b> .....	100
----------------------	-----

<b>Referências</b> .....	109
--------------------------	-----

## Introdução

Um Sistema Interativo (SI) é um conjunto de programas computacionais feitos para serem manipulados pelo(s) usuário(s) com a finalidade de resolver um problema específico. Ele é constituído de uma parte não interativa (aplicação) e de outra interativa (apresentação). A aplicação possui aspectos estáticos (dados) e dinâmicos (tratamentos). Os dados constituem as informações a serem manipuladas pelos tratamentos do sistema durante a resolução de um problema. A apresentação corresponde aos módulos utilizados na exposição dos dados ao usuário e na recepção de estímulos e informações do usuário, possibilitando a troca de informações entre eles [Furtado, 1997].

A modelagem de um SI possibilita a visualização e documentação de suas características (como estrutura, comportamento e formas de interação), além de facilitar sua construção segundo uma arquitetura. A construção de um SI de acordo com uma arquitetura se deve ao fato de que a sua estrutura deve ser organizada de tal maneira a possibilitar ganhos de qualidade e um melhor desempenho na construção e no uso do sistema. As arquiteturas são estruturas de organização do sistema segundo um modelo, onde estão inseridos e especificados os seus componentes, as interfaces e as formas de comunicação entre estes [Savidis & Stephanidis, 2001], que permitem a identificação das etapas de tradução entre a linguagem do usuário e a linguagem abstrata da aplicação [Nanard, 1990].

Além da organização arquitetural, é importante destacar que o projetista e/ou desenvolvedor deve(m) seguir métodos de desenvolvimento de sistemas para orientá-lo(s) sobre procedimentos necessários à modelagem. Em termos gerais, os métodos de desenvolvimento de sistemas descrevem as atividades necessárias para a especificação e construção de sistemas, que satisfaçam a um conjunto de critérios de qualidade e usabilidade estabelecidos. Os métodos geralmente são apoiados por ferramentas, tal como ferramentas CASE, cuja função é facilitar na modelagem e/ou implementação de sistemas interativos através da automação de parte dos procedimentos necessários (tal como: geração automática de um modelo a partir de outro). Dentre estas ferramentas, destacam-se também os geradores de interface, que possibilitam ao projetista e/ou desenvolvedor a construção das interfaces de um sistema a partir de especificações ou modelos, proporcionando uma redução considerável da carga de trabalho e do tempo necessários para o desenvolvimento destas. Geralmente, as

ferramentas de desenvolvimento de sistemas interativos exigem dos projetistas e/ou desenvolvedores conhecimento dos modelos e/ou métodos que as mesmas implementam. Isto acontece porque as ferramentas, geralmente, estão restritas a um determinado método usando certos modelos. Tal fato é decorrente das dificuldades de implementação e restrições tecnológicas atuais, que limitam os formalismos e práticas de formalização utilizados em cada ferramenta a um conjunto de estereótipos bem definidos.

Outro fator importante a ser ressaltado é a ausência de ferramentas que auxiliem o projetista na documentação, na modelagem e no desenvolvimento de sistemas interativos, permitindo o reuso de conceitos de modelagem. Isto é, o aproveitamento de especificações e formas de representações (chamadas de conceitos de modelagem) utilizadas na definição conjunta ou particular de métodos e/ou modelos, e na identificação de características de um SI específico. Além disto, as ferramentas não possuem funcionalidades para evoluir um sistema, após sua utilização efetiva pelo usuário no ambiente real de trabalho. Evoluir um sistema pode significar a necessidade de atender a uma evolução tecnológica (como de mudança de um estilo de interação), ou de torná-lo adaptativo ao usuário e/ou seu contexto de utilização, dentre outros. Manter a modelagem e funcionalidades de um SI adequadas em relação às mudanças tecnológicas, e às necessidades do usuário, às vezes requer modificações significativas no formalismo de um método e/ou do(s) modelo(s) que o representa(m). Como por exemplo, um sistema obsoleto com relação à demanda do usuário pode requerer uma evolução no seu método de desenvolvimento para integrar atividades mais centradas no usuário e/ou modelos mais apropriados para a especificação deste.

Diante destes problemas, foram estabelecidos objetivos para este trabalho, que permitiram a definição de princípios que nortearam a construção de soluções para tais problemas.

## **I) Objetivos**

Este trabalho objetiva auxiliar o projetista e/ou desenvolvedor na modelagem e no desenvolvimento de sistemas interativos, através da construção, reutilização e aplicação de recursos de software. Recursos de software são métodos e modelos responsáveis respectivamente pela compreensão de características e execução das atividades necessárias

para o desenvolvimento de sistemas interativos. Isto significa que, através do referencial teórico desenvolvido neste trabalho, o projetista e/ou desenvolvedor poderá desenvolver um SI para um certo domínio aplicando um novo método e modelos construídos, provavelmente, a partir de recursos já existentes. Este referencial teórico se refere a uma metodologia, chamada META, que descreve os procedimentos, que o projetista e/ou desenvolvedor deverá seguir para definir e aplicar estes recursos. Os objetivos específicos relativos ao auxílio no processo de modelagem e implementação de sistemas interativos são:

- i) Melhorar o suporte à documentação, manutenibilidade e evolução de sistemas interativos, através da execução de procedimentos necessários para a construção de especificações de conceitos de modelagem;
- ii) Permitir ao projetista obter experiências de modelagem e desenvolvimento, através do reuso de especificações de conceitos de modelagem, e acesso à documentação e;
- iii) Identificar características comuns a métodos e modelos, definindo especificações de conceitos de modelagem.

Foi escolhido como domínio de aplicação deste trabalho, o ensino de Pesquisa Operacional. A Pesquisa Operacional trata do uso de métodos científicos para a tomada de decisões, consistindo na construção e resolução de modelos matemáticos representativos do sistema em foco [Silva et al., 1998]. O objetivo específico em relação a este domínio é desenvolver um módulo interativo para auxiliar na aprendizagem de Pesquisa Operacional, que deverá ser integrado a um sistema de educação a distância. A escolha deste objetivo foi fundamentada principalmente devido à ausência de métodos para desenvolver sistemas nesta área. Além disto, o fato dos alunos terem dificuldade na aplicação de Pesquisa Operacional a problemas práticos motivou a necessidade de se desenvolver um módulo que fosse fácil de usar e que promovesse a aprendizagem contextual e colaborativa.

Um melhor detalhamento dos objetivos deste trabalho será descrito no capítulo 3, antes da descrição da proposta e após a descrição do estado da arte, onde é apresentada mais claramente a problemática.

## II) Hipóteses

Para que os objetivos deste trabalho fossem alcançados, foram utilizados como fundamentos as seguintes proposições:

- A estratificação dos conceitos de modelagem em três camadas de diferentes níveis de abstração (meta-nível, nível de construção de conceitos e nível de instanciação de conceitos) facilita a construção dos mesmos através da reusabilidade [Vanderdonck et al, 2003]. Isto porque os conceitos pertencentes às camadas de maior nível de abstração podem ser partes da especificação de conceitos de camadas de níveis inferiores;
- A possibilidade de (re)definir modelos e métodos facilita a manter a coerência da modelagem de um SI evolutivo, face às necessidades de mudanças tecnológicas e de necessidades dos usuários e;
- A integração de recomendações ergonômicas num método de desenvolvimento de sistemas de aprendizagem ajuda a construir sistemas mais fáceis de usar e aprender [Hack et al, 1999].

## III) Solução

O presente trabalho faz uma aplicação da metodologia META para apoiar a construção e a aplicação de conceitos de modelagem necessários ao desenvolvimento de um módulo de aprendizagem de Pesquisa Operacional. A solução descrita neste texto visa ajudar tanto o especialista em Interação Humano-Computador (IHC) a fazer a especificação da interface, definindo modelos que lhe são próprios, como o engenheiro de software a elaborar métodos e fazer a especificação do domínio (parte não interativa). Para simplificar o discurso, o nome *projetista* se referirá a estes dois tipos de usuário da metodologia.

A metodologia META descreve os procedimentos que auxiliam o projetista em como construir e utilizar as especificações de métodos e modelos na construção do módulo de

aprendizagem de Pesquisa Operacional. Para ilustrar tal proposta, neste trabalho foi construído um método de desenvolvimento deste módulo, chamado MAPEAR (Método de Ajuda a concepção de sistemas de Aprendizagem eRgonômicos). Este novo método reusou alguns componentes de especificação de um método (MACIA - Método de Auxílio à Concepção e realização de InterfAces [Furtado, 1997]) já existente, que possibilitaram a especificação deste módulo considerando sua usabilidade e manutenibilidade. Este método foi escolhido para reuso devido à facilidade de execução do mesmo para a obtenção de interfaces ergonômicas. Além disto, algumas fases foram inseridas para contemplar as características funcionais pedagógicas que o módulo deveria ter.

#### **IV) Metodologia de trabalho**

Este trabalho foi executado segundo a metodologia de trabalho descrita a seguir na respectiva ordem cronológica dos fatos:

- Estudo de métodos, modelos e arquiteturas utilizados no desenvolvimento de sistemas interativos;
- Estudo de métodos e sistemas de aprendizagem em geral;
- Estudo de métodos e sistemas de aprendizagem de Pesquisa Operacional;
- Descrição da metodologia META em relação à produção e aplicação de conceitos (métodos e modelos) no desenvolvimento e na documentação de sistemas interativos [Furtado & Cardoso Junior, 2003];
- Definição e representação de conceitos em diversos níveis de abstração (construção e aplicação);
- Aplicação da metodologia META para ilustrar a elaboração de um novo método de desenvolvimento de sistemas para a construção do módulo de aprendizagem de Pesquisa Operacional e;

- Utilização do novo método de desenvolvimento de sistemas (MAPEAR) para especificar e documentar a especificação do módulo de aprendizagem. Tal documentação abrange, principalmente, modelos de desenvolvimento de software.

## **V) Estrutura do trabalho**

Quanto à organização, o trabalho encontra-se dividido em três partes:

- Na primeira parte, foi apresentado o estado da arte, ou seja, o conhecimento bibliográfico utilizado. A primeira parte está dividida em dois capítulos, o primeiro capítulo trata do processo de modelagem de sistemas interativos, enquanto que o segundo apresenta métodos e sistemas de aprendizagem;
- Na segunda parte, representada pelo capítulo 3, foram apresentadas a proposta da metodologia META para a solução dos problemas de modelagem e implementação de sistemas interativos e a proposta do método MAPEAR para o desenvolvimento do módulo de aprendizagem de Pesquisa Operacional;
- Na terceira parte, formada pelos capítulos 4 e 5, foram definidos e exemplificados os conceitos que são utilizados pela metodologia META e o método MAPEAR para o desenvolvimento do módulo de aprendizagem de Pesquisa Operacional, respectivamente e;
- Na quarta parte, constituída pela conclusão e anexos, são apresentados considerações finais, contribuições, perspectivas futuras e complementos em relação ao trabalho realizado.



# **1. Processo de Modelagem de Sistemas Interativos**

## **Introdução**

O desenvolvimento de sistemas interativos geralmente é apoiado por um processo de modelagem que está organizado segundo uma seqüência de fases. A forma de encadeamento das fases pode variar de acordo com o ciclo de vida [Pressman, 1995] adotado. Em cada fase é empregado um conjunto de modelos cuja função é gerar e formalizar informações e/ou conhecimentos em um nível de detalhe e abstração particular. Além dos modelos também podem ser empregadas arquiteturas no processo de modelagem para implementar sistemas em formatos específicos de detalhamento e de organização de suas estruturas funcionais.

Neste capítulo, serão descritas as fases que são geralmente encontradas num processo de desenvolvimento de sistemas interativos, os modelos e arquiteturas aplicados, além de ser apresentado um estudo comparativo realizado entre arquiteturas segundo a organização funcional de suas estruturas.

## **1.1 Fases dos ciclos de vida de desenvolvimento de sistemas interativos**

As fases geralmente empregadas ao longo dos ciclos de vida de desenvolvimento de sistemas interativos são: levantamento das necessidades, análise e projeto, implementação, e integração e testes.

### **1.1.1 Levantamento das necessidades**

Na fase de levantamento das necessidades é feito um estudo do estado atual do ambiente organizacional ao qual se pretende prestar auxílio informacional. Neste estudo, são apuradas as deficiências do ambiente em termos de tecnologia da informação e necessidades do usuário e cliente do sistema que será desenvolvido em termos de satisfação das funcionalidades realizadas. Esta fase requer, portanto, que sejam realizados: estudos de campo no ambiente em que é pretendido prestar auxílio informacional; entrevistas com pessoas importantes para o desempenho das funcionalidades da organização e levantamento do estado da arte em termos de tecnologia, conhecimentos e recursos informáticos.

É importante a documentação do estado atual da organização e de suas necessidades em termos de processos de negócio e de interação, conjunto de procedimentos e formas de interação que devem ser efetivados para que a organização atinja seus objetivos. Isto viabiliza

a utilização das informações documentadas como um mecanismo de consulta em fases posteriores e como um meio de intercâmbio de informações entre o analista e os entrevistados. É importante a utilização de modelos como material de documentação e como meio de intercâmbio de informações com o usuário.

### **1.1.2 Análise e projeto**

Esta fase consiste na aplicação, a nível conceitual, de tecnologia, conhecimento e recursos da informática, no aprimoramento e na resolução de deficiências identificadas nos processos de negócio e interação documentados na fase de levantamento das necessidades. Através da análise do material documentado, é possível observar a possibilidade de reorganizar os processos de negócio da organização e aprimorar os processos de interação, com o emprego da informática, de tal forma a obter ganhos e satisfação do usuário. A partir de então, é possível identificar a interatividade e os comportamentos que o sistema deve assumir para suprir as necessidades da organização. Estes comportamentos e estereótipos de interação devem ser capturados e formalizados através do uso de modelos, e gradativamente refinados e tratados de tal forma a se obter entidades passíveis de representação para tratamento computacional.

### **1.1.3 Implementação**

Esta fase é concretizada através da aplicação de alguma linguagem computacional para representar as entidades obtidas na fase de análise e projeto. Isto é feito a partir da codificação das especificações das entidades, representadas sob a forma de modelos, respeitando as convenções e limitações da linguagem utilizada e da infraestrutura física e lógica que suportará o sistema.

### **1.1.4 Integração e testes**

Nesta fase ocorre a implantação e avaliação do sistema no ambiente organizacional. É realizada a incorporação do sistema desenvolvido ao parque tecnológico existente através da criação dos vínculos necessários entre o novo sistema e outros sistemas da organização. À medida que o sistema vai sendo implantado, testes e avaliações são realizados para medir o quanto o sistema está desempenhando bem as suas funções. Estas avaliações são feitas segundo critérios de qualidade e usabilidade pré-estabelecidos e critérios de interatividade. Os critérios de qualidade consistem em regras e padrões desejáveis, no sistema e nos seus

processos de desenvolvimento, para a valorização do mesmo. Por sua vez, os critérios de usabilidade estabelecem a facilidade de uso do sistema, enquanto que os critérios de interatividade definem o quanto ele é adequado em relação aos meios e a forma de comunicação utilizada pelo usuário durante o seu uso. Após a implantação do sistema, ele deverá ser avaliado quanto à integração em relação ao ambiente para o qual ele foi desenvolvido. Isto deve ser feito para que possam ser estabelecidas as medidas necessárias ao estabelecimento da colaboração requerida entre os sistemas que formam o ambiente, para o bom desempenho do mesmo. Os testes e avaliações podem ser feitos com o uso de modelos e protótipos. É importante lembrar que eles devem ser feitos durante todo o desenvolvimento de um SI.

## 1.2 Modelos

Os modelos são importantes instrumentos para a consecução dos objetivos de cada fase do ciclo de vida de desenvolvimento de sistemas interativos. Os tipos de modelos que geralmente são utilizados no desenvolvimento de sistemas interativos são [Furtado, 1997]: modelos conceituais e modelos de arquitetura. A figura 1.1 apresenta um exemplo de aplicação de modelos ao longo das fases de um ciclo sequencial e iterativo de desenvolvimento de um SI.

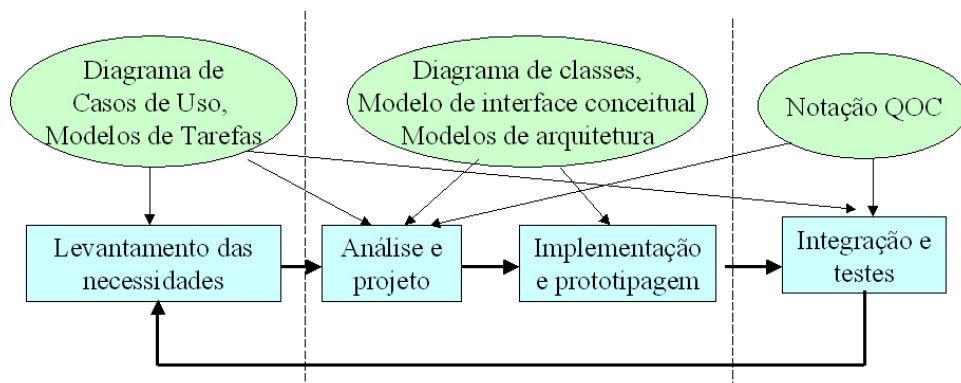


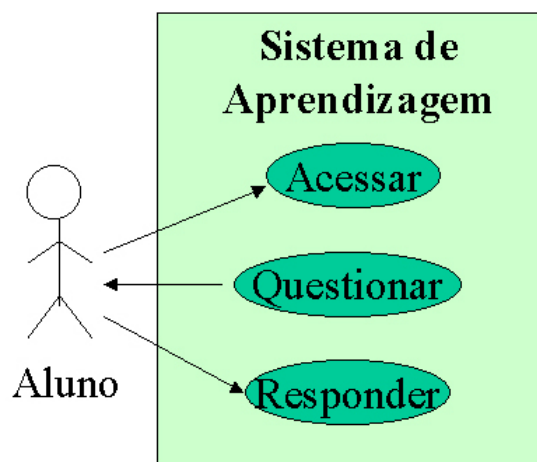
Figura 1.1 – Aplicação de modelos ao longo das fases de desenvolvimento de software

### 1.2.1 Modelos conceituais

Os modelos conceituais servem para a avaliação e concepção de sistemas e/ou de suas interfaces, tais como: Diagrama de Casos de Uso [Booch et al, 2000], Modelo de Tarefas

[Furtado, 1997], Modelo de Interface Conceitual [Furtado, 1997], Diagrama de classes [Booch et al, 2000] e notação QOC [MacLean et al, 1991].

O Diagrama de Casos de Uso serve para a especificação da conversação entre o usuário e o sistema. Através dele representam-se os comportamentos do sistema e do usuário durante a interação. O seu uso é bastante interessante na fase de levantamento de necessidades para capturar informações relacionadas aos comportamentos de indivíduos necessários para a efetivação dos processos de negócio da organização. Ele também pode ser usado



*Figura 1. 2 – Caso de Uso de um aluno interagindo com um sistema de aprendizagem a distância*

na fase de análise e projeto, para identificar os comportamentos desejados do sistema a ser desenvolvido, e na fase de integração e testes, para comparar o comportamento do sistema implantado com o que é desejado. A figura 1.2 ilustra exemplos de Casos de Uso de um aluno interagindo com um sistema de aprendizagem a distância. De acordo com a figura, o aluno pode acessar um curso de educação a distância para responder questões que lhe são solicitadas.

O Modelo de Tarefas especifica o conjunto de atividades que o usuário e/ou computador deve executar para que um determinado objetivo seja alcançado [Furtado, 1997]. Ele pode ser utilizado nas seguintes fases: i) levantamento de necessidades, para descrever os procedimentos necessários para a realização dos processos de negócio da organização; ii) análise e projeto, para especificar as formas de interação que estarão disponíveis para o usuário utilizar o sistema; e iii) integração e testes, para validar o comportamento do sistema em relação às funções para as quais ele foi projetado. Este modelo pode ser hierárquico, representado por árvores aonde os nós de hierarquia superior são tarefas constituídas por outras, que constituem os nós da hierarquia inferior subjacente. Dentre estes modelos são citados o MAD [Scapin & Pierret-Golbreich, 1990] e o TAOS [Medeiros, 2003] entre outros. A figura 1.3 ilustra a modelagem de tarefas MAD, expressando que para um aluno responder a um sistema de aprendizagem (tarefa principal em foco) ele deve realizar as

seguintes sub-tarefas na seqüência a seguir: identificar o problema, escolher a resposta e enviá-la.

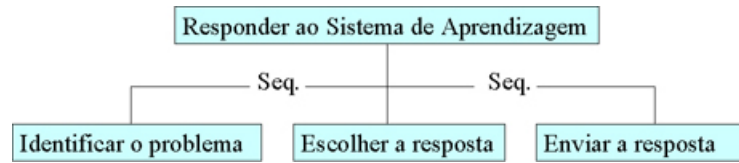


Figura 1.3 – Modelo de Tarefas relativas à tarefa de o aluno responder a um sistema de aprendizagem

O Modelo de Tarefas também pode ser constituído por um fluxo de atividades, iniciado a partir de um ponto de onde o fluxo que representa a atividade inicial parte. Daí em diante, de acordo com as decisões tomadas, o fluxo é seguido por um ou mais fluxos diferentes, que representam as tarefas subsequentes que devem ser executadas de acordo com a satisfação das condições necessárias. Tais fluxos são representados no Diagrama de Atividades [Booch et al., 2000] ilustrado na figura 1.4. Nesta figura, o Diagrama



Figura 1.4 – Diagrama de Atividades de um aluno respondendo a um sistema de aprendizagem

de Atividades descreve o conjunto e a seqüência de atividades necessárias para que o aluno responda a um sistema de aprendizagem, o mesmo caso apresentado na figura 1.3.

O Modelo de Interface Conceitual consiste na representação das interfaces de um sistema a nível conceitual [Furtado, 1997]. Ele é representado por um conjunto de espaços, sub-espaços e objetos de interação detentores de características particulares e organizados segundo uma ordem apropriada. Os espaços correspondem cada um a toda a área espacial que o usuário pode observar e/ou interagir. Os sub-espaços são cada uma das sub-áreas espaciais que constituem um espaço. Os objetos de interação representam os dispositivos com os quais os usuários podem interagir. A figura 1.5 apresenta um Modelo de Interface Conceitual representando um formulário de identificação de um usuário para acesso a um curso de educação a distância. Este formulário é composto dos seguintes objetos de interação: um objeto do tipo entrada de dados para o nome do usuário, outro objeto do mesmo tipo para a

senha do usuário, e um terceiro do tipo clique para a submissão das informações inseridas nos dois primeiros objetos.

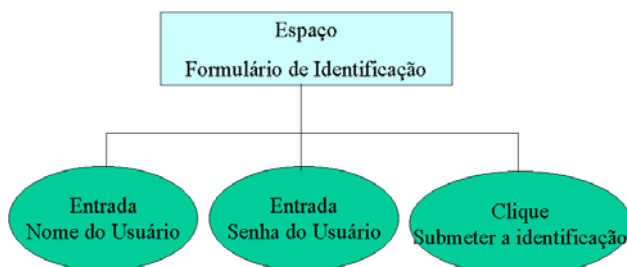


Figura 1.5 – Modelo de Interface Conceitual de um formulário de identificação

O diagrama de classes [Fowler & Scott, 2000] modela uma visão da estrutura estática de um sistema, identificando as entidades que constituem o sistema através de seus dados (atributos), comportamentos (operações) e relacionamentos em três diferentes perspectivas: conceitos do domínio do sistema; classes utilizadas como conectores entre classes ou componentes em um determinado software; e classes implementadas. A figura 1.6 ilustra um diagrama de classes referente a um sistema de cadastro escolar. Nele podemos observar que existem pessoas, alunos, professores e disciplinas. Estas classes possuem atributos (matrícula em aluno), operações (incluir em professor) e relacionamentos (herança entre a classe *pessoa* e as classes *professor* e *aluno*). Existem classes que representam a associação entre duas classes através de seus relacionamentos. Este é o exemplo da classe associativa *AlunoDisc*, que especifica os alunos que estão matriculados numa certa disciplina.

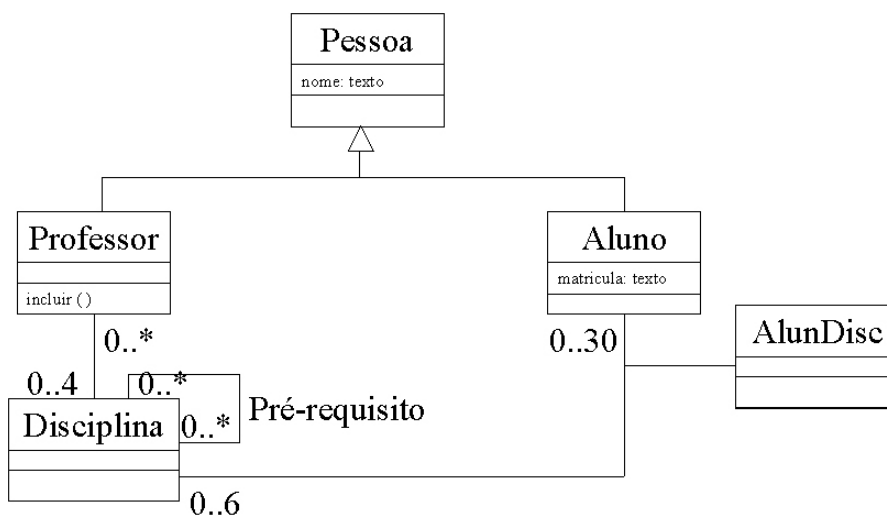


Figura 1.6 – Diagrama de classes de um sistema de cadastro escolar

A notação QOC (Questões, Opções e Critérios) é utilizada para avaliar os critérios de usabilidade [Furtado & Cardoso Júnior, 2003]. Na notação QOC, questões são utilizadas para esclarecer importantes aspectos relacionados ao design, opções são as soluções encontradas e critérios são objetivos em termos de critérios ergonômicos (acessibilidade, aceitabilidade e usabilidade) que devem ser utilizados na avaliação e escolha das soluções. Isto credencia o uso da notação QOC nas fases de análise e projeto, e integração e testes. A figura 1.7 ilustra a aplicação da notação QOC para a elucidação da dúvida a respeito “a quem” ou “a que” a operação *refazer* está relacionada. As opções de respostas são usuário individual e documento. Estas opções de resposta devem ser avaliadas de acordo com os critérios: recuperação do usuário, facilidade de aprendizagem e facilidade de implementação.

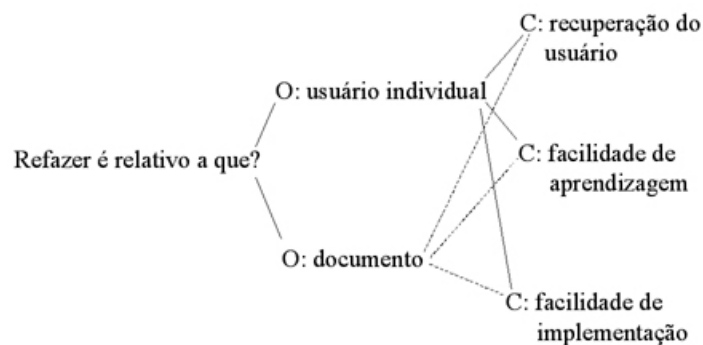


Figura 1.7 – Notação QOC da operação *refazer*

### 1.2.2 Modelos de arquitetura

Os modelos de arquitetura de sistemas interativos são utilizados como estruturas abstratas de alto nível para organizar o sistema segundo um determinado padrão, sem especificar detalhadamente como os componentes constituintes destas estruturas estão inseridos nelas e como eles se comunicam [Coutaz, 1993]. Dentre os modelos de arquitetura, existem: Seeheim, Arco, MVC, PAC, PAC-Amodeus e Três Camadas.

O modelo Seeheim [Coutaz, 1993] se caracteriza como uma organização de um SI em diversos componentes: Aplicação, Interface da Aplicação, Controle de Diálogo e Apresentação. *Aplicação* representa o domínio de aplicação do sistema em forma de funções e dados, enquanto que *Apresentação* representa a visão que o usuário do sistema tem da interface do sistema com o usuário, ficando a cargo do *Controle* fazer o processamento seqüencial e o intercâmbio de dados entre a *Aplicação* e a *Apresentação*, além da execução de

seqüências de visões da interface do usuário. Esta estrutura promove a independência física e conceitual entre *Aplicação* e *Apresentação*.

O modelo Arco [Coutaz, 1993] também é caracterizado pela disposição do SI em componentes: *Domínio*, *Adaptador de Domínio*, *Diálogo*, *Apresentação* e *Dispositivos de Interação*. O *Domínio* é responsável por armazenar e executar os códigos e os dados do domínio de aplicação do sistema, os *Dispositivos de Interação* são os objetos de interação do usuário com o sistema, enquanto que o *Diálogo* tem como funcionalidade coordenar o intercâmbio de dados entre o domínio e os objetos de interação da interface do usuário, controlando o processo de diálogo. O *Adaptador de Domínio* e a *Apresentação* são responsáveis pela independência funcional entre o *Domínio*, o *Diálogo* e os *Dispositivos de Interação* da interface do usuário implementada.

O modelo MVC [Normand, 1992] é representado pela estruturação do SI em uma composição multi-agente, sendo cada agente dotado de partições: *Modelo*, *Controle* e *Visão*. O *Modelo* contém as funções do domínio de aplicação do sistema que executam as informações obtidas do *Controle* que capta e gere entradas em forma de dados e ações do usuário, enquanto a *Visão* gera e controla a geração das saídas, obtidas dos processamentos realizados no Modelo, na interface do usuário. O *Controle* e a *Visão* acessam o *Modelo* diretamente para o envio da entrada e a recepção da saída respectivamente, não podendo o *Modelo* acessar diretamente a *Visão* e o *Controle* [Savidis & Stephanidis, 2001].

O modelo PAC [Bass & Coutaz, 1991] também é uma forma de estruturação multi-agentes em que cada agente é formado por partições: *Apresentação*, *Abstração* e *Controle*. A *Apresentação* é a visão da interface do usuário com o agente do sistema, a *Abstração* representa funções e dados relacionados aos agentes, enquanto que o *Controle* mantém a coerência e a independência funcional entre a *Apresentação* e a *Abstração* do agente ao mesmo tempo em que representa o meio de intercâmbio de dados e informações entre ambos. Também é característica do PAC, a organização dos agentes numa estrutura hierárquica, de tal maneira que um agente não terminal é formado, tanto a nível de abstração como de apresentação, por outros agentes numa estrutura semelhante a uma árvore, aonde os agentes de nível mais baixo (folhas) são Dispositivos de Interação ou componentes atômicos da interface com o usuário.



O modelo PAC-Amodeus [Coutaz, 1993] [Nigay, 1994] aproveita tanto a estrutura na forma de divisões em componentes do meta-modelo Arco como na forma multi-agentes do meta-modelo PAC. Os componentes do PAC-Amodeus são similares a do meta-modelo Arco: *Núcleo Funcional* (Domínio), *Controle Funcional* (Controle), *Dispositivos de Interação*, *Adaptador do Núcleo Funcional* (Adaptador de Domínio) e *Apresentação*. A diferença está no fato de que o *Controle de Diálogo* do PAC-Amodeus está estruturado internamente em forma de agentes PAC, cujas extremidades, *Abstração* e *Apresentação*, se comunicam respectivamente com o *Adaptador do Núcleo Funcional* e com a *Apresentação*. A figura 1.8 ilustra o modelo PAC-Amodeus.



Figura 1.8 – Modelo de Arquitetura PAC-Amodeus

O modelo Três Camadas [Oliveira, 2002] consiste na divisão do sistema em três camadas que se relacionam: *Apresentação*, *Negócio* e *Persistência*. A camada de apresentação corresponde à parte do código relativa à apresentação da interface, ao seu controle e às suas características e, tornando-a equivalente à *Apresentação* e ao *Controlador de Diálogo* do PAC-Amodeus. A camada de negócio diz respeito somente ao código responsável pelas funcionalidades da aplicação, o que a torna equivalente à parte do Núcleo Funcional do modelo PAC-Amodeus responsável pelo tratamento das informações a nível de domínio. A camada de persistência é a parte do código responsável pelo estabelecimento, manutenção e encerramento da conexão do sistema com o banco de dados, o que a torna similar à parte do *Núcleo Funcional* do modelo PAC-Amodeus responsável pelo armazenamento e recuperação de dados. A figura 1.9 ilustra o modelo de arquitetura Três Camadas.



