



TIMEFEX: UMA ABORDAGEM PARA A MODELAGEM DE  
COMPORTAMENTO EM LINHAS DE PRODUTOS DE SOFTWARE

Wagner Schau de Castro

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientadora: Cláudia Maria Lima Werner

Rio de Janeiro

Maio de 2011

TIMEFEX: UMA ABORDAGEM PARA A MODELAGEM DE  
COMPORTAMENTO EM LINHAS DE PRODUTOS DE SOFTWARE

Wagner Schau de Castro

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO  
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA  
(COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE  
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE  
EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

---

Prof.<sup>a</sup> Cláudia Maria Lima Werner, D.Sc.

---

Prof. Toacy Cavalcante de Oliveira, D.Sc.

---

Prof.<sup>a</sup> Debora Christina Muchaluat Saade, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

MAIO DE 2011

Castro, Wagner Schau de

TimeFex: Uma Abordagem para a Modelagem de Comportamento em Linhas de Produtos de Software / Wagner Schau de Castro. – Rio de Janeiro: UFRJ/COPPE, 2011.

XIII, 125 p.: il.; 29,7 cm.

Orientadora: Cláudia Maria Lima Werner

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2011.

Referências Bibliográficas: p. 92-98.

1. Reutilização de Software 2. Linha de Produtos de Software 3. Modelagem de Características 4. Validação. I. Werner, Cláudia Maria Lima. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

*Esta tese é dedicada aos meus pais, Elizabeth e Luiz,  
e àqueles que de alguma forma exerceram estes papéis  
ao longo de minha vida.*

## **Agradecimentos**

A Deus, por me conceder esta pequena fresta de luz do Conhecimento. E aos familiares e amigos que compartilham da expiação terrena praticando o amor ao próximo.

Ao meu pai Luiz Carlos, por ter insistido sabiamente em me dar todo o apoio de que precisei mesmo diante da minha inocente relutância.

À minha mãe e ao Marco Antônio que, mesmo à distância, transmitiram seus votos de confiança, seus conselhos e palavras de conforto.

À minha noiva Monique, pela paciência nos momentos de pressão, pela calma e, principalmente, pela “psicologia”.

Aos meus familiares, por me proporcionarem o conforto e o afeto de que tanto precisei.

Agradeço especialmente à professora Cláudia Werner, pelo empenho e cuidado com todos os seus orientados, por sua sabedoria, e por seus ensinamentos que vão além do processo de orientação de mestrado e permeiam os conselhos da vida.

Aos meus colegas do grupo de pesquisa, pelas opiniões e críticas, e pelos momentos de descontração nas reuniões e confraternizações.

Aos colegas Marcelo Schots e Marco Di Benedetto, por suas sugestões e contribuições, sem as quais este trabalho não seria o mesmo.

Aos companheiros de laboratório, Cláudia Susie, Luis Felipe, Sérgio, Gabriel, Márcia, Leonardo e Henrique.

Aos professores das cadeiras ofertadas durante o curso, por me proporcionar a confiança de ter me tornado um profissional mais crítico e, principalmente, mais curioso.

À CAPES pelo apoio financeiro ao longo do curso, ao PESC pelo auxílio para a participação de eventos, e aos funcionários do PESC pela paciência e constante dedicação junto aos alunos e professores.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

TIMEFEX: UMA ABORDAGEM PARA A MODELAGEM DE  
COMPORTAMENTO EM LINHAS DE PRODUTOS DE SOFTWARE

Wagner Schau de Castro

Maio/2011

Orientadora: Cláudia Maria Lima Werner

Programa: Engenharia de Sistemas e Computação

A Engenharia de Linhas de Produtos de Software (ELPS) propõe o desenvolvimento de sistemas a partir de um núcleo de ativos reutilizáveis para o atendimento de determinados segmentos de mercado. Com isto, busca promover o reúso em larga escala trazendo benefícios como o aumento da produtividade, a amortização de custos e o aumento da qualidade. Neste cenário, permeiam domínios de aplicação que agregam informações temporais à especificação de sistemas de software. Assim, surge a necessidade de analisar a variabilidade de aplicações de forma consistente com os comportamentos que as características da linha podem desempenhar em cada aplicação.

Neste trabalho é apresentada a abordagem TimeFEX, uma proposta para a análise conjunta entre variabilidade de características e de comportamentos através de restrições baseadas em intervalos de tempo. Esta solução é constituída por atividades, artefatos e ferramentas desenvolvidas para uma análise de variabilidade de comportamentos consistente com a expressa pelo modelo de características. Neste contexto, foi realizado um estudo de observação no qual foi possível caracterizar a análise de variabilidade de comportamentos e avaliar os benefícios do uso da abordagem proposta e da orientação a intervalos de tempo.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

TIMEFEX: AN APPROACH FOR MODELING BEHAVIOR IN SOFTWARE  
PRODUCT LINES

Wagner Schau de Castro

May/2011

Advisor: Cláudia Maria Lima Werner

Department: Computer and Systems Engineering

The Software Product Line Engineering (SPLE) paradigm proposes system development from a core of reusable assets to meet the goals of certain market segments. In this way, it provides a large scale reuse to achieve benefits like productivity increase, cost reduction and quality improvement. In this scenario, there are several application domains that aggregate temporal information to the software specification, and brings the need of software variability analysis in a consistent way with the different behaviors that software features can perform in each application.

This paper presents the TimeFEX approach, a solution for a joint analysis of software features and behaviors by interval-based restrictions modeling. This solution consists of a process, artifacts and modelling tools developed for consistent behavior variability analysis and feature modelling. In this context, an observational study is used to describe the behavior variability analysis process and evaluate the benefits of using the proposed approach and time interval orientation.

# Índice

Capítulo 1 Introdução.....	1
1.1 Motivação.....	1
1.2 Objetivos.....	3
1.3 Organização.....	4
Capítulo 2 Modelagem de Comportamento Orientada a Características.....	6
2.1 Introdução.....	6
2.2 Engenharia de Domínio e Linhas de Produtos de Software.....	7
2.3 Modelagem Orientada a Características .....	11
2.4 Interação entre Características e Comportamento.....	13
2.5 Requisitos de Variabilidade de Comportamento Orientada a Características	14
2.6 Abordagens de Modelagem Comportamental.....	15
2.6.1 Diagramas de Estado* .....	15
2.6.2 FeatuRSEB.....	17
2.6.3 PLUS.....	19
2.6.4 Modelagem Racionalizada de Características.....	21
2.7 Análise comparativa entre as abordagens de modelagem de comportamento	25
2.8 Comentários Finais.....	28
Capítulo 3 A Abordagem TimeFEX.....	30
3.1 Introdução.....	30
3.2 Intervalos de Tempo.....	31
3.2.1 Considerações sobre Intervalos e Características.....	33
3.3 Processo.....	33
3.3.1 A Fase de Engenharia de Domínio.....	37
3.3.1.1 Análise de Composição.....	37
3.3.1.2 Análise de Comportamento.....	39
3.3.2 Engenharia de Aplicação.....	45



3.3.2.1	Composição de Características.....	45
3.3.2.2	Composição de Comportamento.....	46
3.4	A notação TimeFEX.....	50
3.4.1	Pacote Principal.....	51
3.4.2	Pacote Relacionamento.....	52
3.4.3	Pacote Regras de Composição.....	53
3.5	Análise Comparativa.....	54
3.6	Comentários Finais.....	57
Capítulo 4	Ambiente de Reutilização de Suporte à Abordagem.....	58
4.1	Introdução.....	58
4.2	O Ambiente Odyssey.....	59
4.3	O Ambiente TimeFEX.....	60
4.4	API de tratamento de leiautes temporais.....	66
4.5	Comentários Finais.....	69
Capítulo 5	Experimentação.....	70
5.1	Introdução.....	70
5.2	Planejamento.....	71
5.3	Execução.....	72
5.3.1	Primeira Etapa.....	73
5.3.2	Segunda Etapa.....	75
5.4	Análise dos Resultados.....	75
5.4.1	Primeira Etapa.....	77
5.4.2	Segunda Etapa.....	79
5.4.3	Considerações sobre os resultados.....	81
5.5	Ameaças à Validade.....	84
5.6	Comentários Finais.....	85
Capítulo 6	Conclusão.....	87
6.1	Epílogo.....	87
6.2	Contribuições.....	88

6.3	Limitações.....	89
6.4	Trabalhos Futuros.....	90
	Referências Bibliográficas.....	92
	Apêndice A Especificação do Domínio.....	99
	Apêndice B Odyssey-FEX.....	102
	Apêndice C Tabela de Transições do Modelo de Allen.....	105
	Apêndice D Cenários de Experimentação.....	106
	Apêndice E Esquema de representação de Leiautes.....	116
	Apêndice F Saída de Depuração.....	118
	Apêndice G Esquema de Depuração.....	120
	Apêndice H Declaração de Consentimento.....	122
	Apêndice I Formulário de Caracterização.....	123
	Apêndice J Formulário de Avaliação – Primeira Etapa.....	124
	Apêndice K Formulário de Avaliação – Segunda Etapa.....	125

# Índice de Figuras

Figura 1.1 – Elementos da abordagem TimeFEX .....	4
Figura 2.1 – Componentes de Processo da Engenharia de Domínio.....	8
Figura 2.2 – Modelo de Características. Adaptado de (KANG <i>et al.</i> , 1990) .....	12
Figura 2.3 – Diagrama de Estados Convencional. Adaptado de (GONZALEZ & LUNA, 2008).....	16
Figura 2.4 – Variabilidade de Estados e Transições.....	16
Figura 2.5 – Diagrama de Estados* (GONZALEZ & LUNA, 2008).....	17
Figura 2.6 – Notação da abordagem RSEB (OLIVEIRA, 2006) .....	18
Figura 2.7 – Notação da abordagem PLUS. ....	20
Figura 2.8 – Descrição de cenários de casos de uso. Adaptado de (ERIKSSON <i>et al.</i> , 2005).....	21
Figura 2.9 – Elementos da notação UCM. Adaptado de (BROWN <i>et al.</i> , 2006).....	22
Figura 2.10 – Exemplo de Mapa de Caso de Uso. Adaptado de (BROWN <i>et al.</i> , 2006) .....	23
Figura 2.11 – Exemplo de Modelo de Características.....	24
Figura 2.12 – Mapeamento entre Cenários e Características. ....	24
Figura 2.13 – <i>Stubs</i> estáticos e dinâmicos .....	25
Figura 3.1 – Leiaute temporal inválido .....	32
Figura 3.2 – Processo da abordagem TimeFEX.....	34
Figura 3.3 – I-Hotelaria, Ambiente Administrativo .....	35
Figura 3.4 – I-Hotelaria, Ambiente de Exibição de Reservas .....	36
Figura 3.5 – Diagrama de Características da plataforma I-Reserve .....	38
Figura 3.6 – Modelo Comportamental com relacionamentos identificados.....	40
Figura 3.7 – Modelo Comportamental expresso por restrições.....	42
Figura 3.8 – Leiaute temporal com Características Intervalares .....	43
Figura 3.9 – Modelo Comportamental contendo Características Intervalares .....	43
Figura 3.10 – Regra de Composição de Comportamento.....	45
Figura 3.11 – Sub-rede emergente do modelo de domínio.....	47

Figura 3.12 – Modelo Comportamental para composição .....	48
Figura 3.13 – Composição inconsistente de Relacionamentos Temporais .....	49
Figura 3.14 – Composição de Relacionamentos Temporais com leiaute inválido.....	50
Figura 3.15 – Taxonomia hierárquica de características .....	51
Figura 3.16 – Classes do Pacote Relacionamento .....	52
Figura 3.17 – Classes do Pacote Regras de Composição .....	53
Figura 3.18 – Leiaute Temporal com Múltiplas Granularidades.....	55
Figura 3.19 – Rede Temporal com Interação Temporal Livre em Granularidade .....	56
Figura 3.20 – Modelo de Característica com Interação Temporal Livre em Granularidade .....	57
Figura 4.1 – Ambiente de Desenvolvimento para Reutilização .....	59
Figura 4.2 – Ambiente de Modelagem de Comportamento .....	61
Figura 4.3 – Regra de Composição Comportamental (exclusiva).....	63
Figura 4.4 – Decisão sobre uso de conhecimento temporal na EA. ....	64
Figura 4.5 – Composição de Características .....	64
Figura 4.6 – Composição de Comportamento .....	65
Figura 4.7 – Diagrama de Classes, leiautes temporais orientados a objetos .....	66
Figura 4.8 – Estrutura de objetos de um leiaute temporal .....	68
Figura B.1 – Categorias de Características da Notação Odyssey-FEX (OLIVEIRA, 2006).....	102
Figura B.2 – Ligações entre Características da Notação Odyssey-FEX (OLIVEIRA, 2006).....	104
Figura D.1 – Linha de Produtos de Compra Coletiva .....	108
Figura D.2 – Exemplo de Ciclo de Vida de uma Oferta.....	108
Figura D.3 – Linha de Produtos de Sistemas de Reservas em Hotéis .....	112
Figura D.4 – Exemplo de configuração de tarifa.....	113
Figura D.5 – Modelo Comportamental (Cenário I).....	115
Figura D.6 – Modelo Comportamental (Cenário II).....	115

# Índice de Tabelas

Tabela 2.1 – Comparação entre abordagens de modelagem de comportamento.....	28
Tabela 3.1 – Propriedades Intervalares.....	31
Tabela 3.2 – Relacionamentos do modelo de Allen .....	32
Tabela 3.3 – Legenda de Cores da notação Odyssey-FEX.....	39
Tabela 3.4 – Tradução de relacionamentos temporais .....	41
Tabela 3.5 – Relacionamentos Alternativos .....	44
Tabela 3.6 – Comparação entre as abordagens de modelagem de comportamento e a abordagem TimeFEX .....	54
Tabela 4.1 – Representações distintas entre diagramas.....	62
Tabela 4.2 – Exemplo de leiaute de entrada.....	69
Tabela 5.1 – Definição do estudo experimental .....	71
Tabela 5.2 – Resultado da caracterização dos participantes.....	76
Tabela 5.3 – Formulário de Avaliação da Primeira Etapa.....	77
Tabela 5.4 – Formulário de Avaliação - Segunda Etapa .....	79
Tabela C.1 – Relações do Modelo de Allen .....	105

### 1.1 Motivação

Durante o desenvolvimento e a manutenção de software, características comuns entre aplicações são exploradas na busca pela reutilização do conhecimento e dos artefatos utilizados ou concebidos em momentos anteriores para a construção de novos produtos.

Esta prática, conhecida como Reutilização de Software (FRANKS & KANG, 2005), pode ser adotada com diversas finalidades e em diferentes níveis, onde se destacam benefícios como: (i) a redução do esforço e o conseqüente aumento da produtividade, resultante das diversas utilizações de um mesmo produto de trabalho, (ii) a amortização de custos de inspeção e teste, favorecendo o aumento da qualidade, (iii) a redução do *time-to-market*, (iv) a facilitação na definição de padrões e normas, (v) a promoção da interoperabilidade e compatibilidade, e (vi) novas oportunidades de negócio obtidas pela redução de custos na customização de produtos para atendimento a novos nichos de mercado (GRISS *et al.*, 1994).

A Engenharia de Domínio (ED) e a Engenharia de Linhas de Produtos de Software (ELPS) constituem abordagens amplamente utilizadas para a promoção da reutilização de software. A adoção destas e outras práticas envolvem a Análise de Domínio (AD), que é responsável pela coleta, manutenção e evolução das informações específicas da área tratada, para que estas sejam utilizadas no desenvolvimento de artefatos reutilizáveis. O conhecimento resultante desta análise é formalizado em um conjunto de artefatos chamado de modelo de domínio.

Diversos sistemas semelhantes podem ser entendidos sob a perspectiva de uma família de sistemas (PARNAS, 1976). Na AD, características de software presentes entre os sistemas que compõem uma família são descritas e expressas em um modelo de características. A partir desta descrição, cada sistema da família é formado a partir de um subconjunto das características presentes neste modelo.

A identificação de diferentes composições de características que derivam produtos de uma linha e a observância da validade destas composições se chama

Análise de Variabilidade. Esta variabilidade de composições consistentes é expressa no modelo de características através de regras de composição.

Dentre os diversos aspectos observados na AD, está o comportamento dos produtos de uma família de sistemas e como suas características se interrelacionam e interagem no tempo.

A modelagem deste comportamento pode envolver a representação de informações de natureza temporal que são de fundamental importância em diversas áreas do conhecimento, tais como, sistemas de informação, verificação de software, inteligência artificial e outros ramos que envolvem a modelagem de processos (ALLEN, 1983). Esta importância é ressaltada por algumas práticas de engenharia de reutilização, que dão maior ênfase a perspectiva temporal durante o processo de AD através de representações gráficas de relacionamentos temporais, fator este que facilita o entendimento e a compreensão do domínio trabalhado (CURTIS, 1984).

Com o foco na perspectiva temporal para a ELPS, foi realizada uma análise de trabalhos da literatura que propõem abordagens de modelagem comportamental orientada a características de software (GONZALEZ & LUNA, 2008), (SZASZ & VILANOVA, 2008), (GRISS *et al.*, 1998), (ERIKSON *et al.*, 2005), (BROWN *et al.*, 2006). Durante a avaliação destes trabalhos, foram identificadas deficiências que caracterizam estas soluções como parciais, comprometendo a expressividade do modelo de domínio e a análise de variabilidade de comportamentos da família de sistemas que este modelo representa. Dentre as deficiências observadas se destacam:

- **Variabilidade de comportamentos:** abordagens existentes não possibilitam a instanciação de produtos com comportamentos distintos para uma mesma configuração de características;
- **Notações específicas:** grande parte das abordagens existentes se restringe ao uso de notações próprias para a modelagem de características, o que dificulta a utilização das mesmas com notações amplamente utilizadas para este tipo de modelagem;
- **Nível de abstração:** utilização de abstrações incompatíveis com a praticada durante a análise de domínio orientada a características; e
- **Validação:** ausência de mecanismos para a validação dos comportamentos presentes nos produtos instanciados a partir da linha.

Estes apontamentos em conjunto impedem que a modelagem de comportamentos ocorra de forma livre da composição estrutural e hierárquica expressa pelo modelo de características, incorrendo na principal deficiência apontada neste trabalho para as soluções analisadas: a ausência de variabilidade comportamental efetiva em relação à estrutura hierárquica de características que é desenvolvida durante a AD.

## **1.2 Objetivos**

Este trabalho tem como objetivo constituir uma abordagem orientada a características para a modelagem e análise da variabilidade de comportamento em linhas de produtos de software. Este desenvolvimento busca contornar deficiências encontradas em trabalhos equivalentes e se baseia em duas potencialidades não encontradas nestes trabalhos: a capacidade de representação de variabilidade de comportamento entre produtos de mesma composição de características, ou seja, o tratamento de situações onde produtos com as mesmas características podem entregá-las a partir de comportamentos distintos; e o uso de notações existentes e de ampla utilização para a modelagem de características, em contraposição à criação de mais uma notação cuja semântica é estendida especificamente para o tratamento da dimensão temporal.

Para este objetivo, o desenvolvimento da solução é fundamentado em duas perspectivas distintas e complementares: a análise e composição de características, e a análise e composição do comportamento que estas desempenham. A abordagem por sua vez é baseada em um processo formado por atividades estabelecidas sobre estas duas perspectivas. Este processo tem como suporte para a sua aplicação um ambiente de reutilização baseado na plataforma Odyssey (ODYSSEY, 2010). A modelagem de características persiste neste ambiente em artefatos complementares que são utilizados no momento da instanciação de um novo produto. Esta instanciação é condicionada por regras de validação temporal e de conformidade em relação à composição de características e composição de comportamento.

