

7 CONCLUSÕES E CONSIDERAÇÕES FINAIS

Após a descrição detalhada de todo o processo de modelagem nos capítulos anteriores, usando a técnica proposta sob a estratégia dirigida por comportamento, podem ser apresentadas algumas conclusões e considerações finais. Para isto são indicados os objetivos atingidos, as contribuições, as comparações com outras técnicas e os trabalhos futuros.

Este capítulo final foi dividido como segue: a seção 7.1 mostra o cumprimento dos objetivos levantados na introdução para o desenvolvimento do presente trabalho; a seção 7.2 apresenta as contribuições gerais e as particularidades do presente trabalho; a seção 7.3 apresenta algumas considerações gerais em relação ao processo descrito neste trabalho; a seção 7.4 mostra outras técnicas que seguem a estratégia dirigida por comportamento, e descreve as principais semelhanças e diferenças com a técnica proposta e a seção 7.5 apresenta os lineamentos para futuros desenvolvimentos que podem ser realizados a partir do trabalho até aqui apresentado.

7.1 O CUMPRIMENTO DOS OBJETIVOS

Pode-se concluir que o principal objetivo deste trabalho foi plenamente atingido. Foi detalhado todo um processo de modelagem conceitual de sistemas de informação de um ponto de vista dinâmico/comportamental, e um primeiro passo dentro do *design* orientado a objetos com a derivação de um modelo estrutural orientado a objetos.

Quanto aos objetivos secundários, estes também foram satisfeitos:

1. O estado da arte em MOO foi descrito e analisado criticamente, detectando-se alguns problemas. Os principais problemas indicados foram:
 - o de argumentar a favor da orientação a objetos, que seria uma visão mais “natural” da realidade,
 - o encapsulamento no início da modelagem favorece a continuidade estrutural com o custo de colocar a MOO mais perto do *design* e
 - a ênfase majoritária na estratégia dirigida por dados na MOO, o que traz como consequência uma sobrevaloração do aspecto estático em relação ao aspecto dinâmico dos sistemas.
2. A proposta resolve alguns destes problemas, porque:
 - os objetos deixam de ser *a forma* de ver a realidade, mudando-se da visão orientada a objetos para uma perspectiva dinâmico/comportamental,

- o encapsulamento é realizado como último passo do processo de modelagem ao derivar o modelo estrutural de objetos e
 - a estratégia dirigida por comportamento é usada para conduzir a técnica de modelagem proposta, com isto, a estrutura do sistema de objetos é derivada do comportamento que este apresenta.
3. Foi criada uma síntese, especificamente no modelo da visão global, de uma visão sistêmica, que permite contextualizar e conceber o sistema sob modelagem (no diagrama de contexto do sistema) e de uma visão dinâmica, que permite definir as relações do sistema sob modelagem com outros sistemas na forma de estímulos e respostas (no diagrama da visão global).
 4. Foi desenvolvida uma extensão denominada *High-Level Statecharts* (HLS) a partir do formalismo *statecharts*, que fornece maior poder de expressão, necessário para a modelagem conceitual de sistemas de informação na perspectiva dinâmica/comportamental. Notações específicas para cada extensão foram desenvolvidas.
 5. Conforme o processo de modelagem conceitual proposto, o encapsulamento é introduzido como um primeiro passo do *design*, e apenas com o propósito de obter um modelo estrutural orientado a objetos. Esta derivação não é necessária se a implementação não segue o paradigma da orientação a objetos.
 6. Com as extensões propostas no formalismo HLS (particularmente no que se refere a variáveis, eventos, ações e conjuntos de estados concorrentes) viabiliza-se uma derivação relativamente simples e direta de um modelo estrutural de objetos.

7.2 AS CONTRIBUIÇÕES E AS PARTICULARIDADES DA PROPOSTA

Em termos gerais, o presente trabalho contribui nos seguintes assuntos:

- **Definição do processo de modelagem conceitual:** O trabalho contribui principalmente na definição do *processo* da MOO, isto é, a forma em que é conduzida a modelagem sob a estratégia dirigida por comportamento. Contrariamente à maioria das técnicas descritas na literatura, que privilegiam os aspectos notacionais e sintáticos dos modelos.
- **Mudança na ênfase do estático para o dinâmico:** O trabalho contribui na viabilização da mudança da ênfase nos objetos, no aspecto estático, na estrutura dos objetos e na estratégia dirigida por dados da MOO; esta mudança acontece para o comportamento, o aspecto dinâmico, as relações dinâmicas e a estratégia dirigida por comportamento da MOO.
- **Amadurecimento da MOO sob a estratégia dirigida por comportamento:** O trabalho fornece uma proposta específica, seguindo a estratégia dirigida por com-

portamento, ao propor um processo de modelagem conceitual completo de sistemas do ponto de vista dinâmico/comportamental e um primeiro passo de *design* com um modelo estrutural orientado a objetos.