Figura 1.9 – Modelo de Arquitetura Três Camadas

### 1.2.3 Arquitetura

As arquiteturas são estruturas de organização do sistema segundo um modelo aonde estão inseridos e especificados os seus componentes, as interfaces e as formas de comunicação entre estes [Savidis & Stephanidis, 2001]. A nível de ilustração, a seguir serão apresentadas as arquiteturas: H4, Unificada e Seis Camadas.

A Arquitetura H4 [Depaulis et al., 2001], ilustrada na figura 1.10, encontra-se dividida em cinco camadas: *Apresentação*, *Monitor*, *Diagets*, *Questionários* e *Núcleo Funcional*. A camada *Apresentação* é o meio pelo qual o usuário interage com o sistema. O *Monitor* é responsável pela recepção de dados e ações (*tokens*) produzidas pelos usuários e pelo repasse dos tokens entre as camadas de *diagets*, começando pela de nível inferior. Um *diaget* (*dialog* + *gadget*) é um dispositivo atômico de diálogo que muda de estado à medida que recebe e/ou envia *tokens*, facilitando e coordenando a comunicação no sistema. A camada *Diagets* possui uma divisão interna de camadas hierárquicas de *diagets*, cuja função é assimilar os *tokens*, esperados de acordo com o estado do *diaget* receptor, e fazer a chamada ao questionário, que é uma chamada de função abstrata em relação à forma de execução da tarefa a ser realizada e utilizada para modificar o estado do *Núcleo Funcional*, assim que o estado de chamada for alcançado pelo *diaget* receptor, apropriado para os *tokens* recebidos. Caso os *tokens* não sejam assimilados pelo *diaget*, para o qual foi endereçado pelo *Monitor*, eles são retornados para o *Monitor* que os repassa para um *diaget* de nível superior. A camada *Questionários* é formada pelos questionários dos *diagets*, sendo a responsabilidade de cada questionário, de passar os

*tokens* do seu respectivo *diaget* para o *Núcleo Funcional*, através da chamada das funções para o tratamento daqueles *tokens*. O *Núcleo Funcional* faz o processamento dos mesmos gerando um *token* de saída. O *token* gerado é passado através do questionário ao seu respectivo *diaget* que o repassa para o *Monitor*, e este o envia para um *diaget* de camada superior. O *diaget* da camada mais alta somente pode receber *tokens* do *Monitor* para enviá-los ao *Núcleo Funcional*, não sendo permitido o percurso inverso. Durante a transição, processamento e geração de *tokens*, mudanças significativas ocorrem na camada *Apresentação* para informar ao usuário o estágio de desenvolvimento da tarefa realizada.

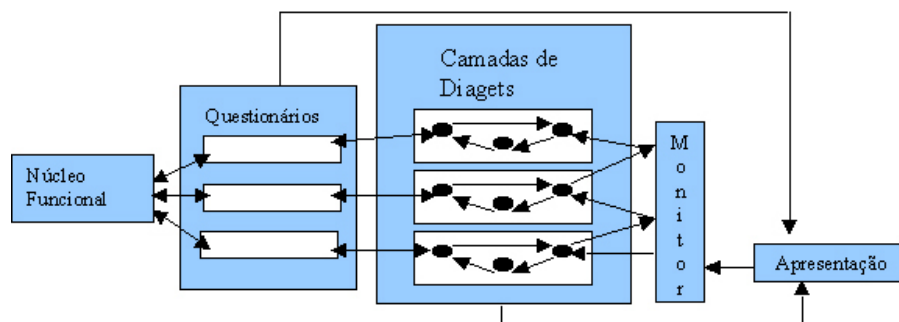
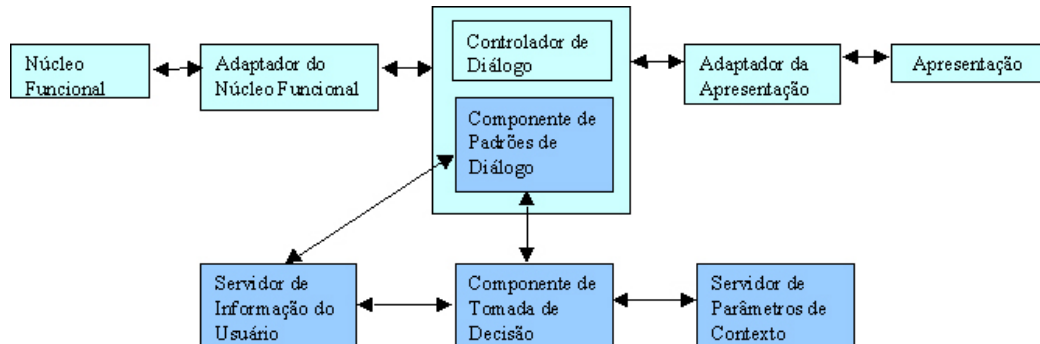


Figura 1.10 – Arquitetura H4

A Arquitetura Unificada de Software de Interface do Usuário [Savidis & Stephanidis, 2001], exibida na figura 1.11, se caracteriza pela capacidade de adaptação da interface às características do usuário, e do ambiente ou do contexto de uso. A estratificação desta arquitetura ocorre como uma disposição ortogonal entre a sua estrutura horizontal e a vertical. A estrutura horizontal da Arquitetura Unificada é representada pelo meta-modelo Arco, enquanto que a estrutura vertical, que torna possível a adaptação, é formada pelo Servidor de Informações do Usuário, Servidor de Parâmetros de Contexto, Componente de Tomada de Decisão e Componente de Padrões de Diálogo. Quando o usuário inicia a interação com o sistema, através de sua identificação, o seu perfil é identificado pelo Servidor de Informações do Usuário, enquanto que as informações sobre o ambiente ou contexto de uso são identificadas pelo Servidor de Parâmetros de Contexto. O perfil do usuário e as informações sobre o contexto de uso são enviados para o Componente de Tomada de Decisão que, baseado nestas informações, escolhe o padrão de diálogo entre o sistema e o usuário. O padrão de diálogo escolhido é informado para o Componente de Padrões de Diálogo que implementa o diálogo escolhido. O perfil do usuário e do contexto de uso também podem ser inferidos, pelo

Servidor de Informações do Usuário e de Contexto de Uso respectivamente, em tempo de uso do sistema.



*Figura 1.11 – Arquitetura Unificada de Software de Interface do Usuário*

A Arquitetura Seis Camadas [Tarby, 1993], apresentada na figura 1.12, é formada pela: i) primeira camada, que é composta de uma base de dados com os relacionamentos necessários para o controle de tipo e integridade para criação e supressão de dados; ii) segunda camada, que é composta por objetos naturais, alocados sobre a base de dados, que fazem chamadas e agem sobre o banco de dados, além de tornarem o sistema perceptível e interativo para o usuário através de suas apresentações; iii) terceira camada, que contém as operações que manipulam um ou mais objetos naturais por vez; iv) quarta camada, que consiste nos procedimentos Diane+ (minimais, previstos e efetivos) que permitem efetuar tratamentos que manipulam os objetos naturais; v) quinta camada, que consiste no conjunto de objetivos a serem alcançados, correspondendo ao conjunto de tarefas que a aplicação pode executar; e vi) sexta camada, que corresponde ao conjunto de estratégias disponíveis para a consecução do objetivo principal almejado. Durante o uso de um sistema implementado através desta arquitetura, o usuário utiliza uma estratégia para alcançar um objetivo principal. Esta estratégia consiste na conclusão de sub-objetivos, cada um relacionado a um conjunto de tarefas. Para a conclusão das tarefas, de acordo com o posto de trabalho e o tipo de usuário, existem os procedimentos minimais, previstos e efetivos. Estes procedimentos, por sua vez, são constituídos de operações, cada uma responsável pelo tratamento e manipulação de um ou mais objetos naturais.

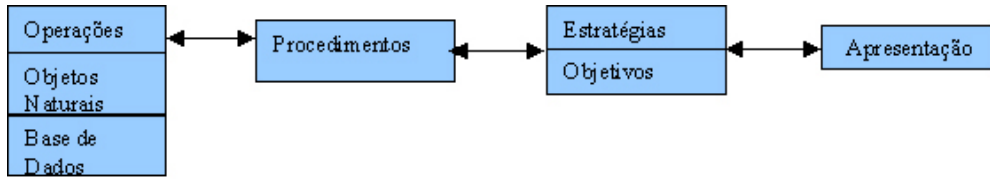


Figura 1.12 – Arquitetura Seis Camadas

Após estudos sobre as arquiteturas, observou-se que elas são compostas de componentes organizados em estruturas, que podem ter equivalentes, segundo as suas funções. Esta equivalência está ilustrada na figura 1.13 através de um quadro comparativo entre três exemplos de arquiteturas de software: Arquitetura H4 [Depaulis et al., 2001], Arquitetura Unificada de Software de Interface do Usuário [Savidis & Stephanidis, 2001] e Arquitetura Seis Camadas [Tarby, 1993].

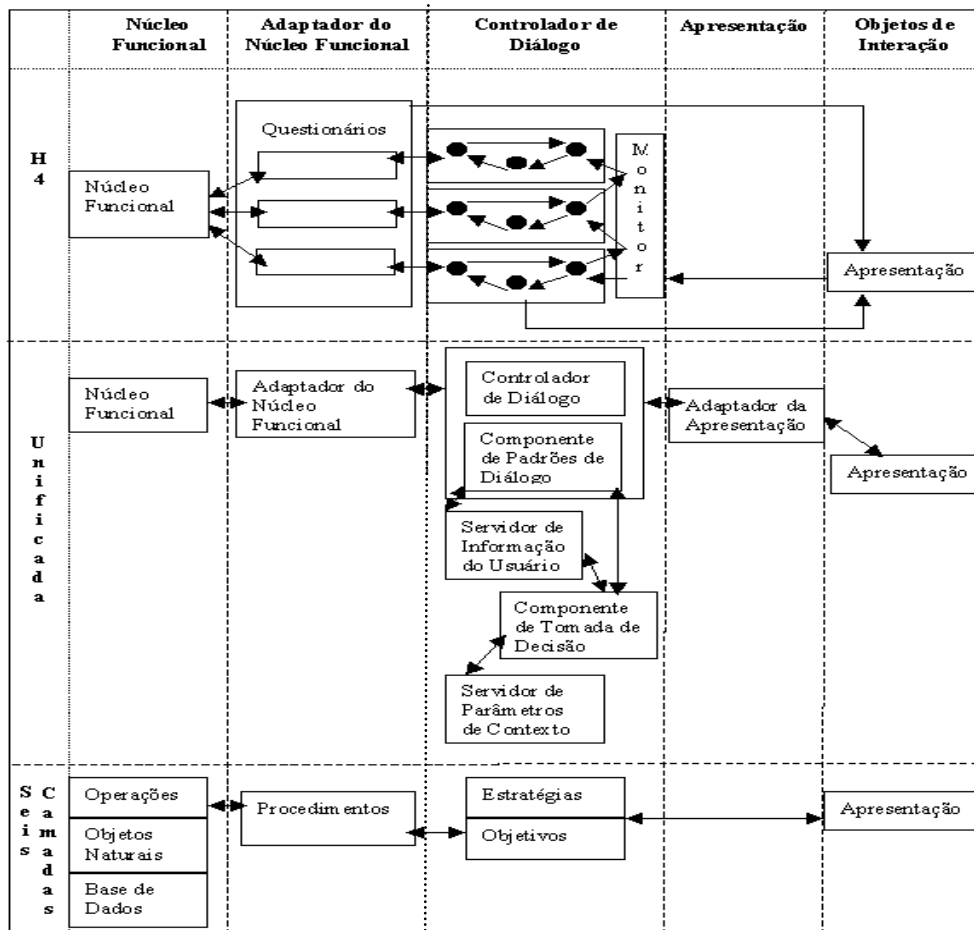


Figura 1.13 – Quadro comparativo entre as arquiteturas H4, Unificada e Seis Camadas

## **Conclusão**

Com a análise comparativa entre as arquiteturas observou-se que existem equivalências entre elas, e que tais equivalências podem ser generalizadas em relação a qualquer arquitetura ou modelo de arquitetura citados. A partir de cada equivalência pode-se identificar conceitos, que representem características comuns, que podem ser utilizados para a especificação de qualquer arquitetura ou modelo de arquitetura. Isto ocorre através dos relacionamentos existentes entre conceitos mais genéricos e conceitos mais específicos do modelo de arquitetura ou da arquitetura que se deseja especificar. Tal processo ocorre através da derivação dos últimos a partir dos primeiros, com aproveitamento de relacionamentos e atributos, o que originou as hipóteses 1 e 2 deste trabalho. No próximo capítulo, serão descritos e exemplificados métodos e sistemas de aprendizagem, e será feito um estudo comparativo entre os sistemas de aprendizagem.

## **2. Métodos e Sistemas de Aprendizagem**

### **Introdução**

A aprendizagem mediada por computador consiste na aquisição de conhecimentos através de métodos e dispositivos capazes de suportá-la [Furtado, 2001] [Reis, 2001]. Os métodos, geralmente baseados em teorias, delineiam como a aprendizagem ocorrerá. Os dispositivos são caracterizados pelos recursos, ferramentas ou meios utilizados para a efetivação da aprendizagem, definindo o que será necessário utilizar. Muitas vezes não existe uma delimitação muito clara entre os métodos e os dispositivos, algo que se pode observar em alguns sistemas de aprendizagem (dispositivos), que encerram em si métodos.

Neste capítulo, serão apresentados inicialmente métodos de aprendizagem e sua aplicação. Logo após, alguns sistemas de aprendizagem serão descritos, enfatizando como estão classificados em relação aos tipos de processos interativos de aprendizagem, suas características pedagógicas e ergonômicas.

### **2.1 Métodos de Aprendizagem**

Os métodos de aprendizagem são descrições sobre como ocorre o processo de aprendizagem do indivíduo. Alguns deles abordam a aprendizagem como sendo um processo no qual o agente responsável pelo seu desencadeamento e execução é o próprio indivíduo, e se detém a enfocá-lo como um processo individual. Outros já consideram a aprendizagem como um processo no qual o meio tem uma influência preponderante e geralmente engloba a participação de outros indivíduos, abordando-a como um processo coletivo. Também existem métodos que consideram o aluno como tendo um papel passivo no processo de aprendizagem, enquanto outros consideram-no como construtor do seu próprio aprendizado.

#### **2.1.1 Aprendizagem baseada em problemas**

A aprendizagem baseada em problemas consiste na exposição e contextualização de alunos a problemas próximos da realidade, numa abordagem interdisciplinar, aonde são incentivados debates entre os alunos [Fekete et al., 1998]. Neste método de aprendizagem, o

aluno constrói soluções para seus problemas, contando com o apoio de recursos que o auxiliem.

### **2.1.2 Aprendizagem através de perguntas e respostas**

Este método é efetivado através de questionamentos feitos a ou pelos alunos no intento de que os mesmos apresentem ou obtenham respostas. O aluno assume um papel (re)ativo em relação a estímulos provenientes do meio, sendo o autor dos estímulos o responsável pela construção da aprendizagem do aluno.

### **2.1.3 Aprendizagem através de imagens**

A aprendizagem através de imagens consiste na exposição dos alunos a imagens ou representações gráficas que encerrem em si as informações ou conhecimentos que devem ser transmitidos aos alunos de maneira contextualizada [Liebman, 1998] [McFadzean, 1999]. O aluno também pode construir imagens durante o processo de aprendizagem, o que o define como observador e/ou participante em relação ao processo.

### **2.1.4 Aprendizagem construtivista**

É através da realização de tarefas em seus respectivos contextos, que ocorre a aprendizagem construtivista. O aprendiz adota um objetivo e realiza tarefas para a consecução do mesmo, e a medida que comete erros, se questiona e procura soluções, formula e reformula hipóteses de como alcançar o objetivo pretendido [Corry, 1996] [Schank & Cleary, 1995]. Neste processo, o aluno constrói o seu próprio conhecimento através de experiências vivenciadas.

### **2.1.5 Aprendizagem colaborativa**

A aprendizagem colaborativa ocorre quando alunos aprendem através do compartilhamento de idéias e conhecimentos entre si [John-Steiner & Mahn, 1997]. Neste processo, a aprendizagem ocorre principalmente quando um aluno é auxiliado por outro mais experiente, ou então pelo próprio professor, na resolução de problemas que ele não conseguiria resolver sozinho. A aprendizagem também ocorre quando o aluno se apropria de recursos e meios, que encerram em si conhecimento humano e significado cultural, para



utilizá-los no alcance dos seus objetivos. Importantes trabalhos têm mostrado aprendizagem significativa em grupo através da utilização de ferramentas de comunicação (como fóruns e chats):

- Segundo MENEZES [Menezes et al., 2000], um curso de educação a distância para a formação de multiplicadores para o Programa Nacional de Informática Educativa foi considerado satisfatório. Este curso era formado por grupos cujos membros se comunicavam entre si através do uso de fóruns e e-mails com o intuito de construir sites sobre os seus respectivos assuntos de estudo e pesquisa e;
- De acordo com RAMOS [Ramos, 1996], o uso de fóruns proporciona a aprendizagem autônoma. Isto foi evidenciado no trabalho realizado com um grupo de 15 alunos sobre a compreensão dos mesmos em relação ao impacto social, político e pedagógico da disseminação do uso da tecnologia da telemática.

## **2.2 Sistemas de aprendizagem e estudos ergonômicos**

Em sistemas de aprendizagem pode-se identificar estruturas computacionais especialmente organizadas, que quando utilizadas facilitam a aprendizagem [Reis, 2001]. Neste trabalho, serão focalizados sistemas computacionais que interagem com o usuário.

A classificação de sistemas interativos utilizados na aprendizagem, no que diz respeito a seus processos interativos de aprendizagem, pode ser a seguinte:

- Meios de aprendizagem direta, nos quais os sistemas são os únicos responsáveis pela aprendizagem dos alunos;
- Meios de intermediação e suporte à aprendizagem, nos quais os sistemas são utilizados somente como um meio de apoio à aprendizagem, estando a responsabilidade pela aprendizagem dos alunos restrita a usuários dos sistemas e;
- Meios de aprendizagem direta com intermediação e suporte à aprendizagem, nos quais a responsabilidade pela aprendizagem dos alunos encontra-se distribuída entre usuários dos sistemas e o próprio sistema, sendo o sistema participador e meio de apoio à aprendizagem.

Também pode-se classificar os sistemas interativos quanto as suas características ergonômicas. Por características ergonômicas, entende-se como estudos, métodos e teorias realizados e/ou construídos no contexto humano e/ou tecnológico que trata de melhorar a interação humano-computador [Coutaz, 1993]. As características ergonômicas identificadas são as seguintes:

- A aplicação de arquiteturas e de modelos de arquitetura, consistindo na organização dos sistemas segundo um determinado padrão, que pode proporcionar vantagens ergonômicas, como a modularidade no caso do modelo de arquitetura PAC;
- A aplicação de metodologias, que são conjuntos de passos pré-definidos a serem executados para a obtenção de um sistema interativo com determinadas características, dentre as quais podemos citar a usabilidade;
- A aplicação e reuso de estruturas de projeto, que são características e conceitos comuns de modelagem, que geralmente são reusados durante o projeto e desenvolvimento de sistemas, que podem e devem ser usados como referência para a obtenção de sistemas ergonômicos;
- A aplicação de recomendações ergonômicas no desenvolvimento dos sistemas, consistindo no respeito a especificações bem definidas de como a interface deve se comportar e ser apresentada;
- Comunicabilidade, que consiste em facilitar o diálogo entre o sistema e o usuário, através do layout e dos objetos de interação da interface, que devem informar ao usuário a seqüência e os tipos de ações que ele pode efetuar para atingir os seus objetivos, além de deixar o usuário ciente do que está acontecendo;
- Aplicação das técnicas GOMS, acrônimo de *Goals, Operations, Methods and Selection rules* [John, 1996], úteis para analisar o conhecimento de como fazer uma tarefa em termos de objetivos, operadores, métodos e regras de seleção;

- Propostas de design [Borges et Baranauskas, 1998], caracterizadas pelo fato de que a usabilidade (facilidade de uso) da interface é aprimorada através da análise da interação do usuário e de suas opiniões, aonde é observada como a interface deve ser para facilitar o seu uso e;
- Teoria da ação [Norman, 1986], segundo a qual é interessante mapear as intenções do usuário na interface para que a mesma seja explícita, consistente e inteligível, proporcionando uma coerência conceitual com o modelo mental do usuário construído em relação ao sistema.

A seguir, serão apresentados alguns sistemas de aprendizagem e em seguida será feito um quadro comparativo entre eles com relação aos itens acima descritos.

A utilização do MOdelador de DEcisão de Interação Visual (VIDEMO) [Beroggi, 1999] como ferramenta computacional de modelagem analítica é de considerável valor como metodologia de ensino da Pesquisa Operacional no que tange a tomada de decisões. O usuário tem liberdade na escolha do problema a ser abordado e da abordagem de acordo com a sua percepção do mesmo, e ataca o problema através do uso da criatividade e intuição para o desenvolvimento de um modelo mental. Neste caso, o modelo mental é uma representação conceitual do problema através do uso de figuras geométricas e setas. Posteriormente, o modelo mental é representado em forma de modelos analíticos, representações formais a nível de equações e inequações, de tal forma a minimizar as perdas na transição. Após o desenvolvimento do modelo analítico, é desenvolvido o modelo de resolução, conjunto de passos para resolver o modelo analítico em meios computacionais. Através da experimentação do modelo e comparação com dados reais, ele será validado ou calibrado, podendo haver o reinício do processo de modelagem caso necessário.

Em Labidi et al. [2000] é apresentado um ambiente de aprendizagem denominado MATHNET, que faz uso de sistemas tutoriais inteligentes implementados sob a forma de sistemas multi-agentes numa rede de computadores, cuja função é promover a aprendizagem do aluno através de uma estratégia pedagógica de colaboração devidamente alocada ao mesmo. O sistema fornece instrução aos alunos e proporciona a comunicação entre eles através de chats e envio de mensagens, sendo respeitadas as características pedagógicas dos alunos

durante a apresentação de informações e controlada a comunicação entre eles durante a aprendizagem.

Através de Furtado et al. [2000] é apresentado um ambiente baseado em agentes para aprendizagem Colaborativa e Adaptativa em Didática (CADI) que auxilia no aprendizado do professor, em como agir em sala de aula, através da exposição do mesmo a casos semelhantes ao que ele está passando, identificando e disponibilizando problemas provenientes destes e suas possíveis soluções, ambos representados através de cenários em formato de imagens e textos. Também é disponibilizado um dispositivo de comunicação para a comunicação entre os professores e facilitada a busca de casos e cenários de interesse do professor através do agente Contador de Histórias. Este ambiente faz parte de um ambiente de educação a distância chamado Cadinet [Furtado et al., 2001].

Através do software educacional GENE [Borges & Oliveira, 2000], alunos do nível médio são auxiliados no aprendizado da genética. O ambiente tem como princípio pedagógico o construtivismo, onde a aprendizagem está baseada na descoberta e exploração, além do uso de recursos multimídia, simulação e animação que provaram, após uma série de testes e reconstruções da interface, serem imprescindíveis para o estímulo e realização da aprendizagem. Neste ambiente, o aluno manipula as informações genéticas de uma célula animal na fase embrionária e observa o resultado destas manipulações ao longo da vida do ser gerado a partir desta célula.

O ambiente multi-agente *moonline*, proposto em Gava et Menezes [2000], consiste num sistema de apoio à aprendizagem extra-classe, através do uso de recursos computacionais e da Internet, para a elucidação de perguntas e questionamentos dos alunos. Estas perguntas podem ser respondidas por monitores e professores, de tal forma que somente as perguntas não respondidas pelos monitores são respondidas pelos professores. A classificação e o encaminhamento de perguntas a monitores e professores de acordo com os seus perfis, ou a procura e apresentação de perguntas já respondidas similares a perguntas feitas por alunos, são outras características deste ambiente constituídas graças ao emprego de agentes.

O Ambiente para aprendizado em Programação LIneAR (AMPLIAR) [Bach et al., 2001], desenvolvido através de um projeto de interface para a obtenção de usabilidade e de

comunicabilidade, proporciona o aprendizado e o entendimento do método SIMPLEX pelo usuário através do emprego de questões sob a forma de interjeição. Além de proporcionar o uso de questionamentos, soluções são apresentadas de acordo com as interjeições utilizadas e contextualizadas num processo de suavização do aprendizado por antecipação e auxílio às possíveis dúvidas do aluno.

O ambiente para o Agente Pedagógico de Aprendizagem Colaborativa em Harmonização Ecológica (APACHE) [Reis, 2001] foi desenvolvido tendo como base a teoria de aprendizagem socio-cultural de Vygotsky, caracterizando-se como um ambiente social de aprendizagem. O processo de aprendizagem desenvolvido neste ambiente consiste no uso de agentes artificiais e humanos distribuídos para, através de interações (manipulações de ferramentas e imagens, argumentações, contestações, escrita e leitura) e recursos tais como a Web (chats e correios eletrônicos) e simulações, desenvolver na criança a noção de preservação, equilíbrio e desenvolvimento sustentável do meio ambiente e de seus recursos naturais de forma participativa.

A partir do quadro comparativo (ver figura 2.1) entre os sistemas de aprendizagem anteriormente descritos, observou-se a dificuldade em encontrar sistemas que apresentem vantagens suficientes para exaurir os possíveis recursos existentes a nível de processos interativos de aprendizagem e características pedagógicas. Isto se deve principalmente ao fato de que tais recursos podem ser utilizados através de um número indefinido de combinações, o que vai depender do público para o qual ele vai ser destinado e do objetivo a ser alcançado. No que concerne as características ergonômicas, poucos são os sistemas de aprendizagem que integram algum tipo de conceito de IHC para constituir sistemas de melhor usabilidade.

Além disto, observou-se que com a evolução dos processos interativos de aprendizagem, da pedagogia e da ergonomia, a atualização de um sistema de aprendizagem em relação aos recursos disponibilizados é muito difícil. Também observou-se uma escassez de notificação do uso de arquiteturas ou modelos de arquiteturas, metodologias e de estruturas de projetos (características comuns reusáveis) na obtenção de sistemas com características apropriadas. Isto evidencia uma sobrecarga de trabalho para o projetista desenvolver e evoluir sistemas de aprendizagem com qualidade. Hoje em dia, tenta-se minimizar este problema através da unificação de conceitos, padronização de modelos e processos, como RUP e UML,

a fim de não exigir do projetista um vasto conhecimento a nível de conceitos de modelagem. Outra tentativa de facilitar o trabalho do projetista, que é a proposta deste trabalho, é a definição de estruturas de projetos já existentes, a fim de que o mesmo possa usá-las para agilizar o processo e continuar garantindo a usabilidade e/ou outra vantagem ergonômica que já havia sido obtida e testada com o emprego de tais estruturas.

Sistema de Aprendizagem	VIDEMO	MATHNET	CADI	GENE	Moonline	AMPLIAR	APACHE
Tipo de processo interativo de aprendizagem	Aprendizagem direta	Meio de aprendizagem direta com intermediação e suporte ao aprendiz	Meio de aprendizagem direta com intermediação e suporte ao aprendiz	Aprendizagem direta	Meio de intermédio e suporte ao aprendiz	Aprendizagem direta	Meio de intermédio e suporte ao aprendiz
Características pedagógicas	Uso da aprendizagem construtivista e de aprendizagem através de imagens	Uso da aprendizagem colaborativa	Uso da aprendizagem construtivista, da aprendizagem colaborativa e da aprendizagem através de imagens	Uso da aprendizagem construtivista	Uso da aprendizagem através de perguntas e respostas	Uso da aprendizagem através de perguntas e respostas	Uso da aprendizagem colaborativa
Características ergonômicas	Não considera	Não considera	Não considera	Uso de propostas de design e teoria da ação	Não considera	Uso do GOMS e da teoria da comunicabilidade de para usabilidade	Não considera

*Figura 2.1 – Quadro comparativo de sistemas de aprendizagem*

## Conclusão

Através do estudo comparativo entre os sistemas de aprendizagem, observou-se que recursos são oferecidos em termos de processos interativos de aprendizagem e métodos pedagógicos, o que possibilita uma grande variedade de combinações da aplicação dos mesmos para o desenvolvimento de diferentes versões de sistemas de aprendizagem, cada uma destinada a um público específico para o alcance de objetivos determinados. Em quase todos eles, se ressalta a importância da aplicação de problemas práticos e da solução colaborativa dos mesmos. No entanto, pouco tem sido feito na aplicação de conceitos de IHC para melhorar as características ergonômicas dos sistemas de aprendizagem. Estas conclusões deram origem a terceira hipótese que fundamenta este trabalho. Também observou-se que estes recursos são constantemente atualizados de tal forma que, aproveitá-los adequadamente em sistemas de aprendizagem, requer uma constante evolução dos sistemas de aprendizagem.

Como apoio ao desenvolvimento e evolução dos sistemas de aprendizagem, deve-se reutilizar estruturas de projeto já concebidas. Neste contexto, o próximo capítulo irá descrever a solução encontrada para apoiar o projetista a construir e aplicar conceitos de modelagem, através da reutilização destas estruturas de projetos organizadas em diversos níveis de abstração.

### **3. Metodologia de apoio à modelagem e concepção de Sistemas Interativos**

#### **Introdução**

Através do estudo realizado no primeiro capítulo sobre processos de modelagem e concepção de sistemas observou-se diversas dificuldades que afligem o projetista na construção de modelos e de sistemas. Estas dificuldades serão reconsideradas na primeira seção deste capítulo, seguidas da definição dos objetivos e solução deste trabalho. No contexto dos sistemas de aprendizagem, observou-se a necessidade do desenvolvimento de um módulo de aprendizagem para a disciplina de Pesquisa Operacional, tendo em vista as dificuldades observadas no segundo capítulo. Estas dificuldades serão descritas novamente na terceira seção deste capítulo, seguidas da definição dos objetivos e solução propostas. Finalmente será descrita a metodologia proposta para apoiar a construção e aplicação de métodos e modelos.

#### **3.1 A problemática na especificação de sistemas interativos**

Após análise sobre processos de especificação de sistemas interativos descrita no primeiro capítulo, verificou-se a ausência de metodologias e/ou ferramentas capazes de auxiliarem o projetista a construir um sistema sem exigência de um conhecimento sobre os métodos e modelos suportados. Isto é devido ao fato de que para as ferramentas gerarem um sistema, o projetista deve usar somente os modelos que elas implementam. Algumas ferramentas CASE permitem que o projetista não use seus modelos facultativos, desde que use todos os modelos que são imprescindíveis. Mas o projetista não pode criar novos modelos a partir dos já existentes a fim de adaptar a sua realidade. Isto significa a inexistência de apoio, por parte das ferramentas, ao reuso de especificações utilizadas na definição de métodos e modelos (conceitos de modelagem). Tal característica é importante porque mantém uma padronização na especificação de conceitos, promovendo um maior controle nas especificações realizadas e proporcionando ganhos de qualidade e de produtividade. A especificação utilizada é aquela que melhor descreve um determinado conceito, reduzindo o trabalho de especificação em termos de quantidade e complexidade. Outro ponto importante verificado após a análise foi a dificuldade que o projetista tem para evoluir um SI que está em



uso. Isto se deve ao fato de existir a necessidade de compreender que diversos aspectos relacionados à interação do usuário evoluem, tais como: estereótipos de utilização dos dispositivos de hardware e software, critérios de utilização dos órgãos sensoriais e motores, formas de emprego do aparelho cognitivo e adaptabilidade ao contexto de uso. Além disto, os formalismos e procedimentos de desenvolvimento de aplicação utilizados na obtenção de sistemas, devem ser alterados de tal forma a incorporar ganhos advindos de novas formas de interação e práticas de desenvolvimento (atividades e hábitos produtivos).

### **3.2 Objetivos e soluções para facilitar a especificação de sistemas interativos**

Os problemas mencionados acima dizem respeito principalmente ao projetista, do qual é exigido um conhecimento para usar as ferramentas existentes e com pouca assistência para reusar trabalhos já realizados. Para facilitar o trabalho de especificação de sistemas interativos, este trabalho visa:

- Ajudar o projetista a elaborar seu próprio método e/ou modelos de desenvolvimento de software ou a escolher aquele(s) que seja(m) mais adequado(s) à situação desejada. A metodologia proposta na próxima seção apóia este trabalho através do fornecimento de procedimentos que descrevem como elaborar e escolher métodos e modelos;
- Melhorar o suporte à documentação, manutenção e evolução de sistemas interativos através de procedimentos que foram elaborados neste trabalho para tornarem métodos e modelos mais fáceis de serem compreendidos. Este trabalho tem como resultado a especificação de todo um sistema interativo, auxiliando na sua documentação e conseqüente manutenção e;
- Permitir que o projetista obtenha experiências de modelagem de seus colegas através do reuso de especificações de métodos e de modelos. Tais especificações são representadas através de conceitos de modelagem, que são organizados em diversos níveis de abstração hierárquicos. Esta organização possibilita facilmente o reuso das especificações através da herança entre conceitos.