Sendo assim, constitui-se também o objetivo de desenvolver uma forma eficaz e flexível para expressar comportamentos de linhas de produtos de software tendo como base a orientação a características. E finalmente, tanto este desenvolvimento quanto a orientação a características constituem os elementos participantes de uma avaliação realizada sob a forma de um estudo de observação.



### 1.3 Organização

Este trabalho se inicia pelo Capítulo 2, onde são apresentados os conceitos da reutilização de software utilizados no escopo tratado, mais especificamente, na ED e na ELPS. Sobre estas duas disciplinas são apresentados conceitos envolvidos na modelagem orientada a características e uma notação seminal às outras notações utilizadas para este tipo de modelagem. Neste mesmo capítulo, é apresentada a questão de como a interação entre características pode acarretar problemas, e como esta questão se relaciona com o comportamento em linhas de produtos de software. E finalmente, o capítulo apresenta requisitos para a representação de variabilidade de comportamento em linhas de produtos de software orientada a características e descreve uma análise comparativa entre abordagens pesquisadas na literatura.

A apresentação dos elementos específicos da abordagem TimeFEX é iniciada no Capítulo 3 e espelha o esquema apresentado na Figura 1.1. Neste capítulo, são descritos os conceitos envolvidos na definição de intervalos de tempo e como estes se relacionam com a definição de características de software. Em seguida, são descritas as atividades de um processo para a aplicação da técnica, este representado no topo da Figura 1.1.

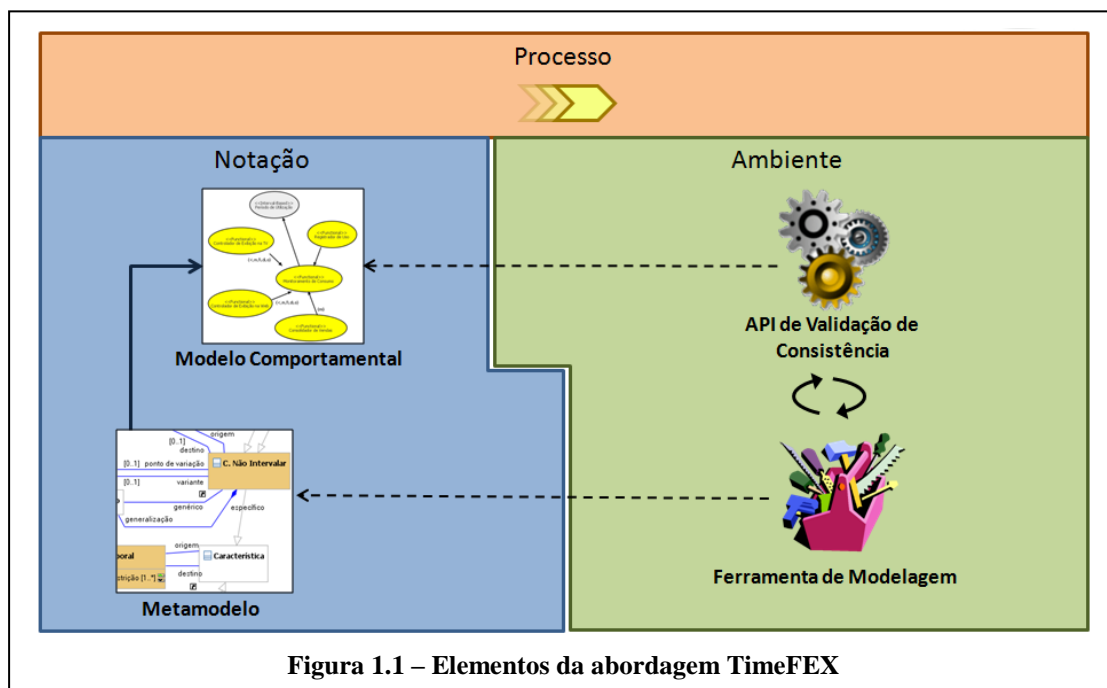


Figura 1.1 – Elementos da abordagem TimeFEX

A descrição destas atividades de processo é dividida entre as Engenharias de Domínio (ED) e de Aplicação (EA), e se fundamenta na análise realizada sobre um

conjunto de sistemas reais no domínio de reservas em hotéis. E finalmente, o Capítulo 3 agrega o desenvolvimento de Modelos Comportamentais, artefato específico da abordagem TimeFEX e utilizado para a descrição de comportamentos de LPS.

O Capítulo 4 trata de descrever o desenvolvimento e o funcionamento de um ambiente de reutilização construído para suportar as atividades de processo propostas. Este ambiente é constituído por uma ferramenta de modelagem e por uma API de validação de consistência. A ferramenta de modelagem proposta se utiliza de um metamodelo concebido a partir de adequações realizadas sobre a notação Odyssey-FEX (OLIVEIRA, 2006) para o desenvolvimento de modelos de características e de modelos comportamentais. A API de validação de consistência trata de validar comportamentos obtidos a partir do Modelo Comportamental.

No Capítulo 5, é apresentado o planejamento, execução e análise dos resultados de um estudo de observação. Neste estudo são explorados aspectos como validade, limitações, potencialidades e pontos de melhoria para a abordagem proposta. Ao final deste capítulo, são apresentadas conclusões e os fatores que ameaçam a validade do estudo apresentado.

E finalmente, no Capítulo 6, são apresentadas as contribuições, limitações e oportunidades de trabalho futuro envolvendo os elementos da abordagem proposta.

# Capítulo 2

## Modelagem de Comportamento Orientada a Características

### 2.1 Introdução

Para alcançar os benefícios esperados da reutilização de software, são necessárias abordagens estruturadas para o acúmulo de informações, descrições, soluções e métodos que apoiem o desenvolvimento de artefatos reutilizáveis baseados neste acúmulo. Neste sentido, a Engenharia de Domínio (ED) e suas derivações se tornaram processos-chave para o estabelecimento de ambientes de reutilização de software e para a especificação de arquiteturas, modelos e ativos reutilizáveis de forma sistemática no desenvolvimento de produtos correlatos.

Diversas práticas de reutilização se baseiam na orientação a características como uma forma de representar as informações de domínio em alto nível de abstração, tais como (KANG *et al.*, 1990) (GRISS *et al.*, 1998) (KANG *et al.*, 2002) (CZARNECKI *et al.*, 2004). Estas práticas utilizam a modelagem de características como uma ferramenta eficaz para a análise de domínio, dada a facilidade com que o termo é compreendido e comunicado entre os diversos envolvidos no processo de desenvolvimento (LEE & MUTHIG, 2006) (KANG *et al.*, 1998).

A Análise de Variabilidade compreende a identificação e representação das diferentes configurações de sistemas passíveis de serem obtidos ou adaptados a partir das composições de características extraídas deste modelo. O resultado deste trabalho é fundamental para a evolução e a manutenção do ambiente de reutilização e sua infraestrutura (LEE & MUTHIG, 2006).

Para diversos domínios de aplicação, tais como sistemas de informação, sistemas hipermídia, e sistemas de tempo real, a perspectiva temporal exerce um papel fundamental na definição do software (BOLOUR *et al.*, 1982) (BUCHANAN & ZELLWEGER, 1993) (BERTINO & FERRARI, 1998) (KIRNER & DAVIS, 1996), importância esta que é extensível à análise de domínio realizada nestas áreas de aplicação. Neste sentido, o comportamento esperado para cada produto de uma família figura entre os aspectos observados durante a ED.

Neste contexto, a utilização de esquemas gráficos facilita a compreensão de problemas cuja complexidade é caracterizada pela existência de relacionamentos temporais (CURTIS, 1984)(CARROLL *et al.*, 1980). Esta constatação converge com a concepção de abordagens orientadas a características, onde a valorização da perspectiva temporal se faz presente sob a forma de modelos. Entre as abordagens utilizadas para a representação do comportamento de linhas de produtos, se destacam técnicas orientadas a casos de uso (ERICKSON *et al.*, 2005) (GRISS *et al.*, 1998), a utilização de diagramas da UML (OMG, 2010) com variabilidades (GONZALEZ & LUNA, 2008) (SZASZ & VILANOVA, 2008) (MOON *et al.*, 2008) e a representação através de mapas de casos de uso (BROWN *et al.*, 2006).

Este capítulo analisa estas abordagens de forma comparativa, tendo em vista requisitos especificados para que a modelagem de comportamento orientada a características expresse uma variabilidade comportamental, de forma complementar e flexível em relação à composição de características.

O restante deste capítulo é iniciado pela Seção 2.2, que introduz conceitos da Engenharia de Domínio e da Engenharia de Linhas de Produtos de Software. A seguir, na Seção 2.3, são apresentados os aspectos relativos à orientação a características e aos elementos recorrentes entre as notações que se utilizam desta orientação. Na Seção 2.4, é descrita a categoria dos problemas derivados da interação entre características e como o comportamento de software e sua variabilidade se inserem neste contexto. Na seção 2.5, são apresentados os requisitos para representação da variabilidade de comportamento de software orientada a características. Na Seção 2.6, são apresentadas notações e abordagens pesquisadas na literatura para a representação desta variabilidade e, na Seção 2.7, é realizada uma análise comparativa entre estas. E finalmente, na Seção 2.8, são apresentados comentários sobre a análise realizada pelo capítulo.

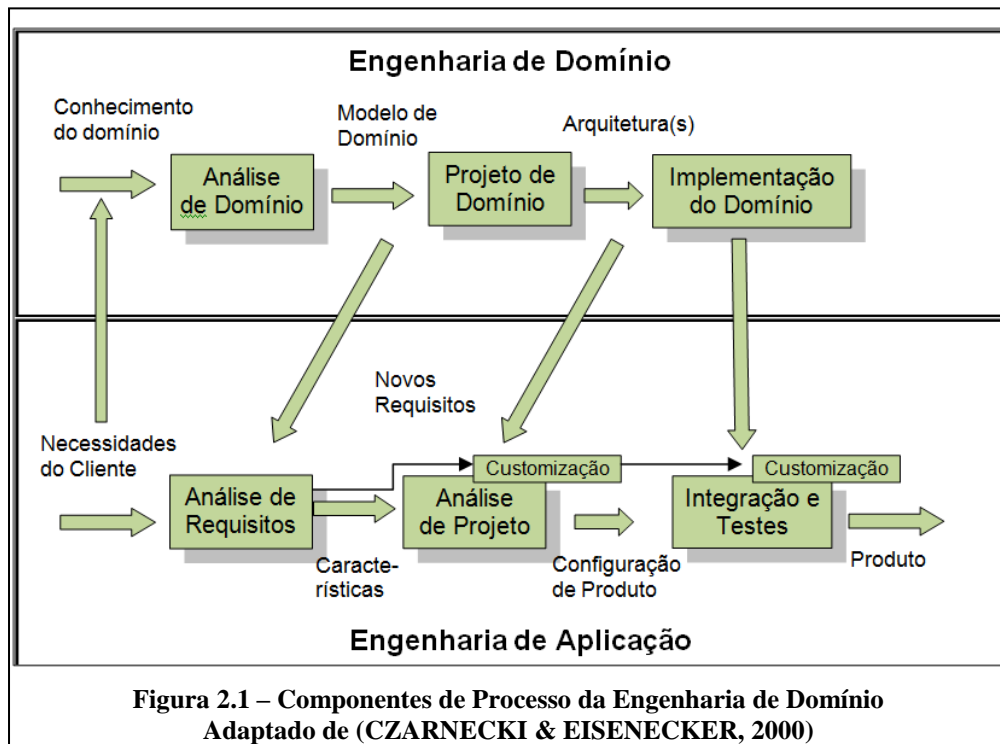
## **2.2 Engenharia de Domínio e Linhas de Produtos de Software**

A ED é um processo amplamente utilizado na engenharia de software para promover a reutilização sistemática no desenvolvimento de sistemas voltados ao atendimento de um determinado domínio de aplicação.

Segundo CZARNECKI & EISENECKER (2000), a ED tem como objetivo “a coleta, organização e armazenamento de experiências anteriores na construção de sistemas [...] em um domínio específico através de ativos reutilizáveis. E para isto, deve

fornecer meios adequados para a reutilização destes ativos (recuperação, avaliação, distribuição, customização, etc...) na construção de novos sistemas.”.

Diversos trabalhos apresentam a ED a partir de três componentes de processo, a saber: a Análise do Domínio, o Projeto do Domínio e a Implementação do Domínio, conforme esquematizado na Figura 2.1 e descrito a seguir.



**Análise do Domínio:** esta atividade trata especificamente da identificação, coleta, organização e representação das informações relevantes do domínio baseadas no estudo de sistemas existentes e o seu histórico de desenvolvimento, na captura de conhecimento de especialistas, no estudo teórico sobre o domínio, e nas tecnologias emergentes na área (KANG *et al.*, 1990). O resultado deste trabalho é formalizado em um modelo do domínio.

A identificação de pontos comuns entre aplicações se baseia na análise de um conjunto de sistemas existentes e futuros que atendem à área tratada, a fim de se obter uma descrição formal das semelhanças e diferenças existentes entre estes sistemas. Neste processo, são apontadas as características presentes em todas as aplicações, e uma variabilidade resultante das características que integram apenas uma parcela destas aplicações. E por fim, características específicas de um único produto e cuja reutilização não se faz compensatória são tratadas separadamente durante a Engenharia de

Aplicação (EA). Neste sentido, a variabilidade define a capacidade de customização de novos sistemas e de transformação de sistemas existentes pertencentes à família (van GURP *et al.*, 2001).

Apesar de haver uma indefinição sobre quais artefatos compõem o modelo de domínio (HARSU, 2002), alguns elementos são recorrentes na literatura, tais como: a definição de escopo e de contexto do domínio, análise dos pontos comuns entre as aplicações que atendem à área, o levantamento do dicionário de termos específicos, e a modelagem conceitual normalmente expressa através de um modelo de características.

E finalmente, a análise de domínio pode ser entendida como uma atividade correlata à análise de requisitos e ao desenvolvimento de projeto em alto nível, porém se destaca por não tratar de uma única aplicação, mas sim de toda uma família de aplicações.

**Projeto do Domínio:** o objetivo desta atividade é a definição de uma arquitetura genérica para a família de sistemas (CZARNECKI & EISENECKER, 2000). Este processo envolve tanto a observação de requisitos funcionais quanto a de requisitos não-funcionais, tais como robustez, confiabilidade, desempenho entre outros.

A definição desta arquitetura genérica envolve a utilização de padrões arquiteturais (BUSCHMANN *et al.*, 1996) ou uma composição destes padrões em diferentes níveis de abstração. Além disto, a definição desta arquitetura genérica ocorre ao longo da ED, corroborando com a prática de avaliá-la durante estágios iniciais do desenvolvimento da LPS, facilitando a resolução de problemas conforme estes são identificados mais cedo (ABOWD *et al.*, 1997).

**Implementação do Domínio:** esta atividade é responsável pelo desenvolvimento de artefatos reutilizáveis a partir dos modelos e da arquitetura genérica fornecidos pela análise e projeto do domínio. A infra-estrutura onde estes artefatos persistem compreende os mecanismos para mantê-los e atualizá-los, geradores de aplicações e linguagens desenvolvidas especificamente para o domínio em questão.

De forma análoga a ED, a Engenharia de Linhas de Produtos de Software (ELPS) propõe a reutilização de software em larga escala, trazendo benefícios convergentes com os esperados por outras práticas de reutilização, tais como, redução de custos, *time-to-market* e o aumento da qualidade (LINDEN *et al.*, 2007).

A ELPS trata de gerir um grupo de sistemas que compartilham um conjunto comum e gerenciado de características que satisfazem as necessidades de um segmento particular de mercado e que são desenvolvidos de forma prescritiva a partir de um núcleo comum de artefatos (NORTHROP, 2002). Esta prática é subdividida em três processos que se intercomunicam, a saber:

- **Desenvolvimento de Ativos Reutilizáveis:** este subprocesso, análogo à ED, trata do desenvolvimento voltado *para* a reutilização, ou seja, elementos como: a variabilidade entre os produtos da linha, os padrões arquiteturais utilizados, as restrições dos produtos e a relação dos artefatos reutilizáveis existentes são utilizados para a manutenção do núcleo de artefatos, realizando a redefinição do escopo da linha, a reespecificação dos planos de produção e o desenvolvimento de novos artefatos reutilizáveis a serem utilizados na EA;
- **Desenvolvimento de Produto:** o desenvolvimento de produto, análogo à EA, trata do desenvolvimento *com* reutilização. Ou seja, artefatos reutilizáveis, desenvolvidos anteriormente na engenharia de domínio, são utilizados como ponto de partida para desenvolver produtos previstos para a linha. Além disto, as especificidades do produto que fogem do escopo da linha, são tratadas caso a caso, tornando o produto único em relação ao restante da linha; e
- **Gestão da Linha:** esta atividade trata dos aspectos técnicos e organizacionais, tais como, treinamento de pessoal, mitigação de riscos e plano de comunicação, para que as outras atividades tenham os recursos necessários e obtenham êxito segundo os parâmetros de processo e conformidade estabelecidos para a linha.

## 2.3 Modelagem Orientada a Características

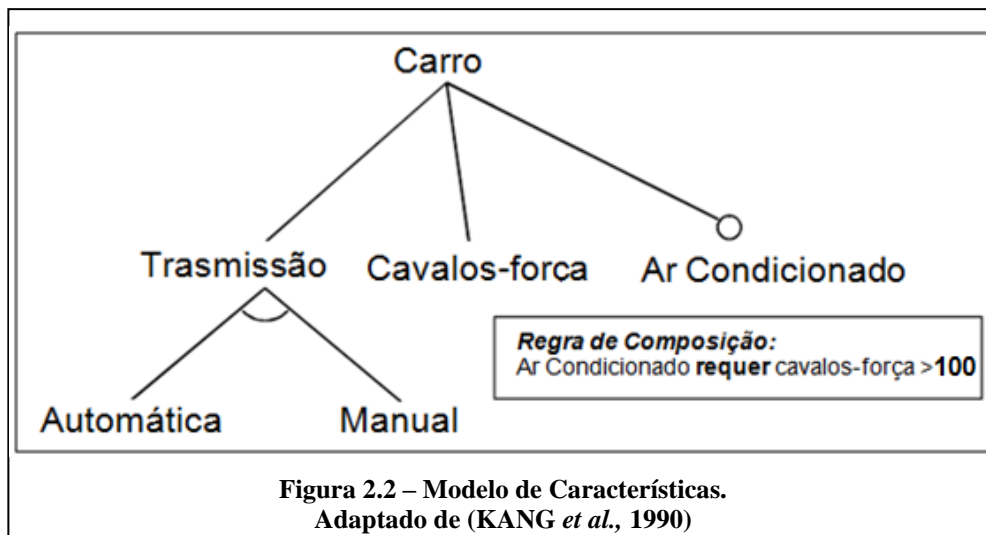
No paradigma de modelagem orientada a características, requisitos, descrições e especificidades da área são entendidos e tratados sob a forma de características de software. A utilização deste conceito como unidade de análise está presente em diversas práticas e abordagens de reutilização (KANG *et al.*, 1990) (KANG *et al.*, 1998) (KANG *et al.*, 2002) (LEE & MUTHIG, 2006), o que configura um extenso conjunto de definições distintas, parciais e sobrepostas, desenvolvidas para o termo ao longo dos anos.

Em (KANG *et al.*, 1990), é apresentada uma definição amplamente utilizada por metodologias posteriores. Neste trabalho, uma característica de software é entendida como um “aspecto visível ao usuário ou propriedade do domínio”. A generalidade desta definição proporciona o uso do termo “característica” como uma ferramenta de representação em alto nível de abstração, utilização esta motivada pelo fato de usuários e desenvolvedores se referirem aos produtos de software em termos das “características” que estes entregam ou possuem (KANG *et al.*, 1998).