- **Demonstração de que o comportamento pode preceder a estrutura dos objetos:** O trabalho demonstra empiricamente a idéia de que o comportamento exibido por um sistema pode preceder a estrutura dos objetos que podem representá-lo. É esta idéia que está por trás da estratégia dirigida por comportamento.
- **Identificação de padrões de comportamento e interação:** O trabalho contribui para a comprovação de que é possível identificar os objetos como padrões de comportamento (modelo de ciclos de vida), e as associações estáticas entre os objetos como padrões de interação (estímulos e eventos nos modelos dinâmicos). Em outras palavras, que os modelos estáticos de objetos são uma visão (até certo ponto arbitrária) dos modelos dinâmicos do sistema.
- **Determinação das propriedades dos objetos a partir das relações:** O trabalho contribui mostrando que os atributos dos objetos identificados são determinados pelas associações estáticas nas quais ele participa. Dado que estas associações derivam-se dos estímulos e eventos, muitos atributos são identificados mais genericamente como tipos de objetos associados. Conseqüentemente, o modelo estrutural de objetos para um sistema qualquer, obtido com a técnica proposta, será sempre mais “plano”, isto é, com menos atributos e mais classes do que um modelo obtido usando uma estratégia dirigida por dados, em que os atributos sejam representados exclusivamente como tais.
- **Introdução da concorrência na modelagem conceitual:** O trabalho contribui para a modelagem *conceitual* de sistemas no sentido de favorecer o paralelismo interno nos modelos, evitando a imposição de seqüências desnecessárias de ações ou operações do sistema. Isto é possível graças ao alto grau de concorrência introduzido na modelagem com HLS.

Em termos mais específicos, o presente trabalho possui as seguintes particularidades:

- **Enfoque de estado:** Usa particularmente o enfoque de estado para identificar objetos, contrariamente à maioria das técnicas encontradas na literatura, que usam enfoques de abstração para esta mesma identificação.
- **Estados atômicos passivos:** Aumenta a precisão da modelagem dinâmica ao reconhecer apenas estados atômicos passivos nos diagramas HLS.
- **Objetos sem concorrência interna:** Define objetos de comportamento mais simples ao supor que estes não apresentam concorrência interna. A forma em que é realizada a integração dos diagramas HLS (ou partes concorrentes dos mesmos) do modelo de processos no modelo de ciclos de vida impede esta concorrência interna.
- **Estímulo tratado concorrentemente:** Introduce o conceito de *tratamento concorrente do estímulo* ao decompor o estímulo, em função do número e dos tipos de

parâmetros que apresente, em conjuntos de eventos que são tratados concorrentemente no modelo de processos. Desta forma, não é imposta nenhuma sequência para o tratamento do estímulo pelo sistema e, especificamente, pelo processo respectivo.

- **Resposta gerada concorrentemente:** Introduz também o conceito de *geração concorrente da resposta* ao compor a resposta com base em um conjunto de ações que são executadas concorrentemente no modelo de processos. As ações são distribuídas como operações em diferentes objetos, contribuindo parcial e concorrentemente à geração da resposta do sistema. Desta forma, é evitada qualquer sequência desnecessária para a geração desta resposta.
- **Herança definida por comportamento:** Utiliza o comportamento ou, mais especificamente, os comportamentos alternativos e independentes que as classes podem assumir como base para a especialização das superclasses e para a definição das hierarquias de herança, contrariamente à maioria das propostas de MOO na literatura, que usam principalmente a herança como um mecanismo de “economia” conceitual de atributos e/ou operações¹.
- **Estímulo como fonte semântica de associações:** Considera as variáveis usadas como parâmetros nos estímulos (e também presentes nos eventos componentes dos mesmos) como a principal fonte de associação dos objetos. Dois ou mais parâmetros que fluem juntos em um mesmo estímulo definem uma associação estática entre os objetos identificados a partir deles.
- **Identificadores dos objetos:** Incluindo-se os identificadores para os tipos de dados compostos (que se correlacionam com as variáveis de referência dos processos) chega-se, naturalmente, à conclusão de que estes mesmos identificadores e variáveis de referência tornam-se os identificadores dos objetos, contrariamente a algumas propostas de técnicas de MOO na literatura, em que a inclusão de identificadores para os objetos parece mais uma imposição notacional do que uma clara necessidade semântica.