### **3.3 Problemática na aprendizagem de Pesquisa Operacional**

O abstracionismo matemático de Pesquisa Operacional é responsável pelas dificuldades de aprendizagem dos alunos em relação a esta ciência [Ingolfsson, 1999]. Para resolver estes problemas, o capítulo 2 mostrou que foram criadas estratégias e dispositivos de ensino e aprendizagem. Dentre as estratégias e dispositivos de ensino e aprendizagem, observou-se o emprego de sistemas de aprendizagem, tais como VIDEMO [Beroggi, 1999] e AMPLIAR [Bach et al., 2001]. Neste capítulo também identificou-se diversas limitações em tais sistemas, dentre as quais pode-se citar: ausência de suporte à aprendizagem colaborativa na resolução de problemas contextualizados e inexistência de uma metodologia de suporte ao desenvolvimento dos mesmos, que considerasse fatores ergonômicos de Interação Humano-Computador (IHC), a fim de melhorar a usabilidade dos sistemas.

### **3.4 Objetivos e solução para facilitar a aprendizagem de Pesquisa Operacional**

Para facilitar a aprendizagem de Pesquisa Operacional, este trabalho visa encontrar uma solução para estimular a reflexão do aluno através da resolução de problemas práticos. Esta solução consiste em definir e desenvolver um módulo de aprendizagem em Pesquisa Operacional que promova a aprendizagem do aluno através da resolução colaborativa e explicação de exercícios de maneira contextualizada.

Outra solução foi a de desenvolver um módulo de forma bem estruturada, seguindo por exemplo um modelo de arquitetura. Acredita-se que uma possível solução para isto é dar condições para o projetista seguir um método que o auxilie na modelagem das informações e a verificar onde bem integrar conceitos ergonômicos no processo de desenvolvimento de um software.

### **3.5 Ilustração das soluções elaboradas**

A figura 3.1 ilustra uma visão simplificada das soluções elaboradas neste trabalho, através da metodologia de apoio à construção e à aplicação de recursos de software necessários ao desenvolvimento e documentação de um módulo de aprendizagem de Pesquisa Operacional. Esta metodologia, chamada META, foi definida para apoiar o trabalho do

projetista na elaboração de um novo método de desenvolvimento de sistemas ergonômicos (chamado MAPEAR). Este método foi aplicado para o desenvolvimento do módulo de aprendizagem de Pesquisa Operacional. A idéia é que este módulo seja integrado ao sistema de educação a distância, chamado CADINET, o qual seria usado pelo aluno durante a realização de um curso em Pesquisa Operacional.

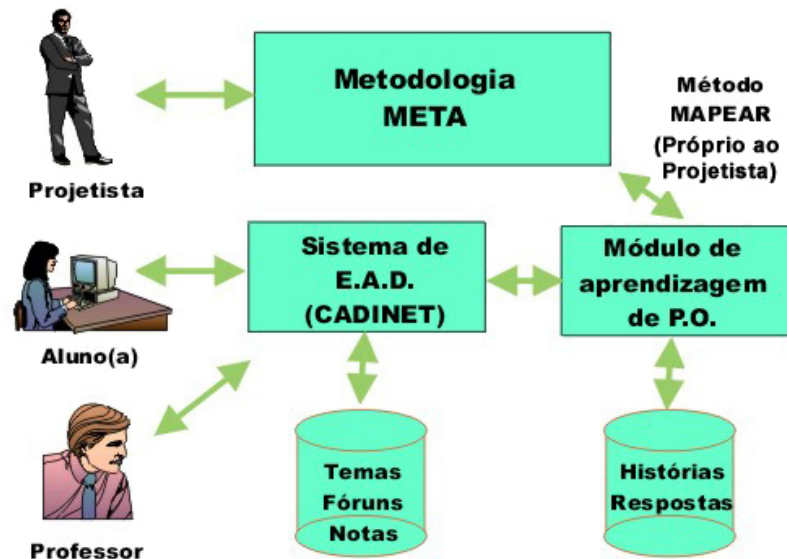


Figura 3.1 – Visão simplificada das soluções integradas

Ao interagir com o CADINET, conforme está ilustrado na figura 3.2, o aluno tem acesso aos temas do curso em Pesquisa Operacional e ao material didático disponível, como exercícios.

Usuários On-Line no CADInet

Início > Sala de Aula > Unidades

Aluno UFC Sair do Sistema

Disciplina : Pesquisa Operacional

Unidades	Temas da unidade
<a href="#">Unidade</a>	<a href="#">Programação dinâmica</a> <a href="#">Programação inteira</a> <a href="#">Programação linear</a>

[ UNIFOR - Universidade de Fortaleza ]

Figura 3.2 – Apresentação dos temas disponíveis em relação a uma unidade do curso em Pesquisa Operacional

O aluno escolhe um tema e navega para a tela de descrição do mesmo, ilustrada na figura 3.3. Ao clicar em “Trabalho de RPG”, o aluno deve resolver o trabalho e enviar a solução para o professor através do sistema ao clicar em “Enviar resposta”.

The screenshot shows the CADInet web interface. At the top, there is a navigation bar with the CADInet logo and the text 'CADInet'. Below the navigation bar, there is a breadcrumb trail: 'Início > Sala de aula > Unidades > Tema > Trabalho do tema >'. The user's name 'Fernanda Goersch Fontenele' and a 'Sair do Sistema' (Log out) button are visible. The main content area displays the following information:

<b>Disciplina: Pesquisa Operacional</b>	
Tema:	Programação Inteira
Trabalho:	Trabalho de RPG
Descrição:	Este trabalho tem como objetivo de avaliar e despertar o interesse do aluno em programação inteira de maneira construtiva e colaborativa, através da utilização de RPG's.
Nota:	Este trabalho vale nota
Registrar resposta:	<a href="#">Enviar resposta</a>
Prazo de entrega:	01/09/01

Below the table, there is a link 'Trabalho de RPG' and a button 'Arquivo com a descrição do trabalho'. At the bottom of the page, there is a footer: '[ UNIFOR - Universidade de Fortaleza ]'.

Figura 3.3 – Apresentação do tema escolhido e das operações relacionadas a ele

Para resolver o trabalho, o aluno poderá acessar o módulo de aprendizagem, cujo protótipo está ilustrado na figura 3.4, para participar de histórias através da escolha de respostas e explicação das mesmas. Para tanto, o aluno lê um trecho da história, apresentada na caixa de texto superior da janela, e observa as perguntas e o cenário a ela relacionados. Para a pergunta existente em cada trecho são apresentadas alternativas numeradas de resposta na caixa de texto inferior do lado esquerdo da tela, na área de resposta. O aluno então resolve o problema, após a sua modelagem e resolução em um software apropriado, e descreve o processo de modelagem, o modelo e a interpretação dos resultados de sua solução na caixa de texto inferior do lado direito da tela, na área de explicação. Após a explicação, o aluno seleciona o item da resposta correta nos botões de seleção e clica em avançar para que o próximo trecho da história seja apresentado. Após avançar, o aluno também poderá voltar a qualquer trecho já visitado para alterar a resposta fornecida e, conseqüentemente, mudar o seu percurso na história. Durante a resolução do exercício, o aluno poderá usar as ferramentas de comunicação do CADINET para interagir com outros alunos (como o fórum). Após a solução

do exercício, o aluno poderá enviar suas respostas para o ambiente e, depois de avaliadas pelo professor, o aluno poderá consultar suas notas.

The screenshot shows the CADInet POPAI interface. At the top, there is a navigation bar with the CADInet logo and the text 'POPAl'. Below this, a breadcrumb trail reads: 'Início > Sala de Aula > Unidades > Tema > Trabalho de Tema > Resolução de RPG'. A status bar indicates 'Usuários On-Line no CADInet'. The main content area is titled 'Aluno UNIFOR' and includes a 'Sair do Sistema' button. The scenario text reads: 'Você é um caixeiro viajante e está no começo de uma longa jornada. Deixará a sua família por algum tempo para fazer a sua rota de trabalho. Como seus recursos financeiros são limitados, e você não deseja passar muito tempo longe de sua família, você deve escolher a rota mais curta, passando pelas cidades pré-estabelecidas (ver cenário gráfico). Qual é a rota que deve ser feita para que você retorne num espaço de tempo tal que a sua família não passe necessidade?'. Below the scenario is a table with two columns: 'Resposta' and 'Explicação'. The 'Resposta' column lists three possible routes. The 'Explicação' column provides a model and a calculation. At the bottom, there are radio buttons for 'Ítem 1' through 'Ítem 5', and buttons for 'Voltar', 'Avançar', and 'Fórum'.

Resposta	Explicação
1) cidadeA-cidadeB, cidadeB-cidadeD, cidadeD-cidadeC, cidadeC-cidadeE, cidadeE-cidadeA	Modelo:
2) cidadeA-cidadeC, cidadeC-cidadeD, cidadeD-cidadeB, cidadeB-cidadeE, cidadeE-cidadeA	Min 702 YAB + 454YAC + 842YAD + 554YAE + 324YBC + 1093YBD + 294YBE + 1137YCD + 338YCE + 799YDE
3) cidadeA-cidadeD, cidadeD-cidadeE, cidadeE-cidadeC, cidadeC-cidadeB, cidadeB-cidadeA	

Figura 3.4 – Resolução e explicação de problemas contextualizados

Como já foi dito, este módulo foi modelado graças à aplicação da metodologia META, que será explicada a seguir.

## 3.6 A metodologia META

### 3.6.1 O processo de construção e aplicação de recursos de software

A metodologia META descreve um processo de construção e aplicação de todos os recursos necessários ao desenvolvimento e documentação de sistemas interativos, tais como métodos e modelos. Este processo ocorre em dois níveis:

- (Re)Construção, nível em que os recursos necessários ao desenvolvimento e à documentação de um SI são construídos. A construção de um método de

desenvolvimento de sistemas consiste em definir seus atributos, tais como: ciclo de vida, fases, modelos e domínio. A construção de modelos de desenvolvimento de sistemas consiste em definir seus atributos, como por exemplo: valores, objetivos e aplicações. Neste processo, ocorre também a reconstrução dos recursos, quando o SI gerado deve evoluir e;

- *Aplicação*, nível em que os recursos de software construídos são instanciados e aplicados a uma área específica de aplicação. Instanciar um recurso significa atribuir valores específicos aos seus atributos, bem como definir os seus relacionamentos. Neste processo de aplicação, as fases de desenvolvimento de um sistema interativo são realizadas quando o método que o descreve estiver sendo aplicado (instanciado). Assim, um SI é desenvolvido e documentado.

### **3.6.2 Conceitos de modelagem**

A representação dos recursos de software é feita através de uma hierarquia de conceitos (ver figura 3.5):

- Meta-conceitos, que são especificações que definem um conjunto de conceitos (métodos ou modelos), obtidas através da abstração de características comuns de métodos e modelos já existentes na literatura. Tais especificações devem ser abrangentes o suficiente para facilitar o reuso das mesmas na definição de conceitos, mas delimitadas ao ponto de não perderem consistência;
- Conceitos, que são especificações que definem recursos utilizados para representar determinadas características e aspectos de um SI e/ou do seu desenvolvimento (como definições de métodos e modelos) e;
- Instâncias de conceitos, que são aplicações de conceitos na identificação e concretização de um determinado SI.

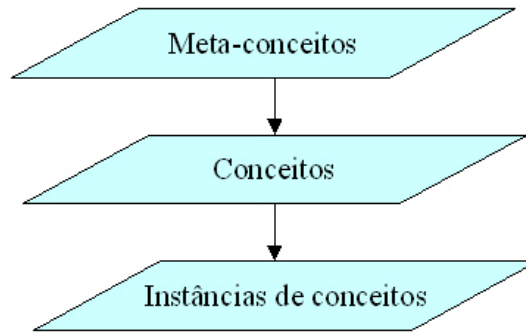


Figura 3.5 – Conceitos de modelagem

### 3.6.3 Fases da metodologia META

A figura 3.6 ilustra os processos de (re)construção e aplicação de recursos de software descritos pela metodologia META, e que abrangem 4 fases: inicial, de definição, de instanciação e de utilização e evolução de um SI. Cada fase é caracterizada como detentora de um objetivo a ser alcançado através de um processo e realizado por um determinado ator. Durante a realização de uma fase, pode-se usar ou não recursos obtidos na fase anterior e produzir produtos finais que podem ser meta-conceitos, recursos de software, instâncias de modelos ou sistemas interativos.

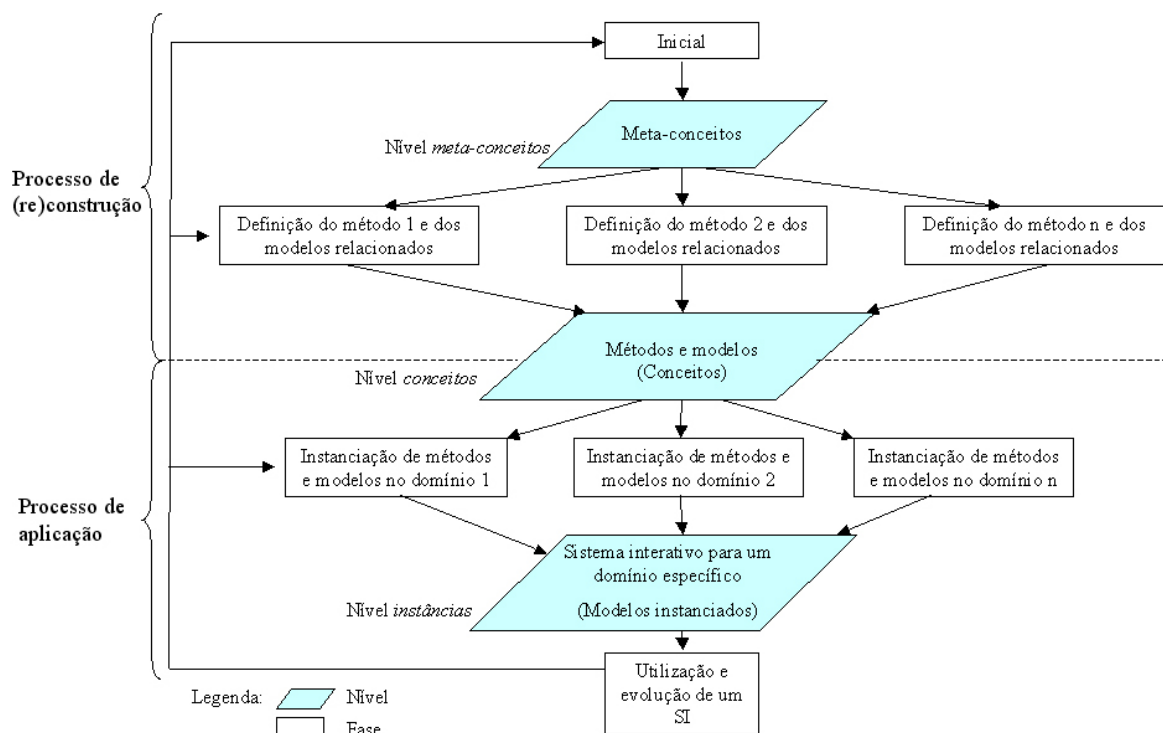


Figura 3.6 – A metodologia META

De uma forma geral, para explicar a figura 3.6, pode-se dizer que é possível definir os métodos e/ou modelos (conceitos) de desenvolvimento de SI a partir de meta-conceitos de métodos e/ou modelos, respectivamente. Em seguida, é possível instanciar os modelos criados, gerando a documentação de um SI para um certo domínio de aplicação. A evolução de um SI desenvolvido pode requerer a redefinição de seu método de desenvolvimento, de seus modelos e/ou até mesmo dos meta-conceitos utilizados. De uma forma mais específica, sugere-se a leitura de cada uma das fases descritas a seguir.

### **3.6.3.1 Fase inicial**

A fase inicial consiste na (re)construção e/ou análise de meta-conceitos a partir de características comuns existentes em métodos e modelos. A (re)construção de meta-conceitos é feita por um projetista (ator) interessado em especificar aspectos relativos à interatividade do usuário ou às características estruturais e funcionais das aplicações.

Durante a construção de um meta-conceito, ele abstrai e formaliza características representativas de um determinado grupo conceitual (como: métodos, modelos de arquiteturas ou modelos de tarefas), gerando especificações conceituais destas características para este meta-conceito. Por exemplo, considere que cinco métodos de desenvolvimento de sistemas têm em comum o fato de serem executados através de fases e de possuírem objetivos. Estas características comuns podem ser utilizadas para a especificação do meta-conceito *método* que conterà os atributos *fase* e *modelos* associados. Este meta-conceito poderá ser usado na especificação de um novo conceito (no caso, um novo método).

### **3.6.3.2 Fase de definição**

Nesta fase, um projetista define métodos e/ou modelos que representem ou apoiem o desenvolvimento de um sistema a partir de meta-conceitos. A definição de métodos pode ser feita utilizando-se especificações de meta-conceitos de métodos para a geração do arquétipo da sua estrutura conceitual e/ou os próprios métodos já elaborados. Tal arquétipo servirá como base para a definição do método através da definição dos conceitos (como: fases e modelos) que fazem parte de sua estrutura. Na figura 3.7, o método 1 é definido contendo N fases e M modelos associados.



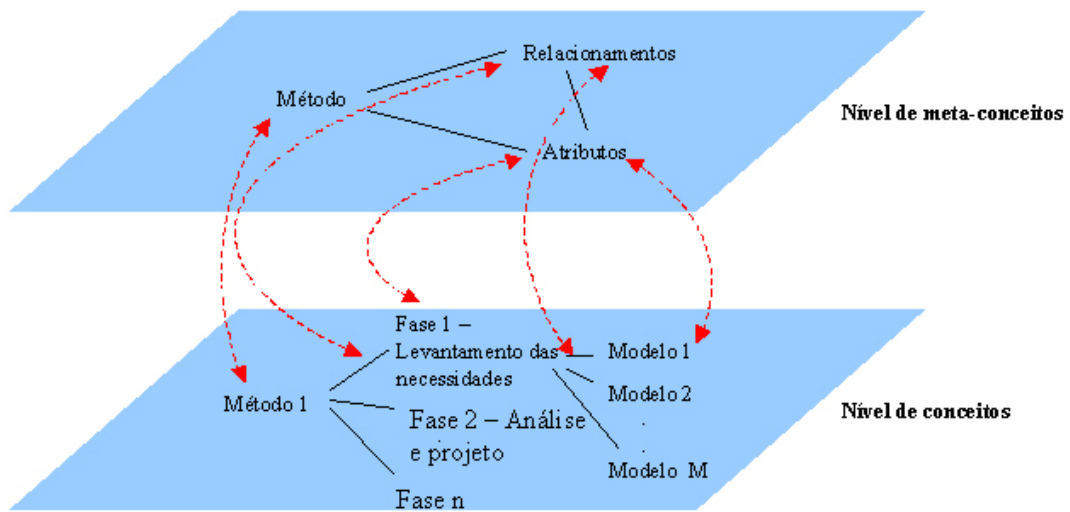


Figura 3.7 – Definição de um método de desenvolvimento de sistemas interativos

A definição dos modelos pode ser feita utilizando especificações de meta-conceitos de modelos e/ou os próprios modelos já elaborados. Por exemplo, a figura 3.8 ilustra que os atributos (camadas e objetos) constituem o modelo de arquitetura X e foram gerados a partir dos atributos do meta-conceito *modelo de arquitetura*, de tal forma a aproveitar características deles na constituição de suas próprias especificações.

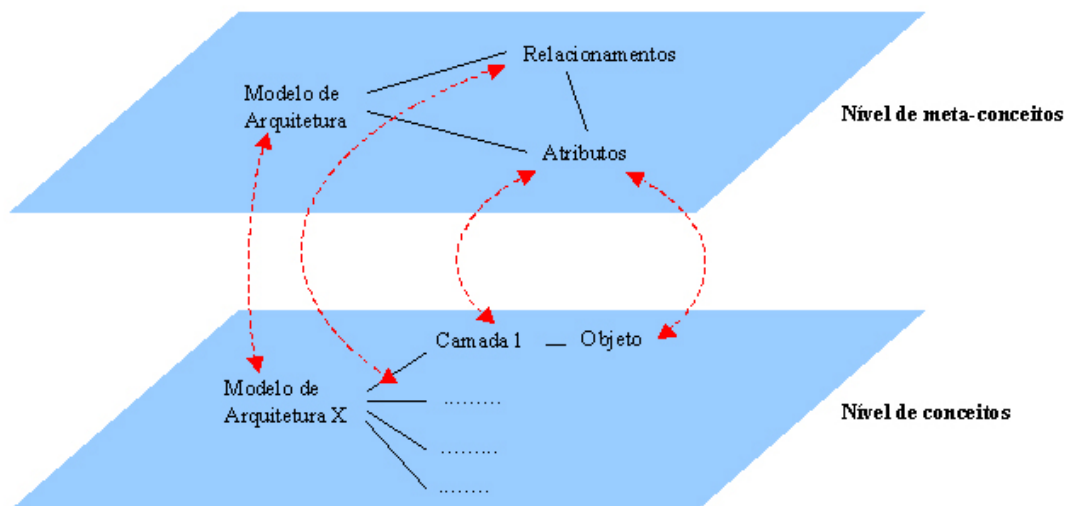


Figura 3.8 – Definição de um modelo de arquitetura

Os resultados desta fase são os seguintes:

- Definição de um método de desenvolvimento de sistemas e/ou;

- Definição de modelos para modelar e documentar um sistema interativo.

### 3.6.3.3 Fase de instanciação

Durante esta fase, o projetista analisa as definições de métodos e modelos, e escolhe aqueles que forem mais adequados para serem aplicados em uma determinada área de domínio. A aplicação dos métodos e/ou modelos escolhidos ocorre através da utilização de suas especificações na derivação de artefatos. Tais artefatos são necessários para representar e tornar compreensível tanto as características de um determinado sistema (instâncias de modelos), quanto a dinâmica e o encadeamento dos processos utilizados durante a modelagem do mesmo (instâncias de métodos).

A figura 3.9 apresenta a instanciação do método 1, ao ser aplicado ao domínio de ensino de Pesquisa Operacional, através da restrição e especialização do uso de suas fases. Isto é necessário para o mapeamento apropriado das sutilezas intrínsecas ao domínio do problema, que servirá para balizar a construção de abordagens adequadas para o tratamento do mesmo.

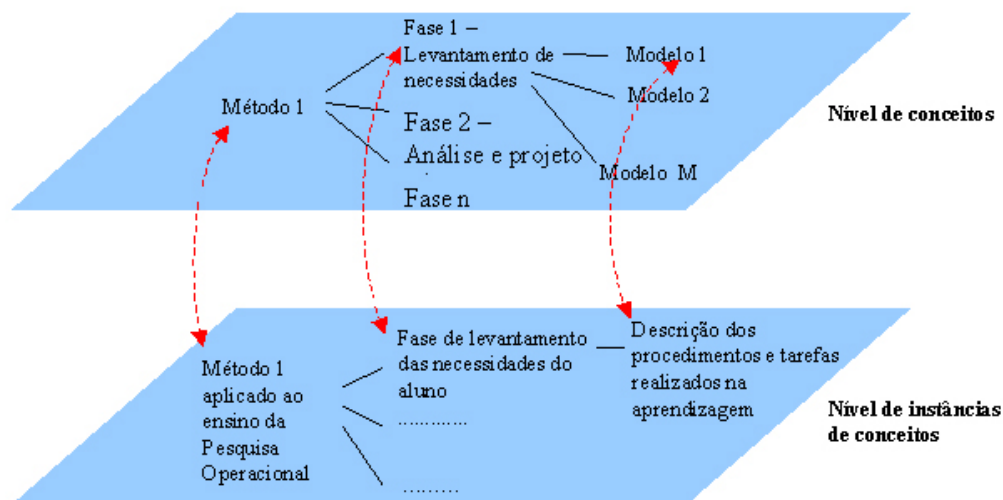
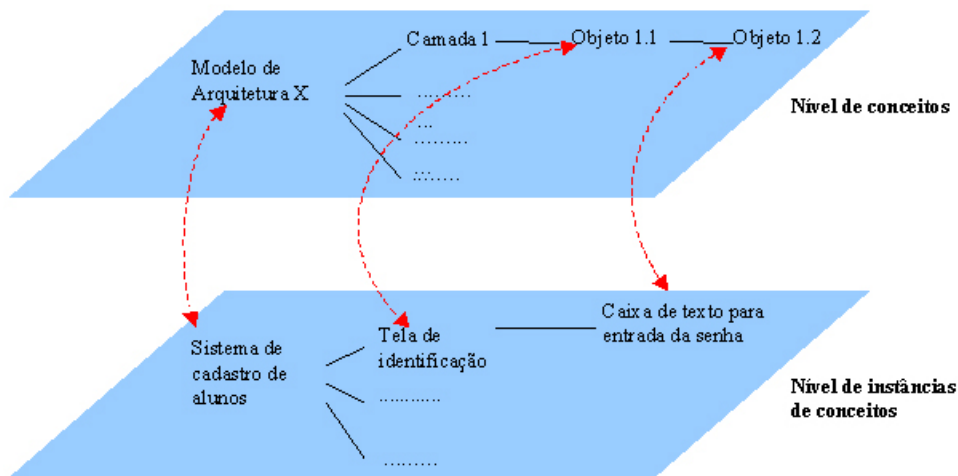


Figura 3.9 – Instanciação do método 1 para o domínio de ensino de PO

A figura 3.10 ilustra a instanciação de um modelo de arquitetura em uma representação da estrutura organizacional de implementação de um sistema de cadastro de alunos. Isto ocorre através da modelagem dos elementos que constituem o sistema, de tal forma que eles obedeçam aos critérios estabelecidos pela especificação dos conceitos a partir dos quais foram gerados. No caso ilustrado, tanto a tela de identificação quanto a caixa de

texto para entrada da senha contida nela devem se portar de acordo com os respectivos objetos pertencentes a camada 1 do modelo de arquitetura X.



*Figura 3.10 – Instanciação de modelos*

Em relação à instanciação de métodos e modelos, deve-se ainda observar que a escolha de especificações conceituais, adequadas para aplicação no domínio do problema, nem sempre é possível. Se não for, o projetista deve voltar para a fase anterior para definir métodos e/ou modelos mais apropriados, o que pode implicar na (re)definição de meta-conceitos.

#### **3.6.3.4 Fase de evolução**

Esta fase é realizada somente quando há necessidade de se realizar reajustes no sistema interativo em utilização. Estes reajustes são provenientes do acompanhamento da evolução das necessidades e capacidades dos usuários, individuais e em grupo. Nesta fase, o projetista deve entrar em interação com os usuários e verificar que fatores devem ser modificados. Após discussão com sua equipe, ele deve voltar às fases anteriores para realizar ajustes necessários nos componentes de modelagem, incorporando fatores e características observados na prática do desenvolvimento de sistemas. A modificação e/ou construção de componentes de modelagem proporciona uma (re)construção dos conceitos através de refinamentos sucessivos nos diversos níveis de abstração.

### **3.6.4 Vantagens da proposta**

Através das fases anteriormente descritas, é importante notar as vantagens que fundamentam a metodologia META:

- O projetista pode desenvolver e documentar um sistema de acordo com seus conhecimentos de modelagem, ao usar os recursos de modelagem que ele definiu;
- O projetista pode aproveitar os modelos, métodos e características comuns dos modelos e métodos já existentes para construir seus próprios modelos e métodos;
- O uso de características comuns (meta-conceitos) a métodos ou modelos em suas definições facilita a especificação destes conceitos. Isto ocorre porque o reconhecimento dos meta-conceitos a serem utilizados se torna evidente pela sua abrangência de uso e a especificação utilizada na definição dos meta-conceitos pode ser reusada na especificação dos recursos de software e;
- O processo não impõe seqüência rígida de passos. O projetista não precisa passar sempre pela construção. Se ele for diretamente aplicar os recursos de software significa que ele irá reusar especificações conceituais de recursos feitos por outros projetistas ou continuar a aplicar os recursos de software que ele já definiu.

### **3.6.5 Utilização da metodologia META: Integrando os níveis**

Nesta seção, serão ilustrados os processos integrados da metodologia META para construir o método MAPEAR, que apoiou a modelagem do módulo de aprendizagem de Pesquisa Operacional.

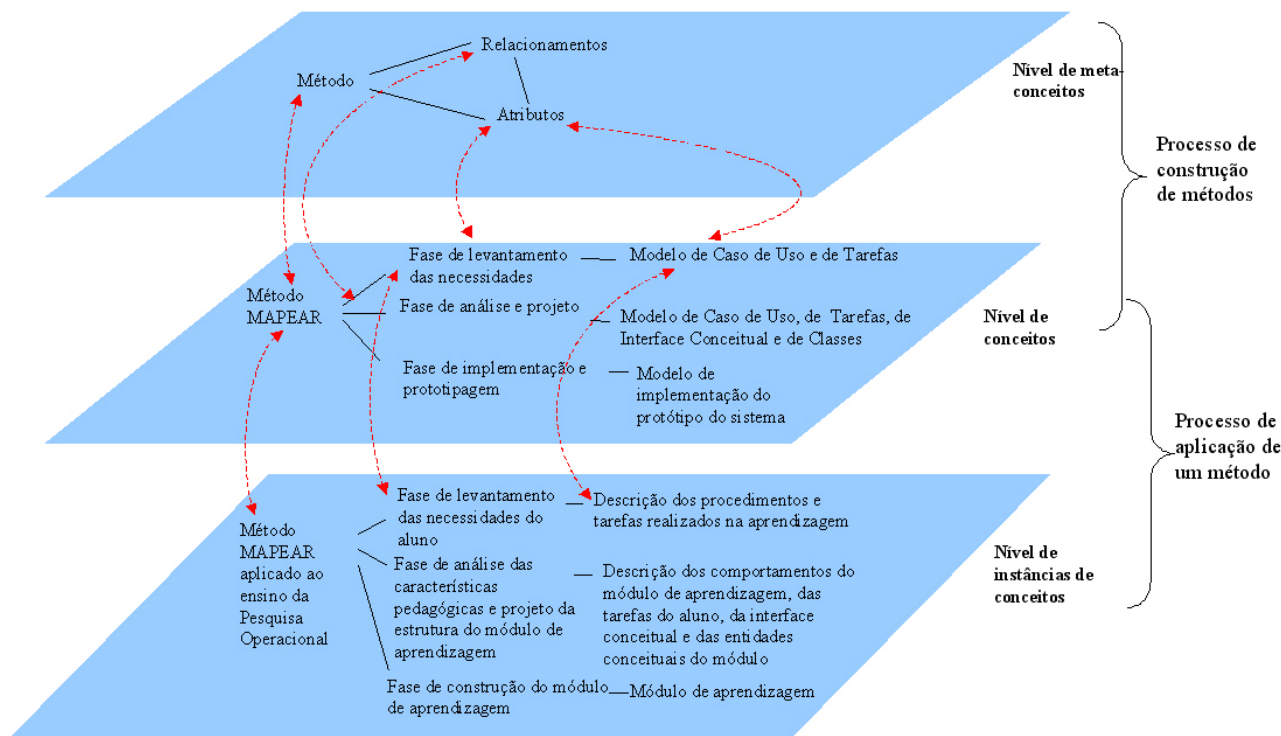


Figura 3.11 – Geração e aplicação do método MAPEAR

Como já foi mostrado até agora, o projetista pode aproveitar a definição conceitual de métodos de desenvolvimento de sistemas interativos, seus relacionamentos e atributos, para especificar um novo método de desenvolvimento, conceituando seu ciclo de vida, suas fases e seus modelos, para em seguida instanciá-lo em uma determinada área de aplicação. Na figura 3.11, o método de desenvolvimento de sistemas (MAPEAR) é especificado a partir do meta-conceito *método*, que será descrito no próximo capítulo, sendo especificadas suas fases e os modelos utilizados em cada fase. Estando o método especificado conceitualmente, o projetista pode aplicá-lo a um domínio a partir da instanciação dos modelos necessários em cada fase. Através da aplicação do método, serão produzidas descrições relacionadas ao domínio do problema, que auxiliarão na identificação e descrição da estrutura e do comportamento do sistema. Com a descrição estrutural e comportamental do sistema, o projetista estará apto a implementá-lo de acordo com o modelo de implementação (diagrama de classes em relação à perspectiva de implementação) do protótipo.