A modelagem de características é a atividade de identificação de aspectos visíveis e externos aos produtos de uma linha e a sua organização em um modelo de características (LEE & MUTHIG, 2006). Neste modelo, características comuns a todos os produtos da linha são classificadas como características mandatórias. Já as características que integram apenas parte dos sistemas da família são modeladas como características opcionais ou alternativas.

O modelo de características trata essencialmente de um grafo hierárquico composto por nós, que representam características, e que são associados por ligações do tipo “OU”, “E” e “composto por”. Além destes tipos, algumas notações apresentam relacionamentos complementares, tais como “generalização/especialização” e “implementado por” (OLIVEIRA, 2006) (GRISS *et al.*, 1998). A seguir é apresentado na Figura 2.2 um modelo de característica desenvolvido a partir da notação FODA (KANG *et al.*, 1990).





Esta notação seminal representa características opcionais acompanhadas por um círculo vazio, como é o caso da característica *Ar Condicionado*. Características alternativas possuem associações ligadas por um arco, como apresentado no exemplo pelas características *Automática* e *Manual*. Nesta e em outras notações de natureza hierárquica, o nó-raiz da árvore representa o produto como um todo, que no exemplo utilizado se trata da característica *Carro*.

Além da estrutura hierárquica, este modelo agrega regras de composição que condicionam a existência de uma característica à existência ou estado de outra característica pertencente à uma mesma configuração de produto. No exemplo apresentado, a característica *Ar Condicionado* requer que *Cavalos-força* adquira um valor superior a 100. O conjunto das regras presentes no modelo expressa a variabilidade de sistemas passíveis de serem obtidos ou adaptados a partir desta estrutura.

A análise da variabilidade representada no modelo é voltada para a forma como diferentes características podem, em conjunto, compor configurações válidas de produtos de software consistentes. Ou seja, de acordo com o modelo apresentado, uma instância de carro com ar-condicionado e 50 cavalos-força de potência constitui uma configuração inconsistente.

Neste sentido, a utilização de características para representar requisitos e regras de negócio contribui para facilitar a análise de variabilidade. Isto ocorre dada a naturalidade de se comunicar similaridades e diferenças entre sistemas através de quais características estes sistemas possuem ou não em comum (LEE & MUTHIG, 2006).

## 2.4 Interação entre Características e Comportamento

Conforme a quantidade de características de sistema aumenta, se diversificam as formas como estas interagem. Porém, estas interações podem ser benignas ou prejudiciais ao desenvolvimento do próprio sistema e às expectativas do usuário (CALDER *et al.*, 2003). Problemas relativos à interação entre características se fazem presentes quando um conjunto de características de um mesmo sistema passa a não atender o que cada característica busca atender separadamente. Este fator é ainda agravado pela existência de interações que envolvem múltiplas características de forma direta e indireta.

Problemas relativos à interação entre características são extensamente tratados em diversas áreas do conhecimento, tais como: a área de telecomunicações (VELTHUIJSEN & BOUMA, 1994), a ELPS (ALFÉREZ *et al.*, 2009), a manutenção de software (SHIRI *et al.*, 2007) e a reutilização baseada em componentes (GRISS, 2000). Nestas análises, a forma como uma parte do comportamento do software interfere em outra pode ser entendida e modelada explicitamente como uma forma de interação entre características.

A modelagem de requisitos funcionais de sistemas complexos muitas vezes implica na ênfase prematura em aspectos comportamentais, normalmente capturados sob a forma de casos de uso e cenários (AMYOT, 2003). Estes aspectos comportamentais estão presentes e se fazem representados pelo modelo de domínio (TRACZ, 1994), o que leva à constatação da necessidade de representação de requisitos de natureza temporal por abordagens orientadas a características, considerando a questão de como estas interagem.

Segundo Buhr (1999), o comportamento é um importante aspecto para o entendimento e concepção de arquiteturas de software. Esta importância se estende, por definição, à especificação de arquiteturas genéricas que é realizada no projeto do domínio durante a ED.

Dada a realidade da prática de que novos produtos podem desempenhar comportamentos distintos e de que produtos existentes podem ter o seu comportamento adaptado durante sua execução, a expressão de variabilidade de comportamento em nível de modelo de domínio se faz mandatória em casos onde há complexidade temporal. No entanto, abordagens tradicionais onde o modelo de características é utilizado de forma exclusiva são insuficientes dada a incapacidade de representação da

variabilidade de requisitos de software de forma ampla por este modelo (BÜHNE *et al.*, 2004) (ALFERT, 2005). Além disto, domínios onde aspectos comportamentais são essenciais ao processo de análise agregam complexidade, pois enfrentam problemas relativos à interação entre características.

## 2.5 Requisitos de Variabilidade de Comportamento Orientada a Características

A partir de uma pesquisa realizada na literatura, foram identificadas abordagens de modelagem de comportamentos de LPS orientada a características. Estes trabalhos apresentam soluções para a representação de variabilidade de comportamentos, porém com potencialidades e características distintas. A partir do contraste entre estas abordagens, foram identificados requisitos não atendidos ou atendidos parcialmente, conforme listados a seguir:

- **Variabilidade Complementar:** as informações do modelo de domínio que compreendem os comportamentos de seus produtos devem contribuir e serem conformantes com a variabilidade da linha em nível de composição de características. Ou seja, a variabilidade definida para a composição de comportamento deve ser complementar, compatível e mapeável com a variabilidade definida para a composição de características;
- **Validação Temporal:** o comportamento definido para um produto da linha deve ser passível de validação. Ou seja, produtos não podem ter seu comportamento definido sobre relações temporais inválidas. Um exemplo desta ocorrência é a definição de um evento A que tem sua execução definida de forma anterior e posterior a outro evento B;
- **Múltipla Granularidade:** a análise deve possibilitar a definição de comportamentos em diferentes granularidades. Ou seja, a disposição temporal deve envolver desde a totalidade de características do modelo a passos internos de uma mesma característica;
- **Interação Temporal Livre em Composição:** a composição do comportamento de um produto deve ser variável em relação a uma mesma composição de características e vice-versa. Isto torna a variabilidade de comportamento independente, porém em conformidade com a variabilidade de características. Ou seja, deve ser possível representar diversos comportamentos possíveis para

uma mesma configuração de características, assim como uma variabilidade de características para uma mesma configuração de comportamento;

- **Interação Temporal Livre em Granularidade:** a análise deve possibilitar que elementos de granularidades distintas se relacionem na definição de comportamentos; e
- **Interação Temporal Livre em Hierarquia:** a análise deve possibilitar a definição de comportamentos que envolvam elementos distintos de forma independente de sua disposição hierárquica no modelo de características.

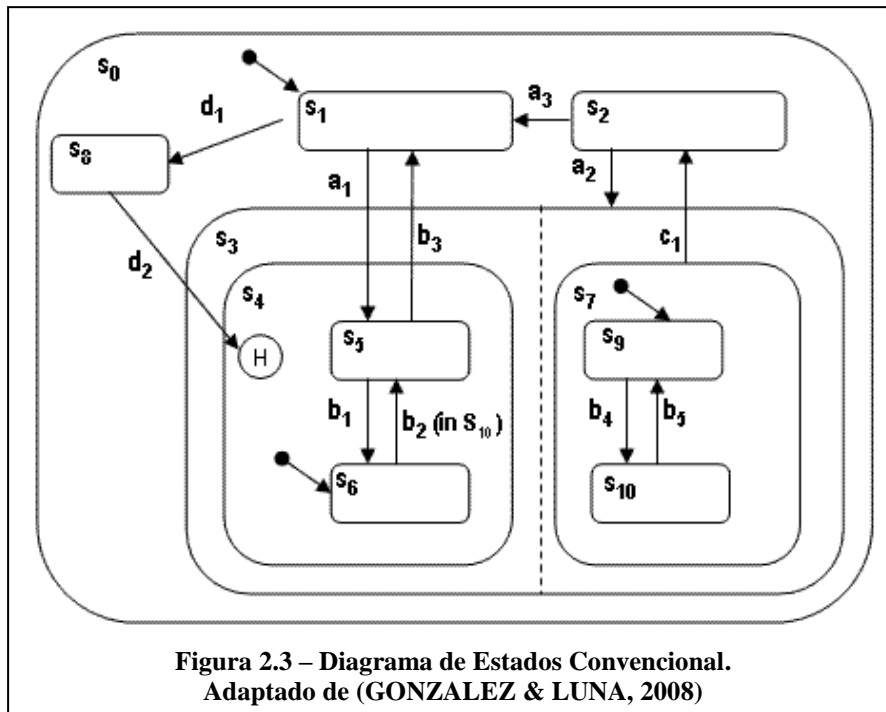
## 2.6 Abordagens de Modelagem Comportamental

Para representar o comportamento orientado a características a partir de um modelo, são necessárias abordagens e ferramentas formais para contemplar a dimensão temporal e agregar o comportamento de software ao modelo de domínio. A seguir são apresentadas as abordagens levantadas a partir de uma pesquisa da literatura, e que, apesar de proporem soluções díspares, possuem como objetivo comum a representação da variabilidade de comportamentos sob a ótica orientada a características.

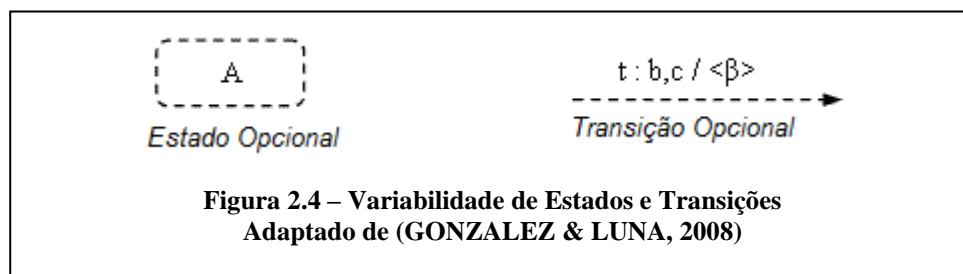
### 2.6.1 Diagramas de Estado\*

O diagrama de estados (Figura 2.3) é um formalismo introduzido para a especificação de sistemas complexos (HAREL, 1987) e, desde a sua concepção, passou a ser amplamente utilizado para a modelagem de comportamento dinâmico de sistemas de software.

Este diagrama é composto fundamentalmente por estados e transições, representados, respectivamente, por retângulos contendo a letra “S” e retas acompanhadas da letra “D”, conforme o diagrama apresentado na Figura 2.3. Um dos principais aspectos deste diagrama é a capacidade de representar estados sequenciais e paralelos. Neste mesmo diagrama utilizado como exemplo, os estados S1, S2 e S8 representam estados alternativos, enquanto os estados S0, S3, S4 e S7 representam estados paralelos ou concorrentes em relação, respectivamente, aos conjuntos de estados {S1, S2, S3, S8}, {S4, S7}, {S5, S6} e {S9, S10}, representados internamente aos mesmos.



Trabalhos apresentados em (SZASZ & VILANOVA, 2008) e (GONZALEZ & LUNA, 2008) tratam da inserção de variabilidade no diagrama de estados através de estados e transições opcionais (Figura 2.4). Nestes trabalhos, a partir de um diagrama de estados contendo elementos opcionais, é possível mapear um subconjunto de seus elementos com uma característica do modelo. Um exemplo de um diagrama de estados contendo elementos opcionais é apresentado na (Figura 2.5).



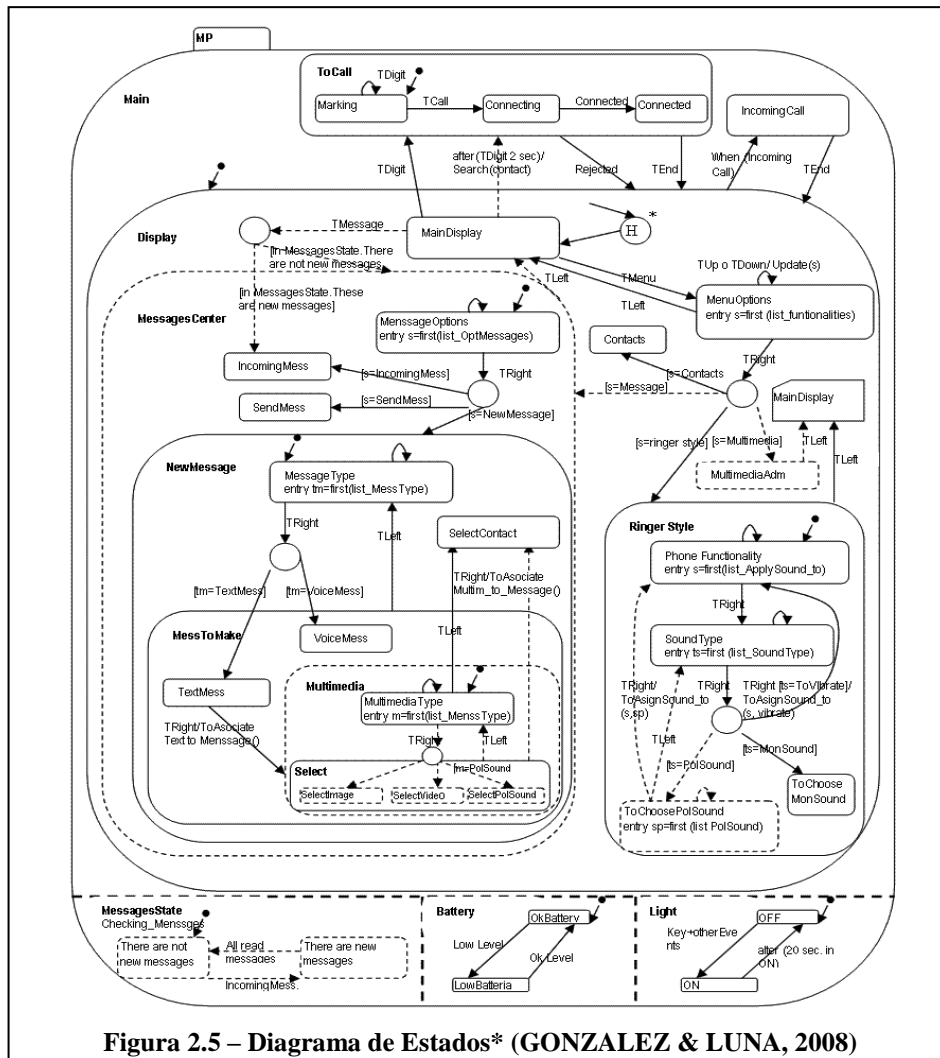


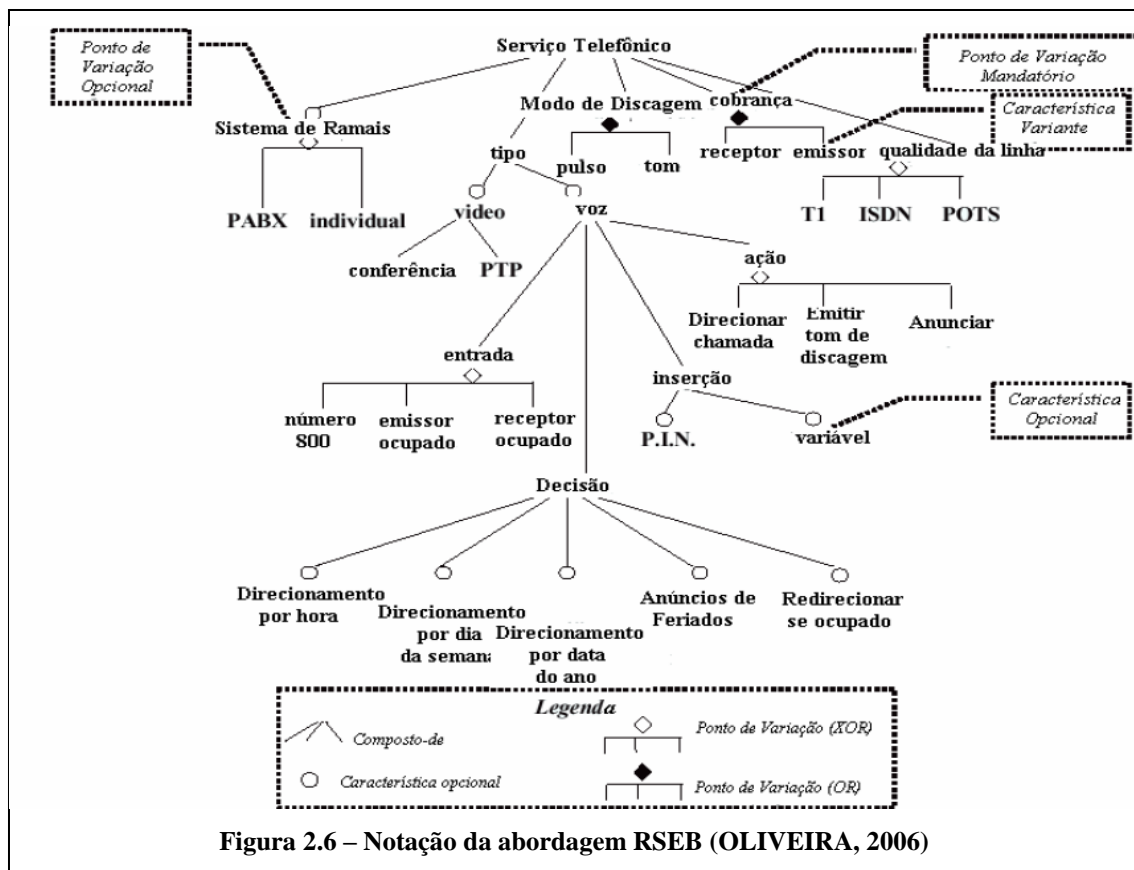
Figura 2.5 – Diagrama de Estados\* (GONZALEZ & LUNA, 2008)

Ambas as propostas apresentam formas para a obtenção de um diagrama de estados concreto, isto é sem variabilidades, a partir de uma composição dos elementos presentes no modelo de características para o qual este foi mapeado.

## 2.6.2 FeaturSEB

A abordagem RSEB (GRISS, 1997), do inglês *Reuse-Driven Software Engineering Business*, é uma forma sistemática, orientada a modelos e casos de uso para a reutilização em larga escala de software orientado a objetos. Esta proposta de ED segue o método proposto em (KRUCHTEN, 1995), onde diferentes modelos são desenvolvidos separadamente, tendo seus respectivos elementos relacionados a partir de um modelo específico para este fim. Especificamente no caso do RSEB, o modelo de casos de uso é utilizado para relacionar os elementos existentes no restante dos modelos.

Apesar do modelo de característica não ser formalmente tratado na abordagem RSEB, a atividade de modelagem em nível de característica foi integrada a este método em uma nova abordagem chamada FeatuRSEB (GRISS *et al.*, 1998). Nesta especialização, o método é acrescido de uma notação própria para a modelagem de características, possibilitando a utilização deste tipo de modelo em conjunto com os outros modelos da abordagem. A nova notação proposta para a modelagem de características é baseada na notação FODA (KANG *et al.*, 1990) e se encontra exemplificada na Figura 2.6.