7.3 CONSIDERAÇÕES GERAIS

Algumas considerações gerais podem ser mencionadas em relação ao presente trabalho:

- **A proposta de modelagem e *design* é uma técnica incompleta de MOO:** A rigor, a proposta de modelagem não é uma técnica de MOO completa. Contudo, as extensões requeridas são menores, dado que a informação necessária para incluí-las já está contida nos modelos resultantes. Estas extensões são: algumas descrições adicionais de comportamento de tipos de objetos, um modelo de colaborações entre

¹ Diferente é a situação no *design* ou na programação orientados a objetos, em que a herança é a base para o reuso.

os objetos e a aplicação do mecanismo de agregação para o modelo estrutural de objetos.

- **As consultas não influem na estrutura dos objetos:** É interessante destacar que todos os estímulos de consulta e, conseqüentemente, os processos que desempenham estas consultas não contribuem em nada para a derivação do modelo estrutural de objetos, isto é, não acrescentam nenhuma informação estática ao sistema na forma de associações deriváveis ou redundantes. Como se está modelando conceitualmente o sistema, a estrutura de objetos não deveria ver-se afetada por considerações de implementação de consultas.
- **O refinamento do modelo estrutural de objetos:** O modelo estrutural de objetos finalmente derivado não é definitivo, ele pode e deve ser refinado utilizando-se os conceitos próprios da modelagem estrutural de objetos, particularmente para identificar composições que, na forma atual da técnica proposta, não são destacadas, e para definir certas classes como atributos de outras.
- **As condições na fase de *design*:** Ao avançar para os seguintes passos na fase de *design* dentro do paradigma da orientação a objetos, é necessário definir os métodos para as operações indicadas no modelo estrutural. Estas operações foram derivadas de ações que eram realizadas nas transições dos processos. Nestas mesmas transições também existem algumas condições, as quais podem servir para definir pré-condições para os respectivos métodos. Adicionalmente, as condições de consistência intraproceto e interprocessos podem requerer mensagens adicionais entre os objetos.

7.4 OUTRAS TÉCNICAS DIRIGIDAS POR COMPORTAMENTO

Como foi indicado no capítulo 1, existem poucas técnicas de MOO conduzidas sob a estratégia dirigida por comportamento. Duas são as principais:

- *Object Behavior Analysis* (OBA) de Rubin & Goldberg [RUB 92] e
- a técnica de análise da metodologia *Object-Oriented Software Engineering* (OOSE) de Jacobson et al. [JAC 92].

Ambas as técnicas são brevemente descritas a seguir e comparadas com a proposta de modelagem.

7.4.1 Técnica OBA

A técnica OBA [RUB 92] tem como idéia central que o comportamento esperado do sistema permite identificar os objetos que iniciam e participam deste comportamento.

7.4.1.1 O Processo de Modelagem

A técnica OBA sugere cinco passos interativos:

1. **Definir o contexto da análise:** Trata com metas e objetivos, recursos, áreas de atividades e plano preliminar da análise.
2. **Entender o problema:** Consiste na definição de cenários de interação para derivar *scripts* de comportamentos. Adicionalmente, são desenvolvidos glossários de partes e serviços. Os atributos são derivados e também definidos em um glossário.
3. **Definir os objetos:** São definidos como objetos as partes envolvidas nos *scripts*, sempre que a parte seja um participante interno do sistema, e não esteja simplesmente iniciando um comportamento. Estes objetos são descritos nos *Cartões de Modelagem de Objeto*.
4. **Classificar os objetos e identificar os relacionamentos:** São definidos os relacionamentos contratuais dos objetos através dos serviços fornecidos/contratados. Os objetos são organizados hierarquicamente de acordo com os conceitos de abstração (generalização), especialização e fatorização.
5. **Modelar a dinâmica do sistema:** São definidos os estados de cada objeto em um glossário. São modelados os estados e os eventos que disparam as transições em diagramas de ciclo de vida. Adicionalmente, as atividades dos *scripts* devem ser sequencializadas/paralelizadas utilizando alguma notação dinâmica adequada.

7.4.1.2 Os Modelos Resultantes

Os modelos resultantes da aplicação da técnica OBA são:

- **Cartões de modelagem de objetos:** Cartões que descrevem, entre outros, os seguintes itens dos objetos: nome, herança, lista de atributos (nome e *script* em que é usado), lista de serviços fornecidos (nome e *script* em que é usado) e lista de serviços contratados (nome, objeto ao qual pertence o serviço e *script* em que é usado).
- **Diagrama de ciclo de vida de objetos:** *Statecharts* tradicionais que descrevem o comportamento padrão de uma instância de cada tipo de objeto.
- **Diagrama de relacionamentos contratuais:** Diagrama que descreve os relacionamentos entre objetos na forma de contratos do tipo *fornece informação e requer ação*.
- **Diagrama de relacionamentos organizacionais:** Diagrama que mostra as hierarquias de herança. Cada tipo de objeto tem seus serviços indicados.
- **Diagrama de seqüenciação de operações:** Diagramas que representam as atividades, descritas nos *scripts*, em forma seqüencial/paralela. Não existe um modelo

definido para tal propósito, podem ser usados *statecharts* [HAR 87] ou redes de Petri [HEU 90].