A construção e aplicação dos modelos são feitas também seguindo o mesmo processo que fundamenta a metodologia META. De acordo com a figura 3.12, atributos (características comuns a modelos de arquitetura) são utilizados para especificar as camadas de um modelo de

arquitetura (três camadas). O modelo é então instanciado, no exemplo, na arquitetura de um sistema de cadastro de alunos.

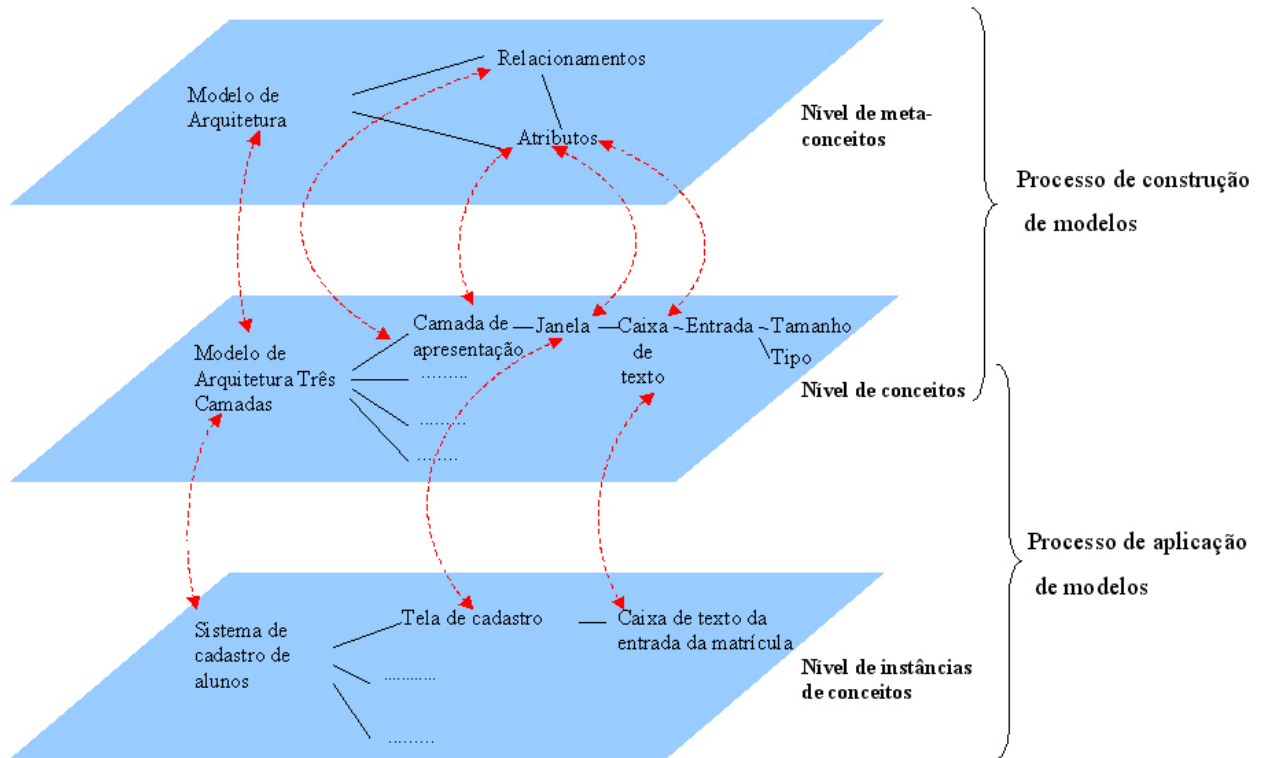


Figura 3.12 – Geração e aplicação de modelos de arquitetura

Através da observação dos processos de construção e aplicação de métodos e modelos, constata-se que tais recursos podem ser definidos de duas formas:

- A partir de meta-conceitos, através de um relacionamento de herança ou composição em que características e comportamentos mais genéricos são utilizados para fundamentar características e comportamentos mais específicos. Por exemplo, um meta-conceito sobre padrões de arquitetura pode ter como característica o fato de receber eventos provenientes da interface do usuário e transformá-los em informações passíveis de serem tratadas pelo computador. Tal característica pode ser então herdada da camada de *apresentação* do modelo de arquitetura *três camadas*, que tem como uma de suas características o fato adicional de definir restrições quanto ao formato e ao tamanho das entradas de dados e;

- A partir de outros conceitos, através de relacionamentos capazes de identificarem a sua constituição e funcionalidade. Por exemplo, o conceito *camada de apresentação* do modelo de arquitetura *três camadas* está relacionado com o conceito *campo de entrada* através do relacionamento *define*, identificando uma das funcionalidades desta camada do modelo.

Observa-se então que um importante aspecto a ser considerado no processo de definição de recursos durante a construção de métodos e modelos, diz respeito aos tipos de relacionamentos entre os conceitos utilizados nas suas definições (anexo A).

## **Conclusão**

A metodologia META proporciona a construção e aplicação de recursos a serem utilizados no desenvolvimento e documentação de sistemas interativos. Ela se utiliza de um processo de especificação incremental, que parte da especificação de conceitos mais abrangentes para a utilização destes na especificação e/ou constituição de conceitos mais específicos. Isto permite que não hajam prejuízos para a compreensão dos mesmos, uma vez que é possível navegar entre conceitos de diferentes níveis de abrangência para a compreensão deles. A navegação permite que o projetista focalize seus esforços no entendimento de aspectos e características de entidades que compõem a estrutura do sistema, a um nível de detalhes e sobre a perspectiva que sejam do seu interesse. Assim foi possível criar o método desejado (MAPEAR), para suprir as deficiências de modelagem: sistemas sem estrutura organizacional (desconsideração a modelos de arquitetura) e sem considerar recomendações ergonômicas. É o que será mostrado a seguir.

## **4. Conceitos de modelagem utilizados no processo de desenvolvimento de sistemas**

### **Introdução**

Em processos de desenvolvimento de sistemas interativos, pode-se identificar conceitos de métodos e de modelos. Geralmente eles são utilizados implicitamente durante a modelagem e implementação de sistemas. Porém, um maior potencial de uso deles se obtém com a representação explícita de suas características e funcionalidades. Isto se deve ao fato de que é pela observação dos detalhes que definem estes conceitos, que se adquire subsídios para compreendê-los e utilizá-los apropriadamente. Então é necessário que as definições de tais conceitos sejam adequadamente representadas em estruturas que facilitem o entendimento e a aplicação dos mesmos. Para tanto foram definidos os conceitos de modelagem e construídos alguns exemplares dos mesmos, cada qual constituindo uma significativa contribuição deste trabalho para apoiar a modelagem de sistemas interativos.

Neste capítulo, serão apresentados os diferentes conceitos de modelagem, suas formas de representação e seus relacionamentos, mostrando em seguida a aplicação dos mesmos na construção de um modelo de arquitetura. Também serão construídos conceitos de modelagem para auxiliarem nos processos de modelagem deste e de outros trabalhos.

### **4.1 Definição de conceitos de modelagem**

Um conceito de modelagem é definido através de seus atributos e relacionamentos, e pode ser apresentado através de várias formas de representação.

Os atributos são características que identificam uma entidade conceitual, como por exemplo os atributos *tipo* e *tamanho* para a entidade conceitual *campo de entrada*. Eles também podem ser descrições de outros aspectos de um conceito, como por exemplo os papéis desempenhados por ele e o(s) problema(s) que ele propõe resolver.

Os relacionamentos são decorrentes dos diferentes tipos de relações que uma entidade conceitual tem com outras, como por exemplo composição, herança, navegação e



colaboração. A composição representa as relações entre as partes e o todo. A herança identifica um conceito a partir de outro(s). A navegação ou associação unidirecional [Furtado, 1997] permite o acesso a um conceito a partir de outro(s). A colaboração define os papéis de cada um dos conceitos envolvidos numa determinada atividade. Além destes relacionamentos apresentados, outros podem ser criados e definidos de acordo com as necessidades de especificação de modelos (ver anexo A).

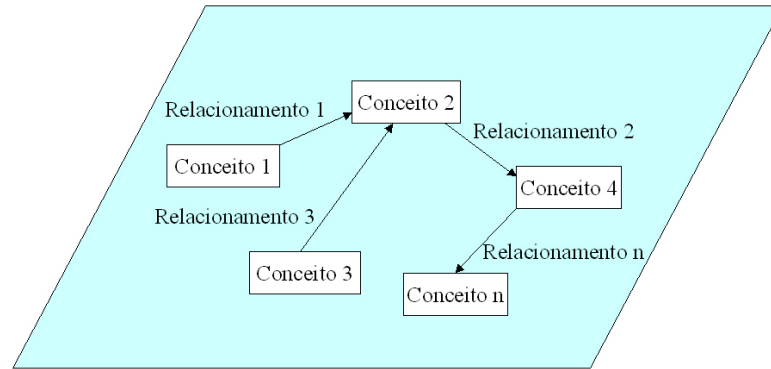
Antes da apresentação detalhada de conceitos de modelagem e de exemplos do seu uso, será descrita uma breve explanação sobre suas características.

## **4.2 Formas de representação de conceitos de modelagem**

Como já foi dito anteriormente, os conceitos de modelagem podem ser apresentados de diversas formas. No entanto, é necessária a definição dos tipos de representação que podem ser utilizadas na apresentação de conceitos.

Os meta-conceitos e conceitos apresentam as seguintes formas de representação:

- Gráfica, que é feita através de redes semânticas devido à facilidade de compreensão e à abrangência de representação da mesma, além deste tipo de formalismo permitir a realização de inferências lógicas. As inferências lógicas são operações realizadas sobre argumentos ou conjuntos de argumentos, de tal forma a se testar a consistência dos mesmos. Isto é bastante útil para se avaliar o grau de confiabilidade de um conceito e, conseqüentemente, a qualidade do mesmo. Em suma, redes semânticas são grafos cujos nós representam entidades conceituais e cujos arcos rotulados representam relacionamentos entre estas entidades. Foi adotada neste texto a representação da rede dentro de um losango, conforme ilustrado na figura 4.1, para representar o nível de abstração em que o conceito está definido.



*Figura 4.1 – Representação gráfica dos conceitos de modelagem em um certo nível de abstração*

- Textual, que é constituída das seguintes informações (ver formulário da figura 4.2):
  - ♦ Problema, que especifica a situação para a qual o meta-conceito ou conceito foi desenvolvido;
  - ♦ Tipo de solução, que descreve a intenção de um meta-conceito ou conceito;
  - ♦ Funcionalidade, que descreve os diferentes papéis que ele deve assumir de acordo com eventos ocorridos.

Nome do conceito de modelagem Problema: ..... Solução: ..... Funções: .....
-----------------------------------------------------------------------------------------

*Figura 4.2 – Formulário de identificação dos conceitos de modelagem*

As formas de representação das instâncias de conceitos se restringem a, no mínimo, um dos diferentes tipos:

- Gráfica, que consiste no uso combinado de imagens, figuras, formas geométricas e textos, de tal maneira que se possa visualizar a aplicação de um modelo para representar uma determinada característica do sistema ou do seu desenvolvimento. Exemplos de representações gráficas de instâncias de modelos foram apresentadas no primeiro capítulo deste trabalho (ver figuras 1.2, 1.3, 1.4, 1.5 e 1.6).

- Textual, que é obtida através de um esquema escrito das informações que descrevem um determinado aspecto do sistema ou de outra(s) entidade(s) (como o modelo do usuário) que foi utilizado no seu desenvolvimento. Exemplos de representações textuais são descrições de cenários utilizados em casos de uso, ou dos próprios casos de uso, e aplicações da notação QOC, ilustrada na figura 1.7 e;
- Computacional, que diz respeito ao código e/ou estrutura de dados, que representa a implementação de um determinado conceito (geralmente de um modelo de arquitetura, arquitetura ou algo relacionado) aplicado a uma determinada área de domínio. A figura 4.3 ilustra a definição computacional da instância *Janela1* do conceito *Janela* de um modelo de arquitetura qualquer.

```
Janela 1: Janela
caixa_texto1, caixa_texto2, ..., caixa_texton: caixa_texto;
botao1, botao2, ..., botaon: botao;
Se botao1.clique entao executar(funcao1.1(caixa_texto1), funcao1.2);
Se botao2.apontar entao executar(funcao2.1);
Se botao2.clique entao {parar(funcao2.1); executar(funcao2.2)};
```

*Figura 4.3 – Representação computacional da instância de um conceito*

A seguir, os conceitos construídos neste trabalho serão apresentados. Os exemplos usados para ilustrá-los se referem a dados pertencentes a um sistema simplificado de gerenciamento de pré-inscrições.

### **4.3 Conceitos de modelagem construídos**

Neste capítulo, foram construídos quatro tipos de meta-conceitos (*modelo de arquitetura, padrões de arquitetura, estrutura de modelos e método*), um conceito (*modelo de arquitetura PAC*) e uma instância de conceitos (*modelo de arquitetura PAC do sistema de gerenciamento de pré-inscrições*). A seguir, serão apresentados as descrições dos tipos de conceitos de modelagem e os exemplares construídos para posterior utilização neste trabalho.

### 4.3.1 Meta-conceitos construídos e representados

Os meta-conceitos são especificações conceituais ou conceitos de alto nível de abstração, que abrangem uma vasta margem de conceitos de menor nível de abstração, servindo para defini-los. Tais especificações conceituais são obtidas através da abstração de características comuns relativas a métodos ou modelos. No caso dos meta-conceitos descobertos neste trabalho, foram observados métodos, estruturas de modelos de arquitetura e conceituais de topologia hierárquica e em rede, e os componentes destas estruturas. As vantagens no uso de meta-conceitos são:

- Possibilidade de reuso, que permite o aproveitamento de características de modelos e/ou método(s), e de aspectos arquiteturais de sistemas interativos já construídos, na geração de novos modelos de sistemas interativos. Assim, é possível obter melhorias na qualidade de especificação e documentação, através da padronização na geração e na estrutura das especificações dos métodos e/ou modelos construídos;
- Facilidade de compreensão e de emprego de suas especificações na geração de modelos e/ou métodos, o que reduz o esforço mental e o tempo de trabalho do projetista durante a especificação destes conceitos. Isto ocorre pelo fato de que tais especificações delimitam e representam muito bem os grupos conceituais aos quais estão restritas. Isto também é decorrente do fato delas permitirem a anexação ou alteração (especialização) de conceitos e relacionamentos em suas especificações, para a representação de conceitos pertencentes aos grupos por elas delimitados, sem que haja perda de consistência em relação à especificação original e;
- Reuso de representações concisas na definição de métodos e/ou modelos, ou seja, aproveitamento de estereótipos conceituais capazes de proporcionarem uma identificação e compreensão satisfatória do conceito representado com a apresentação da mínima quantidade de informações possível.

A seguir, serão apresentados os meta-conceitos construídos, e descritas suas características, em suas devidas formas de representação. Estes meta-conceitos consistem em uma contribuição significativa deste trabalho para o domínio de IHC. Isto não quer dizer que será possível se modelar um SI de acordo com qualquer método e/ou modelo e assim nortear a

construção de qualquer tipo de sistema interativo. No entanto, tal limitação poderá ser gradualmente diminuída com a evolução destes meta-conceitos.

#### 4.3.1.1 Meta-conceito de Modelo de Arquitetura

O meta-conceito *modelo de arquitetura* apresenta a forma como pode e deve ser estruturada a arquitetura de um SI e como deve ser especificado um modelo de arquitetura, orientando o projetista na construção e aplicação deste tipo de modelo. Sua representação textual pode ser observada na figura 4.4.

<p>Modelo de arquitetura:</p> <ul style="list-style-type: none"> <li>- Problema: Como deve ser organizada a estrutura da arquitetura de um SI ?</li> <li>- Solução: Fornecer ao projetista um molde para a especificação de modelos de arquiteturas.</li> <li>- Função: Delimitar a constituição e especificação dos modelos de arquitetura de sistemas a um estereótipo bem definido, cujas camadas e/ou objetos constituintes obedecem a determinadas características comuns (Padrões de Arquitetura).</li> </ul>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

*Figura 4.4 – Representação textual do meta-conceito modelo de arquitetura*

A figura 4.5 ilustra a representação gráfica do meta-conceito *modelo de arquitetura*. Ela identifica um modelo de arquitetura formado por camadas e/ou objetos inter-relacionados em forma de rede ou hierárquica. Os relacionamentos em rede são representados através do auto-relacionamento indefinido do conceito *objeto*, ou dos relacionamentos *tem* e *relaciona* do conceito *objeto* com o conceito *estrutura*. Por exemplo, no caso do modelo de arquitetura *três camadas*, são identificadas objetos (classe) constituídas de estruturas (conjuntos formados de atributos, métodos e operações), detentoras de conexões (associação, generalização, dependência, agregação e composição), que agrupam tais classes em pacotes de serviço. Quanto aos relacionamentos hierárquicos, eles estão representados através do relacionamento *tem* entre os conceitos *objeto* e *estrutura*, do relacionamento *tem* entre os conceitos *estrutura* e *conexão*, e do relacionamento *agrupa* entre os conceitos *conexão* e *objeto*. Por exemplo, o objeto (modelo de arquitetura *PAC*) é constituído de objetos (agentes *PAC*) que têm estruturas (apresentação, controle e abstração) detentoras de conexões (navegação e/ou composição), que agrupam tais agentes em espaços de composição e/ou navegação bem definidos.

Também é característica de tais objetos o fato deles obedecerem a características comuns de organização e funcionamento (Padrões de Arquitetura) observadas em arquiteturas

e modelos de arquiteturas. Isto pode ser observado através dos relacionamentos *obedece\_a* entre os conceitos *parte*, *camada* e *padrão de arquitetura*. Por exemplo, as classes que constituem a camada de apresentação, de um sistema implementado de acordo com o modelo de arquitetura *três camadas*, devem se comportar como adaptadores. Isto ocorre porque esta camada deste modelo obedece ao conjunto de critérios estabelecidos pelo Padrão de Arquitetura *Adaptador da Apresentação*. Outra característica deste meta-conceito é que um objeto pode pertencer ou não a uma camada, identificada através do relacionamento *pertence\_a* entre os conceitos *objeto* e *camada*. Por exemplo, os agentes que constituem o modelo de arquitetura *PAC* não pertencem à camada alguma, uma vez que este modelo não está organizado em camadas. Tal meta-conceito possibilita a especificação de modelos de arquiteturas cujos objetos constituintes encontram-se relacionados através de uma estrutura em rede ou hierárquica.

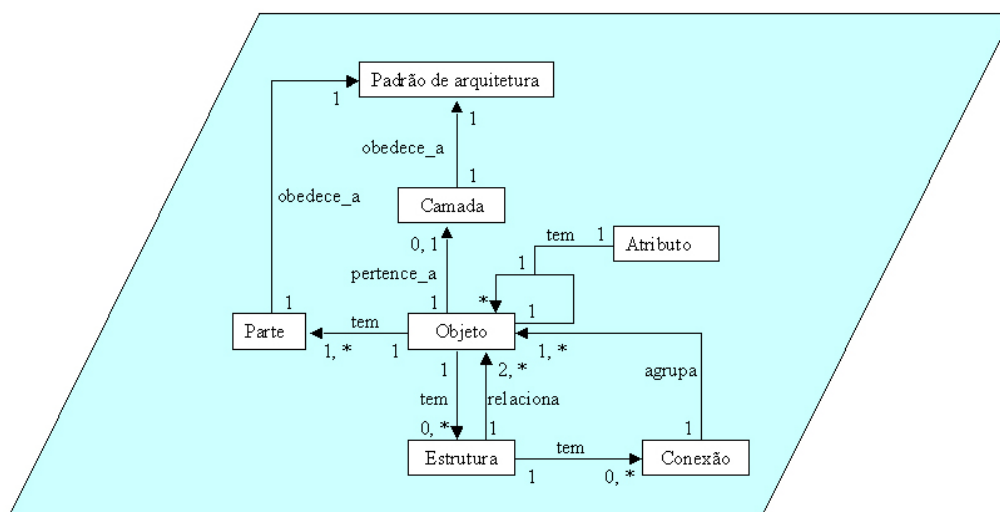


Figura 4.5 – Representação gráfica do meta-conceito modelo de arquitetura

#### 4.3.1.2 Meta-conceitos de Padrão de Arquitetura

Os meta-conceitos do tipo Padrão de Arquitetura, ilustrados na figura 4.6, identificam como camadas e objetos, que constituem uma arquitetura devem se comportar. Os meta-conceitos que representam estes Padrões de Arquitetura são os seguintes:

- Apresentação, que descreve o comportamento dos objetos de interação e da camada relativa à apresentação da interface do usuário;

- Controlador de diálogo, que informa como devem se comportar a camada e os objetos responsáveis pelo tratamento das informações, a nível de controle e coordenação da interface do usuário e dos dispositivos de tratamento e armazenamento de informação;
- Núcleo funcional, que identifica as funções que devem ser exercidas pela camada e pelos dispositivos onde serão armazenados e executados informações e tratamentos persistentes;
- Adaptador da apresentação, que detalha as funcionalidades que devem ser exercidas pela camada e pelos objetos responsáveis pela independência funcional e adaptação, entre a camada de apresentação e a de controle e coordenação da interface do usuário e;
- Adaptador do núcleo funcional, que instrui sobre os papéis a serem exercidos pela camada e pelos objetos responsáveis pela independência funcional e adaptação, entre a camada de controle e coordenação da interface do usuário e a de armazenamento e execução de informações e tratamentos persistentes.

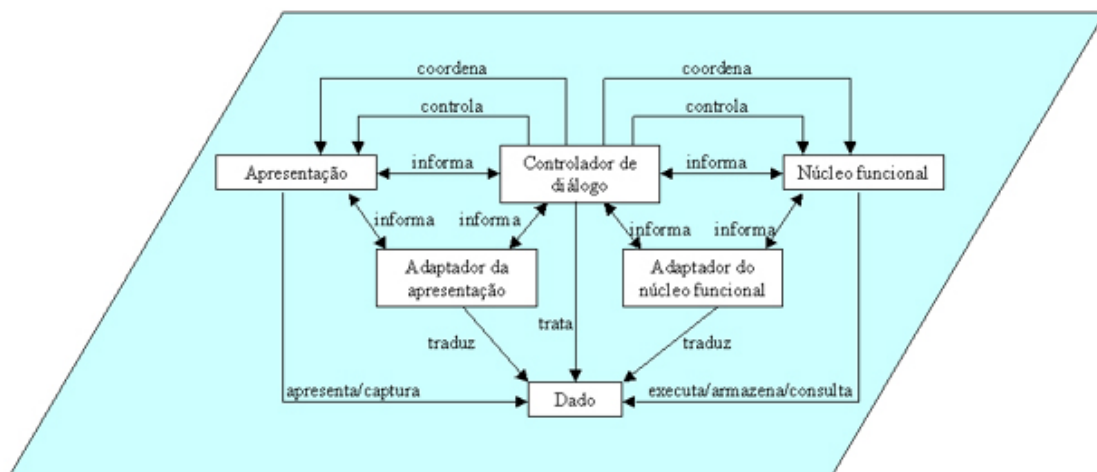


Figura 4.6 – Representação gráfica integrada dos meta-conceitos Padrões de Arquitetura

Neste trabalho, foi feita uma descrição detalhada textual e gráfica de cada Padrão de Arquitetura no anexo B.

### 4.3.1.3 Meta-conceitos de Estrutura de Modelos Conceituais

O meta-conceito *estrutura de modelos conceituais* identifica como é que devem ser organizadas as estruturas de modelos conceituais do tipo hierárquico e em rede. Sua função é apoiar o projetista na especificação destes tipos de modelo, o que está descrito na sua representação textual ilustrada na figura 4.7.

<p>Estrutura de modelos conceituais:</p> <ul style="list-style-type: none"> <li>- Problema: Como deve ser organizada a estrutura de um modelo conceitual de um SI ?</li> <li>- Solução: Fornecer ao projetista um molde para a especificação de modelos conceituais.</li> <li>- Função: Delimitar a constituição e especificação dos modelos conceituais de sistemas a um estereótipo bem definido, cujos objetos possuem estruturas que pode ou não relaciona-los através de conexões que podem agrupa-los segundo uma semântica bem definida.</li> </ul>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figura 4.7 – Representação textual do meta-conceito estrutura de modelos conceituais

A partir desta representação textual, observa-se uma considerável semelhança entre os meta-conceitos *estrutura de modelos conceituais* e *modelo de arquitetura*. As diferenças entre eles são que no caso do meta-conceito *estrutura de modelos*, os objetos que constituem as estruturas dos modelos não precisam pertencer a camadas e nem possuir partes que obedecem a padrões de arquitetura. Outra diferença é que um objeto do meta-conceito *estrutura de modelos* possui no mínimo uma estrutura, que pode não ter conexão alguma para agrupar objetos e, conseqüentemente, não relacionar objetos. A representação gráfica do meta-conceito *estrutura de modelos* está ilustrada na figura 4.8. Ela identifica um modelo conceitual formado por objetos inter-relacionados em forma de rede ou hierárquica. Os relacionamentos em rede podem ser observados através do auto-relacionamento indefinido do conceito *objeto*, ou dos relacionamentos *tem* e *relaciona* do conceito *objeto* com o conceito *estrutura*. Por exemplo, no caso do diagrama de caso de uso, são identificados por objetos (casos de uso) constituídos de estruturas (cenários) que os descrevem e relacionam ou não entre si, proporcionando ou não conexões entre eles. Quanto aos relacionamentos hierárquicos, eles podem ser observados através do relacionamento *tem* entre os conceitos *objeto* e *estrutura*, do relacionamento *tem* entre os conceitos *estrutura* e *conexão*, e do relacionamento *agrupa* entre os conceitos *conexão* e *objeto*. Por exemplo, o objeto (modelo de interface conceitual) é constituído de objetos (espaços) que têm estruturas (navegação e composição) detentoras de conexões (seqüencial, paralelo, simultâneo e alternado), que



agrupam objetos (espaços) que constituem ou que são acessados pelos espaços detentores das estruturas.

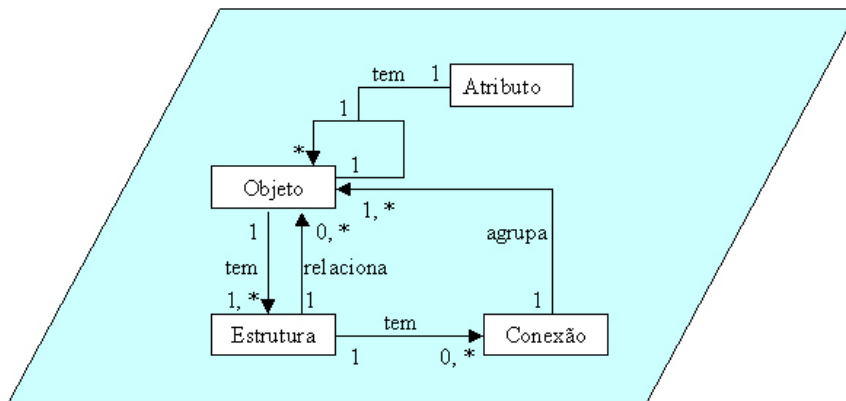


Figura 4.8 – Representação gráfica do meta-conceito estrutura de modelos conceituais

#### 4.3.1.4 Meta-conceitos de método

O meta-conceito *método* esclarece como ocorre a modelagem de um SI durante a sua construção. Ele tem o designo de auxiliar o projetista na especificação de métodos, conforme está descrito na sua representação textual, ilustrada na figura 4.9.

<p>Método:</p> <ul style="list-style-type: none"> <li>- Problema: Como ocorre a modelagem de um SI ?</li> <li>- Solução: Fornecer ao projetista um molde para a especificação de métodos de modelagem de sistemas interativos.</li> <li>- Função: Delimitar a constituição e especificação dos métodos de modelagem de sistemas a um estereótipo bem definido, em que os métodos são constituídos de fases que utilizam e produzem recursos para a obtenção de um SI com as características desejadas.</li> </ul>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figura 4.9 – Representação textual do meta-conceito método

A representação gráfica do meta-conceito *método* está ilustrada na figura 4.10. De acordo com ela um método é executado com a realização de suas fases pelo projetista através da utilização e produção de recursos, que representam características de sistemas interativos. As características a serem representadas em cada fase são definidas através dos objetivos a serem alcançados na mesma, resultando na obtenção de um sistema com as características desejadas.

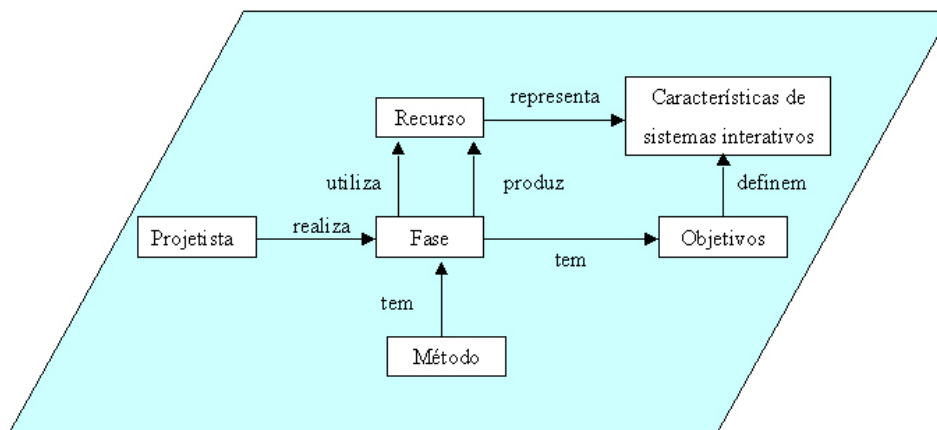


Figura 4.10 – Representação gráfica do meta-conceito método

### 4.3.2 Conceitos

Os conceitos são especificações que definem por exemplo um método ou um modelo. Os conceitos podem utilizar meta-conceitos em suas especificações, geralmente através do estabelecimento de relacionamentos de herança (relações “é”) ou de composição (relações “tem”) entre eles. A utilização da especificação conceitual apresenta como vantagem uma fácil e rápida compreensão de métodos e modelos, além da possibilidade de refinamento destes conceitos através da evolução de suas especificações. A compreensão dos conceitos possibilita eficácia e eficiência na utilização dos mesmos por indivíduos inexperientes. Esta maior acessibilidade a métodos e modelos proporciona então uma maior abrangência e rápida atualização dos conhecimentos de modelagem do projetista em relação à diversidade e inovação destes conceitos. A possibilidade de refinamento destes conceitos através da evolução de suas especificações democratiza o saber obtido durante o uso e reuso dos mesmos, o que aperfeiçoa e diversifica as suas aplicações em relação aos aspectos a serem modelados e aos autores da modelagem. Por exemplo, podem ser construídos novos conceitos a partir de conceitos já existentes, aonde são representados diferentes pontos de vista para um mesmo conceito. Para uma melhor compreensão dos conceitos e de suas características, foi definido o conceito do modelo de arquitetura PAC através da especificação conceitual da definição das características estruturais e funcionais deste modelo.

Um exemplo de conceito é o modelo de arquitetura PAC. Na figura 4.11 é apresentada a representação textual do conceito do modelo de arquitetura PAC. Sua função é apoiar o projetista na modelagem e no desenvolvimento de sistemas de tal forma que os mesmos sejam organizados em estruturas modulares, capazes de reagirem aos estímulos do usuário e do

sistema em tempo hábil e manterem a coerência e manterem a consistência entre as representações internas e externas do sistema. Isto é feito a partir de funções que descrevem os passos necessários para o desenvolvimento e a modelagem de sistemas neste tipo de arquitetura.