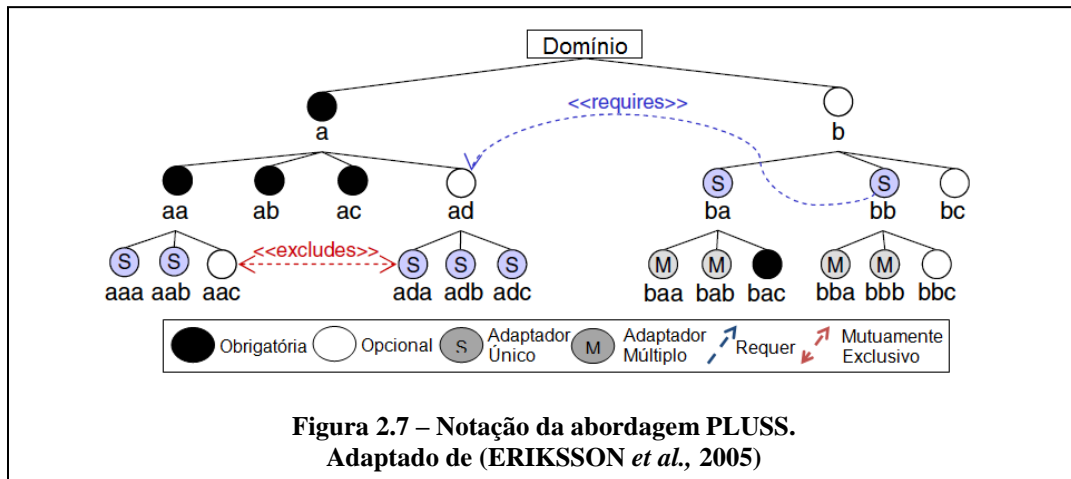
No FeatuRSEB, a variabilidade existente entre as características de software passa a se relacionar e estar em conformidade com uma variabilidade que é expressa em nível de casos de uso. Portanto, a utilização de modelos de casos de uso, em conjunto com os modelos de objetos e de características, possibilita a representação da variabilidade do comportamento existente em LPS.

### 2.6.3 PLUSS

Em (ERIKSSON *et al.*, 2005) e (ERIKSSON *et al.*, 2006), é descrita a abordagem PLUSS, do inglês *Product Line Use case modeling for Systems and Software engineering*, para a representação de variabilidade comportamental em nível de modelagem de domínio. Para esta representação, é proposta uma solução baseada em descrições de casos de uso relacionadas a modelos de características obtidos a partir de uma notação homônima baseada nas notações FeatuRSEB (GRISS *et al.*, 1998) e FODA (KANG *et al.*, 1990). Assim como no FeatuRSEB, a abordagem PLUSS se baseia fortemente na utilização de casos de uso expressos em um único modelo para toda a LPS.

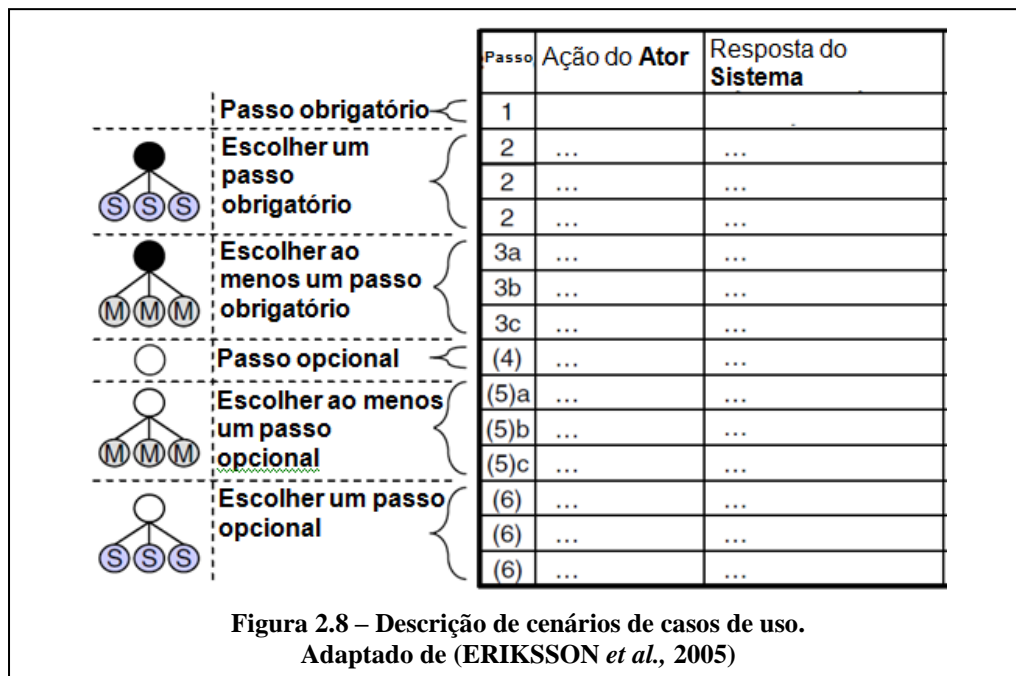
Na notação PLUSS, exemplificada na Figura 2.7, são utilizados conceitos comuns a outras notações, tais como regras de composição, características opcionais e características obrigatórias, denotadas por setas tracejadas, círculos brancos e círculos pretos, respectivamente. Em conjunto a estas estruturas, é acrescido o conceito de adaptadores, que correspondem a um conjunto de passos possíveis, internos aos cenários dos casos de uso, e ao qual uma característica pode corresponder. Estes adaptadores são classificados entre: Adaptadores Múltiplos (denotados por um círculo com a letra “M”) e Adaptadores Únicos (denotados por um círculo com a letra “S”), que por sua vez são análogos aos relacionamentos “ao-menos-um-de-muitos” e “exatamente-um-entre-muitos”, respectivamente, ambos existentes na notação FODA. No exemplo apresentado na Figura 2.7, a composição da característica opcional *ad* envolve a seleção de ao menos um adaptador entre os adaptadores únicos *ada*, *adb* e *adc*.





Na Figura 2.8, é demonstrado como as diferentes características (opcionais e obrigatórias) e os tipos de adaptadores podem se relacionar com os passos utilizados na representação dos cenários de casos de uso. A composição dos passos de cenários de execução é realizada a partir das seguintes categorias existentes na notação utilizada pelo método:

- I. Caso onde apenas o número aparece isoladamente em um único passo (ex. passo de número um), tratando-se, portanto, de um passo obrigatório;
- II. Caso onde um mesmo número aparece repetidas vezes em um conjunto isolado de passos (ex. passos de número dois), referindo-se a que um único passo deve ser escolhido como obrigatório entre os passos possíveis do conjunto;
- III. Caso onde um mesmo número aparece repetidas vezes seguido de uma letra (ex. passos de número três), tratando-se do caso onde ao menos um passo deve ser escolhido como obrigatório;
- IV. Casos onde um número aparece isoladamente em um único passo entre parênteses (ex. passos de número quatro), tratando-se de um único passo opcional;
- V. Casos onde um número aparece repetidas vezes entre parênteses e vem acompanhado de letras (ex. passos de número cinco), onde, neste caso, um ou mais passos devem ser escolhidos como opcionais; e
- VI. Caso onde um número aparece repetidas vezes entre parênteses, tratando-se de um único passo que deve ser escolhido como opcional para o cenário.



Com isso, a variabilidade do comportamento existente entre as diferentes configurações de produto é expressa e representada sob a forma de casos de uso e cenários. Estes últimos são formados por passos alternativos e opcionais, que variam de acordo com a composição de características na instanciação do produto, determinando assim um comportamento que varia entre um produto e outro de uma mesma LPS.

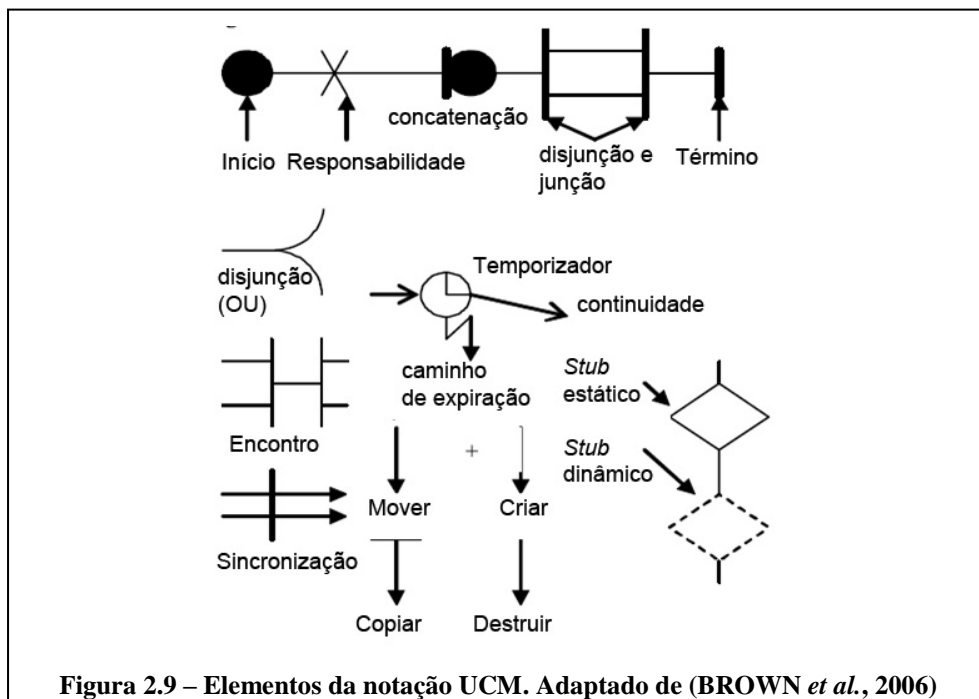
### 2.6.4 Modelagem Racionalizada de Características

A modelagem racionalizada de características (BROWN *et al.*, 2006), do inglês *Rationalised Feature Modelling*, se baseia na utilização de mapas de casos de uso (ITU, 2008), do inglês *Use Case Maps* (UCM), em conjunto com a modelagem de características.

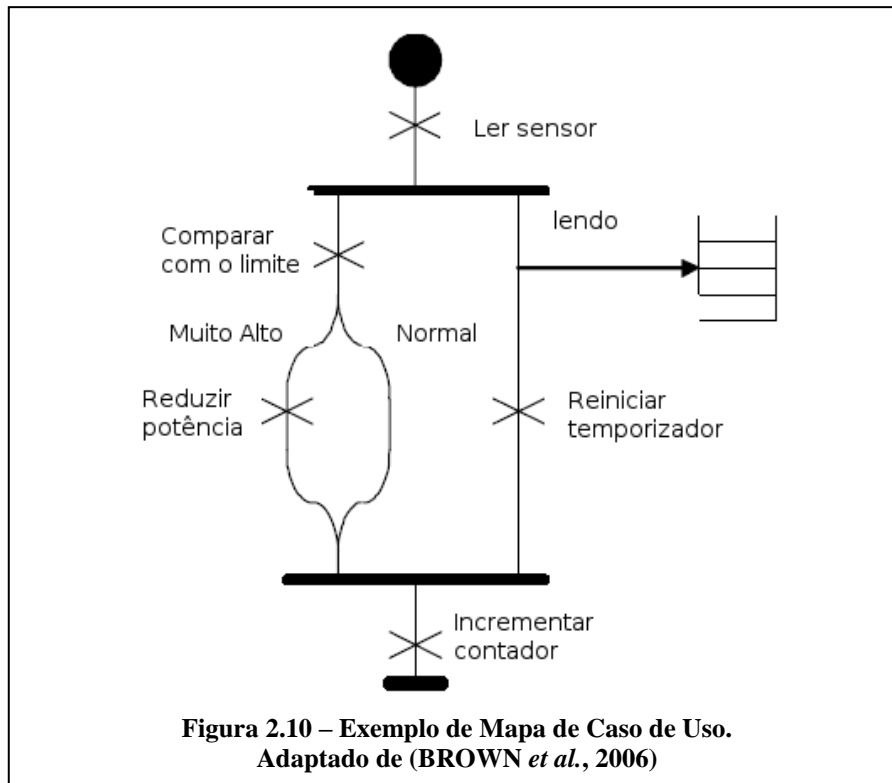
Mapas de Casos de Uso são descrições orientadas a caso de uso, desenvolvidas para representar os requisitos funcionais de um sistema de forma abstrata e orientada a passos por meio de cenários. Um cenário trata da descrição parcial do uso de um sistema como um conjunto parcialmente ordenado de responsabilidades através das quais um sistema transforma dados de entradas em dados de saída, conforme satisfaz pré-condições e pós-condições estabelecidas. Estas responsabilidades, por sua vez, são representadas por atividades que correspondem a algo que deve ser feito, tal como uma operação, ação, tarefa, função, etc. Com isso, um cenário se concentra na representação

dos relacionamentos entre estas responsabilidades em um ou mais casos de uso (AMYOT, 2003).

Na Figura 2.9, são apresentados os principais elementos que compõem a representação gráfica dos cenários. Um exemplo de utilização destes elementos é apresentado na Figura 2.10 em um modelo referente a um cenário simplificado de sistema de sensoriamento. O primeiro círculo preenchido é um ponto de partida e captura aspectos como pré-condições e gatilhos de execução. A última barra perpendicular ao caminho causal representa o fim da execução e captura aspectos como pós-condições de execução. Nos caminhos, as responsabilidades são indicadas pelas cruzes.



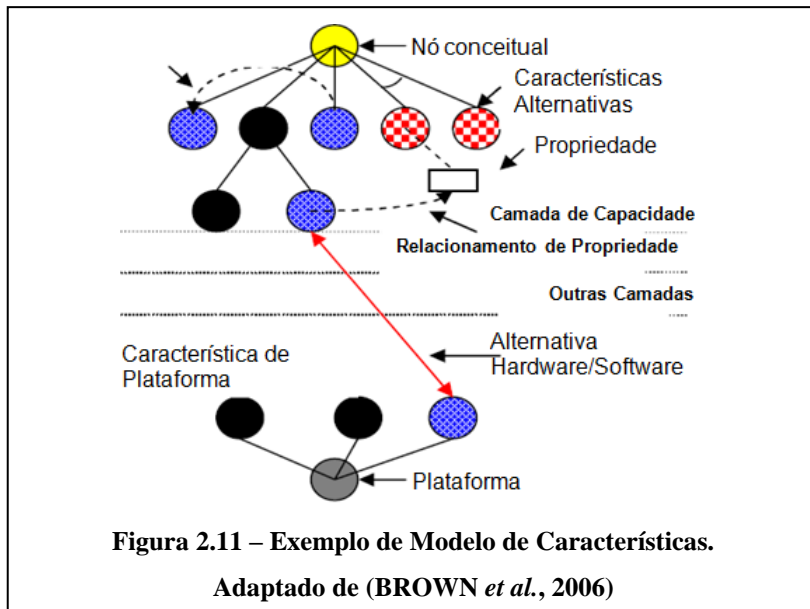
Caminhos podem possuir divisões do tipo inclusivas, como o exemplo das execuções concorrentes após a leitura do sensor, e divisões alternativas que são por sua vez acompanhadas de condições, como é o caso do exemplo onde após “Compara com o limite” é avaliado o valor entre as condições “Muito Alto” e “Normal”.



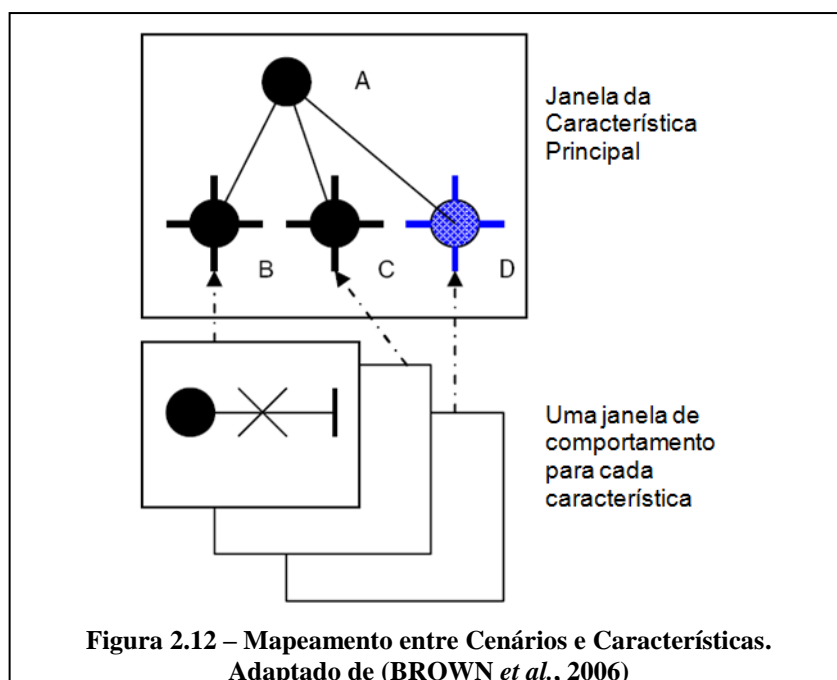
A linguagem utilizada para a descrição de Mapas de Caso de Uso integra o padrão internacional para Notação de Requisitos de Usuário (ITU, 2008), do inglês *User Requirements Notation* (URN). Este padrão é voltado para a elicitación, análise, especificação e validação de requisitos de forma geral, sendo utilizado, principalmente, na área de telecomunicações, processos de negócio e serviços, e adequado para sistemas de informação em geral (ITU, 2008).

Para utilizar a descrição de Mapas de Casos de Uso em conjunto com a modelagem de características, a abordagem de modelagem racionalizada prevê a utilização de uma notação gráfica própria exemplificada pela Figura 2.11. Nesta notação, características obrigatórias, opcionais e alternativas são expressas por círculos de cor preta, azul, e vermelha, respectivamente. Além disto, relacionamentos hierárquicos comuns como “provido por” e “formado por” são representados por uma seta e por uma linha preenchida, respectivamente.

Regras de composição de mútua exclusão de dependência são representadas graficamente por um arco pontilhado com setas nas duas terminações e por um arco pontilhado com uma única seta, respectivamente.

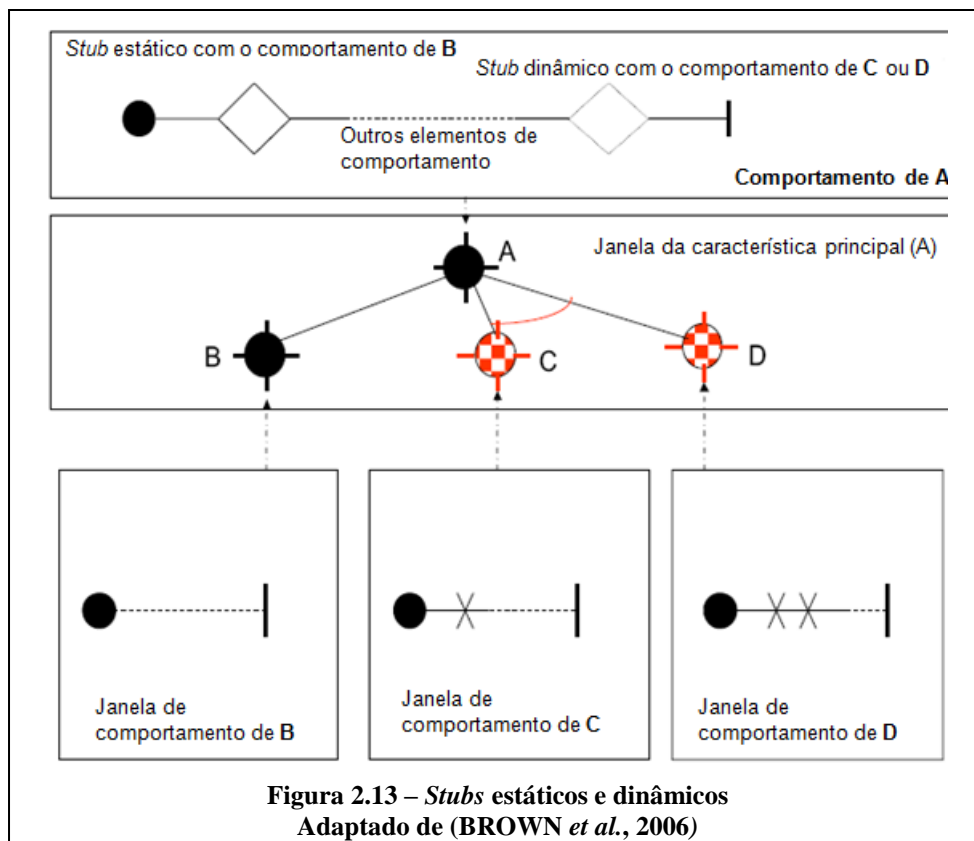


Para a junção destas duas representações, são atribuídos mapas de caso de uso a algumas características do modelo, conforme apresentado na Figura 2.12. Neste processo, mapas de caso de uso somente são atribuídos às características cujo comportamento é único, ou seja, está associado a somente um cenário de execução. Isto impossibilita, por exemplo, que características de alto nível, que tendem a possuir um comportamento variável, possuam diversos mapas de caso de uso relacionado a elas. Outra exceção é o caso das características não-funcionais que por definição não possuem um comportamento explícito.