- **Glossário de atributos:** Glossário que descreve os atributos em termos do nome, definição, contrato da parte que utiliza o atributo, serviços acessador e atualizador do atributo, tipo de valoração (mono ou multivalorado), domínio e a determinação ou não pelo atributo dos estados para a parte.
- **Glossário de estados:** Glossário que descreve o nome dos estados, a definição em função de atributos, uma descrição e a referência aos *scripts*.
- **Glossário de nomes de partes:** Glossário que descreve as partes (iniciadores e/ou participantes de cada comportamento descrito nos *scripts*) em termos do nome, definição, *script* em que aparece a parte e papel desempenhado (iniciador, participante ou iniciador/participante).
- **Glossário de serviços:** Glossário que descreve os serviços em termos do nome, definição, parte e *script* em que aparece serviço.
- **Scripts do sistema:** Tabelas de descrição dos comportamentos esperados do sistema. Cada *script* possui um cabeçalho que inclui, entre outros, o nome, as pré-condições para a aplicação do *script* e as pós-condições resultantes da conclusão do *script*. Os comportamentos são descritos com um iniciador (que inicia a ação a ser descrita), a ação realizada, o participante (que interage com o iniciador, recebendo a ação) e o serviço (que é realizado pelo participante).

7.4.1.3 O Comparativo

As principais equivalências entre a técnica OBA e a proposta de modelagem são mostradas na tabela 7.1.

Em síntese, as principais semelhanças entre ambas as técnicas são:

- **Construção de ciclos de vida:** Ambas as técnicas constroem os ciclos de vida a partir dos componentes contidos no modelo de comportamento inicial do sistema.
- **Descrição dos atributos:** Ambas as técnicas usam dicionário de dados ou glossários e cartões de modelagem para descrever os atributos.

TABELA 7.1 - Tabela comparativa de conceitos da técnica proposta em relação à técnica OBA.

CONCEITO	EQUIVALENTE NA TÉCNICA OBA	OBSERVAÇÕES
ação	serviço	
agente externo	parte apenas com o papel de iniciador	
associações estáticas		deriváveis dos relacionamentos contratuais
atributos	atributos	
concorrência nos processos	seqüenciação/paralelização de atividades	representação com <i>state-charts</i> ou redes de Petri
dicionário de dados	glossários de atributos e cartões de modelagem de objetos	
estímulo	ação	modelada no <i>script</i>
evento	implícito no passo das pós-condições de um <i>script</i> para as pré-condições de outro <i>script</i>	corresponde mais precisamente à transição
composição da resposta		
decomposição do estímulo		
hierarquia de herança	derivadas em função de atributos e serviços	
identificação de objetos	participantes que desempenham algum serviço contratado	
identificador único		
integração de ciclos de vida	integração de (ou partes de) <i>scripts</i> em ciclos de vida dos objetos	
limites do sistema	implícito ao definir os iniciadores que não são participantes	
parâmetros	incluídos nas ações	
processo		
resposta		
supra-sistema		

As principais diferenças dizem respeito a:

- **O modelo fundamental:** Na técnica OBA, são os *scripts* de comportamento que tratam com muito detalhe cada ação iniciada desde o exterior do sistema. Os *scripts* devem posteriormente ser particionados de acordo a mudanças de estado que ocorram nos objetos participantes. A seqüência/concorrência na execução destas partições deve ser definida posteriormente. Os HLS apresentam a vantagem de tratarem todos os elementos (estímulos, estados e respostas) já em forma seqüencial/concorrente.
- **O nível de abstração na descrição do comportamento do sistema:** Na técnica OBA, a descrição do comportamento do sistema é tratada no mesmo nível de abstração dos serviços dos objetos potenciais. Na técnica proposta, a abstração é maior ao considerar os processos como receptores dos estímulos. Esta nível de abstração parece mais apropriado face a modelagem conceitual.
- **A identificação dos objetos:** Na técnica OBA, a identificação dos objetos é feita através dos participantes que desempenham serviços. Na técnica proposta, é feita a partir das variáveis tratadas pelo sistema diminuindo com isto a influência do modelador, que na técnica OBA deve definir inicialmente o objeto que interage com o iniciador de uma ação.

- **A interação com os agentes externos:** A técnica OBA centra-se em interações unidirecionais entre o iniciador (agente externo potencial) e o sistema. Na técnica proposta, esta interação é bidirecional, isto é, vai do estímulo do emissor até a resposta para o receptor fornecendo maior riqueza de interação do sistema com o ambiente.