<p>Modelo de arquitetura PAC:</p> <ul style="list-style-type: none"> <li>- Problema: Como modelar e/ou desenvolver um sistema que possua uma interface constituída de elementos organizados em estruturas bem definidas (modulares) e capazes de reagirem às ações do usuário no menor espaço de tempo possível, mantendo a coerência entre as representações internas e externas do sistema?</li> <li>- Solução: Dar ao projetista a especificação de um modelo de arquitetura constituído por agentes de interface reativos e de composição elementar.</li> <li>- Funções: <ul style="list-style-type: none"> <li>F1 – Organizar a interface do usuário sob a forma de agentes.</li> <li>F2 – Estabelecer os agentes da interface como sendo reativos e compostos ou simples.</li> <li>F3 – Caracterizar os agentes compostos como formados por outros agentes, compostos ou simples, recursivamente agrupados.</li> <li>F4 – Caracterizar cada agente simples como não constituído de outros agentes, mas somente da abstração (núcleo funcional), do controle (controlador de diálogo) e da apresentação relativos a si mesmo.</li> </ul> </li> </ul>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

*Figura 4.11 – Representação textual do conceito do modelo de arquitetura PAC*

No exemplo a seguir, o conceito do modelo de arquitetura PAC é construído a partir do conceito de agente PAC, em observância ao meta-conceito *modelo de arquitetura*. A figura 4.12 ilustra graficamente esta definição através do relacionamento *tem/acessa* entre os conceitos *agente PAC simples* e *agente PAC composto*, e do *agente PAC composto* para ele mesmo. A observância ao meta-conceito *modelo de arquitetura* também é reforçada pelo relacionamento *é* dos conceitos *agente PAC simples* e *agente PAC composto* com a entidade conceitual *objeto* do meta-conceito *modelo de arquitetura*. A figura 4.12 também ilustra a definição do conceito *agente PAC simples* a partir dos meta-conceitos *núcleo funcional*, *controlador de diálogo* e *apresentação* através do relacionamento de composição. Tal definição é identificada a partir dos relacionamentos *tem* entre os conceitos *agente PAC simples*, *apresentação*, *controlador de diálogo* e *núcleo funcional*. Isto ocorre devido à necessidade de cada agente ser constituído de partes interdependentes capazes de suportarem os critérios de representação interna, controle e apresentação destes padrões de arquitetura. As entidades conceituais dos meta-conceitos *Padrão de Arquitetura* e *Modelo de Arquitetura* servem então como moldes para a construção das partes que constituem cada agente e dos próprios agentes entre si. Estes moldes fornecem as suas funcionalidades genéricas para suportar a construção de funcionalidades específicas às partes de cada um dos agentes PAC simples. Por exemplo, o projetista define ou se informa de como um sistema deve ser

organizado para a obtenção das vantagens próprias do modelo de arquitetura de agentes PAC através das representações gráficas e textuais deste conceito. A partir de então ele pode definir ou aprender sobre como deve se comportar cada uma das partes que constituem a entidade conceitual mais simples, o agente PAC simples, utilizado na definição deste conceito. Isto é feito através da utilização das entidades conceituais dos meta-conceitos *Padrão de Arquitetura* (Núcleo Funcional, Controlador de Diálogo, Apresentação) e *Modelo de Arquitetura* (Objeto) relacionadas. A partir do estudo ou da realização da construção do conceito *agente PAC simples*, o projetista compreende ou efetua a construção do conceito *agente PAC composto* através do relacionamento de composição entre eles. Isto significa que o agente PAC tem como atributos partes construídas a partir dos meta-conceitos ou então outros agentes PAC.

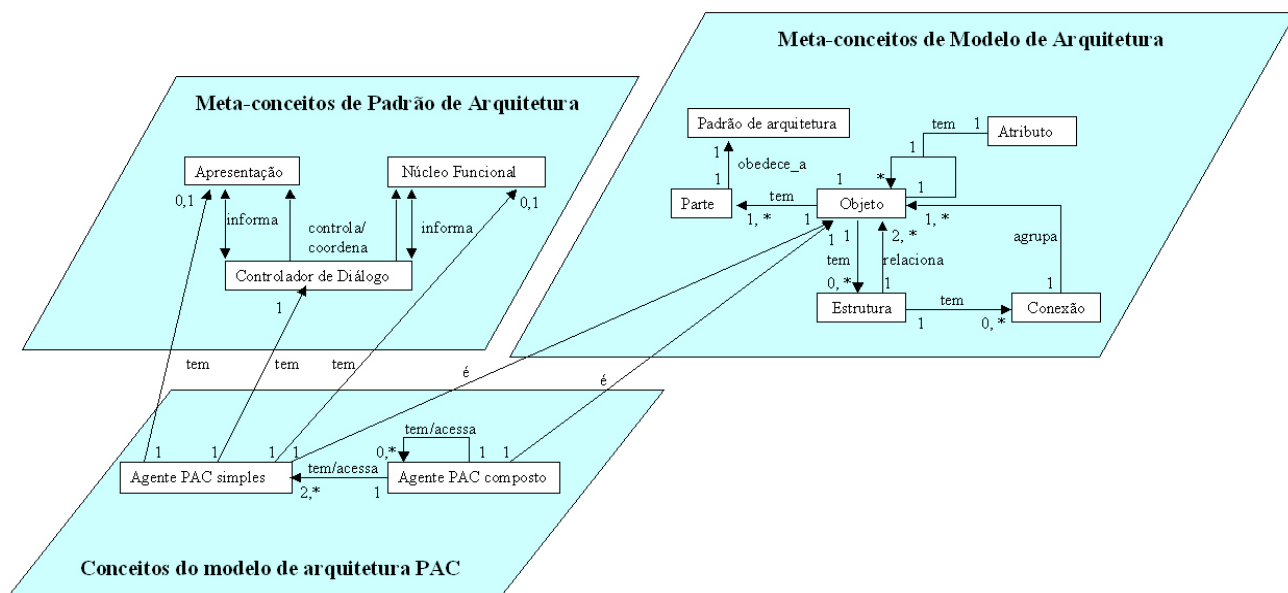


Figura 4.12 – Representação gráfica do conceito do modelo de arquitetura PAC

#### 4.3.3 Instâncias de conceitos

Instâncias de conceitos são aplicações dos conceitos em um determinado domínio. Um conceito ao ser aplicado em um determinado domínio para modelar uma determinada informação deste passa a ser uma instância deste conceito. Por exemplo, considere o conceito *janela* com os seguintes atributos: tamanho, cor e campos. Para o domínio de identificação, a modelagem da *janela de identificação* poderia ser feita informando todos os atributos do conceito janela. Outra instância deste conceito poderia ser a *janela de cadastro de alunos*.

As instâncias de conceitos têm relacionamentos entre elas, identificando e detalhando aspectos ou características do sistema a serem apresentados. Por exemplo, instâncias do conceito *janela*, que são as janelas de *identificação* e *cadastro de alunos*, se relacionam para estabelecer critérios de segurança no sistema através de restrições de acesso no fornecimento de informações. As instâncias de conceitos também mantêm relações (vínculos) com os conceitos utilizados na sua construção, permitindo a reutilização destas na construção de instâncias de novos conceitos, que são evoluções de conceitos já instanciados. Por exemplo, as informações das instâncias de um modelo de tarefas *MAD* podem ser reusadas na construção de instâncias de um outro tipo de modelo de tarefas, para o mesmo domínio, que considere dispositivos de interação na sua estrutura de representação.

Como exemplo num processo de modelagem e documentação, na figura 4.13 pode-se observar uma instância do modelo de arquitetura de agentes PAC referente a um sistema de gerência de pré-inscrições de alunos. Nela, observa-se as representações a nível de apresentação e abstração das partes constituintes dos agentes que constituem a área de identificação do sistema. A partir dos objetos de interação *caixa de texto* da apresentação dos agentes responsáveis pela identificação do usuário no sistema, o nome e a senha do usuário são informados e capturados pelo sistema. Porém a captura das informações fornecidas pelo usuário só é efetuada após a interação do mesmo com o objeto de interação da apresentação do agente responsável pelo desencadeamento do processo de verificação das informações fornecidas. A interação faz com que o agente relacionado ao botão envie uma mensagem ao agente relacionado à área de identificação para notifica-lo da submissão. Com a notificação, o agente responsável pela área de identificação envia chamadas requisitando aos agentes responsáveis pela área de identificação que informem os seus conteúdos. Após a informação do conteúdo existente na área de identificação, o agente responsável pelo sistema faz chamadas a sua abstração para uma análise comparativa entre as informações fornecidas e àquelas necessárias para o acesso ao sistema. Com a validação das informações fornecidas, é executada a navegação da apresentação do agente da área de identificação para a do agente de da área de gerenciamento das pré-inscrições.

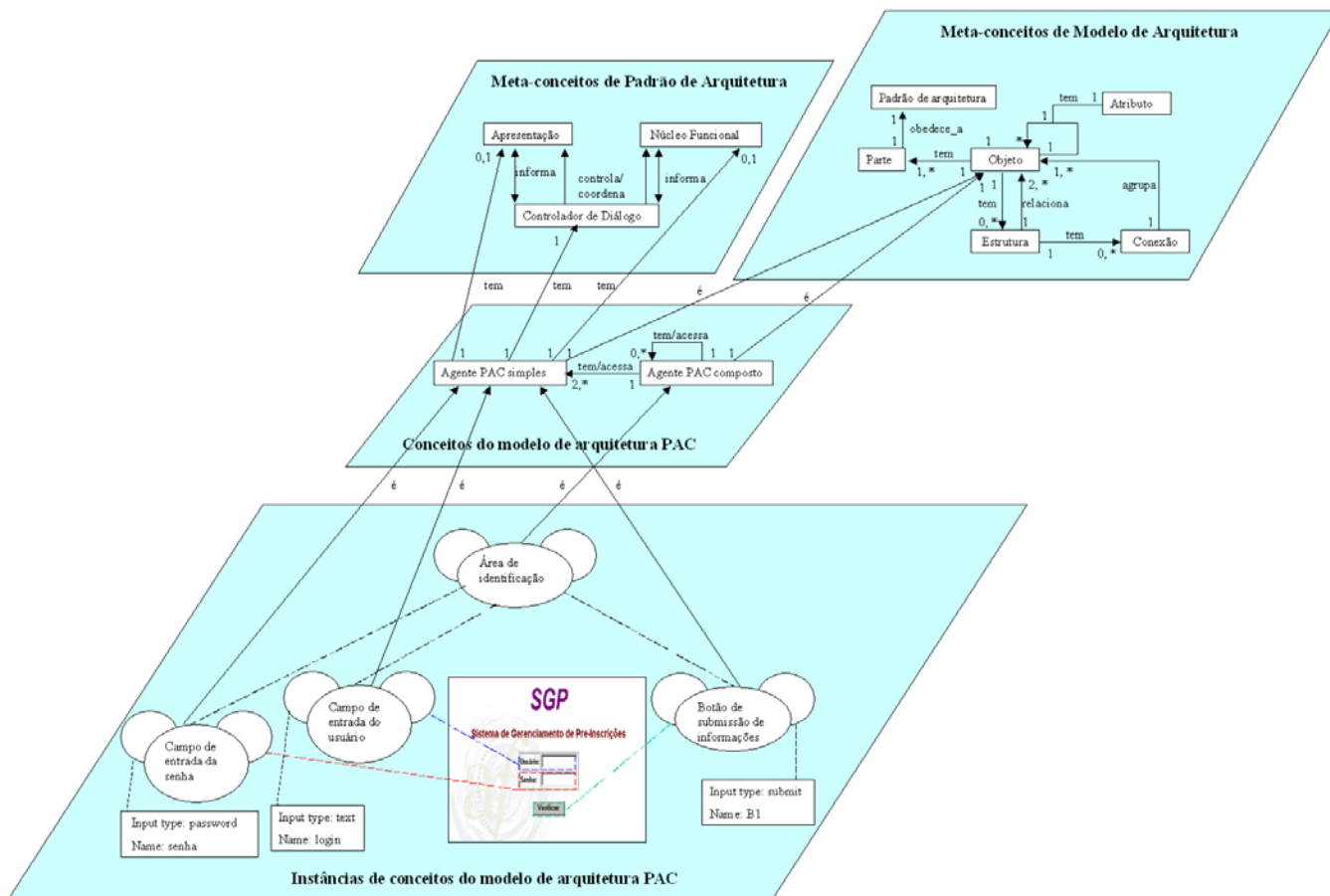


Figura 4.13 – Instância do modelo de arquitetura PAC referente à área de identificação de um sistema de gerenciamento de pré-inscrições

Através da representação gráfica da instância do modelo de arquitetura PAC apresentado na figura 4.13, o projetista identifica e constrói o sistema de tal forma a obter vantagens da modularização (como usabilidade e manutenibilidade).

Através da observação de exemplos de conceitos de modelagem, identificam-se alguns exemplos de relacionamentos existentes entre conceitos pertencentes a um mesmo nível de abstração ou a níveis diferentes. Estes relacionamentos e os seus tipos estão apresentados no anexo A.

## Conclusão

Neste capítulo, mostrou-se que é possível representar sistemas através de estruturas bem definidas. As estruturas definidas foram as seguintes: meta-conceitos de modelos de arquitetura, meta-conceitos de Padrão de Arquitetura (apresentação, adaptador da apresentação, controlador de diálogo, adaptador do núcleo funcional), meta-conceitos de

estrutura de modelos conceituais, meta-conceito de método, conceito de modelo de arquitetura PAC e instâncias do modelo PAC para a área de identificação de um sistema de gerenciamento de pré-inscrições. Este trabalho de definição destas estruturas e da forma de utilização das mesmas constitui uma pesquisa original que foi realizada nesta dissertação. Estas estruturas servem como meio de documentação para auxiliar na manutenção e evolução de um sistema interativo porque identifica detalhes que foram ou podem ser utilizados no seu desenvolvimento. Também pode-se facilitar o processo de modelagem e de desenvolvimento de sistemas através destas estruturas, uma vez que elas possibilitam uma definição conceitual do sistema bem apurada e apropriadamente distribuída antes da implementação.

## **5. Desenvolvimento de um sistema de aprendizagem de Pesquisa Operacional a partir da evolução de um método de desenvolvimento de sistemas**

### **Introdução**

Tendo em vista as dificuldades existentes para a modelagem, manutenção e evolução de sistemas interativos, principalmente no caso dos sistemas de aprendizagem devido às características pedagógicas e ergonômicas que devem estar envolvidas, resolveu-se evoluir o método MACIA [Furtado, 1997] para apoiar o processo de modelagem e documentação do módulo de aprendizagem de Pesquisa Operacional. Esta decisão foi baseada na facilidade de execução do mesmo para obtenção de interfaces ergonômicas. A evolução deste método, gerando o método MAPEAR, foi feita usando os meta-conceitos descritos no capítulo anterior e os fundamentos da metodologia META. Uma vez que não existe ainda uma ferramenta específica e capacitada para apoiar a aplicação da metodologia META na geração automática ou semi-automática de conceitos de modelagem de sistemas ou de partes deles, ela será aplicada manualmente a nível de documentação.

Neste capítulo, será apresentada a aplicação da metodologia META ao longo das fases do processo de modelagem e desenvolvimento de um sistema de aprendizagem de Pesquisa Operacional.

### **5.1 Construção e instanciação do método MAPEAR a partir da evolução do método MACIA**

O processo de especificação do método MAPEAR foi feito com o auxílio de meta-conceitos e de conceitos pré-existentes num processo evolutivo (conforme ilustrado na figura 5.1). Inicialmente o projetista definiu o método MACIA<sup>1</sup> a partir do meta-conceito *método* (ver figura 4.10), identificando os critérios necessários para a realização do mesmo. De posse do meta-conceito *método* e do conceito de método MACIA, o projetista pode aplicá-los de acordo com a sua necessidade na construção de modelos e/ou no desenvolvimento de sistemas interativos. Neste caso de estudo, o projetista incorporou conceitos provenientes do

---

<sup>1</sup> Como se trata da 1ª vez de uso da metodologia META, houve a necessidade de definir o conceito *Método MACIA*. Nas versões subsequentes este conceito já deverá existir.



desenvolvimento orientados a objetos ao método MACIA, durante o desenvolvimento do módulo de aprendizagem em PO. A evolução do método MACIA não consistiu necessariamente no seu descarte e/ou no de meta-conceitos e conceitos relacionados. O que ocorreu foi a criação de novos conceitos que serviram para a modelagem de características ao nível de detalhes de todas as principais entidades computacionais, que irão compor o módulo de aprendizagem ergonômico. Tais entidades, até então, não eram passíveis de serem totalmente capturadas pelos conceitos utilizados no método MACIA.

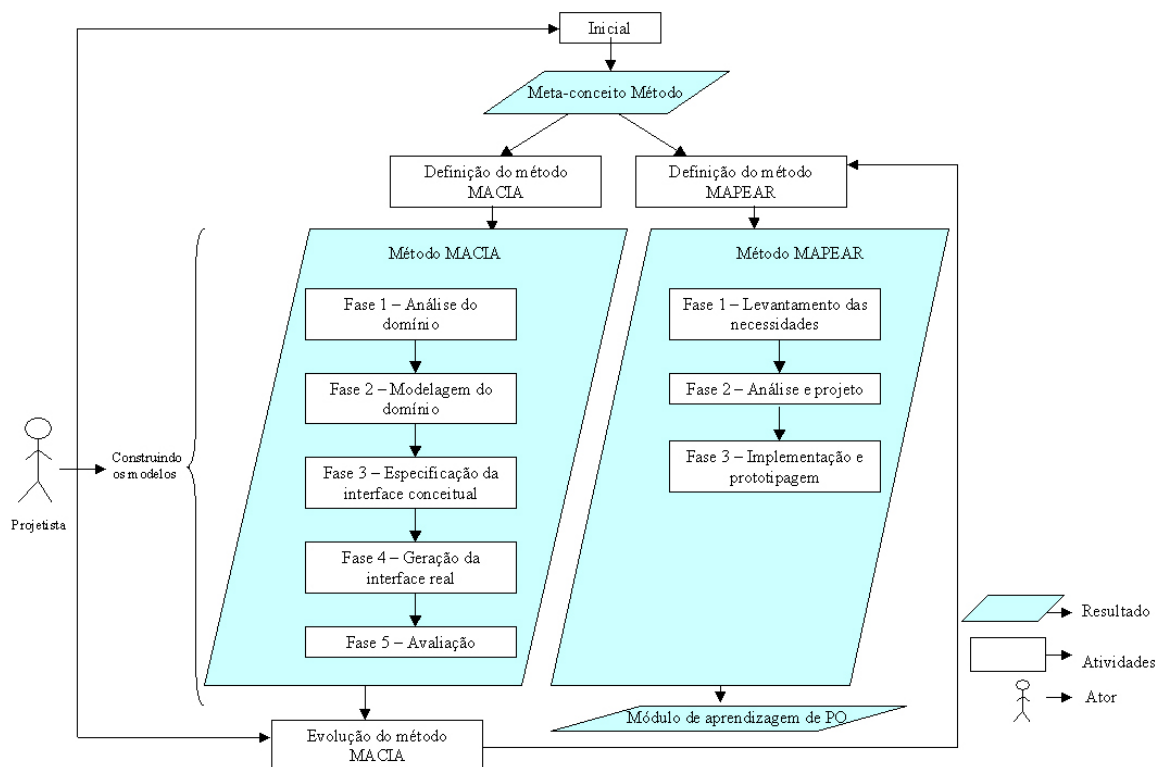


Figura 5.1 – O processo de definição do método MAPEAR

A seguir serão descritas, em detalhes, a definição e instanciação do método MAPEAR a partir da definição e evolução do método MACIA, através das suas utilizações no desenvolvimento de um módulo de aprendizagem de Pesquisa Operacional.

### 5.1.1 Definição do método MAPEAR

A definição do método MAPEAR foi feita a partir do meta-conceito *Método* responsável pela sua definição. A partir deste meta-conceito, adquiriu-se subsídios para a

especificação conceitual do método MACIA cuja representação gráfica está ilustrada na figura 5.2.

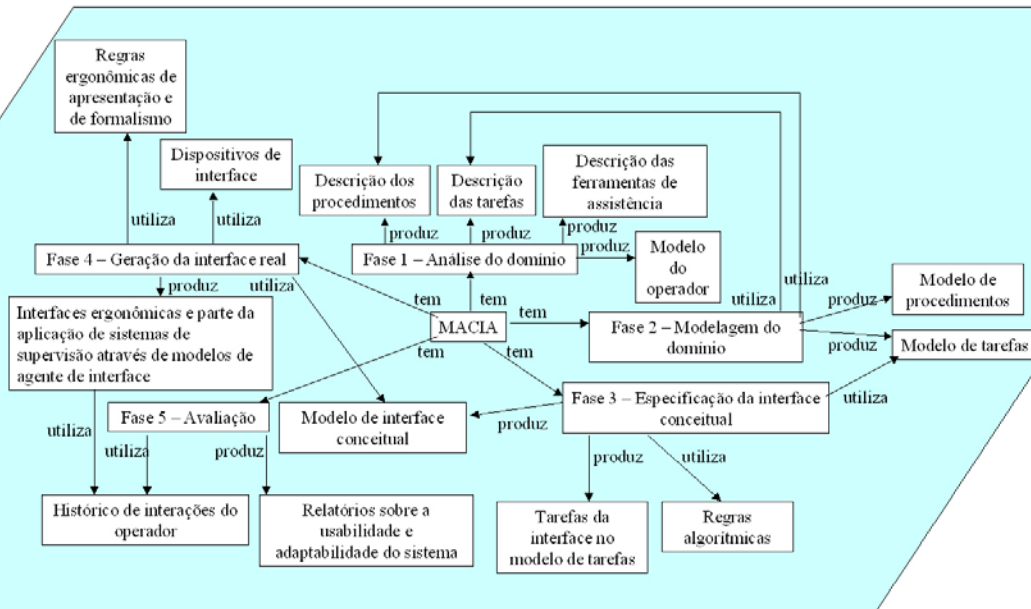
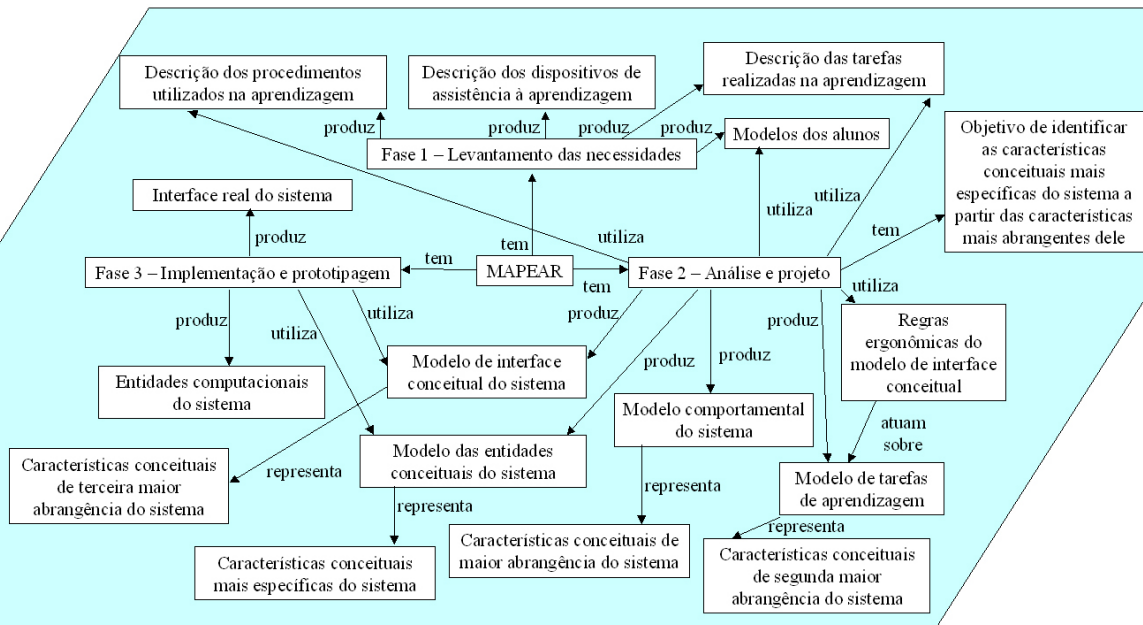


Figura 5.2 – Representação gráfica do conceito de método MACIA

Este método foi feito para ser aplicado no desenvolvimento de interfaces e partes da aplicação de sistemas de supervisão ergonômicos. Após a definição do mesmo, percebeu-se que ele é incompleto do ponto de vista da identificação das entidades envolvidas no desenvolvimento das estruturas responsáveis pela persistência, pelo tratamento de informações e pelo estabelecimento de características pedagógicas em um sistema ergonômico. Portanto, tal método foi evoluído através da inclusão, em suas especificações, de atividades que facilitem a concepção de todas estas estruturas. Este processo de evolução culminou com o desenvolvimento de uma representação gráfica do conceito do método MAPEAR, ilustrada na figura 5.3.



*Figura 5.3 – Representação gráfica do conceito de método MAPEAR*

Opcionalmente, o projetista poderá construir a representação textual do método elaborado para uma melhor visualização e documentação. Neste caso, a definição textual será feita durante a aplicação de cada fase do método MAPEAR no desenvolvimento de um sistema de aprendizagem de Pesquisa Operacional. Esta aplicação mostrada a seguir consiste na instanciação da sua definição conceitual.

### 5.1.2 Instanciação do método

A aplicação ou instanciação do método MAPEAR é feita através, da concretização de cada uma de suas fases: levantamento das necessidades; análise e projeto; e implementação e prototipagem.

#### 5.1.2.1 Levantamento das necessidades

O domínio de aplicação do método está restrito ao ensino e à aprendizagem de Pesquisa Operacional em relação aos alunos da Universidade de Fortaleza. O procedimento utilizado consiste no ministério de aulas em classe e na resolução de exercícios em casa e na sala de aula. Este procedimento é apoiado por dispositivos como programas para a resolução de modelos matemáticos (tal como: Lindo [Schrage, 1991]), e para auxiliarem o aluno na compreensão de conceitos e definições de Pesquisa Operacional.

Através de uma análise crítica do procedimento utilizado, observam-se vantagens e desvantagens. A resolução de provas e exercícios em sala de aula proporciona aos alunos a possibilidade de trocarem idéias entre si e com o professor, porém possui restrições severas em relação ao tempo (tais como: horários de início e término da aula são respectivamente os horários de término e início das atividades) e ao espaço (tais como: os alunos têm de estar situados na sala de aula). Enquanto que a resolução de exercícios em casa oferece aos alunos flexibilidade em termos de tempo, isto é, o aluno pode resolver o exercício em quaisquer horários antes da data limite de entrega do exercício, e espaço, isto é, os alunos podem estar juntos ou separados em qualquer lugar. As restrições em termos de interações sociais dos alunos entre si e com os professores podem ser consideráveis. A partir da identificação e análise destas duas formas de aprendizagem e avaliação, observa-se a necessidade do desenvolvimento de um sistema, que proporcione as vantagens apresentadas em ambas, sem que exista alguma das restrições evidenciadas.

Além da análise do procedimento, também foi realizado um estudo relativo ao perfil dos alunos. Foram observados aspectos e características que influenciam no seu desempenho durante o aprendizado. Dentre eles, destacou-se a disponibilidade de tempo e de atenção. Geralmente os alunos trabalham tempo integral ou pelo menos meio expediente e estão envolvidos no desempenho de diversas atividades. Isto dificulta a resolução de exercícios em casa de forma continuada, provocando rupturas de durações prolongadas no processo de aprendizagem. Então faz-se necessária a existência de um sistema que possibilite ao aluno dar continuidade à resolução dos exercícios de tal forma que os intervalos entre as interrupções e as retomadas possam ser diminuídos.

Com a identificação da necessidade de desenvolvimento de um sistema para suprir os requisitos de aprendizagem através da supressão dos problemas identificados, devem ser apresentadas as características deste sistema para dar prosseguimento à sua construção. Isto é feito logo a seguir.

#### **5.1.2.2 Análise e projeto**

Antes da apresentação das características do sistema requerido, deve-se especificar quais são elas, como obtê-las e como representá-las. Para facilitar este trabalho, será adotada uma abordagem que transita das características mais abrangentes às mais específicas (ver

especificação conceitual da figura 5.3), pois pode-se obter estas últimas a partir do tratamento detalhado das primeiras. A característica mais abrangente do sistema, observada na definição conceitual do método MAPEAR, é o seu comportamento. Para a definição do comportamento do sistema é interessante a utilização de um dispositivo capaz de representá-lo de tal forma a proporcionar uma visualização fácil e detalhada do mesmo. Tal dispositivo chama-se diagrama de caso de uso e a representação gráfica de sua definição conceitual está documentada na figura 5.4. A escolha deste dispositivo está fundamentada no fato de ser este um modelo teoricamente bem embasado e por ter sido amplamente utilizado e validado por profissionais da área de informática para a representação comportamental do sistema. Para a geração de sua especificação conceitual foi utilizado a definição do meta-conceito *estrutura de modelos*, construído no capítulo 4 e ilustrado na figura 4.8.

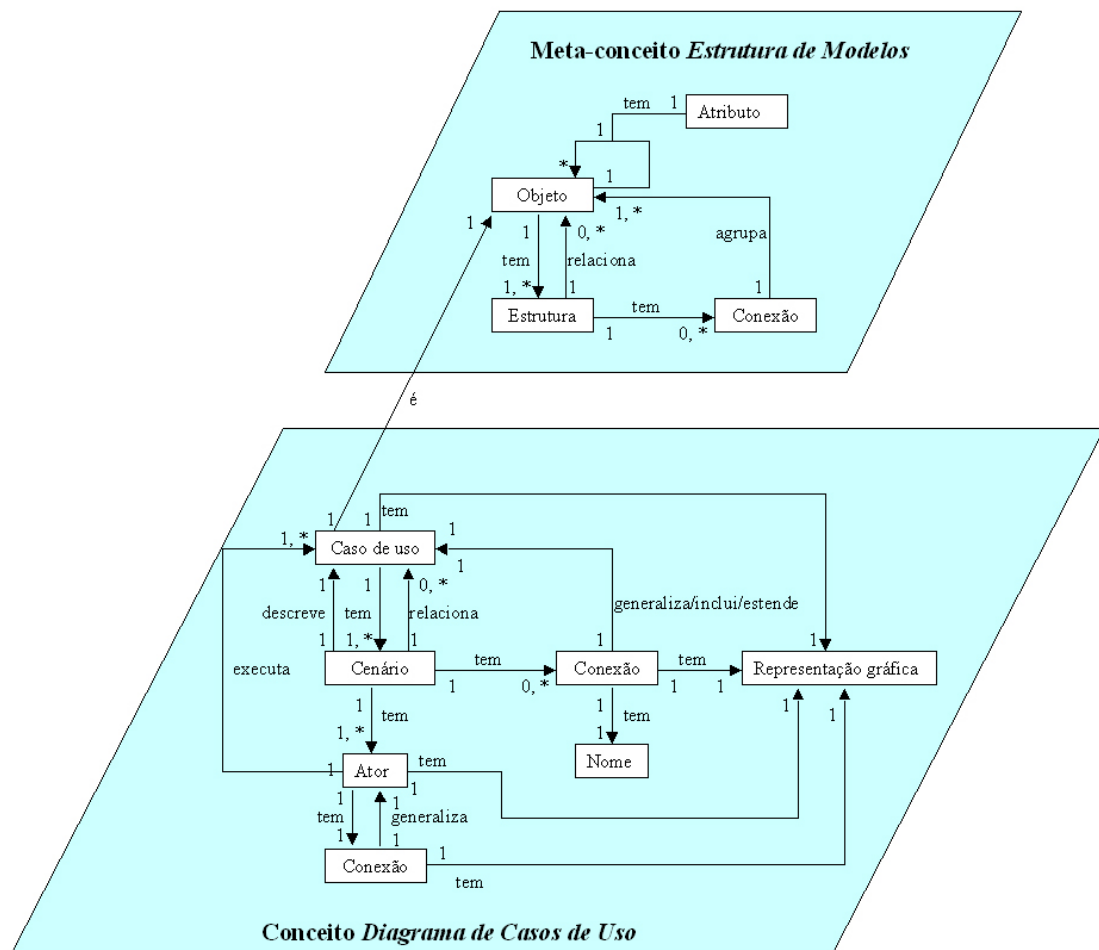


Figura 5.4 – Representação gráfica da definição conceitual do diagrama de caso de uso

Com a definição conceitual do diagrama de caso de uso, fica-se apto a aplicá-lo (instanciá-lo) para a visualização do comportamento do sistema. Na figura 5.5 é apresentado um simplificado diagrama de caso de uso do sistema em desenvolvimento, no caso, o módulo de aprendizagem de Pesquisa Operacional. De acordo com a figura, observa-se que o professor pode avaliar e auxiliar o aluno durante a resolução de problemas. Quanto ao aluno, ele pode participar de histórias através da identificação e resolução de problemas, além de tratar os eventos ocorridos através do seu histórico. O aluno também pode obter auxílio de outro aluno através do CadiNET.