Apesar de cada característica poder ser relacionada a apenas um mapa de caso de uso, a notação prevê a utilização de *stubs* estáticos e dinâmicos, representados na Figura 2.13, respectivamente, por um losango de linha contínua e pontilhada.

Estes elementos representam posições do cenário onde outro mapa de caso de uso pode ser acoplado, seja estática ou dinamicamente. Sendo assim, uma característica-pai, passa a ter agregado ao seu comportamento, o mapa de uma característica-filho obrigatória através de um *stub* estático. Além disso, o comportamento de características-filho alternativas pode ser agregado ao comportamento da característica-pai a partir de *stubs* dinâmicos, conforme apresentado na Figura 2.13.



## 2.7 Análise comparativa entre as abordagens de modelagem de comportamento

As abordagens de modelagem de variabilidade de comportamento apresentadas na Seção 2.6 são analisadas nesta seção a partir dos requisitos listados na Seção 2.5. Neste sentido, procura-se identificar as vantagens e as desvantagens de cada abordagem

e, principalmente, as tendências no atendimento destes requisitos ao confrontá-las de forma comparativa.

- **Diagrama de Estados:** a utilização de diagramas de estados com variabilidades traz vantagens que resultam da independência entre o modelo de características e o diagrama de estados propriamente dito. Isto possibilita que interações temporais entre características e seus componentes sejam feitas de forma livre, tanto em granularidade quanto em hierarquia. Porém, isto ocorre devido ao fato das características de software não serem representadas no diagrama, apenas mapeadas com seus estados. Durante a EA, a instanciação do comportamento de produtos resulta em um diagrama de estados derivado diretamente da composição de características, o que impossibilita uma variabilidade de comportamento em relação a uma mesma composição, tornando a primeira uma derivação pura da segunda. Além disto, dada a presteza do diagrama para a descrição de sistemas complexos (HAREL, 1987), o aumento na quantidade de eventos e ações concorrentes resulta em diagramas complexos e de grande profundidade entre subestados. E finalmente, a utilização da UML prescinde do grau de abstração praticado durante a Análise de Domínio orientada a características. Esta inadequabilidade persiste, por exemplo, no domínio das aplicações concorrentes, onde é perseguida uma independência entre o comportamento e os componentes arquiteturais (BROWN *et al.*, 2006), estes últimos fundamentalmente representados durante a utilização da UML;
- **FeatuRSEB:** nesta abordagem, o mapeamento proposto entre casos de uso e características de comportamento observável impede a modelagem de comportamentos globais, ou seja, casos de uso que envolvam diversas características do modelo. Além disto, a abordagem é caracterizada pelo fato de uma característica poder se relacionar a somente um único caso de uso. Este aspecto representa um impeditivo descrito no trabalho como o “problema de *feedback*”. Isto faz com que o comportamento modelado não seja de fato variável em relação a uma mesma configuração de características. O comportamento de características de mais alto nível de abstração e de hierarquia no modelo tem a sua modelagem comprometida dada uma tendência de que este seja variável. E finalmente, a abordagem não prevê a modelagem de interações

temporais entre elementos internos às características ou qualquer validação sobre o comportamento que é instanciado;

- **PLUSS:** em contrapartida ao seu precursor, o FeatuRSEB, esta abordagem permite a modelagem em nível de passos internos a um cenário de caso de uso. No entanto, não é possível estabelecer interações temporais entre os cenários e os passos de diferentes características de forma independente do seu arranjo hierárquico. Além disto, a modelagem proposta neste trabalho se restringe exclusivamente a representação da variabilidade do comportamento, ou seja, não possui uma perspectiva de composição estrutural da linha, impossibilitando a modelagem de características sem comportamento observável. E finalmente, a validação dos cenários que determinam o comportamento do produto instanciado não é tratada pela abordagem; e
- **Modelagem Racionalizada:** esta abordagem se mostrou ser mais completa no tratamento da granularidade e, principalmente, na semântica das relações temporais entre eventos. No entanto, assim como na abordagem PLUSS, as interações temporais são restritas conforme a hierarquia das características envolvidas no comportamento. Apesar de Mapas de Caso de Uso (UCM) serem passíveis de validação através do uso da linguagem LOTOS (BOLOGNESI & BRINKSMA, 1987) (AMYOT *et al.*, 1999), este uso não é tratado pela abordagem. Este tratamento se mostra necessário, pois a abordagem é derivada de uma adaptação dos Mapas de Caso de Uso para que estes sejam utilizados no contexto de modelagem de características.

A análise comparativa apresentada nesta seção é sumarizada na Tabela 2.1 com base nos requisitos apresentados na Seção 2.5.

Requisitos/Abordagens	DE*	FeatuRSEB	PLUSS	Modelagem Racionalizada
Interação Temporal Livre em Composição	Não	Não	N/A	Não
Validação Temporal	Sim	Não	Não	Não
Variabilidade Complementar	Não	Não	N/A	Não



Múltipla Granularidade	Sim	Não	Sim	Sim
Interação Temporal Livre em Hierarquia	Sim	Não	Não	Não
Interação Temporal Livre em Granularidade	Sim	Não	Sim	Sim

**Tabela 2.1 – Comparação entre abordagens de modelagem de comportamento**

Nesta tabela, é possível observar que um dos principais problemas encontrados nas abordagens analisadas está no determinismo dado ao comportamento do produto a partir da configuração de características. Neste sentido, é observada uma necessidade obtida do seu contraposto, ou seja, diversos comportamentos passíveis de serem obtidos para um mesmo conjunto de características. Todas as abordagens apresentadas divergem neste aspecto, pois se restringem a agregação de informações adicionais fornecidas para cada configuração de forma auxiliar ao processo de instanciação do produto, sem com isso apresentar, por completo, a variabilidade de comportamento.

## 2.8 Comentários Finais

Neste capítulo foram analisadas diferentes abordagens para a modelagem de variabilidade comportamental em nível de características de software com enfoques distintos. O primeiro aspecto observado é que todas as propostas utilizam diagramas complementares ao modelo de características para representar o comportamento e sua variabilidade, tendência esta comum a grande parte das práticas de Engenharia de Software, como por exemplo, abordagens baseadas na UML (OMG, 2010), onde determinadas perspectivas são privilegiadas em cada tipo diagrama. Em abordagens desta natureza, as dimensões de espaço e do tempo permeiam documentos, artefatos ou visões distintas, porém complementares para a manutenção da clareza e da facilidade de compreensão dos seus elementos.

Com exceção da primeira abordagem, apresentada na Seção 2.6.1, as técnicas utilizam notações próprias para a modelagem de características, inserindo elementos específicos para possibilitar a vinculação do caráter temporal, presente em outro modelo, com os elementos do modelo de características. Este fato inviabiliza a utilização imediata destas abordagens com outras notações reconhecidas por sua ampla utilização, tais como (KANG *et al.*, 1990) e (CZARNECKI *et al.*, 2004).

Outro aspecto importante é a recorrência na representação da variabilidade do comportamento através de casos de uso ou cenários. A representação através de casos de uso tende a privilegiar interações e seqüências de ações entre sistemas e atores externos ao sistema (JACOBSON, 1994). Esta prática é deficitária em casos onde há a necessidade de modelar comportamentos internos, correspondentes às partes ou quando o domínio tratado é baseado em sistemas não-reativos.

Outro aspecto importante se dá pelo fato de que nenhuma das abordagens analisadas prevê a utilização das informações comportamentais como prática auxiliar à análise de variabilidade. Ou seja, a observação da modelagem na perspectiva comportamental não é utilizada como insumo para adequação da composição de características ou vice-versa, como por exemplo, pela geração de novas regras de composição inclusivas e exclusivas.

# Capítulo 3

## A Abordagem TimeFEX

### 3.1 Introdução

Tendo sido apontadas deficiências nas abordagens analisadas no Capítulo 2, neste capítulo será apresentada uma abordagem para modelagem e análise de variabilidade de comportamento em LPS orientada a características. Esta abordagem, chamada TimeFEX, tem como foco o tratamento do comportamento em LPS e da sua variabilidade através da utilização de um modelo temporal orientado a intervalos e adaptado ao contexto de ED.

Esta abordagem é constituída por três elementos principais, a transposição conceitual de intervalos de tempo para a descrição de comportamento de software orientada a características, um processo prescritivo para a aplicação da técnica e uma sugestão de metamodelo capaz de agregar o comportamento dos integrantes de uma família de sistemas às informações de modelagem de características.

O capítulo apresenta inicialmente, na Seção 3.2, o arcabouço teórico utilizado na transposição de intervalos de tempo para a descrição comportamental de linhas de produtos de software e uma analogia entre intervalos e características de software, necessária para esta transposição. A seguir, na Seção 3.3, é apresentado um processo prescritivo composto por um conjunto de atividades para a aplicação da técnica, dividido nas fases de ED e EA, e que utiliza, como exemplo, informações reais colhidas de sistemas de software no domínio de redes de hotéis. Na Seção 3.4, é apresentada uma notação utilizada na modelagem de característica e o seu respectivo metamodelo. Na Seção 3.5, é dado prosseguimento a análise comparativa realizada no capítulo anterior agregando a abordagem TimeFEX a esta análise. E finalmente, na Seção 3.6, são apresentados comentários finais sobre os aspectos conceituais da abordagem apresentados neste capítulo.

### 3.2 Intervalos de Tempo

A definição de sistemas computacionais envolve a modelagem do tempo em nível conceitual (LITTLE & GHAFOOR, 1993) (BOLOUR *et al.*, 1982) (ANDERSON, 1982). Sob esta perspectiva, o modelo de Allen (ALLEN, 1983) propõe a utilização de intervalos de tempo como uma forma eficaz para a representação, manutenção e validação do conhecimento a respeito do tempo.

Dentre os conceitos fundamentais utilizados neste tipo de modelo, está o de instante, que é um ponto no tempo com duração igual a zero, e o de intervalo de tempo, que é o espaço temporal obtido a partir de dois instantes (LITTLE e GHAFOOR, 1993) (ALLEN, 1983). Um intervalo, denotado por  $[a,b]$ , é definido como  $\{x \mid a \leq x \leq b\}$ , onde  $a$  e  $b$  são elementos de um conjunto ordenado  $S$ . Segundo a definição utilizada em (ANDERSON, 1982), todo intervalo pertencente a  $S$  obedece às propriedades apresentadas na Tabela 3.1.

$[a,b] = [c,d] \Leftrightarrow (a = c) \wedge (b = d)$
Se $c,d \in [a,b] \wedge e \in S \wedge c \leq e \leq d$ então $e \in [a,b]$
$\#[[a,b]] > 1$

**Tabela 3.1 – Propriedades Intervalares**

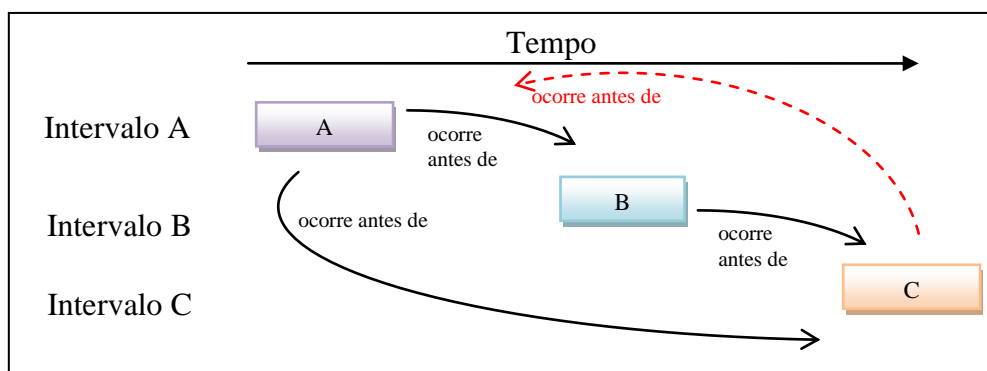
Além de intervalos de tempo isolados, modelos temporais utilizam relacionamentos para representar as formas como estes intervalos interagem (BENTHEM, 1991). No modelo de Allen (ALLEN, 1983) mais especificamente, são utilizados treze tipos de restrições, conforme apresentado na Tabela 3.2. O conjunto formado por intervalos de tempo e seus relacionamentos constitui um grafo referido como leiaute temporal.

Relacionamento	Símbolo	Símbolo Inverso	Exemplo Gráfico
X antes de Y	<	>	XXX YYY
X igual a Y	=	=	XXX YYY
X antecede Y	m	mi	XXXYYY
X sobrepõe Y	o	oi	XXX YYY
X durante Y	d	di	XXX YYYYYYYY
X inicia com Y	s	si	XXX YYYYYYY
X finaliza com Y	f	fi	XXX YYYYYY

**Tabela 3.2 – Relacionamentos do modelo de Allen**

Neste modelo, dois relacionamentos que envolvem um mesmo intervalo possibilitam a inferência de outros relacionamentos a partir de regras transitivas apresentada no Apêndice C. Este processo pode ser exemplificado de forma simples pela relação em que A “ocorre antes de” B e que B “ocorre antes de” C ( $A < B < C$ ), o que leva a inferência do relacionamento que define que A “ocorre antes de” C ( $A < C$ ). Estes três relacionamentos são representados na Figura 3.1 por setas curvas e contínuas.

A partir de um leiaute temporal, é possível verificar a sua validade, ou seja, se este possui informações conflitantes ou inconsistentes. As restrições fornecidas pelo Modelo de Allen são mutuamente exclusivas (ALLEN, 1983), portanto, dois intervalos não podem se relacionar simultaneamente por mais de uma restrição ou por conjuntos disjuntos de restrições. Neste caso, um leiaute se torna inválido caso possua relacionamentos temporais divergentes, sejam estes relacionamentos inferidos a partir de outros intervalos ou não.



**Figura 3.1 – Leiaute temporal inválido**

No leiaute apresentado na Figura 3.1, a inserção de um novo relacionamento representado pela seta tracejada torna o mesmo inválido. O relacionamento transitivo A “ocorre antes de” C ( $A < C$ ) conflita com a inserção do relacionamento C “ocorre antes de” A, pois o intervalo A não pode ocorrer simultaneamente antes e depois do intervalo C, o que torna o referido leiaute inválido.

### **3.2.1 Considerações sobre Intervalos e Características**

É importante observar que intervalos de tempo, delimitados por pontos de início e fim, e a utilização de instantes de tempo constituem abstrações amplamente utilizadas e de fácil compreensão na descrição de problemas de natureza temporal (LADKIN, 1990). Este fator torna possível a comunicação em um alto nível de abstração compatível com a praticada durante a Análise de Domínio orientada a características.

Em (BOSCH, 2000), uma característica de software é definida como “uma unidade lógica de comportamento que é especificada a partir de um conjunto de requisitos funcionais e de qualidade”. Neste entendimento, a abstração da característica de software sob a forma de um comportamento e seus respectivos requisitos converge para uma utilização de intervalos de tempo em casos onde estes requisitos tratam de aspectos comportamentais, ou seja, dos mesmos aspectos para os quais são utilizados abstrações de intervalo e instante de tempo.

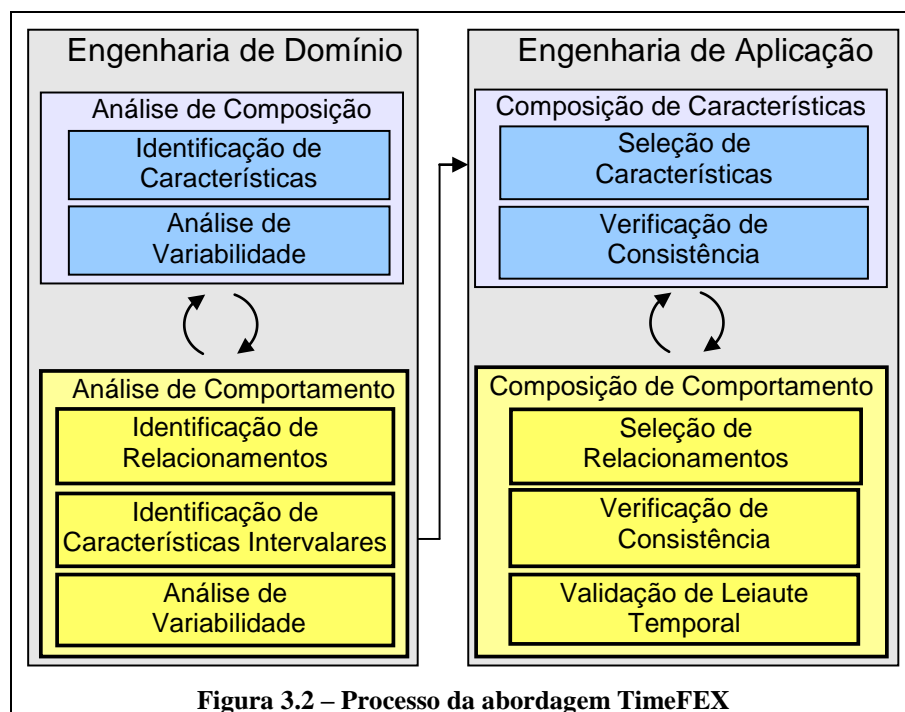
A verificação formal de consistência, que pode ser realizada sobre o leiaute temporal baseado em intervalos (Seção 3.2), possibilita que técnicas de ED baseadas no Modelo de Allen, validem o comportamento temporal obtido a partir do leiaute temporal instanciado, conforme o requisito de validação apresentado no Capítulo 2.

E finalmente, assim como na proposta deste trabalho, a transposição do modelo de Allen é realizada de forma extensiva, auxiliar e transdisciplinar em trabalhos como (GUESGEN, 1989), (ABRAHAM & RODDICK, 1999), (KNAUFF, 1999) e (FREKSA, 1992).

## **3.3 Processo**

Uma boa forma para se entender uma abordagem e seus objetivos é através da apresentação de um processo de aplicação. O processo prescritivo da abordagem TimeFEX é composto por atividades e tarefas distribuídas em duas fases, conforme apresentado na Figura 3.2, a saber: a ED, constituída pelas atividades de Análise de

Composição e Análise de Comportamento; e a EA, formada pelas atividades de Composição de Características e de Composição de Comportamento. As quatro grandes atividades apresentadas na figura são compostas por tarefas. As atividades e as tarefas que integram o foco da abordagem estão representadas com a borda em destaque no esquema apresentado na Figura 3.2.



As atividades de modelagem previstas pela abordagem se dão de forma complementar ao longo de todo o processo: sob a perspectiva da composição de características de produtos e sob o ponto de vista da composição de comportamentos determinados a partir de relacionamentos temporais. Estes comportamentos são atribuídos aos produtos que envolvem estas características.

Para a explanação das fases deste processo, será utilizado o conhecimento coletado acerca de uma linha de produtos real chamada **I-Reserve** (UMAIDEIA, 2010b). Esta linha se trata de uma plataforma de sistemas para reserva de diárias em hotéis, confirmação de reservas e publicação de tarifas na web e em redes internas de TV em hotéis. Os sistemas que compõem esta plataforma tiveram o seu desenvolvimento iniciado em 2006 pela empresa UmaIdea (UMAIDEIA, 2010c) e se encontram em constante evolução e implantação em novos hotéis.