Em [BUS 94e] é apresentada a aplicação da técnica OBA ao problema da IFIP. A figura 7.1 apresenta o exemplo específico de um *script* para a submissão de um artigo dentro deste problema.

Nome do Script	Paper.Submission		
Autor	GBR		
Pré-condição	true		
Pós-condição	paper_exists(P), author_exists(A ₁ , A ₂ , ..., A _n)		
Referência	Core Activity Area - Program Administration		
Iniciador	Ação	Participante	Serviço
Program Committee	include paper P of authors A ₁ , A ₂ , ..., A _n	Paper	inclusion
Paper	verify existence of A ₁	Author	verification
Paper	include A ₁ if not exists	Author	inclusion
Paper	verify existence of A ₂	Author	verification
Paper	include A ₂ if not exists	Author	inclusion
...	...	...	...
Paper	verify existence of A _n	Author	verification
Paper	include A _n if not exists	Author	inclusion

FIGURA 7.1 - Exemplo específico de um *script* para a submissão de um artigo no problema da IFIP, conforme a técnica OBA.

7.4.2 Técnica de Análise na Metodologia OOSE

A metodologia OOSE [JAC 92] (uma simplificação da metodologia *Objectory*) cobre as fases de análise, *design* e implementação. Trata de modelos de requisitos, de análise, de *design*, de implementação e de teste.

A técnica de análise, dentro desta metodologia, baseia-se na idéia de que o modelo do sistema pode ser construído em torno das interações dos diferentes tipos de usuários com o sistema.

7.4.2.1 O Processo de Modelagem

A técnica de análise da metodologia OOSE sugere os seguintes passos:

1. **Identificação de atores:** Definição dos tipos de usuários do sistema ou papéis que os mesmos podem desempenhar em sua interação com o sistema. Os atores são externos ao sistema, fixando assim os limites do mesmo.

2. **Construção dos casos de uso:** Para cada ator identificado é construído um caso de uso do sistema que descreve um aspecto da funcionalidade do sistema iniciada pelo ator. É um modelo que tipifica o “diálogo” entre o ator e o sistema.
3. **Descrição das interfaces:** Se as interfaces no modelo de casos de uso é muito complexa, pode ser necessário descrevê-las com o propósito de validá-las com os usuários.
4. **Modelagem dos objetos do domínio do problema:** Construção de uma visão lógica do sistema usando os objetos do domínio do problema. Os objetos podem ser identificados a partir do modelo de casos de usos ou diretamente de uma descrição textual disponível dos requisitos do sistema.
5. **Refinamento do modelo de requisitos:** Revisão do modelo de requisitos para melhorar o reuso e prepará-lo para o passo seguinte. O principal refinamento é a geração de casos de uso *abstratos*. Partes comuns dos casos de uso *concretos* (gerados no passo 2) são fatorizados nestes casos de uso abstratos.
6. **Construção do modelo de análise:** Consiste na distribuição do comportamento descrito no modelo de casos de uso entre os objetos e suas associações estáticas. Estes objetos podem ser de interface (traduzem as entradas dos atores em eventos para o sistema e as operações em saídas do mesmo), entidades (objetos do domínio do problema aos quais são acrescentados atributos) e de controle (recebem comportamento que não é de interface, nem é do domínio do problema).

7.4.2.2 Os Modelos Resultantes

Os modelos resultantes da aplicação da técnica de análise da metodologia OOSE são dois:

1. **Modelo de requisitos:** Consiste em:
 - **Descrições de interfaces:** Descrições detalhadas das interfaces necessárias no modelo de casos de uso.
 - **Modelo de casos de uso (*use case model*):** Modelo que mostra o uso do sistema por atores (papéis que os usuários podem desempenhar). Cada caso de uso descreve um fluxo completo de eventos, que é iniciado pelo ator, da perspectiva do usuário do sistema.
 - **Modelo do domínio do problema:** Modelo que descreve os objetos do sistema e suas associações de herança derivadas do domínio do problema.
2. **Modelo de análise:** Modelo derivado do modelo de casos de uso que descreve as associações de conhecimento (*acquaintance associations*) entre os três tipos de objetos (de interface, entidades e de controle).

7.4.2.3 O Comparativo

As principais equivalências entre a técnica de análise da metodologia OOSE e a proposta de modelagem são mostradas na tabela 7.2.

TABELA 7.2 - Tabela comparativa de conceitos da técnica proposta em relação à técnica de análise da metodologia OOSE.