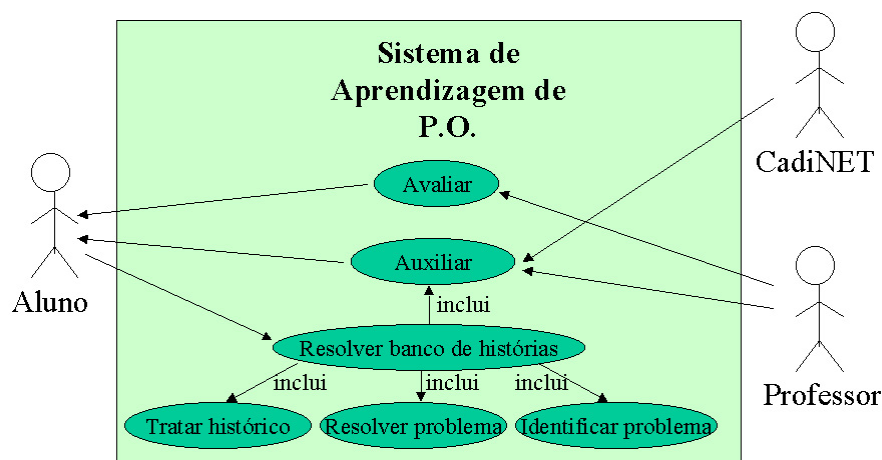


Figura 5.5 – Diagrama de caso de uso do sistema de aprendizagem de Pesquisa Operacional

Para que o sistema se comporte da maneira descrita é necessário que determinadas tarefas sejam realizadas pelo sistema e pelo usuário. É necessário, portanto utilizar-se um mecanismo que possibilite visualizar e documentar as tarefas a serem realizadas. Este mecanismo é denominado modelo de tarefas e a representação gráfica de uma de suas definições conceituais (MAD - Método Analítico de Descrição)<sup>2</sup> está documentada na figura 5.6. A escolha deste mecanismo foi norteada pela forte fundamentação teórica, ampla utilização, validação, facilidade de uso e aplicação de sua estrutura. Para a definição deste mecanismo foi utilizado o meta-conceito *estrutura de modelos* na sua geração e na sua especificação conceitual.

<sup>2</sup> Cada atributo de tarefa (como pré-condição) teria que ter ligação com o campo atributo do meta-conceito, no entanto para simplificar a ilustração, omitiu-se estas relações.

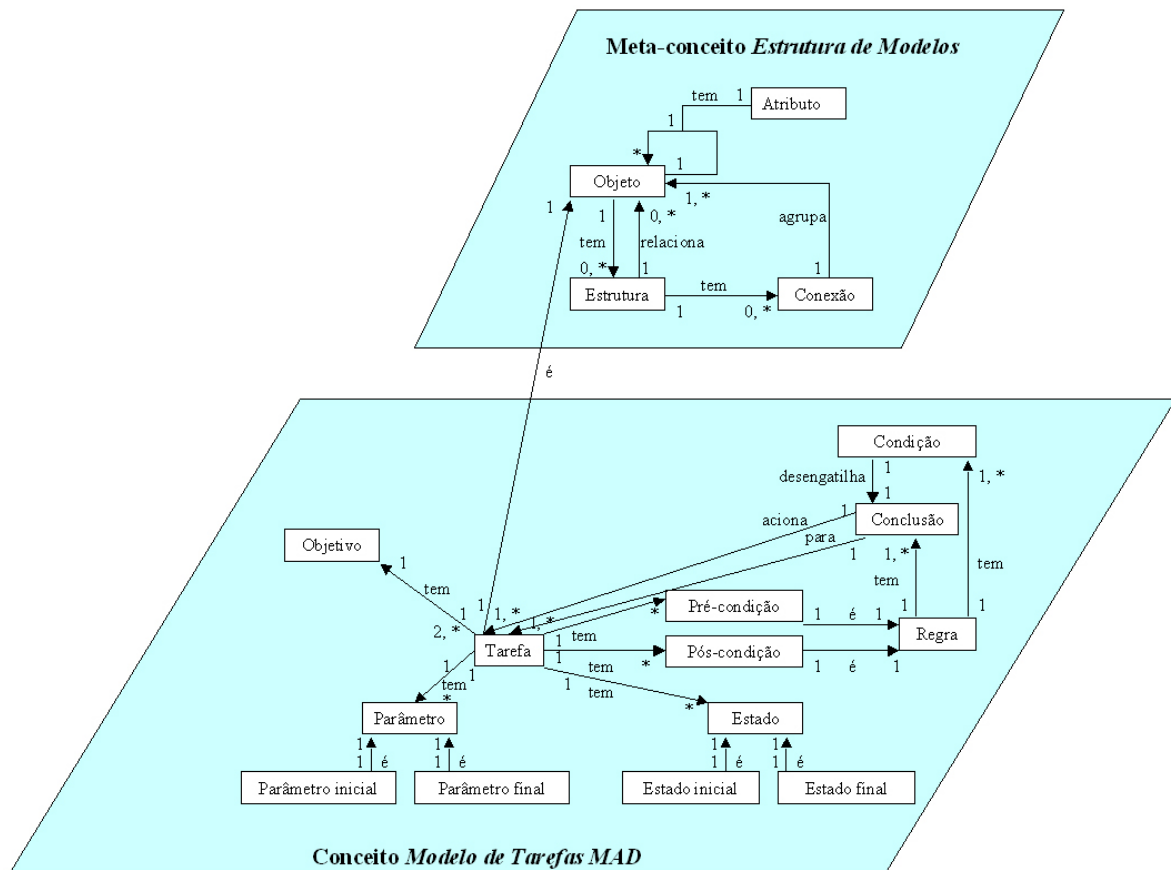


Figura 5.6 – Definição conceitual do modelo de tarefas MAD

Já que foi realizada uma definição conceitual para o modelo de tarefas, agora pode-se utilizá-lo para visualizar e documentar as tarefas que devem ser realizadas para que o sistema se comporte como definido. Na figura 5.7 é apresentado o modelo de tarefas MAD relacionado ao sistema. O modelo de tarefas MAD do sistema de aprendizagem de Pesquisa Operacional consiste na escolha e participação de alunos em histórias que constituem um histórico que pode ser tratado (analisado e enviado). A participação na história ocorre através da identificação (pela leitura ou visualização do cenário) e resolução (visualização, escolha e explicação da resposta) de problemas, além da participação em fóruns de discussão. O histórico enviado será utilizado pelo professor para avaliar o aluno. Já que se sabe quais são as tarefas que devem ser realizadas, é adquirida a aptidão necessária para desvendar e modelar estruturas mais específicas, a interface conceitual e as entidades que formam o sistema.

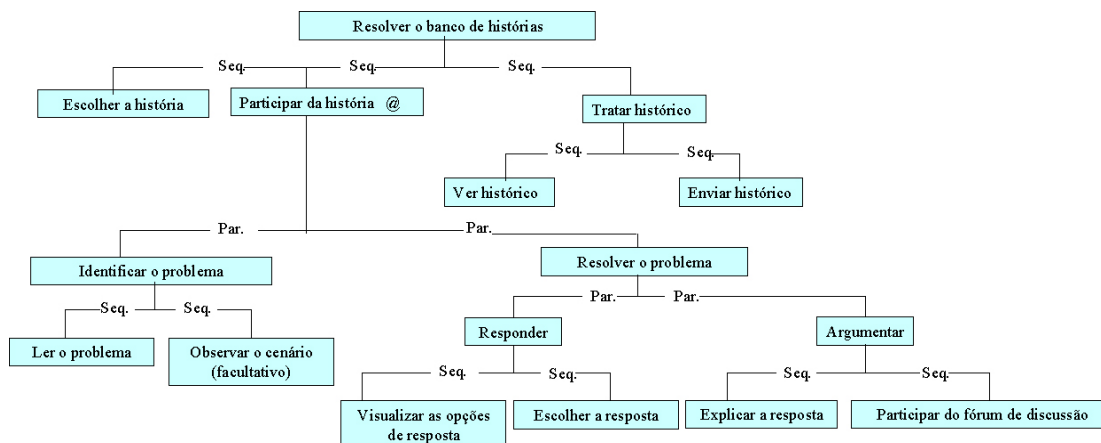


Figura 5.7 – Modelo de tarefas MAD do sistema de aprendizagem de Pesquisa Operacional

A interface conceitual de um sistema é uma abstração capaz de capturar características da interface, permitindo a sua visualização e modificá-la antes da sua implementação. Para que seja possível a sua utilização, é necessário utilizarmos um dispositivo capaz de capturá-la e representá-la, o modelo de interface conceitual (MIC). A definição conceitual do MIC está documentada na figura 5.8. Ela foi obtida a partir do meta-conceito *estrutura de modelos*, também ilustrado nesta figura. A razão de uso deste modelo se deve ao fato de que ele representa muito bem características e comportamentos de interfaces visuais do sistema com o usuário, além do fato dele poder ser facilmente instanciado a partir de instâncias de modelos de tarefas [Furtado, 1997].

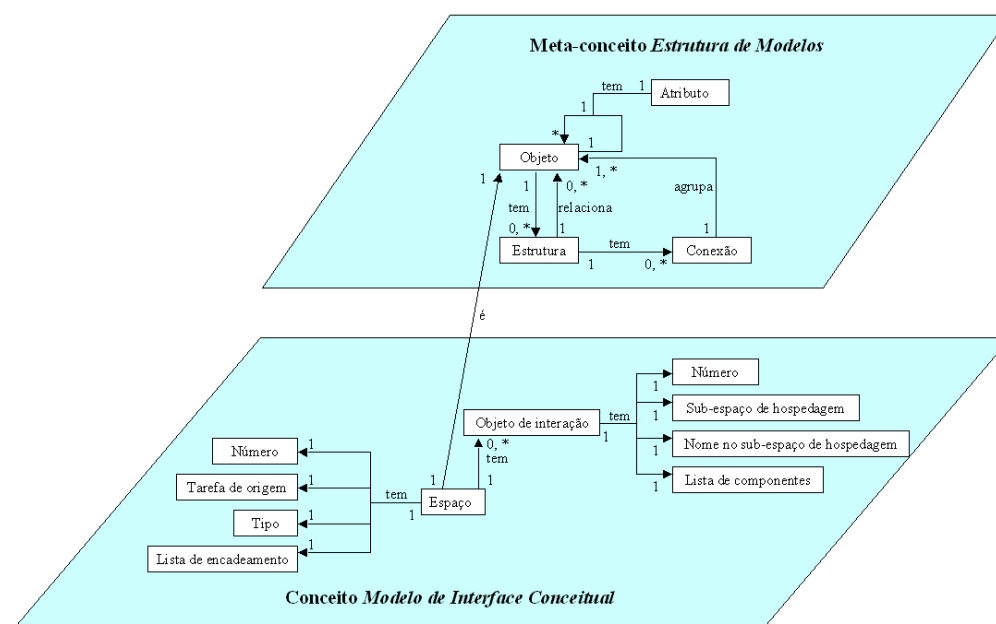


Figura 5.8 – Definição conceitual do Modelo de Interface Conceitual



Com a instanciação do modelo de tarefas, a definição conceitual do MIC e aplicação do processo de geração de interfaces conceituais a partir de modelos de tarefas, é possível construir e documentar o MIC do sistema de aprendizagem de Pesquisa Operacional (ver figura 5.9).

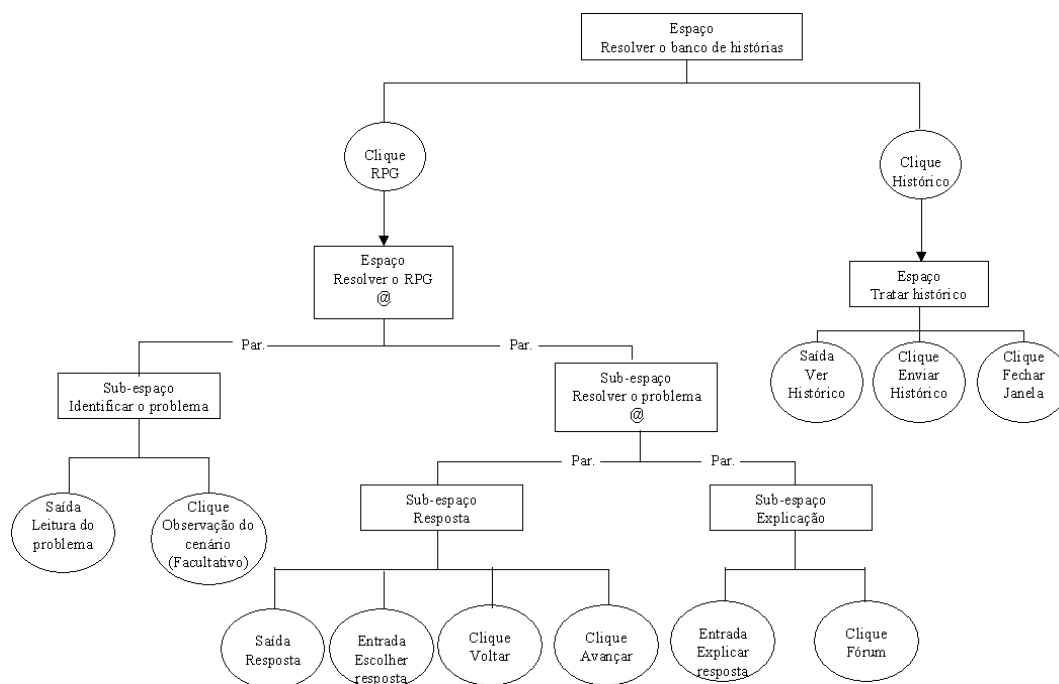


Figura 5.9 –MIC do sistema de aprendizagem de Pesquisa Operacional

Neste processo, o MIC é obtido através da aplicação de regras ergonômicas no modelo de tarefas, como pode-se observar na definição conceitual do método MAPEAR na figura 5.3. A tarefa “Resolver o banco de histórias” dá origem ao espaço de navegação “Resolver o banco de histórias”. As tarefas “Escolher a história” e “Participar da história” dão origem ao espaço de composição “Resolver o RPG”, enquanto que a tarefa “Tratar histórico” origina o espaço de composição “Tratar histórico”. A tarefa “Identificar problema” origina um sub-espço de mesmo nome que agrupa os objetos de interação “Leitura do problema” e “Observação do cenário” originados respectivamente das tarefas “Ler o problema” e “Observar o cenário”. A tarefa “Resolver o problema” dá origem a um sub-espço de mesmo nome, formado pelos sub-espços “Resposta” e “Explicação” derivados respectivamente das tarefas “Responder” e “Argumentar”. As tarefas “Visualizar as opções de resposta” e “Escolher a resposta” (sub-tarefas da tarefa “Responder”) dão origem aos objetos de interação do sub-espço “Resposta”, tais como: “Resposta” (saída), “Escolher Resposta” (entrada),

“Voltar” (clique) e “Avançar” (clique). Concluindo, as sub-tarefas da tarefa “Argumentar” (“Explicar a resposta” e “Participar do fórum de discussão”), originam os objetos de interação do sub-espço “Explicação”: “Explicar resposta” (entrada) e “Fórum” (clique).

Uma vez que já foi construída a interface conceitual do sistema, resta encontrar todas as entidades conceituais do sistema. Um dispositivo que pode ser utilizado para identificar e modelar todas as entidades conceituais do sistema é o diagrama de classes. Este modelo foi escolhido devido ao respaldo teórico do mesmo na comunidade científica e a ampla validação da aplicabilidade de sua estrutura para a modelagem conceitual dos domínios de aplicação de sistemas, capaz de retrata-los com exatidão e flexibilidade. Sua definição conceitual está ilustrada na figura 5.10, sendo também desenvolvida a partir do meta-conceito *estrutura de modelos*.

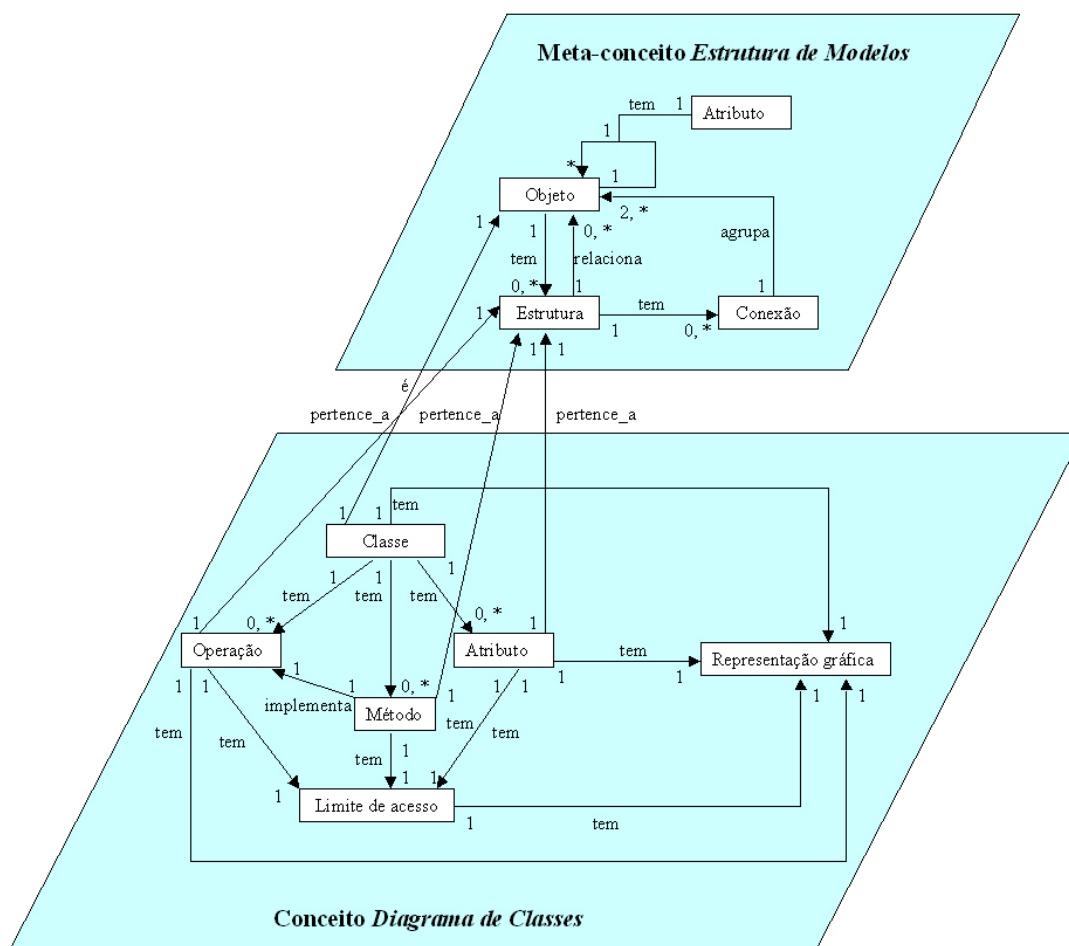


Figura 5.10 – Definição conceitual do Diagrama de Classes

Com a realização e/ou observação do conceito *diagrama de classes* adquire-se informação sobre a estrutura necessária para a construção de um diagrama de classes do módulo de aprendizagem de Pesquisa Operacional. Para tanto, é necessário que inicialmente sejam identificadas as classes que compõem o diagrama. Isto pode ser feito através da análise do diagrama de casos de uso, do modelo de tarefas e do MIC. Por apresentar um maior nível de detalhes, é interessante o desenvolvimento do diagrama de classes a partir do MIC. Portanto os seguintes passos devem ser executados:

- Cada espaço deverá ser transformado em uma classe limítrofe: *Resolver o RPG*, *Tratar histórico*, *Resolver o banco de histórias*.
- As navegações entre espaços devem ser transformadas em relacionamentos de associação entre as classes limítrofes que representam os respectivos espaços: *ResolverBancoHistórias* tem uma associação com *ResolverRPG* e outra com *TratarHistorico*. Ambas associações recebem o rótulo *acessa*, uma vez que a partir da classe *ResolverBancoHistórias* é possível acessar as classes *ResoverRPG* e *TratarHistorico*. Como existem vários RPG's e os seus respectivos históricos a serem acessados a partir da tela *ResolverBancoHistórias*, a multiplicidade deste relacionamento é de um para muitos no sentido da tela *ResolverBancoHistórias* para a tela *ResolverRPG* e para a tela *TratarHistorico*.
- Os objetos de interação dos sub-espacos de cada espaço devem ser transformados em atributos da classe limítrofe relacionada: i) para a classe *ResolverRPG* existem os atributos *Problema*, *Cenário*, *Resposta*, *escolhaResposta*, *Retorno*, *Avanco*; ii) para a classe *ResolverBancoHistórias* existem os atributos *resolverRPG* e *tratarHistorico*; e iii) para a classe *TratarHistorico* existem os atributos *Enviar*, *Fechar* e *Ver*.
- As reações do sistema, relativas às interações do usuário com cada objeto de interação, devem ser transformadas em operações da classe que representa o espaço aonde a interação ocorreu: i) na classe *ResolverRPG* deve existir uma operação para capturar as ações do usuário (*capturarAcaoUsuario()*), uma operação responsável pela exibição do problema (*mostrarProblema()*), uma operação responsável pela abertura de uma janela de exibição de cenário (*abrirCenario()*), uma operação responsável pela

exibição das alternativas de respostas (*mostrarRespostas()*), uma operação responsável pelo registro da pergunta, resposta escolhida e explicação em um histórico (*registrarHistorico()*), uma operação responsável pela navegação a respostas já escolhidas e suas respectivas perguntas e explicações (*mostrarProblemaAnterior()*) e uma operação responsável pelo avanço para a próxima pergunta e suas alternativas de resposta (*mostrarProblemaPosterior()*); ii) na classe *ResolverBancoHistórias* deve existir uma operação para capturar a ação do usuário (*capturarAcaoUsuario()*), uma operação para abrir a janela de resolução de RPG (*abrirRPG()*) e uma operação para abrir a janela de exibição do histórico (*abrirHistorico()*); e iii) na classe *TratarHistorico* deve existir uma operação para capturar a ação do usuário (*capturarAcaoUsuario()*), uma operação para enviar o histórico (*enviarHistorico()*), uma operação para fechar o histórico (*fecharHistorico()*) e uma operação para exibir o histórico (*mostrarHistorico()*).

- Observar as operações utilizadas para abertura de espaços não representados no MIC, para que sejam criadas as classes limítrofes relacionadas a estes espaços: a operação *abrirCenario()* indica que deve existir a classe limítrofe *Cenário*. Ela também indica que a classe *cenário* deve possuir uma associação com a classe *ResolverRPG* identificada através do rótulo *acessa*. Quanto à multiplicidade desta associação, ela deve ser de um para um, pois a partir da tela *ResolverRPG* é possível acessar somente uma tela de exibição de cenários. A classe *Cenário* deve ter como atributos: o objeto de interação *cenário* para a exibição do cenário, o objeto de interação *imprimir* para a impressão do cenário e o objeto de interação *cancelar* para fechar a janela de exibição do cenário. A partir dos atributos observa-se a existência dos métodos para: capturar a ação do usuário (*capturarAcaoUsuario()*), exibir o cenário (*mostrarCenario()*), imprimir o cenário (*imprimirCenario()*) e fechar janela do cenário (*fecharCenario()*).
- Identificar outras classes do sistema através de informações que precisam ser retidas para que os métodos das classes limítrofes possam ser executados: classe entidade *História* para o acolhimento e o fornecimento de perguntas e respostas para a classe limítrofe *ResolverRPG*, a classe entidade *Aluno* para a identificação do aluno no histórico de tal forma que ele possa dar continuidade à resolução de histórias a partir do ponto de interrupção e ser identificado na avaliação do seu histórico, a classe

entidade *Histórico* para o armazenamento do histórico do aluno na resolução de RPG's e a classe de controle *GerenteHistórico* para controlar o a geração do histórico a partir das informações disponibilizadas pelo usuário nas classes limítrofes ou pelo sistema nas classes de entidade.

- Construir os relacionamentos, do tipo associação, das novas classes construídas com as classes limítrofes que lhe deram origem:
  - Da classe entidade *História* com a classe limítrofe *ResolverRPG*, através de uma associação de multiplicidade de um (lado da classe *ResolverRPG*) para muitos (lado da classe *História*), identificada com o rótulo *captura*. Isto porque a tela *ResolverRPG* deverá promover a participação do aluno em várias histórias, capturando suas informações (perguntas e respostas) de acordo com o RPG escolhido;
  - Da classe controle *GerenteHistórico* com a classe limítrofe *ResolverRPG*, através de uma associação de multiplicidade de um para um, identificada com o rótulo *proporciona*. Isto porque será a tela *ResolverRPG* que proporcionará o meio necessário para captura de boa parte das informações (perguntas, respostas e explicações) necessárias para a construção dos históricos.
- Construir os relacionamentos, do tipo associação, entre as classes encontradas a partir das classes limítrofes:
  - Da classe controle *GerenteHistórico* com as classes entidades *Aluno* e *Histórico*, através de uma associação para cada classe entidade, identificada cada qual com os seus respectivos rótulos (*gera* para o relacionamento da classe *Histórico* e *proporciona* para o da relacionamento da classe *GerenteHistórico*). Estas associações serão de multiplicidade de um (do lado da classe *GerenteHistórico*) para muitos (do lado das classes *Aluno* e *Histórico*). Isto porque o acesso às informações de vários alunos pela classe *GerenteHistórico*, permitirá que tal classe possa gerar os históricos dos alunos a serem avaliados.

- Da classe entidade *Aluno* com a classe entidade *Histórico*, através de uma associação de multiplicidade de um (do lado da classe *Aluno*) para muitos (do lado da classe *Histórico*), identificada com o rótulo *tem*. Isto ocorre em função do fato de que um aluno poderá ter mais de um histórico (um para cada trabalho de RPG).
  - Da classe entidade *História* com a classe limítrofe *Cenário*, através de uma associação de multiplicidade de um (do lado da classe limítrofe *Cenário*) para muitos (do lado da classe entidade *História*), identificada com o rótulo *representa*. Isto é decorrente do fato de que a tela que exibe cenários serve para a identificação e compreensão de histórias.
- Completar as operações (*obterCenário()* na classe limítrofe *Cenário* e *enviarCenário()* na classe entidade *História*) e atributos (*cenário* na classe entidade *História*) necessários para suportar as operações (*imprimirCenário* e *mostrarCenário* na classe limítrofe *Cenário*) e os relacionamentos (*representa* entre as classes *Cenário* e *História*) das novas classes construídas (*História* e *Cenário*).
- Definir a visibilidade (acesso) dos atributos (privada: inacessível a outras classes) e das operações (pública: acessível a qualquer classe relacionada). Esta definição assim foi feita para restringir o acesso aos atributos de uma classe somente às suas operações, proporcionando o encapsulamento das mesmas e conseqüentemente uma maior reusabilidade.

Através da figura 5.11 pode ser visualizado o diagrama de classes, a nível conceitual, para o módulo de aprendizagem de Pesquisa Operacional.

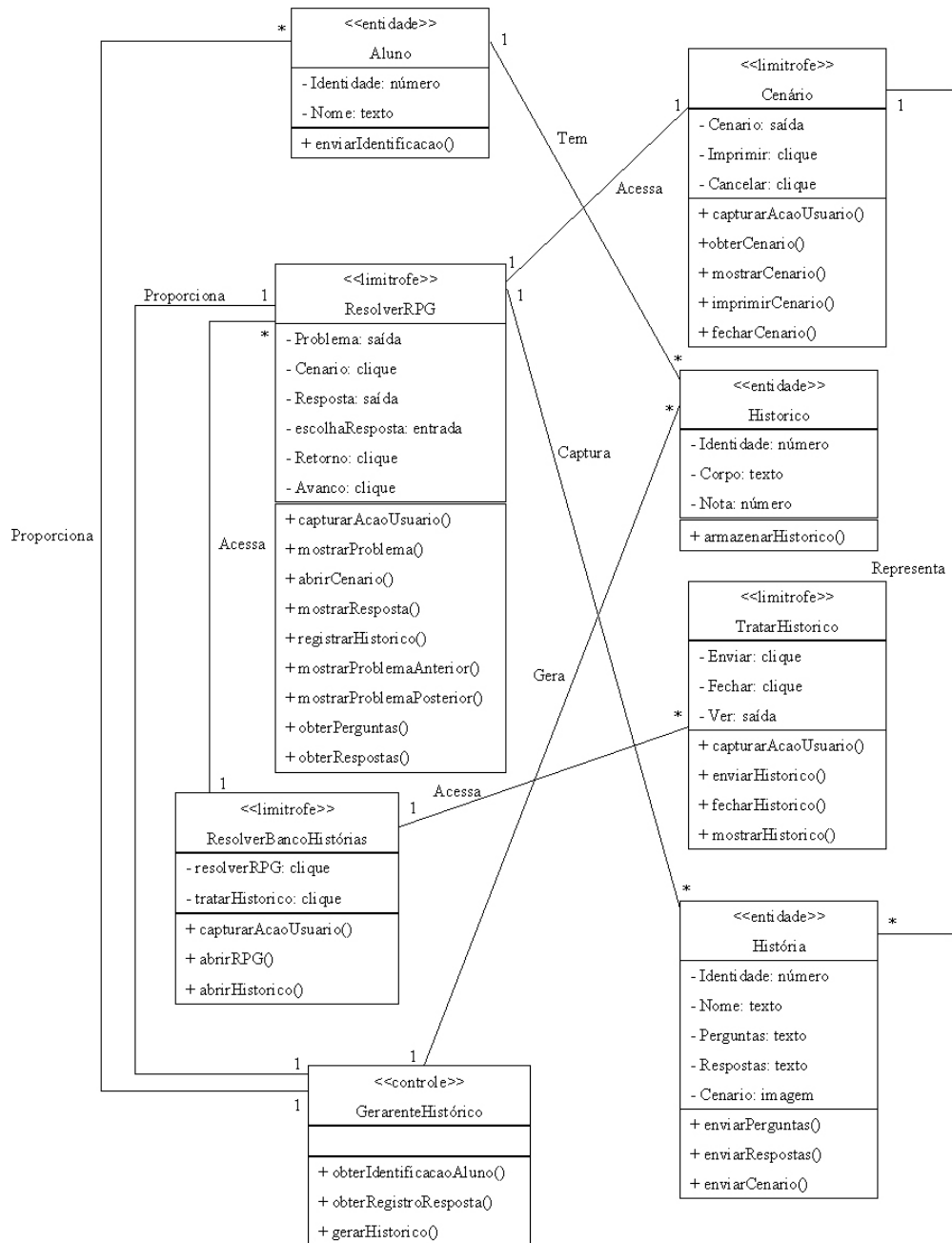


Figura 5.11 –Diagrama de classes do módulo de aprendizagem de Pesquisa Operacional

### 5.1.2.3 Implementação e prototipagem

Com a identificação das entidades conceituais em um diagrama de classes a nível de projeto, são satisfeitas as pendências necessárias para a construção dos recursos relacionados à implementação e prototipagem do sistema. Tais recursos são o modelo de arquitetura e as interfaces reais do sistema.

Para que o modelo de arquitetura do sistema seja construído, é importante a definição e/ou reutilização dos meta-conceitos capazes de suportarem a sua especificação. Estes meta-conceitos são *modelo de arquitetura* e *padrões de arquitetura*, construídos no capítulo 4 e ilustrados nas figuras 4.5 e 4.6. A partir de então obtém-se a orientação necessária para a escolha e definição do modelo de arquitetura necessário. De acordo com a identificação da necessidade de que o sistema deveria evitar as interrupções prolongadas no processo de aprendizagem do aluno e de que deveria ser acessado por vários alunos, foi definido o modelo de arquitetura três camadas na figura 5.12. Isto porque ele permite a implantação do módulo de aprendizagem em um ambiente disponível ao acesso através da Internet, além de facilitar a manutenção do sistema e satisfazer ao princípio de independência de diálogos.