As informações utilizadas nos exemplos deste capítulo tratam de sistemas em funcionamento e algumas das evoluções pretendidas pela empresa para a linha de

produtos. Os detalhes destas informações foram coletados através de entrevistas com os integrantes da empresa e estão descritos no Apêndice A. A seguir é apresentado um breve resumo sobre os principais conceitos envolvidos nos exemplos utilizados neste capítulo.

Cada sistema da linha é composto por um ambiente administrativo no qual os funcionários dos hotéis cadastram e agendam tarifas promocionais de acordo com suas permissões de usuário (Figura 3.3).



Figura 3.3 – I-Hotelaria, Ambiente Administrativo

As tarifas cadastradas, depois de configuradas no ambiente administrativo, são disponibilizadas em aplicação residente no website do hotel em questão (Figura 3.4), onde os usuários da web podem solicitar reservas baseadas nas tarifas em apresentação no site.





**Figura 3.4 – I-Hotelaria, Ambiente de Exibição de Reservas**

O processo de cadastro e publicação de tarifas, utilizado como exemplo neste capítulo, envolve minimamente os seguintes elementos:

- **Unidade Habitacional:** o termo que se refere aos quartos do hotel disponibilizados para hospedagem;
- **Tarifa:** referente à organização dos períodos que regem as estadias dos hóspedes em um hotel. Uma Tarifa agrega a disponibilidade de unidades habitacionais (Agendamento de Validade), o preço da diária (Tabela de Preço), o período em que a tarifa será exibida nos sites de web (Agendamento de Exibição) e demais condições especiais de aquisição. Exemplo de Tarifa são: “Dia dos namorados”, “Natal”, “Reveillon”, etc.;
- **Agendamento de Validade:** período de dias em que ocorrerão as estadias de uma determinada Tarifa. O seu cadastramento é realizado pelos funcionários do hotel no ambiente administrativo da plataforma;
- **Solicitação de Reserva:** pedido enviado pelo hóspede ao hotel pelo site da web. Uma solicitação é vinculada a uma tarifa, os dias em que o usuário deseja permanecer no hotel e a unidade habitacional na qual deseja se hospedar;

- **Agendamento de Exibição:** período de dias em que uma tarifa será divulgada. O seu cadastro é realizado por funcionários do hotel no ambiente administrativo da plataforma;
- **Confirmação de Reserva:** confirmação realizada pelo hotel informando que a Solicitação de Reserva foi atendida; e
- **Tabela de Preços:** indica o preço praticado para uma unidade habitacional em um determinado período de dias da tarifa. O seu cadastro é realizado por funcionários do hotel no ambiente administrativo da plataforma.

### 3.3.1 A Fase de Engenharia de Domínio

Na abordagem TimeFEX, a ED, ou desenvolvimento *para* reutilização, é iniciada pela Análise de Domínio que trata da identificação, coleta, organização, e representação das informações relevantes do domínio (KANG *et al.*, 1990). Uma vez que a abordagem TimeFEX tem como foco a representação do conhecimento temporal e sua formalização no modelo de domínio, a fase de ED se utiliza de informações coletadas com o uso de outras metodologias complementares a este trabalho.

Na abordagem TimeFEX, a ED é composta por atividades definidas sob duas perspectivas distintas. A primeira delas, a Análise de Composição, trata da Identificação de Características e da Análise de Variabilidade que prevê configurações válidas de características definidas a partir de regras de composição inclusivas e exclusivas específicas do domínio tratado. Já na Análise de Comportamento, a perspectiva se volta para os aspectos relativos à composição e variabilidade de comportamentos que envolvem estas características.

E finalmente, por se basearem em perspectivas complementares, as atividades de Análise de Composição e Análise de Comportamento são desenvolvidas de forma concomitante, para que suas informações, presentes no modelo de domínio, estejam em conformidade ao fim da ED.

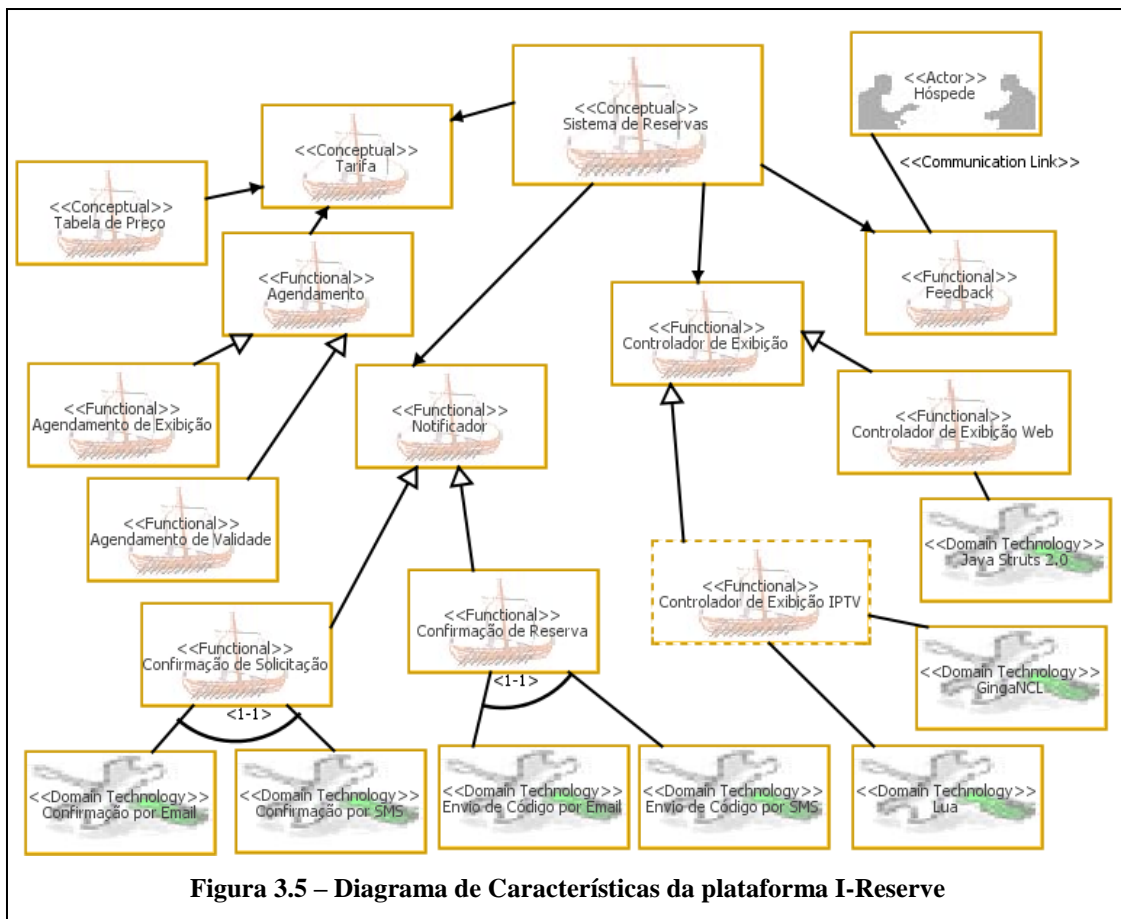
#### 3.3.1.1 Análise de Composição

A Análise de Composição é iniciada pela Identificação de Características. Esta identificação envolve todas as características do modelo de domínio. Ou seja, tanto

características participantes de comportamentos que se pretende analisar e modelar quanto características de software sem comportamento explícito.

Ao longo desta atividade, características são refinadas e organizadas de forma hierárquica em um Modelo de Características. A partir do conjunto de características identificadas, é realizada a Análise de Variabilidade. Neste momento, são definidas e integradas, ao modelo de características, as regras de composição inclusivas e exclusivas responsáveis por reger a composição de novos produtos e a adaptação de produtos existentes segundo as especificidades do domínio tratado.

Para a construção deste modelo, a abordagem TimeFEX possibilita o uso de notações compatíveis com a notação FODA (KANG *et al.*, 1990) e suas extensões. Para exemplificar o desenvolvimento deste artefato, é apresentado na Figura 3.5 um modelo de características construído a partir da notação Odyssey-FEX (OLIVEIRA, 2006). Os detalhes que compõem esta notação são apresentados no Apêndice B.



### 3.3.1.2 Análise de Comportamento

Na Análise de Comportamento, que constitui a primeira atividade de foco da fase de ED, a perspectiva comportamental é ressaltada em detrimento da composição de características, que por sua vez foi tratada anteriormente e de forma específica pela Análise de Composição. Nesta atividade, é desenvolvido um Modelo Comportamental que será utilizado de forma complementar ao Modelo de Características.

No início do desenvolvimento deste modelo, é realizada a transcrição das características presentes no Modelo de Características e que participam dos comportamentos que se deseja analisar. Estas características são transcritas para o novo modelo sob a forma de elipses, conforme apresentado na Figura 3.6.

Para aprimorar o entendimento do Modelo Comportamental no exemplo apresentado, será utilizada uma legenda de cores para o preenchimento das características segundo a Tabela 3.3. Assim, as características transpostas do modelo de características (Figura 3.5), para o Modelo Comportamental (Figura 3.6), passam a obedecer à taxonomia proposta na abordagem Odyssey-FEX apresentada no Apêndice B.

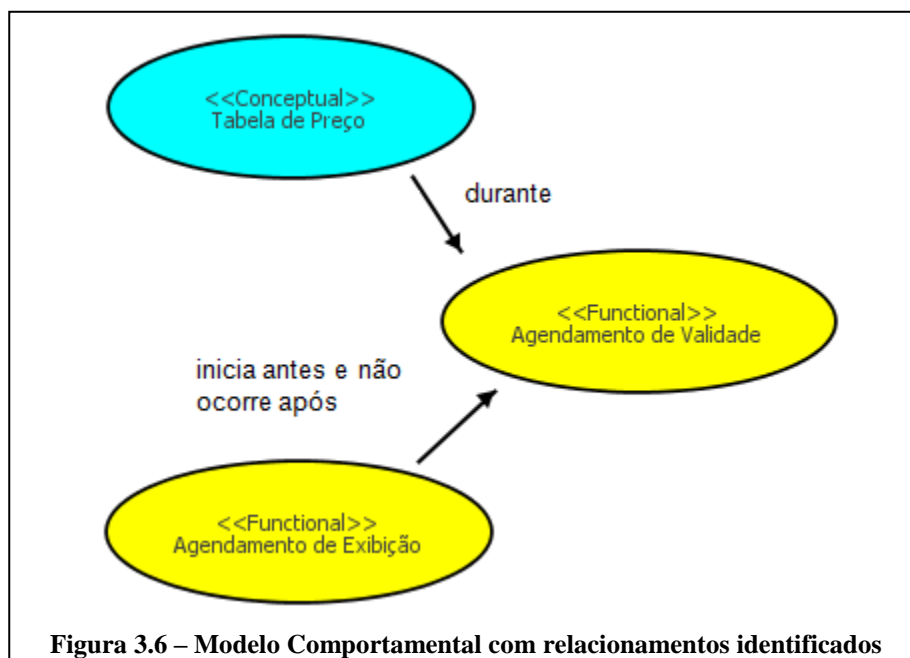
Tipo de Característica	Cor
Característica Conceitual	Ciano
Característica Funcional	Amarelo
Característica de Entidade	Branco
Característica de Ambiente Operacional	Vermelho Claro
Característica de Tecnologia de Domínio	Verde
Característica de Técnicas de Implementação	Azul

**Tabela 3.3 – Legenda de Cores da notação Odyssey-FEX**

No Modelo Comportamental, é desenvolvida uma rede de relacionamentos temporais entre as características de software presentes no modelo. A construção desta rede de relacionamentos é iniciada pela identificação dos requisitos específicos do domínio que são ligados a aspectos comportamentais, tais como ordenação temporal, dependência, precedência e de sincronização de forma geral.

Estes relacionamentos são modelados sob a forma de setas acompanhadas por expressões em linguagem natural, com o intuito de facilitar a compreensão do

comportamento e da dinâmica das interações entre as características envolvidas. Um exemplo desta etapa da modelagem é ilustrado através dos relacionamentos presentes na Figura 3.6.



Segundo as informações do domínio utilizado como exemplo, a característica *Tabela de Preços* é responsável por estipular o valor cobrado por uma *Unidade Habitacional* em um determinado período de dias da *Tarifa*. Este valor é referente às diárias que serão cobradas nos dias em que o hóspede esteve no hotel. Sendo assim, o período de vigência de uma *Tabela de Preços* deve ocorrer, obrigatoriamente, durante o período de *Agendamento de Validade*, já que esta segunda característica determina quais os dias disponíveis para reserva e, conseqüentemente, para futura cobrança de diárias.










Outro requisito exemplificado na Figura 3.6, ocorre entre as características *Agendamento de Exibição* e *Agendamento de Validade*. O *Agendamento de Exibição* se refere ao período de dias em que a divulgação publicitária da *Tarifa* ocorre nos meios de comunicação, e que, por consequência, deve ser iniciado antes do início do *Agendamento de Validade*. Ou seja, é mandatório que a *Tarifa* seja divulgada antes dos dias em que as unidades habitacionais serão de fato utilizadas. Esta disponibilidade, por sua vez, é determinada pelo *Agendamento de Validade*. E finalmente, o *Agendamento de Exibição* não deve perdurar após o *Agendamento de Validade*, pois a *Tarifa* não deve ser divulgada após a passagem dos dias em que as hospedagens ocorreram.

Por envolver elementos do Modelo de Características, a identificação de relações temporais pode levar à necessidade de criação de novas características ou ao refinamento das características existentes, ressaltando a complementaridade entre as duas análises e trazendo a necessidade de realizá-las de forma concomitante.

Uma vez identificados os relacionamentos temporais, as condições expressas em linguagem natural são traduzidas para a forma de relacionamentos temporais seguindo a taxonomia de restrições do Modelo de Allen (ALLEN, 1983), apresentada na Tabela 3.4. Ou seja, as expressões em linguagem natural são analisadas para se identificar as respectivas restrições sobre os intervalos, para que a sua disposição corresponda ao que está sendo expresso originalmente pelo requisito.

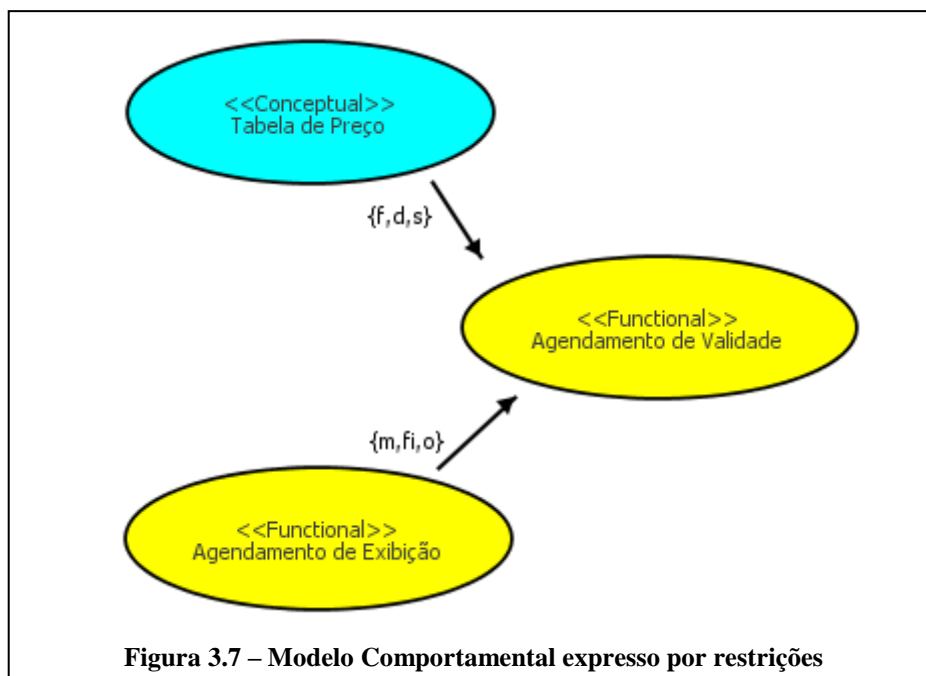
Conforme o exemplo apresentado, o relacionamento *Tabela de Preços* “durante” *Agendamento de Validade* indica que o período referente à primeira característica *inicia com* (s), *ocorre durante* (d) ou *finaliza com* (f) a característica *Agendamento de Validade*, conforme apresentado na Tabela 3.4.

Já o relacionamento *Agendamento de Exibição* “inicia antes e não ocorre após” *Agendamento de Validade*, também apresentado na Tabela 3.4, indica que a primeira característica *antecede imediatamente* (m), *sobrepõe o início* (o) ou *é finalizada* (fi) pela característica *Agendamento de Validade*.

Representação	Restrição	Representação	Restrição
	s		m
	d		o
	f		fi
 Tabela de Preço	 Agend. de Validade	 Agend. de Exibição	

**Tabela 3.4 – Tradução de relacionamentos temporais**

As restrições obtidas são então representadas no Modelo Comportamental entre chaves em substituição aos termos que acompanhavam anteriormente cada relacionamento temporal. O diagrama contendo a tradução do exemplo utilizado é apresentado na Figura 3.7.



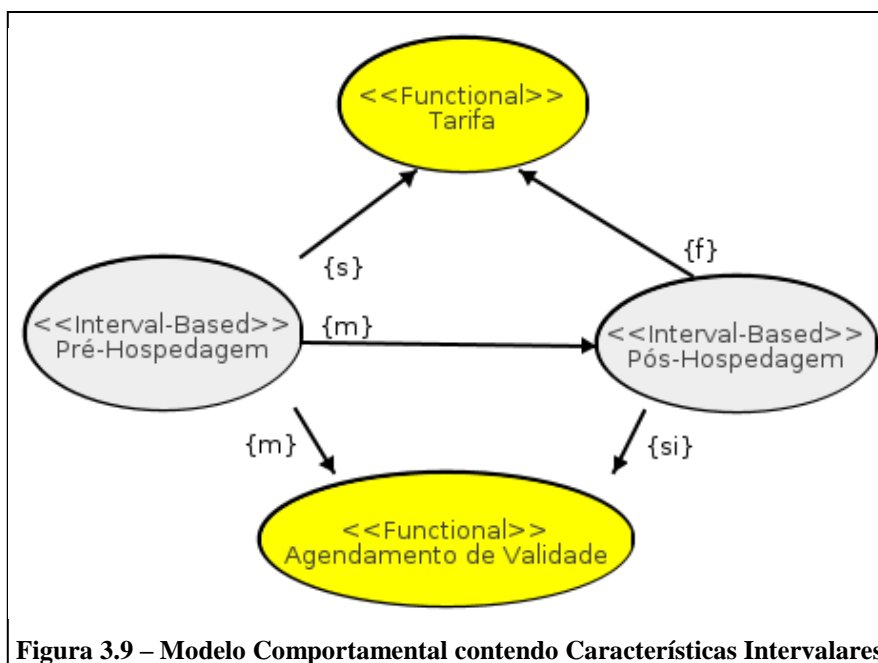
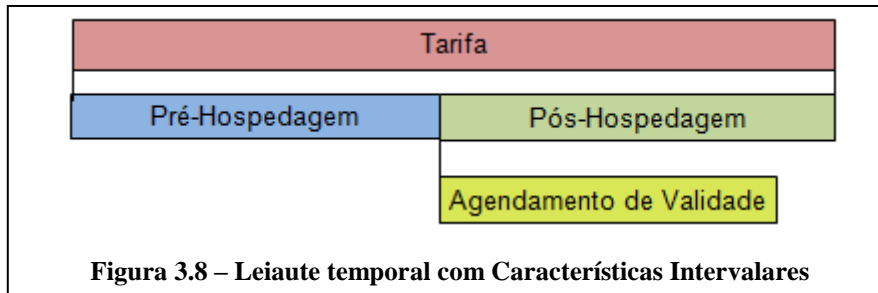
**Figura 3.7 – Modelo Comportamental expresso por restrições**

Durante a identificação e a tradução dos relacionamentos temporais, novas características referentes ao comportamento podem ser identificadas sem estarem necessariamente relacionadas ao Modelo de Características. Estes elementos são utilizados como referenciais temporais por outras características, proporcionando a diversificação de suas relações de forma direta e indireta. Estas novas características, chamadas de *Características Intervalares*, figuram exclusivamente na perspectiva comportamental, ao contrário das restantes, que podem figurar tanto no Modelo de Características quanto no Modelo Comportamental, ou exclusivamente no Modelo de Características.