CONCEITO	EQUIVALENTE NA TÉCNICA DE ANÁLISE DA METODOLOGIA OOSE	OBSERVAÇÕES
ação	operação	
agente externo	ator	mais orientado a papéis de usuários
associações estáticas	associações de conhecimento	
atributos	atributos	
concorrência nos processos	implícito no modelo de casos de uso	não é particularmente privilegiada
dicionário de dados	implícito no modelo de análise	definições descritas como associações
estímulo	<i>input</i> do sistema	
evento	evento	
composição da resposta	feita por objetos de interface	não descrita na fase de análise
decomposição do estímulo	feita por objetos de interface	não descrita na fase de análise
hierarquia de herança	baseada em atributos e operações	
identificação de objetos	de interface (interfaces de comunicação entre o usuário e o sistema), entidades (informação mantida no sistema) e de controle (para cada caso de uso)	todos derivados do modelo de casos de uso
identificador único		
integração de ciclos de vida		
limites do sistema	realizada conjuntamente com a definição dos atores	
parâmetros	implícito nos <i>inputs</i> e <i>outputs</i>	
processo	implícito no caso de uso	
resposta	<i>output</i> do sistema	
supra-sistema		

Em síntese, as principais semelhanças são as seguintes:

- **O modelo estrutural de objetos:** Ambas as técnicas descrevem os objetos e as associações estáticas derivadas dos modelos de interação de maneira semelhante.
- **Os níveis de abstração nas comunicações:** Ambas as técnicas fazem uma distinção entre as comunicações externas (*input* e estímulo, *output* e resposta) e as internas (eventos) ao sistema.
- **A definição dos limites do sistema:** Ambas as técnicas definem agentes externos ou atores, tendo como um dos propósitos o de delimitar o sistema como a linha imaginária entre estes agentes ou atores e o interior do sistema.

As principais diferenças são encontradas em:

- **O modelo fundamental:** Na técnica de análise da metodologia OOSE, o modelo fundamental é o de casos de uso, que centra-se na comunicação usuário-sistema de maneira informal. O modelo HLS é mais formal, rico em elementos de modelagem e poderoso em termos de expressividade do que o *use case model*.
- **O desenvolvimento do modelo estrutural:** Na técnica de análise de OOSE, o modelo é desenvolvido como uma combinação da modelagem de conceitos do domínio do problema e da derivação de objetos usados no modelo de casos de uso. Na técnica proposta neste trabalho, a estrutura é integralmente derivada dos modelos HLS diminuindo assim a influência de modelador na identificação dos objetos.
- **Categorização dos objetos:** Na técnica de análise OOSE, os objetos do modelo da análise são categorizados em objetos de interface, entidade e de controle. Esta distinção não é feita na técnica proposta, porque se julga ser esta uma atividade de *design*.

A figura 7.2 mostra um exemplo de aplicação do modelo de casos de uso da técnica de análise OOSE ao problema da IFIP. Trata-se da submissão de um artigo dentro deste problema.

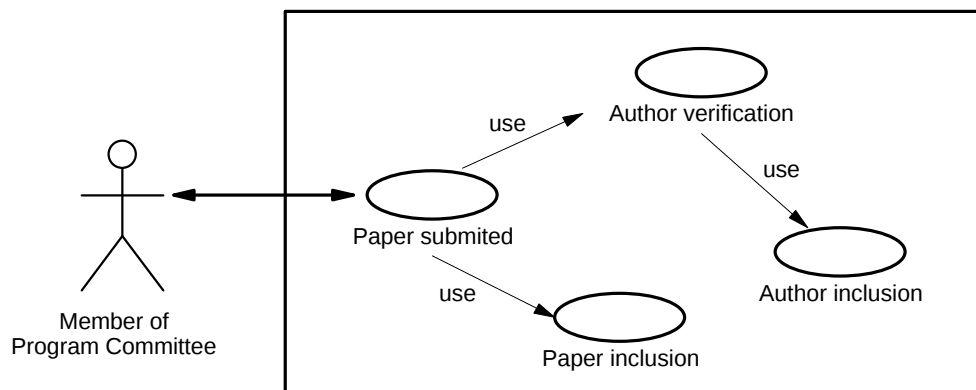


FIGURA 7.2 - Exemplo específico de um modelo de caso de uso para a submissão de um artigo no problema da IFIP, conforme a técnica de análise da metodologia OOSE.

7.5 OS TRABALHOS FUTUROS

A partir do presente trabalho, é possível visualizar vários lineamentos para futuros desenvolvimentos, entre eles cabe destacar os seguintes:

1. **Reutilização de especificações:** Avaliar a possibilidade de considerar a tripla (estímulo, processo, resposta) como unidade de reuso a nível de especificação. Como a lista de tipos de estímulos é configurada a partir dos requisitos de com-

portamento do sistema sob modelagem, esta tripla, usando alguma representação mais abstrata, poderia se tornar um construtor de reuso no nível da modelagem conceitual. Estes construtores deveriam ser devidamente instanciados conforme o estímulo que se esteja modelando e posteriormente especializados a fim de satisfazer os requisitos específicos do problema.