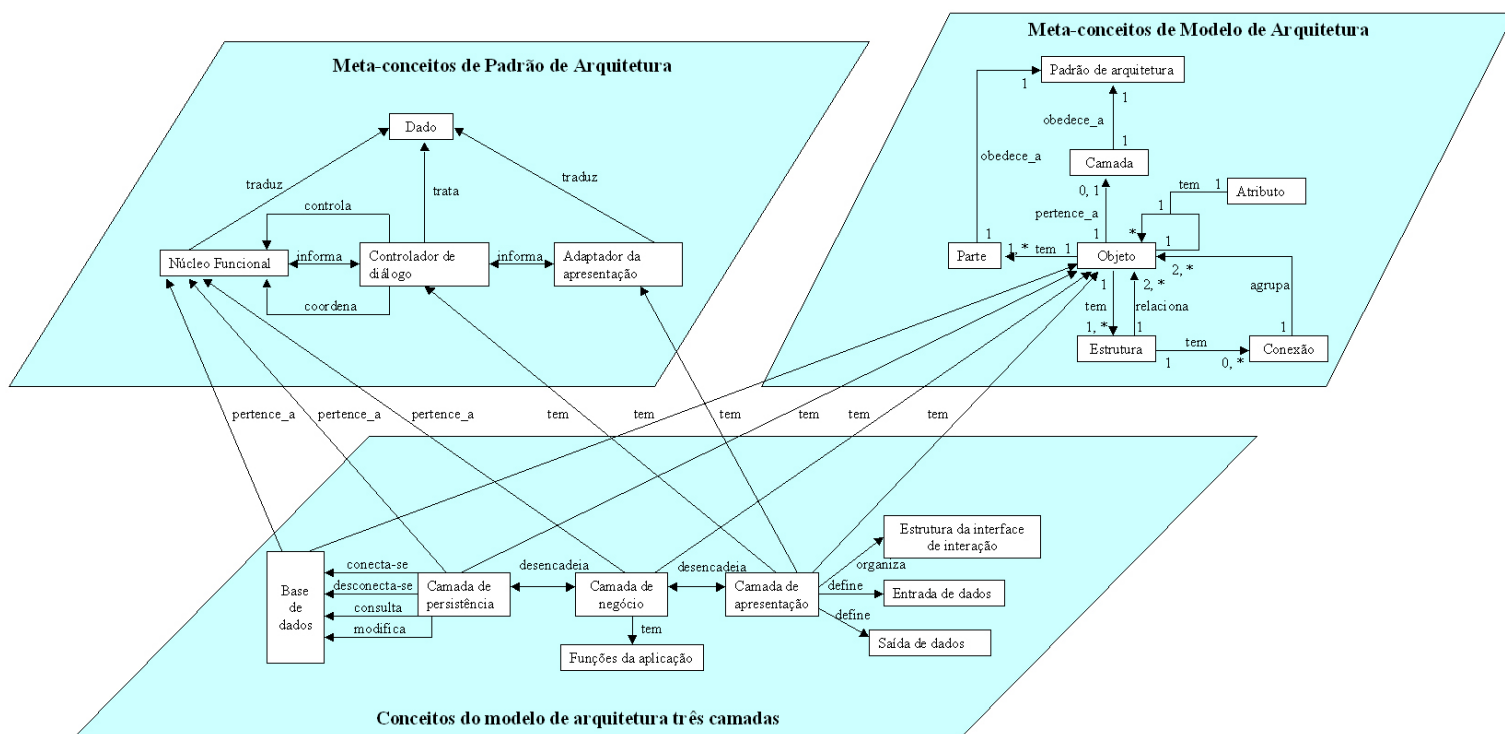


Figura 5.12 – Definição conceitual do modelo de arquitetura três camadas

Através da especificação do modelo de arquitetura *três camadas* identifica-se que a arquitetura do sistema será constituída de três camadas, cada qual constituída de objetos organizados numa estrutura de rede e/ou hierárquica e responsável por funcionalidades relativas à persistência, aplicação, controle ou apresentação no módulo de aprendizagem. Para construir as instâncias deste modelo, foi aproveitado o diagrama de classes desenvolvido na fase de análise e projeto. No entanto, foi necessária a realização de alterações e acréscimos ao diagrama de classes. Para que as instâncias obedecessem ao modelo de arquitetura



especificado, ao princípio de encapsulamento da orientação a objetos e às limitações tecnológicas dos recursos de implementação atuais (linguagens, bancos de dados e outros), foi feito o seguinte:

- Foram criadas classes de controle para fazer o interfaceamento entre as classes limítrofes e as classes de entidade, ou entre das classes de entidade existentes, de tal forma que para cada classe de entidade exista uma classe de controle para encapsulá-la (recomendação de manutenibilidade). Por exemplo, a classe de controle *GerenteAluno* entre a classe de controle *GerenteHistórico* e a classe de entidade *Aluno*; e *GerenteHistória* das classes limítrofes *Cenário* e *ResolverRPG* em relação à classe de entidade *História*. Isto resultou na eliminação das associações diretas que existiam entre as classes que foram interfaceadas, resultando na criação de associações de multiplicidade de um para muitos entre as classes controle do tipo gerente e suas respectivas classes de entidade. A razão desta multiplicidade se deve ao fato de que através de uma classe de controle do tipo gerente se possa gerar e acessar todos os objetos da respectiva classe de entidade. As classes limítrofes e de controle representam a camada de apresentação do módulo;
- Para cada classe de entidade foi criada uma classe de persistência para mapear os seus objetos em tabelas de bancos de dados relacionais, entidades computacionais responsáveis pela realização das funcionalidades de retenção e fornecimento de informações no núcleo funcional. O fato do banco de dados escolhido ter sido do tipo relacional fundamenta-se na difusão, confiabilidade e potencial de uso deste tipo de dispositivo em relação aos demais. Por tanto, as classes de persistência criadas foram: *Aluno*, *Histórico* e *História*;
- Foram criadas classes de controle do tipo transação, para fazer o interfaceamento (camada de persistência) entre as classes de persistência e as classes de entidade (camada de aplicação), sendo as de entidade criadas e carregadas (preparadas para a execução) pelas classes de controle do tipo gerente a partir informações armazenadas nas classes de persistência. Assim foram criadas as classes: *TransacaoAluno*, *TransacaoHistorico* e *TransacaoHistoria*. Para cada uma destas classes foram criadas uma associação com sua respectiva classe controle do tipo Gerente e outra com sua

correspondente classe de persistência. As multiplicidades destes relacionamentos seriam de um para um no que diz respeito à associação entre as classes de controle do tipo gerente e as do tipo transação; e de um para muitos nas associações entre as classes de controle do tipo transação e as respectivas classes de persistência. Isto porque para cada classe de entidade deverá existir uma classe de gerência e outra de acesso ao banco de dados (encapsulamento); e para cada classe de transação deverão ser acessados diversos objetos da respectiva classe de persistência;

- Foram criadas dependências entre as classes de persistência e as respectivas classes de entidade, tendo em vista a necessidade de obtenção de informações relacionadas às classes de entidade para o armazenamento das mesmas em classes de persistência e;
- Foram criadas as navegações, no sentido descendente das classes limítrofes às classes de persistência. Esta decisão foi fundamentada nas restrições de implementação (custos operacionais):
  - Entre as classes limítrofes, o sentido da navegação foi a mesma da observada entre os espaços, para que os sentidos das navegações no modelo de interface conceitual fossem respeitados e facilmente implementados;
  - Entre as classe limítrofes e as classes de controle, o sentido das navegações foi das primeiras para as últimas. Isto foi efetuado para que regras da orientação a objetos e dos padrões de arquitetura, relativas à portabilidade da camada de aplicação, fossem respeitadas;
  - Entre as classes de controle *GerenteAluno* e *GerenteHistórico*, o sentido da navegação criada foi da última para a primeira classe citada, pois é necessária a obtenção de informações que identificam o aluno para a geração da identidade dos históricos.
  - Entre as classes de controle do tipo gerente e as classes de entidade, as navegações estabelecidas tiveram o sentido de orientação das primeiras para as últimas. Isto porque o encapsulamento, das classes de entidade entre si, é obtido

com a atribuição da responsabilidade de acesso às informações para as respectivas classes de controle do tipo gerência.

- Entre as classes de controle do tipo gerente e as do tipo transação, as navegações estabelecidas tiveram o sentido de orientação das primeiras para as últimas. O motivo foi a necessidade de encapsulamento das classes de entidade em relação às respectivas classes de persistência, e o respeito aos princípios de independência de dialogo estabelecidos pelos padrões de arquitetura.
  - Entre as classes de controle do tipo transação e as respectivas classes de persistência, as navegações foram das primeiras para as últimas. Os motivos foram as de ganhos de reusabilidade (portabilidade das classes de persistência) e o respeito aos princípios de independência de diálogos.
- Devem ser adicionadas operações em todas as classes de tal forma a possibilitar o acesso, a consulta e o registro de informações relativas aos atributos das classes.
  - As visibilidades dos atributos e operações adicionados, como definidas no diagrama de classes da fase anterior, devem ser respectivamente privadas e públicas para proporcionar o encapsulamento das classes e maior reuso de código.

O modelo de arquitetura obtido está ilustrado na figura 5.13, aonde é possível observar a representação das entidades computacionais que constituirão o módulo de aprendizagem de Pesquisa Operacional. Também foi construída a interface relativa à principal classe limítrofe, ResolverBancoHistórias, ilustrada na figura 3.4.

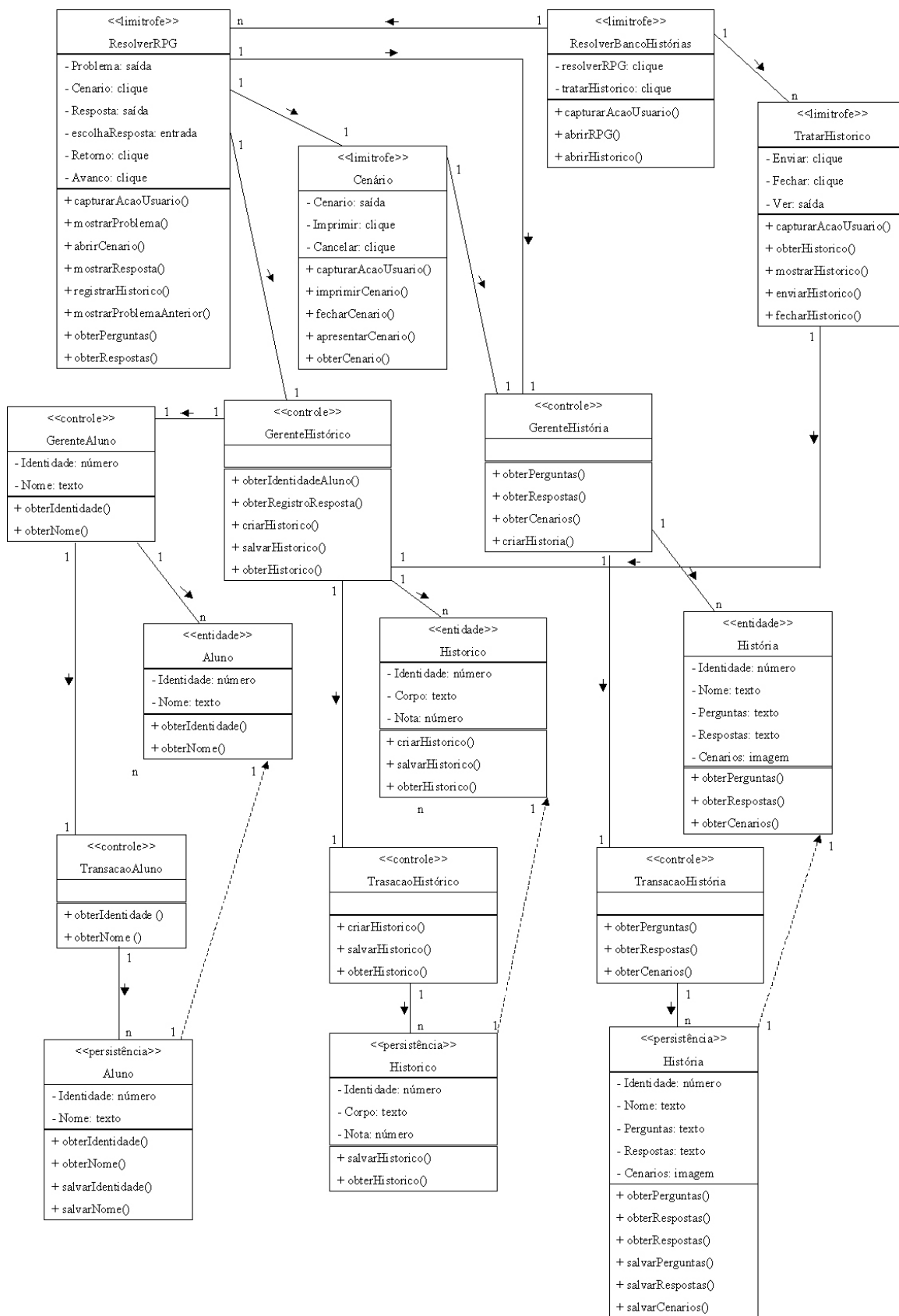


Figura 5.13 – Modelo de arquitetura do módulo de aprendizagem de Pesquisa Operacional

Especificado o modelo de arquitetura, e alterado o diagrama de classe, pode-se fazer as interfaces reais, cujo protótipo já foi apresentado no capítulo 3.

É importante ressaltar que a dinâmica entre as classes, representada através de trocas de mensagens entre suas instâncias, pode ser modelada em diagramas de colaboração. A representação deste diagrama segue o mesmo processo já utilizado pelos outros modelos especificados até aqui. No entanto, sua modelagem não foi especificada por se tratar de detalhes específicos à implementação, que geralmente são próprias a cada programador.

## **Conclusão**

Neste capítulo foi utilizada a metodologia META para apoiar a modelagem e documentação de um módulo de aprendizagem de Pesquisa Operacional. Para tanto, foram construídos e aproveitados conceitos de modelagem já existentes, como o método MACIA, que proporcionou a construção de especificações que foram o ponto de partida para a construção do método (MAPEAR) utilizado na modelagem do módulo. Foi, portanto, observado o reuso, proporcionado pela distribuição dos conceitos em três camadas de abstração e pelo aproveitamento de recursos e trabalhos já realizados através do tratamento de suas especificações.

## 6. Conclusão

### 6.1 Revisão geral

Após estudos realizados em métodos, processos, modelos e arquiteturas utilizados no desenvolvimento de sistemas interativos, observou-se que existem equivalências e que tais equivalências podem ser generalizadas em relação a qualquer arquitetura ou modelos. A partir de cada equivalência, identificou-se neste trabalho conceitos que podem ser utilizados para a especificação de qualquer arquitetura ou modelo de arquitetura. Para tanto foi desenvolvida a metodologia META que descreve os processos de construção de conceitos de modelagem.

Outro aspecto que foi verificado neste trabalho diz respeito à necessidade de modelar sistemas interativos mais flexíveis, seguindo uma organização estrutural para facilitar sua evolução. A evolução de sistemas pode e deve ser apoiada por conceitos relacionados a métodos e modelos. Isto é feito através de novas formas de aplicações dos conceitos já existentes, ou então através da construção de novos conceitos que podem ser refinamentos de conceitos até então disponíveis (evolução de conceitos). Isto funcionou como mais um incentivo para a criação de uma metodologia (META) que permitisse a evolução de sistemas através da aplicação e reconstrução de conceitos.

Outros problemas que foram identificados são relativos ao ensino e à aprendizagem de Pesquisa Operacional. Foi identificada a necessidade de aplicação de um sistema que apoiasse a aprendizagem colaborativa e contextualizada desta disciplina e que fosse fácil de usar. Para tanto era necessário um método ou ferramenta que apoiasse a construção deste tipo de sistema. Baseado nestas dificuldades, foi construído o método MAPEAR, a partir da evolução de outro método (MACIA), aplicado posteriormente na construção de um módulo ergonômico de aprendizagem contextualizada e colaborativa de Pesquisa Operacional.

Com relação às hipóteses levantadas no início deste trabalho, pode-se dizer que:

- A aplicação e construção de recursos de modelagem facilitam a modelagem, a manutenção e a evolução de sistemas interativos. Tais situações foram detectadas

através dos exemplos apresentados no capítulo 5; onde foi possível mostrar a construção de modelos e métodos a partir de outros;

- A estratificação dos conceitos de modelagem em três camadas de diferentes níveis de abstração facilita o uso, a construção e a evolução dos mesmos de maneira a garantir um controle de qualidade. Esta hipótese foi comprovada através de argumentos no capítulo 3 e exemplificada nos capítulos 4 e 5;
- Os conceitos de modelagem mais abstratos servem como guias no desenvolvimento de conceitos menos abstratos. Esta hipótese se baseou na figura original 1.13 e permitiu que os meta-conceitos *padrões de arquitetura*, *modelo de arquitetura*, *estrutura de modelos* e *método* fossem criados no capítulo 4 e;
- A integração de recomendações ergonômicas num método de desenvolvimento de sistemas de aprendizagem e a utilização de ferramentas de comunicação podem ajudar a construir sistemas mais fáceis de usar e aprender. Esta hipótese apoiou a criação do método MAPEAR que considera recomendações ergonômicas para desenvolver o módulo, e a idéia de integrar este módulo a um sistema de educação a distância para viabilizar a aprendizagem colaborativa.

## 6.2 Contribuições

Este trabalho realizou significativas contribuições nas áreas de IHC integrada à engenharia de software através das seguintes realizações:

- a) Discussão sobre processos de desenvolvimento de sistemas interativos e modelos utilizados;
- b) Estudo comparativo e original entre modelos de arquiteturas e entre arquiteturas de um SI;

- c) Construção de uma nova metodologia de desenvolvimento de sistemas interativos (META) de apoio à modelagem, manutenção e evolução de sistemas interativos através da construção e aplicação de conceitos distribuídos em três níveis de abstração;
- d) Definição de conceitos de modelagem que podem ser usados para compor diversos modelos necessários ao projeto de sistemas interativos estruturados e que respeitem ao princípio de independência do diálogo e;
- e) Validação da metodologia META com sua aplicação no apoio ao desenvolvimento de um novo método (MAPEAR) para a construção de um módulo de aprendizagem de Pesquisa Operacional.

### **6.3 Trabalhos futuros**

Os trabalhos futuros se resumem basicamente em dois: implementar e avaliar.

A reutilização dos conceitos de modelagem e a geração e aplicação de qualquer método de desenvolvimento de software poderia ser mais eficiente através de uma ferramenta que permitisse a construção e aplicação dos conceitos. Esta ferramenta não imporiria nenhum modelo ou método para desenvolver um SI. Este seria criado e/ou escolhido pelo projetista. Quanto à contribuição para a área de Pesquisa Operacional, a utilização da ferramenta permitiria automatizar parte dos processos realizados na execução do método MAPEAR. Tal ferramenta proporcionaria um controle adequado na geração e documentação dos conceitos de modelagem, facilitando o reuso de especificações e concretizando as demais vantagens da metodologia. Esta funcionalidade culminaria com a geração do módulo de aprendizagem de Pesquisa Operacional a partir dos conceitos de modelagem. Para se saber se o módulo construído traz as vantagens aqui proferidas de facilidade de aprendizagem e uso, seria necessário ainda a utilização e avaliação do mesmo através da elaboração de um curso de educação a distância usando o CADINET. Outra característica a ser avaliada é capacidade de evolução do módulo.



Outra função desta ferramenta seria a de proporcionar a existência de um banco de conceitos de modelagem, capaz de ser acessado e atualizado através da internet, potencializando as vantagens da proposta aqui descrita e servindo como meio para a validação das ontologias (vocabulário) dos conceitos de modelagem.

## Anexo A

Os tipos de relacionamentos que existem entre os conceitos de modelagem identificam como ocorre o processo de definição e construção dos mesmos. O quadro A.1 ilustra os tipos de relacionamento entre os tipos de conceitos.

	Meta-conceito	Conceito	Instância de Conceito
Meta-conceito	Qualquer tipo de relacionamento	Derivação e herança	Derivação e herança indiretas
Conceito	Abstração	Qualquer tipo de relacionamento	Derivação e herança
Instância de Conceito	Abstração indireta	Abstração	Qualquer tipo de relacionamento

*Quadro A.1 – Tipos de relacionamentos entre os tipos de conceitos*

Uma perspectiva interessante está relacionada aos tipos de relacionamentos existentes quando os conceitos relacionados pertencem a uma mesma camada de abstração ou a camadas diferentes. Quando os conceitos pertencem a uma mesma camada é permitido qualquer tipo de relacionamento, pois o que importa é que eles fiquem definidos em relação ao nível de abstração de suas camadas. Quando os conceitos pertencem a diferentes camadas, estas camadas são adjacentes e o relacionamento ocorre no sentido da camada de maior abstração para a de menor, os relacionamentos observados são de derivação e herança. Num caso semelhante ao anterior, mas com o relacionamento ocorrendo no sentido da camada de menor abstração para a de maior, observa-se o relacionamento inverso ao de derivação e herança que é o de abstração. Quando os conceitos pertencem a camadas diferentes, não adjacentes e o relacionamento ocorre no sentido da camada de maior abstração para a de menor, observa-se o relacionamento de herança e derivação indireta. Tal relacionamento consiste no fato de que características e funcionalidades são herdadas pelo conceito da camada de menor abstração derivado em relação a um ou mais conceito(s) de uma camada intermediária. Este(s) por sua vez são derivados e herda(m) características e funcionalidades de um ou mais conceito(s) da camada de maior abstração. Quando os conceitos pertencem a camadas diferentes, não adjacentes e o relacionamento ocorre no sentido da camada de menor abstração para a de maior, observa-se o relacionamento inverso ao da herança e derivação indireta que é o da

abstração indireta. Tal relacionamento consiste no fato de que características e funcionalidades são abstraídas de conceitos da camada de menor abstração para formar um conceito de uma camada intermediária. Este por sua vez tem as suas características e funcionalidades abstraídas para formar o conceito da camada de maior abstração. Comparando as observações, conclui-se que existe uma simetria no processo de construção dos conceitos, o que traz boas perspectivas para a utilização de um mecanismo semelhante ao da engenharia direta e inversa para (re)constituí-los.

Além dos tipos de relacionamentos, foram identificados alguns dos possíveis relacionamentos existentes entre conceitos pertencentes a um mesmo nível de abstração ou a níveis diferentes. Estes relacionamentos estão exibidos no quadro A.2 a seguir como uma exemplificação em relação ao quadro A.1.

	Meta-conceito	Conceito	Instância de Conceito
Meta-conceito	informa, controla, coordena	é_instância_de (é), tem_instância_de (tem), pertence_a_instância_de (pertence_a)	pertence_a_instância_de, tem_instância_de
Conceito	é_abstração_de (é), é_abstração_de_parte_de (tem), tem_abstração_de (pertence_a)	tem, e_instancia_de, desencadeia, organiza, define, trata, conecta-se, desconecta-se, consulta, modifica	pertence_a_instância_de (pertence_a), é_instância_de (é)
Instância de conceito	tem_abstração_de, é_abstração_de_parte_de	é_abstração_de (é), tem_abstração_de (pertence_a)	composição, associação, generalização

*Quadro A.2 – Os relacionamentos entre os diversos tipos de conceitos*

Através dos exemplos de relacionamentos ilustrados no quadro A.2, observa-se a dificuldade na definição de um padrão em termos de nomes e significados de relacionamentos. No entanto é interessante observar a influência dos relacionamentos *é* e *tem*

em todos os relacionamentos existentes entre conceitos de modelagem pertencentes a níveis diferentes. Isto facilita bastante o estabelecimento de relacionamentos padrões a serem utilizados entre conceitos de níveis diferentes, permitindo o reuso de relacionamentos na geração de conceitos de modelagem. Tal fato permitiria também a geração automática ou semi-automática de conceitos, uma vez que seria possível definir um conceito como molde de parte ou de tudo de um outro conceito.

## Anexo B

Os padrões de arquitetura são meta-conceitos que descrevem como camadas e objetos, que constituem arquiteturas de sistemas interativos, devem estar organizados e se comportar. Eles encontram-se divididos em cinco tipos de estrutura: apresentação, núcleo funcional, controlador de diálogo, adaptador do núcleo funcional e adaptador da apresentação.

O padrão de arquitetura *apresentação* apresenta características e comportamentos da representação externa e dos objetos de interação de uma sistema. Objetos de interação são os dispositivos da interface do usuário que possibilitam a comunicação e o desencadeamento de ações entre o usuário e o sistema. A função deste meta-conceito é dar ao projetista uma orientação quanto à modelagem e programação das funções dos objetos de interação, conforme ilustrado no esquema apresentado na figura B.1.

<p>Apresentação:</p> <ul style="list-style-type: none"> <li>- Problema: O que deve ser feito para construir uma representação externa do sistema ?</li> <li>- Solução: Dar ao projetista um padrão de comportamento dos objetos de interação e da apresentação.</li> <li>- Funções: <ul style="list-style-type: none"> <li>F1 – Receber ou efetivar eventos a nível de representação externa.</li> <li>F2 – Descobrir objetos que sofreram ou efetivaram eventos na representação externa.</li> <li>F3 – Representar externamente os eventos sofridos ou efetivados.</li> <li>F4 – Avisar ao controlador de diálogo, ou adaptador da apresentação, o evento sofrido pelo objeto de interação ou pela apresentação.</li> <li>F5 – Receber do controlador de diálogo, ou adaptador da apresentação, os eventos a serem representados externamente pelo objeto de interação ou apresentação.</li> <li>F6 – Enviar mensagens a determinados objetos visando a mudança dos seus estados externos de acordo com os estados dos objetos que enviaram as mensagens e com o contexto de uso.</li> <li>F7 – Manter certos objetos constituintes do sistema informados sobre ações ou eventos ocorridos em determinados objetos.</li> <li>F8 – Receber mensagens de determinados objetos visando manter a coerência dos estados externos de alguns objetos em relação aos objetos que enviaram as mensagens e ao contexto de uso.</li> </ul> </li> </ul>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

*Figura B.1 – Representação textual do padrão de arquitetura Apresentação*

Um exemplo de obediência ao meta-conceito *apresentação* no comportamento de um objeto de interação é apresentado na figura B.2. Ela ilustra a reação de um objeto ao se passar o cursor do mouse sobre o mesmo. Tal efeito produzido está em concordância com as funções *F2* e *F3* da representação textual deste padrão de arquitetura. Ele reforça a percepção de que o elemento da interface é um objeto de interação.

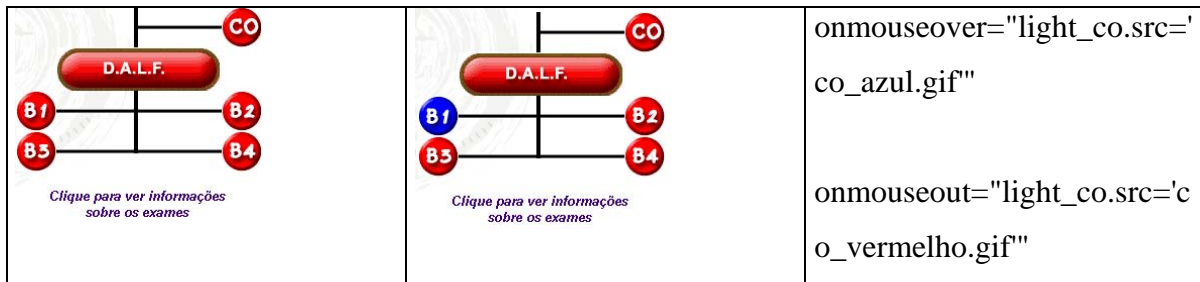


Figura B.2 – Resposta de um objeto de interação à passagem do cursor do mouse e o código utilizado

A figura B.3 ilustra a representação gráfica do meta-conceito *apresentação* através de uma rede semântica que o relaciona a outras entidades conceituais, identificadas em muitas arquiteturas e modelos de arquitetura. O relacionamento *é* entre os conceitos *apresentação* e *conceito* serve para identificar a apresentação como uma representação de algo pertencente ao mundo real. O relacionamento *informa*, do conceito *apresentação* com o conceito *controlador de diálogo* ou com o conceito *adaptador da apresentação*, identifica a troca de informações entre eles (funções F4 e F5). Por sua vez os dados são apresentados através da apresentação, fato representado pelo relacionamento *apresenta* entre os conceitos *dado* e *apresentação* (funções F1 e F3). Os relacionamentos *apresenta* e *captura* entre os conceitos *apresentação* e *dado*, definem a principal funcionalidade da apresentação, que é apresentar e capturar dados da interface do usuário.

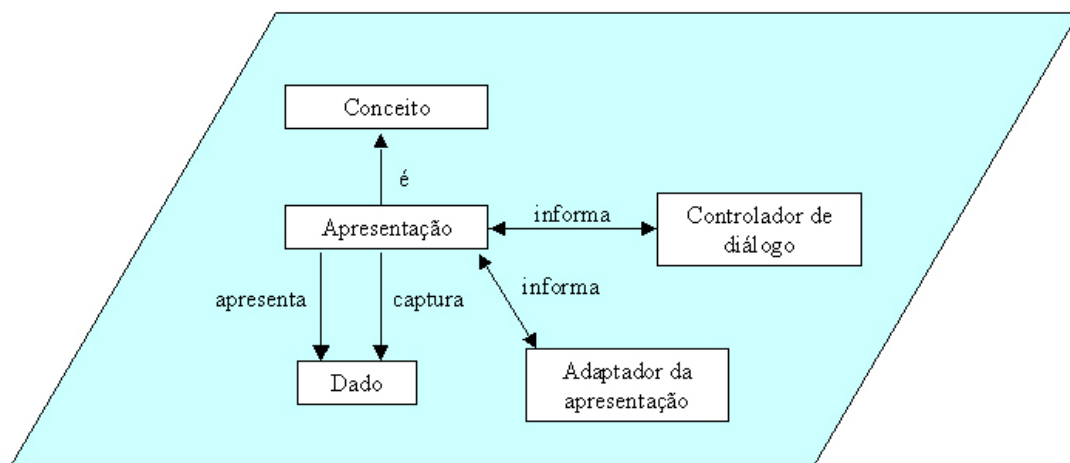


Figura B.3 – Representação gráfica do padrão de arquitetura Apresentação

O padrão de arquitetura *núcleo funcional* representa as características e comportamentos de entidades utilizadas como estruturas de armazenamento e manutenção de informações a serem apresentadas ao usuário ou fornecidas por ele. Sua função é apoiar o

projetista na modelagem e construção de tais entidades. A figura B.4 ilustra a representação textual do núcleo funcional.

Núcleo Funcional:	
-	Problema: O que deve ser feito para construir uma representação interna integrada do sistema?
-	Solução: Dar ao projetista um padrão de modelagem integrada de todas as formas de abstração utilizadas para representar informações e procedimentos utilizados como recursos que possam ter uma vida útil maior do que o tempo de execução do sistema (parte de persistência do sistema).
-	Funções:
	F1 - Definir a organização da estrutura de armazenamento, manutenção e execução de informações e procedimentos do sistema, identificando suas subdivisões ou camadas com seus relacionamentos, funções e composições.
	F2 – Armazenar, consultar e executar informações e procedimentos segundo uma determinada forma.
	F3 – Manter a integridade das informações e dos procedimentos armazenados.
	F4 - Receber chamadas do adaptador do núcleo funcional ou controlador de diálogo às estruturas detentoras dos parâmetros requeridos.
	F5 - Enviar os parâmetros requeridos, em chamadas, para o adaptador do núcleo funcional ou controlador de diálogo.
	F6 - Receber parâmetros, enviados pelo adaptador do núcleo funcional ou controlador de diálogo, para serem alocados e/ou executados em estruturas específicas.

Figura B.4 – Representação textual do padrão de arquitetura Núcleo Funcional

Um exemplo concreto correspondente ao meta-conceito *núcleo funcional* é apresentado na figura B.5. Ela ilustra um recurso de armazenamento de informações utilizado: tabela de registros de banco de dados relacionais (Access). Tal recurso, como qualquer outro designado para o armazenamento de informações, deve caracterizar-se pelo cumprimento das funções existentes no meta-conceito *núcleo funcional*.