Devido a sua semântica restrita, Características Intervalares se associam com as demais características exclusivamente por meio de relacionamentos temporais. Com isto, não impactam na perspectiva de composição de características, sendo portanto, transparente para as atividades de Análise de Composição e de Composição de Características.

Conforme o exemplo utilizado neste capítulo, uma *Tarifa* pode ter sua duração dividida internamente entre as características intervalares *Pré-Hospedagem* e *Pós-Hospedagem*, conforme apresentado na Figura 3.8. Estas duas novas características intervalares são relativas, respectivamente, ao período anterior e posterior ao início do *Agendamento de Validade*. Com isto, estas podem ser utilizadas como referenciais temporais na identificação de novos relacionamentos e características que envolvam as

características *Tarifa* e *Agendamento de Validade*. O Modelo Comportamental obtido a partir da inserção destas duas novas características intervalares é representado pela Figura 3.9.



Após a identificação dos relacionamentos temporais, é realizada a Análise de Variabilidade comportamental. Nesta atividade, é analisada a dinâmica de formação de configurações válidas de relacionamentos temporais. Os leiautes temporais formados por estes relacionamentos conferem os comportamentos possíveis aos produtos instanciados durante a atividade de Composição de Comportamento na Engenharia de Aplicação.







Este processo é análogo à Análise de Variabilidade que ocorre em nível de características. No entanto, regras de composição, que tradicionalmente envolvem características, passam a ser definidas neste modelo sobre relacionamentos temporais do Modelo Comportamental. Esta potencialidade converge para a necessidade de que relacionamentos temporais sejam utilizados como entidades de primeira classe.



As regras de composição de comportamento, definidas pelo negócio e captadas na Análise de Variabilidade, serão responsáveis por definir o conjunto de leiautes temporais válidos e passíveis de instanciação. Com isto, estas regras passam a reger os possíveis comportamentos que os produtos da linha poderão adquirir.

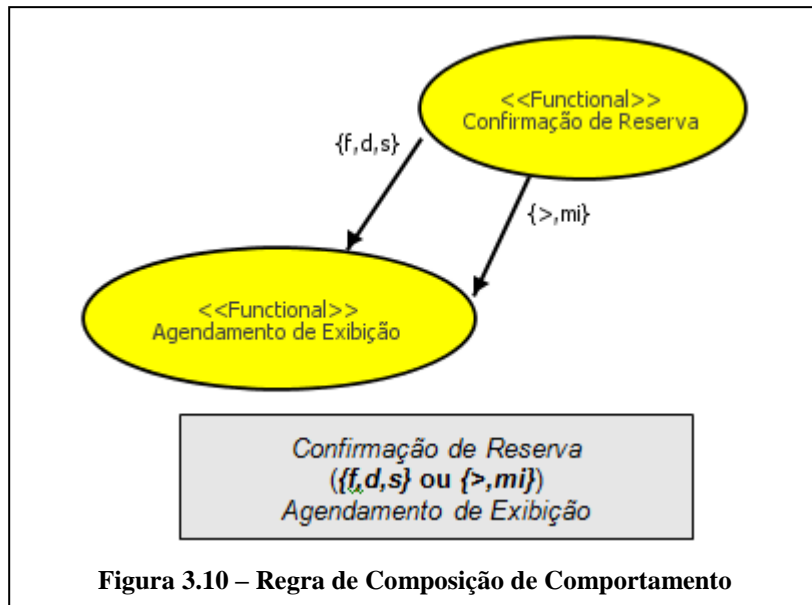
No exemplo utilizado, existem dois possíveis comportamentos para a característica *Confirmação de Reserva*. Em parte dos sistemas, as solicitações de reserva são realizadas de forma individual, já em outra parte, a tarifa é direcionada a um grupo específico de hóspedes, como por exemplo, os funcionários de uma empresa. Na primeira parcela, a *Confirmação de Reserva* é realizada logo após a solicitação de cada hóspede, ou seja, durante o *Agendamento de Exibição*. Porém, na segunda parcela de sistemas, estas confirmações somente são enviadas após a consolidação de todas as solicitações enviadas, ou seja, após o fim do *Agendamento de Exibição*.

Conforme apresentado na Tabela 3.5, a primeira parcela de sistemas descrita acima é representada por configurações onde o relacionamento *Confirmação de Reserva* {s,d,f} *Agendamento de Exibição* se faz presente. Já a segunda parcela, é obtida pelo relacionamento *Confirmação de Reserva* {>,mi} *Agendamento de Exibição*.

Representação	Restrição	Representação	Restrição
	s		mi
	d		>
	f		
			

**Tabela 3.5 – Relacionamentos Alternativos**

Na Análise de Variabilidade, realizada durante a Análise de Comportamento, esta alternância é modelada a partir de uma regra de composição de comportamento de mútua exclusão envolvendo dois relacionamentos temporais. Na Figura 3.10, é apresentado um modelo com os relacionamentos envolvidos na regra e que são acompanhados de uma representação textual da mesma.



### 3.3.2 Engenharia de Aplicação

Assim como na ED, a EA é realizada sobre duas perspectivas distintas e complementares: a composição isolada de características e a sua unificação em comportamentos esperados para o produto.

Em um primeiro momento, a Composição de Características é responsável pelo tratamento de quais características irão compor o produto pretendido. A partir deste conjunto, a Composição de Comportamento é responsável pela seleção dos relacionamentos temporais que definem o leiaute temporal correspondente ao comportamento do produto instanciado. Estas duas atividades são descritas a seguir, respectivamente, pelas Seções 3.3.2.1 e 3.3.2.2.

#### 3.3.2.1 Composição de Características

A atividade de Composição de Características é iniciada pela Seleção de Características, que por sua vez é realizada sobre as características obtidas da Análise de Composição executada durante a ED. Sendo assim, esta seleção é restrita às características presentes no Modelo de Características, excluindo, portanto, Características Intervalares presentes no modelo de domínio.

Em seguida, é realizada a Verificação de Consistência, quando é verificado se as características que compõem a seleção estão em conformidade com as regras de

composição de características inclusivas e exclusivas, bem como os pontos de variação definidos durante a Análise de Composição.

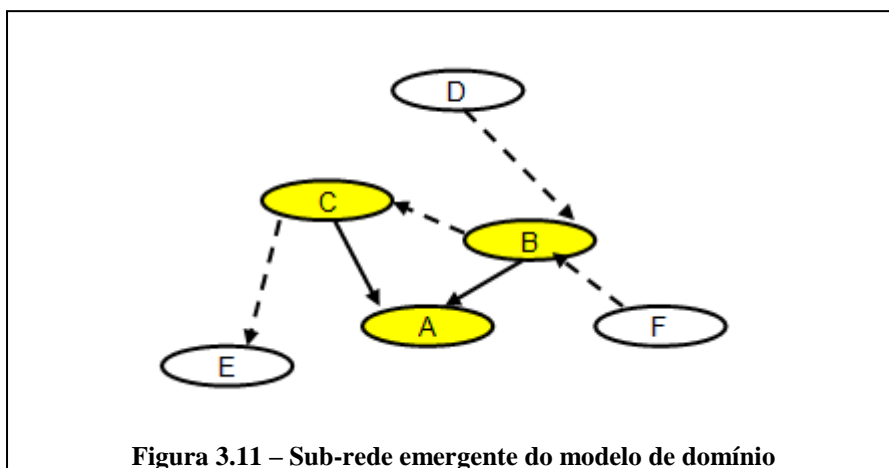
Devido à flexibilidade de utilização de diferentes notações para a modelagem de características em conjunto com a abordagem TimeFEX, esta atividade é igualmente flexível em relação a metodologia utilizada para a composição de características.

### 3.3.2.2 Composição de Comportamento

A atividade de Composição de Comportamento constitui o foco da abordagem no escopo da EA e é responsável por determinar o comportamento que o produto assumirá a partir das informações e regras colhidas do modelo de domínio. Inicialmente, é realizada a Seleção de Relacionamentos, momento em que relacionamentos temporais entre características são escolhidos para compor o leiaute temporal do novo produto. Esta seleção destacará um conjunto de relacionamentos do Modelo Comportamental que definem o comportamento do produto. Ou seja, da mesma forma que um subconjunto de características do modelo de características integram o produto a partir da composição de características, o comportamento do mesmo é regido por apenas um subconjunto de relacionamentos do modelo comportamental a partir da composição de comportamento. Ao conjunto de características e relacionamentos temporais escolhidos dá-se o nome de sub-rede emergente.

A Figura 3.11 apresenta características  $\{A, B, C, D, E, F\}$  e relacionamentos  $\{(B, A), (B, C), (C, A), (C, E), (D, B), (F, B)\}$  que integram um exemplo de modelo de domínio. Neste exemplo, as características  $\{A, B, C\}$ , em destaque, constituem as características selecionadas durante a composição de características e os relacionamentos selecionados  $\{(B, A), (C, A)\}$ , representados pelas setas contínuas, constituem os relacionamentos selecionados para compor a sub-rede emergente.

A seleção de relacionamentos é dependente da Composição de Características realizada anteriormente, dada a necessidade de que os relacionamentos selecionados associem exclusivamente características escolhidas durante a Composição de Características. Ou seja, nenhum relacionamento que envolve características externas esta composição pode compor a sub-rede. Sendo assim, os relacionamentos  $\{(C, E), (D, B), (F, B)\}$  não são candidatos a compor sub-rede, pois envolvem características externas à composição.



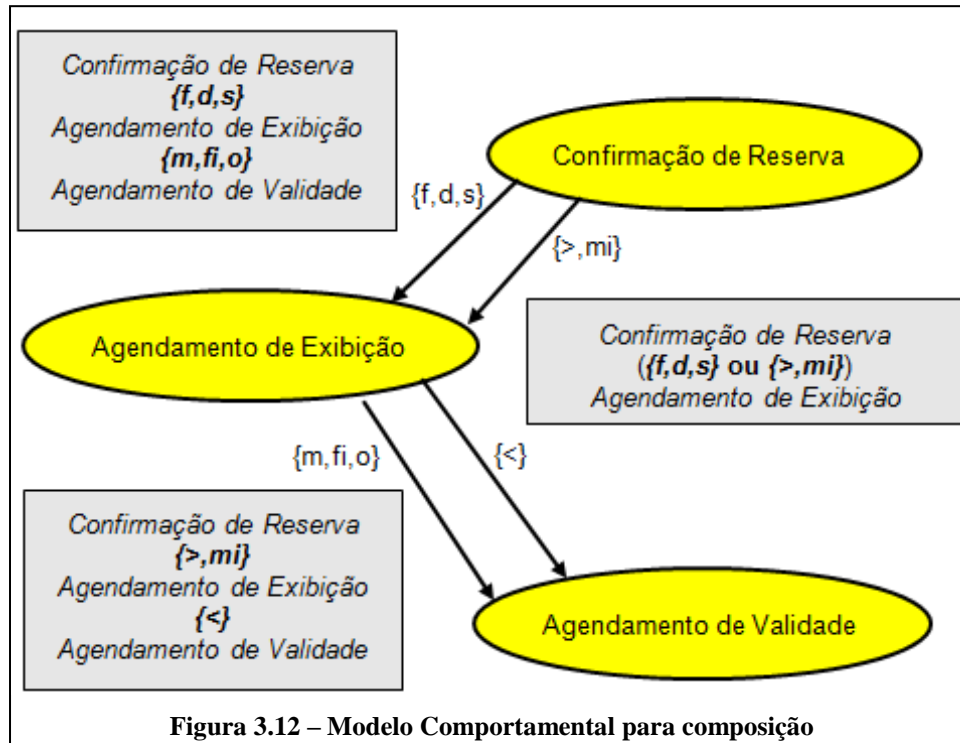
Conforme a dualidade observada anteriormente na ED, a composição de características afeta e é afetada pela composição de comportamento durante a EA. Esta influência pode ser observada quando, por exemplo, há a necessidade de um relacionamento que envolve uma característica externa ao escopo da composição de características anteriormente realizada, levando a uma revisão desta primeira composição. O inverso também pode ser observado quando a exclusão de uma característica, por exemplo, pode tornar a sub-rede emergente inconsistente, já que esta pode conter relacionamentos que envolvem a característica que foi removida. Este aspecto leva à necessidade da concomitância entre as duas composições.

Assim como a verificação de consistência realizada sobre a composição de características, os relacionamentos escolhidos também são inspecionados com o intuito de verificar se estes estão em conformidade com as regras de composição comportamental, inclusivas e exclusivas, estabelecidas pela Análise de Variabilidade durante a Análise de Comportamento.

Esta verificação pode ser exemplificada em configurações obtidas a partir do domínio tratado neste capítulo, com um incremento envolvendo a regra de composição de comportamento apresentada na Figura 3.10. Nesta primeira regra, é determinado que configurações possuam *Confirmação de Reserva* que ocorrem durante  $(\{f,d,s\})$  ou após  $(\{>,mi\})$  o *Agendamento de Exibição*, tratando, respectivamente, de sistemas de reservas individuais e sistemas de reserva em grupo.

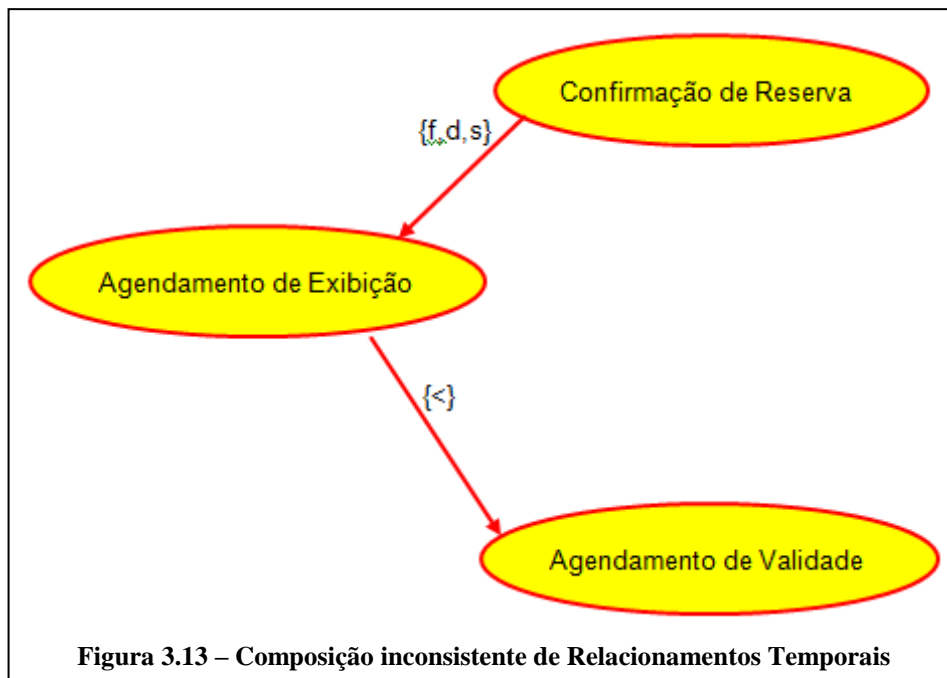
No caso específico das reservas em grupo, o *Agendamento de Exibição* não pode concorrer com o *Agendamento de Validade*, pois as hospedagens só ocorrem após a realização de reservas por todos os integrantes do grupo. Já no caso de reservas individuais, esta restrição não se aplica. Este fato agrega dois novos relacionamentos,

*Agendamento de Exibição* {<} *Agendamento de Validade* e *Agendamento de Exibição* {m, fi, o} *Agendamento de Validade*. Além disto, duas novas regras de composição inclusivas são geradas conjugando corretamente os quatro relacionamentos, conforme apresentado na Figura 3.12.



**Figura 3.12 – Modelo Comportamental para composição**

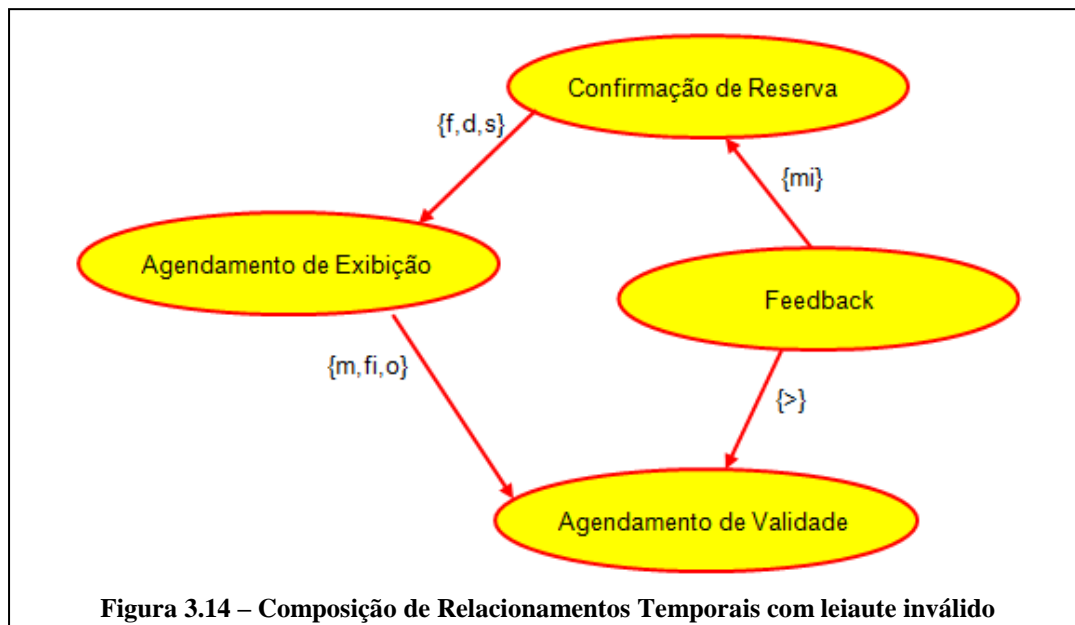
A partir deste modelo é apresentada, na Figura 3.13, uma configuração baseada na composição de comportamento onde os relacionamentos *Confirmação de Reserva* {f, d, s} *Agendamento de Exibição* e *Agendamento de Exibição* {<} *Agendamento de Validade* estão simultaneamente presentes. De acordo com a regra de composição inclusiva “*Confirmação de Reserva* {f, d, s} *Agendamento de Exibição* E *Agendamento de Exibição* {m, fi, o} *Agendamento de Validade*”, presente no modelo comportamental apresentado na Figura 3.12, a ausência do relacionamento temporal “*Agendamento de Exibição* {m, fi, o} *Agendamento de Validade*” na configuração da Figura 3.13, torna a referida configuração inconsistente.



Após a verificação de consistência em relação às regras de composição de comportamento, é realizada a validação do leiaute temporal expresso pelo comportamento que está sendo instanciado para o produto. Isto ocorre, pois o atendimento às regras de composição de comportamento, estabelecidas por regras de negócio, não impede que, ao final da composição, o comportamento instanciado possua inconsistências temporais.