2. **Formalização dos HLS:** Especificar formalmente todas as extensões sugeridas para os *statecharts* e que definem os denominados *High-Level Statecharts* (HLS). Neste trabalho, as extensões foram apresentadas de maneira informal e fortemente baseadas em exemplos para facilitar a compreensão.
3. **Identificação de associações padronizadas:** Estudar as formas que assumiriam os padrões de interações (estímulos, eventos e ações) nos modelos dinâmicos para identificar associações padronizadas, tais como composição, agregação e participação (*membership*).
4. **Experimentação com a técnica de modelagem:** Aplicar o processo de modelagem a problemas de diferentes domínios, idealmente com distintos modeladores, para ganhar informação experimental que permita avaliar mais profundamente a técnica e indique modificações e extensões de caráter prático. O ideal seria realizar esta experimentação em conjunto com o desenvolvimento das ferramentas indicadas mais acima.
5. **Comparar modelos obtidos sob diferentes estratégias:** Analisar diferenças nos modelos estruturais derivados usando a técnica proposta sob a estratégia dirigida por comportamento, em relação aos modelos desenvolvidos usando técnicas de análise dirigidas por dados. A natureza das diferenças pode indicar o grau de influência do modelador na qualidade dos modelos obtidos por estas estratégias diferentes. Por exemplo, a figura 7.3 mostra um modelo de objetos desenvolvido para o mesmo problema² [BUS 94d] (aqui em português), usando a técnica OMT [RUM 91]. As diferenças mais notáveis com o modelo estrutural da figura 6.12 ocorrem nas hierarquias de herança.
6. **Extensões para a técnica de modelagem:** Tornar a proposta de modelagem uma técnica completa de MOO conforme as atividades definidas para ela. Algumas extensões sugeridas são:
 - incluir descrições de comportamento para aqueles tipos de objetos que não as apresentam (todos os que não foram identificados a partir das variáveis de referência do modelo de ciclos de vida);

² Em [BUS 94f] e [BUS 94h] são mostrados outros modelos estruturais de objetos desenvolvidos com técnicas dirigidas por dados. Em [BUS 94g] é apresentado um modelo estrutural desenvolvido com uma técnica dirigida por processos. O caso modelado para todas estas aplicações é o mesmo problema da IFIP.

- incluir modelo de colaborações entre os objetos a partir da informação implícita nas associações estáticas, estímulos, eventos e ações;
- aplicar o mecanismo de agregação, brevemente descrito no capítulo 2, para hierarquizar os objetos do modelo estrutural por níveis de abstração;
- flexibilizar as notações utilizadas nos diferentes modelos, visando uma maior praticidade no seu uso e
- incluir uma atividade para reorganizar o dicionário de dados para torná-lo um *dicionário de classes*, refletindo ou complementando os componentes mostrados no modelo estrutural. Particularmente necessário é a inclusão das variáveis secundárias que não aparecem no dicionário de dados inicial, mas que foram introduzidas nos modelos HLS.

7. **Ferramentas de suporte:** As ferramentas de suporte necessárias para criar um ambiente automatizado de suporte à modelagem conceitual e ao *design* orientado a objetos, seriam as seguintes:

- **Editor de diagramas HLS:** Um editor desta natureza poderia servir de suporte tanto para modelar os processos, como para modelar os ciclos de vida. Com pequenas extensões poderia até permitir editar os diagramas de contexto do sistema e da visão global do sistema.
- **Simulador de HLS:** Este simulador permitiria a verificação dos modelos ao instanciar as variáveis e executar o modelo.
- **Ferramenta de suporte à integração do modelo de processos no modelo de ciclos de vida:** Como existe o problema de sinônimos e homônimos para variáveis e estados nos diagramas, esta ferramenta de integração deveria fornecer informação sobre as possíveis similitudes e desigualdades de conceitos para o modelador decidir caso a caso a ações aplicáveis.
- **Gerador de modelos estruturais de objetos:** Para construir este gerador, seria necessário definir as heurísticas para identificar e mapear adequadamente todos os elementos considerados no modelo estrutural.
- **Gerador de modelos de colaboração entre os objetos:** Aproveitando a informação já contida nos modelos dinâmicos e usando heurísticas que devem ainda ser desenvolvidas, seria possível obter automatizadamente as colaborações entre os objetos do sistema.
- **Gerenciador do repositório:** Este repositório seria útil para manter, no dicionário de dados, todas as variáveis definidas no sistema. Também poderia servir para definir funções de mapeamento.