Código	nome	responsavel	data_nascime	profissao	endereco	cep	fone_contato
42	Gustavo Frota F		08/01/75	Dentista	Ramos Botelho 60175-265		2342565
43	Ana Paula Sari		24/09/1982	Estudante Univ	Rua Visconde c 60 125-100		2423666
45	Manuel de Saia		05/11/1974	Médico	R. Cal. Mozart 60000200		223.9510
46	Rubemir Teobal		06/04/1982	Estudante	Av. J Nº 1760 - 60 348-290		85 2845136
47	Vanessa de Me		26/10/1981	Estudante	R. Martinho Rei 60411-280		272-1086
50	Thiago Ribeiro f		26/06/1900	Estudante de E	Rua Abilio Mari 60455-470		214-2651
51	Ligia Girao		29/10/1966	jornalista	rua Coronel Lini 60 170-241		264 4492
54	Israel Rocha Br		04/07/1971	Psicólogo	Rua Tenente Oti 60360-170		(85) 2351454
54	Paulo Conte		27 04 71	Advogado	Av Sator Dumas 60160-170		234 5759
55	Rodrigo Almeida Jo	Baltazar dos	22/08/1982	Estudante	R. Nestor Faria 60811620		2732575
56	Walter Magno c		02/02/1976	Analista de Sis	Avenida Praxedi 60 833-690		9963
57	Francisca Evile		04/09/74	Aux. de Enferm	Av. Sgt Herminio		287
58	Paulo Afonso C		18/10/1977	Advogado	Av. Beira Mar n 60165-121		240000
59	Paulo Afonso C		18/10/1977	Advogado	Av. Beira Mar n 60165-121		2488484
60	Ana Carolina Al		31 de janeiro de	Estudante	Rua Barbosa de 60170-020		85 2246727
62	RIHILDA JUNGL ANA MARIA JL		07/02/1962	PUBLICITARIA	RUA TRIUNFAR 04137-110		11-5589/324
63	ALGUSTALO E		12 12 1972	PASTOR	RUA PLUTÃO 60440-200		2269666/26368
64	Manuela Medei		26/11/1982	Estagiária/univ	Av. Sapiranga, 60834-140		278 3604/216
65	Polyanna Enred Jose Pinto De		23.05.85	estudante	Tibrcio Cavalc 60125101		2644804
66	LARA BARBOS REGINA CELIA		27/07/1993	ASSESSORA F	RUA TOMAZ A. 60135100		2643605
67	Mantovanni Col		23/03/64	Juiz de Direito	Av. Desembarg 60170001		244 2031
68	andrea bazzem		26/02/09	administradora	barbosa de frez 60150121		2249590
69	Marilia Eunice		15/04/1984	estudante	rua Sotiano Alb 60130160		2272016
70	ANDREA PONT		05-02-93	ESTUDANTE	RUA CEL ALVI 60130001		2442942

Figura B.5 – Exemplo concreto de uma entidade do Núcleo Funcional e de sua aplicação

A figura B.6 apresenta a representação gráfica do meta-conceito *núcleo funcional* relacionado a outras entidades conceituais observadas em arquiteturas e modelos de

arquitetura. O relacionamento *é* entre os conceitos *núcleo funcional* e *conceito* define o núcleo funcional como sendo uma representação de alguma entidade do mundo real. Tal representação se caracteriza pela troca de informações com o controlador de diálogo ou com o adaptador do núcleo funcional. Isto pode ser evidenciado pelos relacionamentos *informa* entre conceitos *controlador de diálogo*, *adaptador do núcleo funcional* e *núcleo funcional* (funções F4, F5 e F6). Outra característica do núcleo funcional é o fato dele armazenar, consultar e executar dados e tratamentos relacionados à persistência do sistema (funções F1, F2 e F3). Esta característica pode ser observada pelos relacionamentos *armazena*, *consulta* e *executa* entre os conceitos *núcleo funcional*, *dado* e *tratamento*.

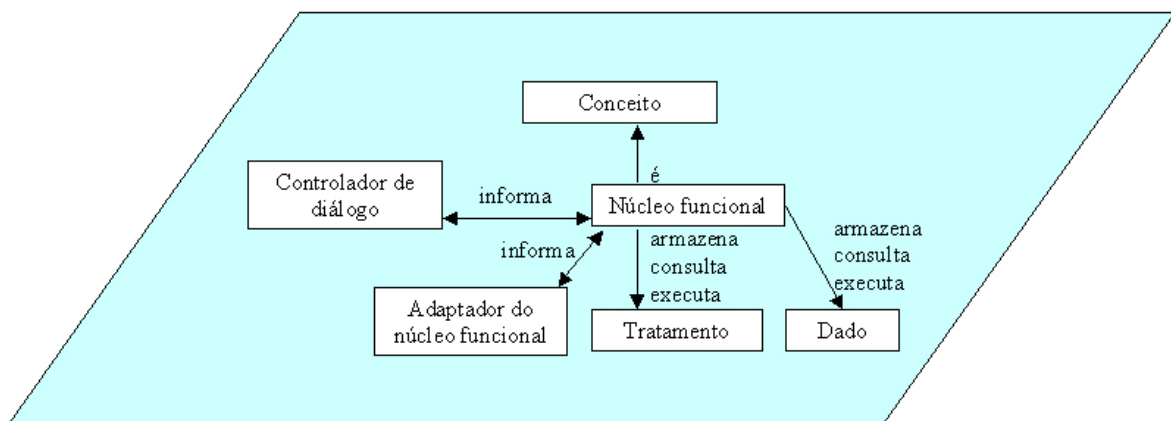
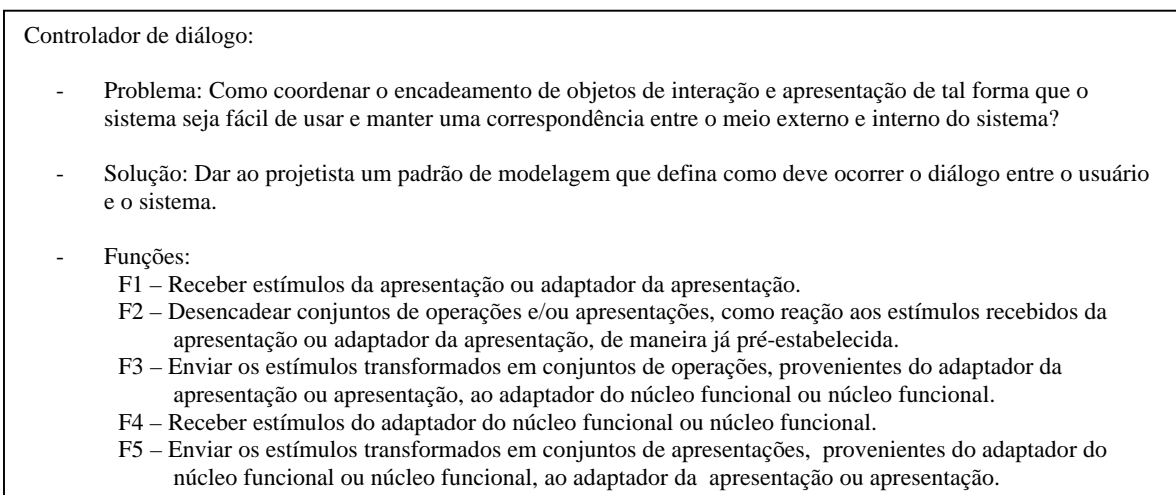


Figura B.6 – Representação gráfica do padrão de arquitetura Núcleo Funcional

O padrão de arquitetura *controlador de diálogo*, que é representado textualmente no esquema da figura B.7, indica como coordenar o encadeamento de objetos de interação e apresentação de tal forma que o sistema seja fácil de usar. Por exemplo, a função F2 da representação textual do meta-conceito *controlador de diálogo* pode ser aplicada no desenvolvimento de um sistema de gerenciamento e controle de uma usina ou indústria. Tal sistema deve possuir um dispositivo capaz de detectar situações de perigo e nortear o usuário através da interface e dos objetos de interação para a resolução do problema. Isto é feito de acordo com o histórico dele e/ou informações sobre sua(s) área(s) de atuação (como operador de máquinas e/ou supervisor de produção) e grau(s) de experiência (como experiente em operação e/ou novato em supervisão). O *controlador de diálogo* também descreve como manter correspondência entre os meios externo e interno do sistema. Um exemplo disto é a aplicação das funções F1, F2, F3, F4 e F5 na geração do sistema de controle e gerenciamento industrial anteriormente descrito. Este sistema também deve possuir um mecanismo capaz de



identificar e evitar a ocorrência de atividades, que possam causar desastres ou prejuízos em relação ao processo de produção e/ou até mesmo à integridade física das pessoas e instalações envolvidas. Isto é feito através da desativação de determinados objetos de interação e da apresentação desta restrição a nível de interface (como a perda de cores, sombreamento e perda de interatividade de parte de uma barra gráfica de controle de geração de energia em uma usina nuclear). A desativação ocorre em resposta aos estados de inviabilidade de execução dos procedimentos assim registrados no núcleo funcional (como o registro de inviabilidade da execução de procedimentos relacionados ao aumento da produção de energia nuclear efetuado em resposta a um vazamento na usina).



*Figura B.7 – Representação textual do padrão de arquitetura Controlador de Diálogo*

Na figura B.8, para os dois exemplos apresentados acima, pode-se identificar os relacionamentos de entidades conceituais com o meta-conceito *adaptador do núcleo funcional*. No primeiro exemplo, observam-se os relacionamentos *controla* e *coordena* do conceito *controlador de diálogo* com as entidades conceituais *apresentação* e *núcleo funcional* como resultado da realização a função F2. No segundo exemplo, a realização das funções F1, F3, F4 e F5 suportam os relacionamentos *informa* do conceito *controlador de diálogo* com as entidades conceituais anteriormente descritas, ou então com os conceitos *adaptador da apresentação* e *adaptador do núcleo funcional*.

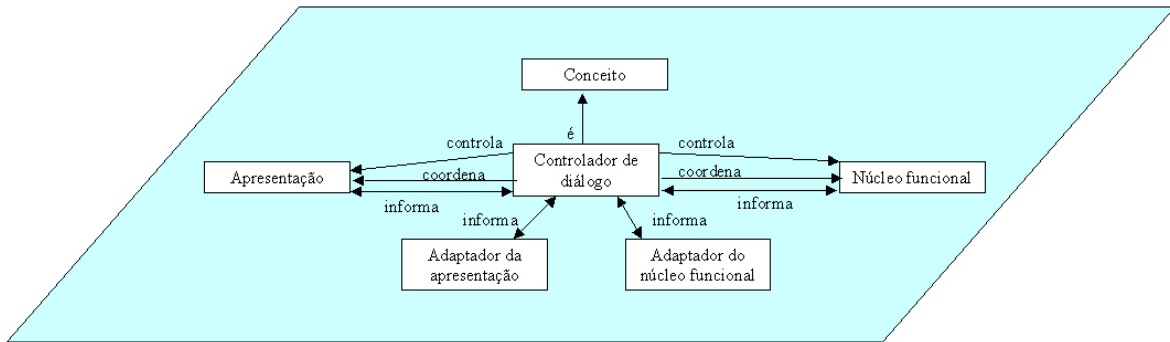


Figura B.8 – Representação gráfica do padrão de arquitetura Controlador de Diálogo

O padrão de arquitetura *Adaptador do Núcleo Funcional* é representado textualmente no esquema da figura B.9. Ele ajuda o projetista a identificar as funcionalidades básicas da camada que deve tornar funcionalmente independentes a apresentação e seus mecanismos de controle e coordenação em relação à parte de persistência e aplicação do sistema. Aplicando suas funções, é possível a realização de alterações nos dispositivos responsáveis pela persistência e aplicação sem modificar o código ou as representações responsáveis pela execução de tratamentos e detecção de eventos a nível de apresentação ou coordenação da apresentação. Tudo isto porque este meta-conceito faz a intermediação entre a apresentação interna do sistema e o controle da interface do usuário.

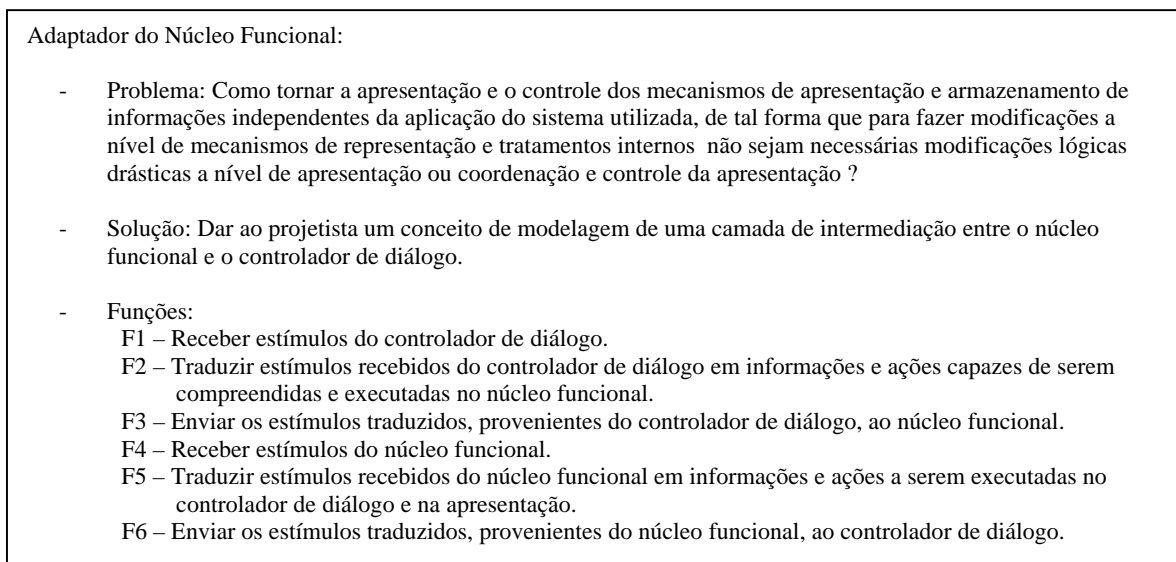
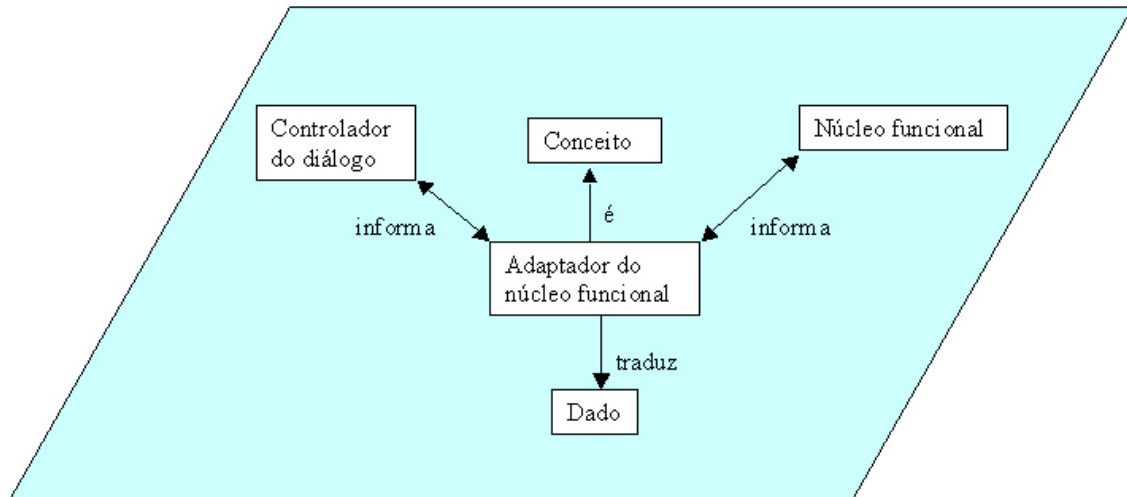


Figura B.9 – Representação textual do padrão de arquitetura Adaptador do Núcleo Funcional

Um exemplo de aplicação do *adaptador do núcleo funcional* é a utilização de suas funções no desenvolvimento de um sistema comercial em um negócio com boas perspectivas

de crescimento. O sistema inicialmente desenvolvido utilizava um banco de dados relacional para o armazenamento de informações textuais de produtos para possibilitar a consulta dos mesmos na loja. Como o negócio tinha boas perspectivas de crescimento, foi planejada a expansão do sistema de tal forma que o mesmo possibilitasse consultas e vendas através da internet. No entanto, para que as vendas fossem efetivadas através da internet seria necessário persuadir o cliente durante a consulta de tal forma que ele tivesse uma boa noção das características e funcionalidade do produto que estava comprando. Para tanto seria necessária a utilização de um banco de dados orientado a objetos, tendo em vista a possibilidade de apresentação de textos, imagens, som e vídeo relacionados ao produtos durante as consultas. Isto influenciou bastante, no desenvolvimento do sistema, o projeto e desenvolvimento de uma camada responsável pela adaptação do mesmo a quaisquer recursos de banco de dados. Com a migração para um banco de dados diferente, a modificação seria feita somente na camada de adaptação, o que reduziria bastante o tempo e a quantidade de trabalho necessários para a modificação do sistema.

Na figura B.10, para o exemplo dado acima, pode-se identificar os relacionamentos entre as entidades conceituais e o meta-conceito *adaptador do núcleo funcional* para a sua definição. No caso do relacionamento *informa* deste meta-conceito com o conceito *controlador de diálogo*, observa-se a realização de troca de informações a ser concretizada através da execução das funções F1 e F6. No que diz respeito ao relacionamento *informa* deste meta-conceito com o conceito *núcleo funcional*, evidencia-se a comunicação entre ambos a ser promovida através da aplicação das funções F3 e F4. Conseqüentemente, uma das funções do adaptador do núcleo funcional é a tradução das informações trocadas com o controlador de diálogo e o núcleo funcional, o que pode ser identificado através do relacionamento *traduz* entre os conceitos *adaptador do núcleo funcional* e *dado* (funções F2 e F5).



*Figura B.10 – Representação gráfica do padrão de arquitetura Adaptador do Núcleo Funcional*

O padrão de arquitetura *Adaptador da Apresentação* consiste em apoiar o desenvolvimento de um sistema em que o código responsável pelo controle e coordenação do diálogo é independente daquele dos dispositivos de apresentação e interação disponíveis. Isto deve ser feito para facilitar modificações a nível de dispositivos de apresentação e interação da interface de acordo com o ambiente de uso do sistema. Por exemplo, um determinado sistema de consulta a locais turísticos foi desenvolvido para ser instalado em diversas plataformas situadas em locais bem diferentes. Então era interessante aproveitar as funcionalidades responsáveis pelo tratamento de informações e tentar modificar somente aquelas responsáveis pela interatividade com o usuário. Foi desenvolvido portanto uma camada de adaptação entre a camada de controle e coordenação do diálogo e a de apresentação e interação do sistema. Na figura B.11 pode-se observar a representação textual do padrão de arquitetura Adaptador da Apresentação.

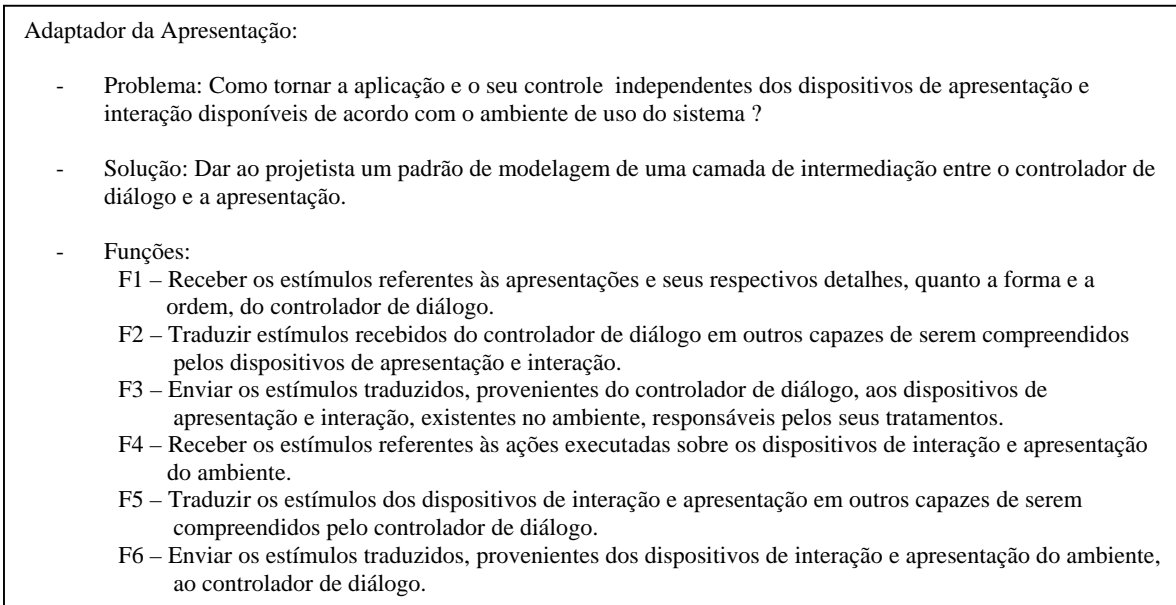


Figura B.11 – Representação textual do padrão de arquitetura Adaptador da Apresentação

Na figura B.12 é apresentada a representação gráfica do meta-conceito *adaptador da apresentação* relacionado a outras entidades conceituais observadas em arquiteturas e modelos de arquitetura. O relacionamento *informa* deste meta-conceito com o conceito *controlador de diálogo*, suporta a troca de informações entre eles através da realização das funções F1 e F6. Também foi observado o relacionamento *informa*, correspondente à troca de informações, deste meta-conceito com o conceito *apresentação* através da execução das funções F3 e F4. O relacionamento *traduz* entre os conceitos *adaptador da apresentação* e *dado* suportam a tradução das informações trocadas (funções F2 e F5).

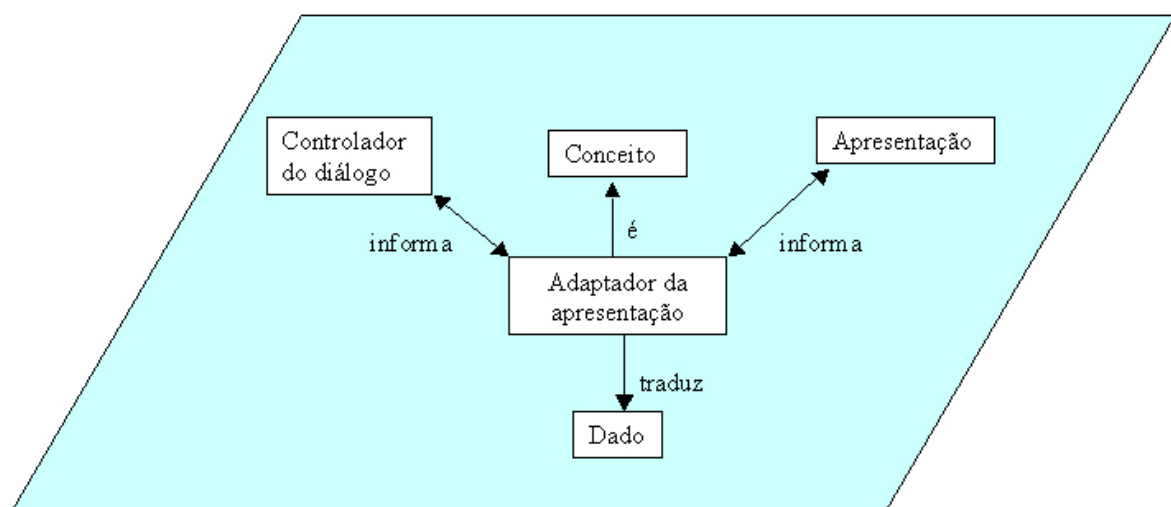


Figura B.12 – Representação gráfica do padrão de arquitetura Adaptador da Apresentação

## Referências

- BACH, Catharine Ferreira, PRATES, Raquel Oliveira, FIGUEIREDO, Rosa Maria de. **GOMSQ: Um método de modelagem de tarefas para sistemas educacionais baseado em comunicabilidade.** In: IHC 2001: IV Workshop sobre Fatores Humanos em Sistemas Computacionais, 2001, 4. Florianópolis – Santa Catarina.
- BASS, Len & COUTAZ, Joëlle. **Developing Software for the User Interface.** USA: Addison-Wesley Publishing Company, May, 1991.
- BERROGGI, Giampero. E.G. The Teacher's Forum: Visual Interactive Decision Modeling (VIDEMO) for Problem Solving – A Hypermedia Concept in Education. **INTERFACES**, The Netherlands, 1999.
- BOOCH, Grady, RUMBAUGH, James, JACOBSON, Ivar. **The Unified Modeling Language User Guide.** USA: Addison-Wesley Publishing Company, August, 2000.
- BORGES, M.A.E., BARANAUSKAS, M.C.C. **Design centrado no usuário ou no aprendiz: uma proposta prática.** In: SBIE98: IX Simpósio Brasileiro de Informática na Educação, 9., 1998, Fortaleza - Ceará.
- BORGES, Marcos A. F. & OLIVEIRA, Simony Pelicer de. **Design de uma ferramenta de apoio ao aprendizado.** In: SBIE 2000: XI Simpósio Brasileiro de Informática na Educação, 11., 2000, Maceió – Alagoas.
- CATAPAN, Araci Hack, CORNELIO FILHO, Plínio, SOUZA, Antonio Carlos de, THOMÉ, Zeina Rebouças Corrêa, CYBIS, Walter de Abreu. **ERGONOMIA EM SOFTWARE EDUCACIONAL: A possível integração entre usabilidade e aprendizagem.** In: IHC'99: II WORKSHOP SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, 2., 1999, Campinas – São Paulo.
- CORRY, M. Overview of Jean Piaget's Theory. **Educational Psychologist**, April 8, 1996. Disponível em: <<http://www.indiana.edu/~educp540/piagsc.html>>. Acesso em: 20 de agosto de 2001.
- COUTAZ, Joëlle. SOFTWARE ARCHITECTURE MODELING FOR USER INTERFACES. . In: MARCINIAK, J. J. **The Encyclopedia of Software Engineering**, 2nd ed. Chichester, England: Wiley & Son Publ., 1993, p. 38-49.
- DEPAULIS, Frabrice, MAIANO, Sabrina, TEXIER, Guillaume. **DTS-Edit: an Interactive Development Environment for Structured Dialog Applications.** In: CADUI'2002 – 4th International Conference on Computer-Aided Design of User Interfaces, 4., 2002, France.
- FEKETE, Alan, GREENING, Tony, KINGSTON, Jeffrey. **Conveying Technical Content in a Curriculum Using Problem Based Learning.** In: Proceedings of 3rd. Australasian Conference on Computer Science Education, 3.,1998, Brisbane, Australia.

- FOWLER, Martin & SCOTT, Kendall. **UML essencial: um breve guia para a linguagem padrão de modelagem de objetos**. Tradução Vera Pezerico e Christian Thomas Price. 2. ed. Porto Alegre: Bookman, 2000.
- FURTADO, Elizabeth. **Mise en oeuvre d'une méthode de conception d'interfaces adaptatives pour des systèmes de supervision à partir des spécifications conceptuelles**. 1997. Thèse de doctorat. Université d'Aix Marseille III. France, 1997.
- FURTADO, Elizabeth, VIEIRA, Flávio Horácio S., LINCOLN, Fernando, MAIA, Márcio. **Auxílio à Solução de Problemas no Processo de Ensino através de Cenários e do Contador de Estórias**. In: SBIE 2000: XI Simpósio Brasileiro de Informática na Educação, 11., 2000, Maceió – Alagoas.
- FURTADO, Elizabeth, LINCOLN, Fernando, VASCO, João, HOLANDA, Raimir. Um Sistema de Aprendizagem Colaborativa de Didática utilizando Cenários. **Revista Brasileira de Informática na Educação**, Florianópolis – Santa Catarina, v. 8, n. 8, p. 53-61, Abr. 2001.
- FURTADO, Elizabeth & CARDOSO JUNIOR, Otoni. **A Framework to Ensure Continuous Accessibility, Acceptability and Usability of Systems**. In: HCI International 2003: 2nd International Conference on Universal Access in Human – Computer Interaction. 10., 2003 Crete, Greece.
- GAVA, Tânia Barbosa de Sales & MENEZES, Crediné Silva de. **Moonline: Um Sistema Multiagentes baseado na Web para Apoio à Aprendizagem**. In: SBIE 2000: XI Simpósio Brasileiro de Informática na Educação, 11., 2000, Maceió – Alagoas.
- INGOLFSSON, Armann & ZALKIND, David. The Teachers' Forum: Two Looks at the Spinner Experiment. **INTERFACES**, Maryland, v. 29, n. 6, p.112 – 126, Nov. - Dec 1999.
- JOHN, Bonnie E. & KIERAS, David E. Using GOMS for User Interface Design and Evaluation: Which technique? **ACM transactions on Computer – Human Interaction**, v. 3, n. 4, p. 287-319, Dec. 1996.
- JOHN-STEINER, Vera & MAHN, Holbrook. Sociocultural Approaches to Learning and Development: A Vygotskian Framework. **Educational Psychologist**, New Mexico, v. 31, n. 3, p. 191-206, 1997. Disponível em: <http://members.home.net/vygotsky/johnsteiner.html>>. Acesso em: 20 de agosto de 2001.
- LABIDI, Sofiane, SILVA, Josenildo C., COUTINHO, Luciano R., COSTA, Nilson S., COSTA, Evandro de Barros. **Agent Based Architecture for Cooperative Learning Environment**. In: SBIE 2000: XI Simpósio Brasileiro de Informática na Educação, 11., 2000, Maceió – Alagoas.
- LIEBMAN, Judith S. Teaching Operations Research: Lessons from Cognitive Psychology. **INTERFACES**, Illinois, v. 28, n. 4, p. 119-132, Jul.-Aug. 1998.

- MACFADZEAN, Elspeth. Creativity in MS/OR: Choosing the Appropriate Technique. **INTERFACES**, Oxfordshire, Sept.-Oct. 1999.
- MACLEAN, A., YOUNG, R.M., BELLOTTI, V.M.E and MORAN, T.P. Questions, options and criteria: elements of design space analysis. **Human-Computer Interaction**, v. 6, n. 3&4, p. 201-250, May 1991.
- MEDEIROS, F.P. **Projeto e Implementação do Módulo TAOS-Graph da Ferramenta ITAOS para Análise e Modelagem da Tarefa**. 2003. Dissertação de Mestrado, Universidade Federal de Campina Grande, Campina Grande – Paraíba, Fevereiro, 2003.
- MENEZES, Crediné Silva de, BAZZARELLA, Luciana Barcelos, CURY, Davidson, CRISTOVÃO, Henrique Monteiro, TAVARES, Orivaldo de Lira. **Formação de Recursos Humanos em Informática Educativa – Uma experiência com Educação a Distância**. In: SBIE 2000: XI Simpósio Brasileiro de Informática na Educação, 11., 2000, Maceió – Alagoas.
- NIGAY, Laurence,. **Conception et modélisation logicielles des systèmes interactifs: application aux interfaces multimodales**. 1997. Thèse de doctorat, Université Joseph Fourier – Grenoble 1, France, 1997.
- NORMAN, V. A. **User Centered System Design: a new perspective human computer interaction**. New Jersey, EUA: Lawrence Erlbaum, 1986.
- OLIVEIRA, Rafael Braga. **O Desenvolvimento de Sistemas Interativos Centrado no Usuário e de Fácil Manutenção Baseado no Processo RUP Usando a UML**. 2002. Monografia, Universidade de Fortaleza, Fortaleza – Ceará, Junho, 2002.
- PRESSMAN, Roger S. **ENGENHARIA DE SOFTWARE**. Tradução José Carlos Barbosa dos Santos. São Paulo: Makron Books, 1995.
- RAMOS, Edla Maria Faust. **ANÁLISE ERGONÔMICA DO SISTEMA HIPERNET BUSCANDO O APRENDIZADO DA COOPERAÇÃO E DA AUTONOMIA**. 1996. Tese de doutorado, Universidade Federal de Santa Catarina, Florianópolis – Santa Catarina, Novembro, 1996. Disponível em: <<http://www.eps.ufsc.br/teses96/edla/index/index.htm>>. Acesso em: 22 de Abril de 2003.
- REIS, Ângela B. **UM AMBIENTE PARA O AGENTE PEDAGÓGICO DE APRENDIZAGEM COLABORATIVA EM HARMONIZAÇÃO ECOLÓGICA – APACHE**. Abril, 2001. Dissertação de Mestrado, Universidade Federal da Paraíba, Campina Grande – Paraíba, Abril, 2001.
- SAVIDIS, Anthony & STEPHANIDIS, Constantine. The Unified User Interface Software Architecture. In: \_\_\_\_\_. **User Interfaces for All – Concepts, Methods, and Tools**. New Jersey, USA: Lawrence Erlbaum Associates Publishers, 2001, cap. 20, p. 389-415.



- SCAPIN, Dominique L. & PIERRET-GOLBREICH, Christine. **TOWARDS A METHOD FOR TASK DESCRIPTION: MAD.** In : Work With Display Units 89, 2., 1990, Amsterdam, The Netherlands: Northe-Holland.
- SCHANK, Roger C. & CLEARY, Chip. **Engines for Education.** New Jersey, USA: Lawrence Erlbaum Associates Publishers, 1995.
- SCHRAGE, Linus. **An Optimization Modeling System.** South San Francisco, USA: Scientific Press, 1991.
- SILVA, Ermes Medeiros da, SILVA, Elio Medeiros da, GONÇALVES, Valter, MUROLO, Afrânio Carlos. **PESQUISA OPERACIONAL.** São Paulo: Atlas, 1998.
- TARBY, Jean-Claude. **Gestion Automatique du Dialogue Homme-Machine à partir de Spécifications Conceptuelles.** 1993. Thèse de doctorat, Université Toulouse I, France, 1993.
- VANDERDONCKT, J., FURTADO, M. E. S., FURTADO, J. J. V. P., WILLIAM, D., **Multi-Model and Multi-layer Development of User Interfaces.** In: Multiple User Interfaces, ed. John Wiley, 2003.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)