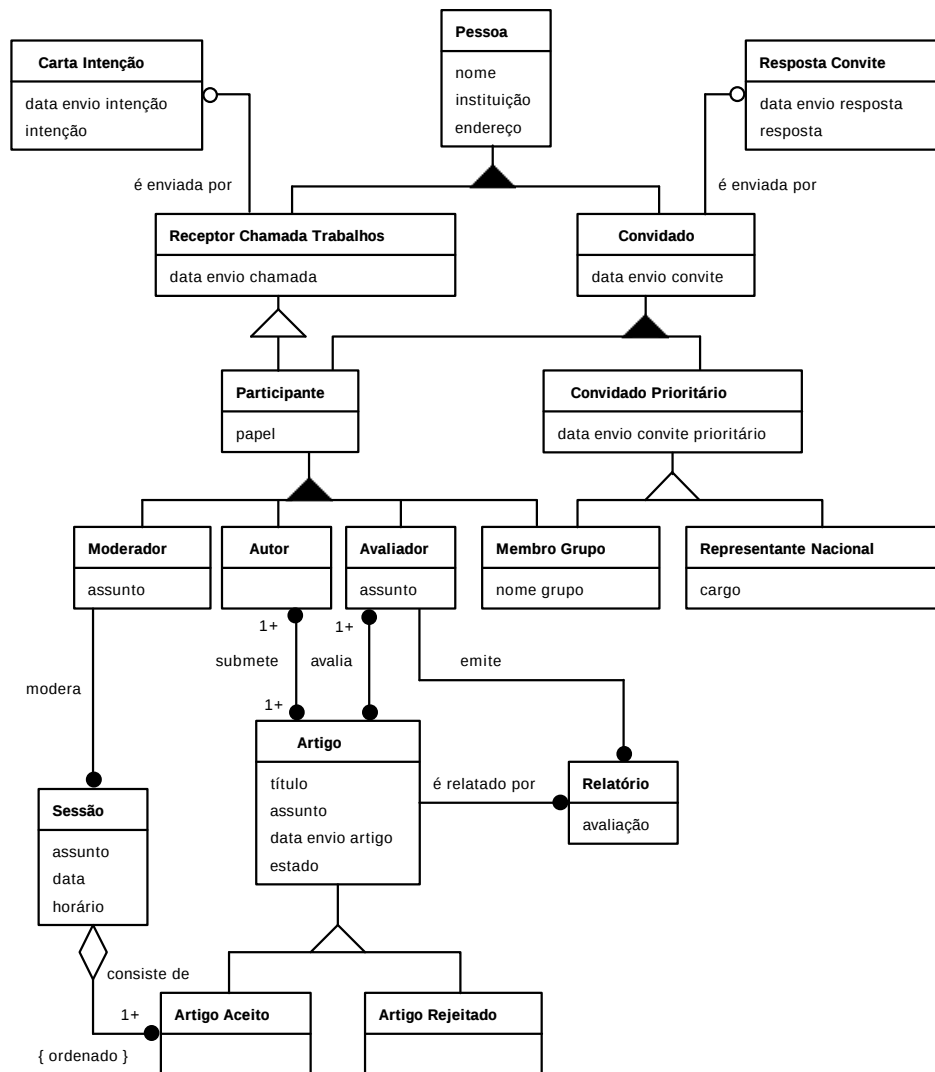


FIGURA 7.3 - Modelo estrutural de objetos para o problema da IFIP, desenvolvido segundo a técnica OMT.

7.6 RESUMO

O presente capítulo apresentou as conclusões e algumas considerações finais em relação à proposta de modelagem descrita neste trabalho. Para isto foi verificado que os objetivos levantados no início do trabalho foram atingidos, principalmente no que se refere à modelagem dinâmica do sistema e postergação do encapsulamento.

Foram descritas as contribuições do presente trabalho. Entre as contribuições gerais estão o amadurecimento da MOO sob a estratégia dirigida por comportamento, a mudança da ênfase da estrutura para o comportamento e a derivação deste comportamento a partir da estrutura. Entre as particularidades, cabe destacar o enfoque de estado usado para identificar

os objetos, o tratamento e geração concorrente de estímulos e respostas, respectivamente, e a derivação das hierarquias de herança a partir do comportamento dos objetos.

Algumas técnicas de MOO dirigidas por comportamento foram descritas e comparadas com a proposta. Em síntese, a técnica OBA e a técnica de análise OOSE mostram mais diferenças do que semelhanças, apesar de conduzirem o processo de modelagem com a mesma idéia central (de que o comportamento é anterior à estrutura) e a mesma estratégia.

Os trabalhos futuros mais relevantes são o desenvolvimento de ferramentas para um ambiente de modelagem com a técnica proposta, a formalização dos HLS, a necessidade de uma maior experimentação com a técnica, a comparação de modelos estruturais gerados com estratégias diferentes e incluir as extensões necessárias para tornar a técnica uma MOO completa.