Inconsistências temporais são baseadas em definições ambíguas entre dois eventos, mais especificamente, entre os inícios e os terminos entre dois intervalos de tempo. Na Figura 3.14, é apresentada uma composição obtida do modelo apresentado na Figura 3.12 com a inserção da característica *Feedback* e dois relacionamentos temporais posicionando a mesma em relação a *Confirmação de Reserva* e *Agendamento de Validade*. Apesar do modelo não possuir regras de composição de comportamento que impeçam esta configuração, este comportamento ainda assim é inconsistente.

De acordo com as regras de transitividade do Modelo de Allen (Apêndice C), duas relações do tipo  $A \{f, d, s\} B$  e  $B \{m, fi, o\} C$ , respectivamente, inferem um relacionamento do tipo  $A \{<, f, fi, o, d, s, =, m\} C$ . De forma análoga a este arranjo é possível realizar a mesma inferência onde  $A, B$  e  $C$  se referem, respectivamente, às características *Confirmação de Reserva*, *Agendamento de Exibição* e *Agendamento de Validade*. Neste caso, passa a ser inferido que *Confirmação de Reserva*  $\{<, f, fi, o, d, s, =, m\}$  *Agendamento de Validade*.



O conjunto de restrições deste novo relacionamento indica que o término da primeira característica é necessariamente anterior ao término da segunda. Com isto, uma definição em que a característica *Feedback* inicia após ( $\{>\}$ ) o término do *Agendamento de Validade* e imediatamente após o término da *Confirmação de Reserva* torna o leiaute desta composição inválido.

### 3.4 A notação TimeFEX

A notação TimeFEX foi desenvolvida para possibilitar o envolvimento dos elementos do modelo de características com intervalos de tempo em uma rede de relacionamentos temporais que expressa um comportamento variável entre um produto e outro. Esta notação foi utilizada no escopo deste trabalho com três objetivos específicos: exemplificar a modelagem do comportamento na explanação da abordagem, apresentada na Seção 3.3, fornecer uma base de representação para a ferramenta de modelagem descrita pelo Capítulo 4, e auxiliar na avaliação da abordagem realizada pelo experimento apresentado no Capítulo 5.

A notação apresentada a seguir é derivada da extensão realizada sobre a notação Odyssey-FEX (OLIVEIRA, 2006) e o seu respectivo metamodelo. Em conformidade com a estrutura original desta notação, foram realizadas adaptações em seus três pacotes de classes, descritos nas seções a seguir: Principal, Relacionamento e Regra de Composição.

### 3.4.1 Pacote Principal

No pacote Principal, se encontra representada a taxonomia de características e os elementos utilizados na sua agregação e organização. Esta organização é realizada através de Espaços de Nomes, que são conjuntos de elementos nomeados e identificados unicamente (OMG, 2010). Os elementos sobre os quais ocorre a organização de características são: Domínios, que tratam de uma composição de Contextos, que possuem agregados de Pacotes. Estes três elementos se tratam de concretizações de Espaços de Nomes organizadas de forma hierárquica. Neste sentido, não houve adequações necessárias para esta forma de organização de características.

Em relação à taxonomia de características e sua hierarquia, apresentadas a partir de um recorte feito no pacote Principal, foi realizada a inserção de dois novos elementos em destaque e de profundidade na árvore apresentada na Figura 3.15. As duas classes, *Característica Intervalar* e *Característica Não-Intervalar*, foram inseridas para compor mais um nível de categorização necessário para a organização dos tipos de relacionamentos possíveis entre cada uma destas duas categorias de características. Estes relacionamentos são apresentados a seguir, na Seção 3.4.2.

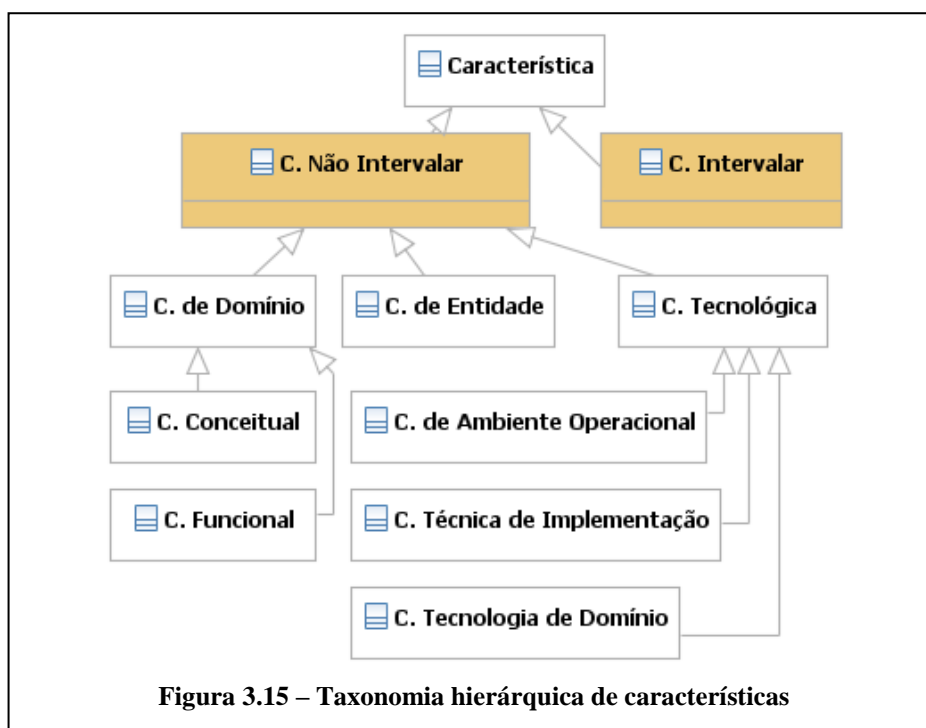


Figura 3.15 – Taxonomia hierárquica de características



### 3.4.2 Pacote Relacionamento

No pacote Relacionamento se encontram representados os elementos que definem as formas de interação e associação entre características. Os novos elementos definidos para o pacote são: a classe *Temporal*, referente ao relacionamento temporal entre duas características (classe *Característica*), que é composta por um conjunto de restrições, e a enumeração *Tipo Restrição*, que representa o dicionário de restrições do Modelo de Allen (Tabela 3.2) utilizadas pela classe *Temporal*. Além disto, as classes *Implementado Por*, *Associação*, *Alternativo* e *Generalização*, que se associam à classe *Característica* na notação Odyssey-FEX, passam a se associar exclusivamente à classe *Característica Não-Intervalar*, garantindo assim a independência dos elementos originais do Modelo de Características em relação a presença de Características Intervalares (classe *Característica Intervalar*) no modelo de domínio. E finalmente, todas as heranças estabelecidas no pacote se tratam de heranças simples.

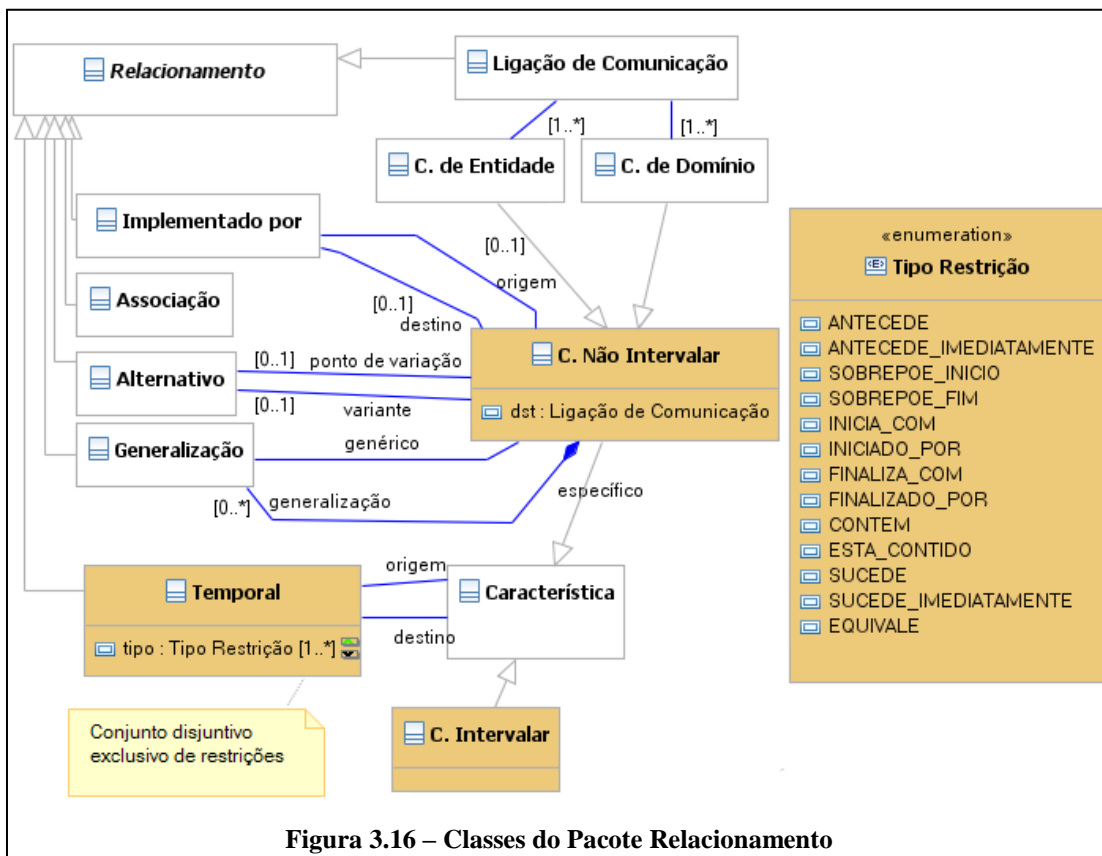


Figura 3.16 – Classes do Pacote Relacionamento

### 3.4.3 Pacote Regras de Composição

No pacote Regras de Composição, encontram-se representadas as classes referentes aos elementos utilizados na construção de regras de composição inclusivas e de mútua exclusão entre características. A alteração realizada sobre este pacote trata essencialmente da discriminação de *Expressões Literais* entre as classes *Característica Não-Intervalar* e a classe *Temporal*. Esta modificação possibilita a utilização de relacionamentos temporais em *Expressões Literais* como argumentos da regra de composição. Ou seja, regras de composição passam a reger a composição de características e relacionamentos temporais. Para isto, são necessárias duas restrições ao metamodelo para manter a consistência deste novo tipo de regra de composição específico para o comportamento:

- R1:** As associações entre a classe *Expressão Literal* e as classes *Temporal* e *Característica Não-Intervalar* são mutuamente exclusivas;
- R2:** As composições de *Regra de Composição* (consequente e antecedente) devem conter *Expressões Literais* com o mesmo tipo de associação, *Temporal* ou *Característica Não-Intervalar*.

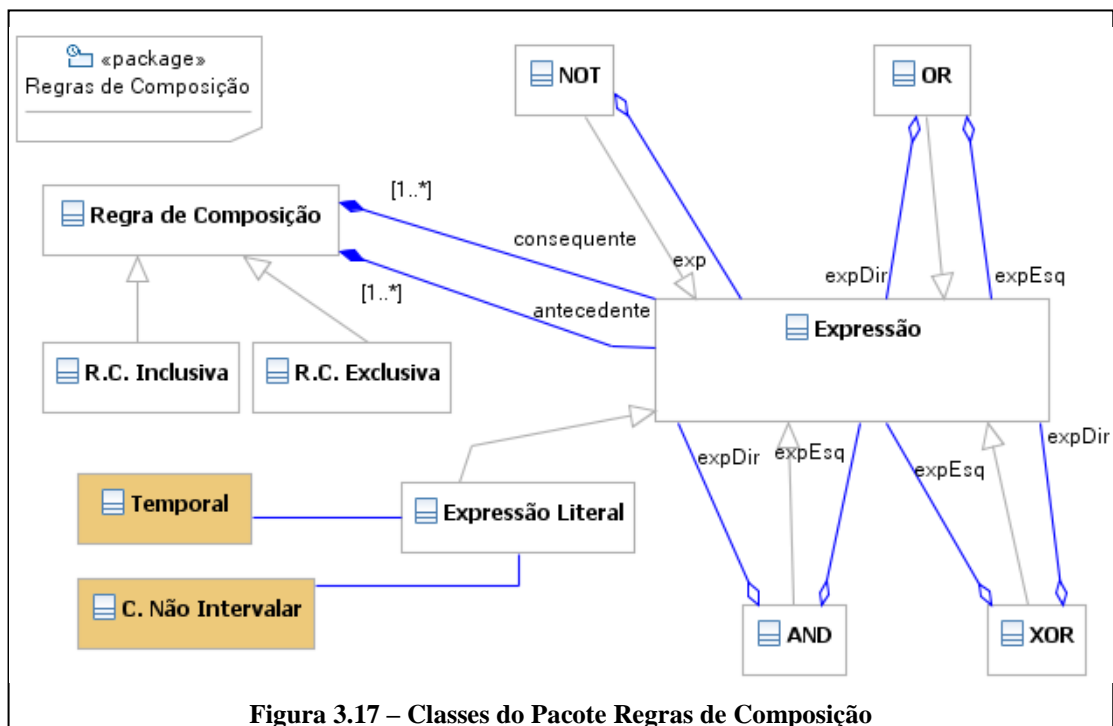


Figura 3.17 – Classes do Pacote Regras de Composição

### 3.5 Análise Comparativa

Em prosseguimento à análise realizada no Capítulo 2, a abordagem TimeFEX foi comparada em relação aos requisitos listados na Seção 2.5 deste mesmo capítulo. Apesar de a abordagem atender à totalidade de requisitos apontados na Tabela 3.6, este atendimento não posiciona a solução, *a priori*, como superior às demais em todos os aspectos. Algumas das abordagens analisadas no Capítulo 2 são direcionadas para determinados domínios de aplicação, como é o caso da Modelagem Racionalizada, ou orientadas a determinadas abordagens de desenvolvimento, como é caso da orientação a casos de uso. Para estas situações mais específicas estas abordagens são potencialmente superiores à abordagem TimeFEX.

Requisitos/Abordagens	DE*	FeatuRSEB	PLUSS	Modelagem Racionalizada	TimeFEX
Interação Temporal Livre em Composição	Não	Não	N/A	Não	<b>Sim</b>
Validação Temporal	Sim	Não	Não	Não	<b>Sim</b>
Variabilidade Complementar	Não	Não	N/A	Não	<b>Sim</b>
Múltipla Granularidade	Sim	Não	Sim	Sim	<b>Sim</b>
Interação Temporal Livre em Hierarquia	Sim	Não	Não	Não	<b>Sim</b>
Interação Temporal Livre em Granularidade	Sim	Não	Sim	Sim	<b>Sim</b>

Tabela 3.6 – Comparação entre as abordagens de modelagem de comportamento e a abordagem TimeFEX

O atendimento do primeiro requisito, a Interação Temporal Livre em Composição, é demonstrado na Análise de Comportamento descrita na Seção 3.3.1.2. No exemplo utilizado, a variabilidade de comportamento entre as características *Agendamento de Exibição* e *Confirmação de Reserva* é expressa no modelo apresentado na Figura 3.10. Neste sentido, configurações equivalentes, ou seja, onde estas duas características estão presentes, podem apresentar comportamentos distintos. Esta variabilidade é dada pelos diferentes relacionamentos temporais envolvidos no exemplo.

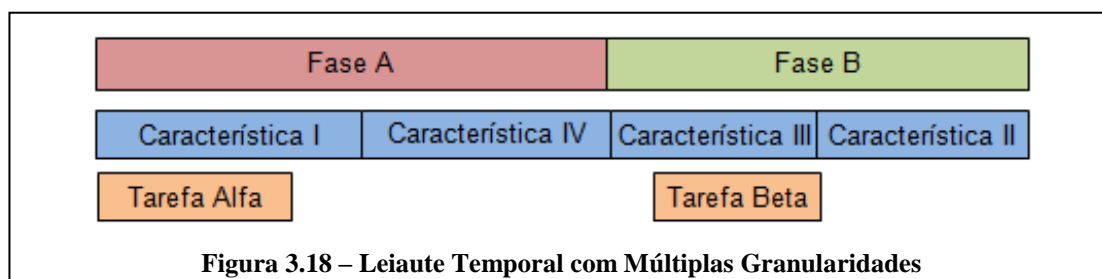
Outro requisito, a Validação Temporal, é observado no processo apresentado imediatamente após a verificação de consistência de comportamento em relação às

regras de composição, apresentada na Seção 3.3.2.2. Com isto, a capacidade de validação de leiaute temporal é agregada à abordagem TimeFEX.

A Variabilidade Complementar é demonstrada ao longo das diversas atividades do processo prescritivo, onde a complementaridade entre as perspectivas de características e comportamento é ressaltada. Estas atividades são caracterizadas por possibilitar a identificação de inconsistências entre o Modelo de Características e o Modelo Comportamental. Durante a EA, por exemplo, uma inconsistência gerada pela remoção de uma característica e a permanência indevida de um relacionamento temporal que a envolve na sub-rede emergente caracteriza esta complementaridade.

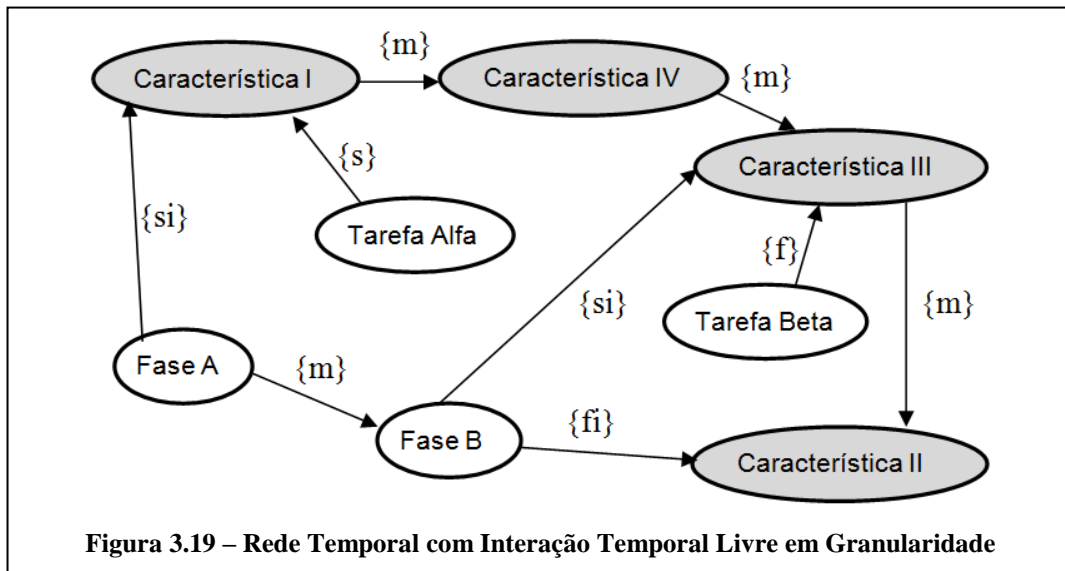
Os requisitos de Múltipla Granularidade e de Interação Temporal Livre em Granularidade são atendidos através do uso de Características Intervalares. Com esta categoria de característica, referenciais internos e externos às características não-intervalares, ou seja, presentes no Modelo de Características, podem se relacionar e integrar a rede de relacionamentos temporais em diferentes níveis de granularidade.

Conforme o exemplo apresentado na Figura 3.18, as características intervalares Fase A e Fase B determinam referenciais de maior granularidade, externos aos elementos *Característica I*, *Característica II*, *Característica III* e *Característica IV*. Além disso, as características intervalares *Tarefa Alfa* e *Tarefa Beta* constituem referenciais de menor granularidade, internos a algumas destas mesmas características. Os relacionamentos que estabelecem este leiaute são apresentados na Figura 3.19. Neste exemplo, a utilização das Características Intervalares viabiliza o atendimento ao requisito de Múltipla Granularidade.

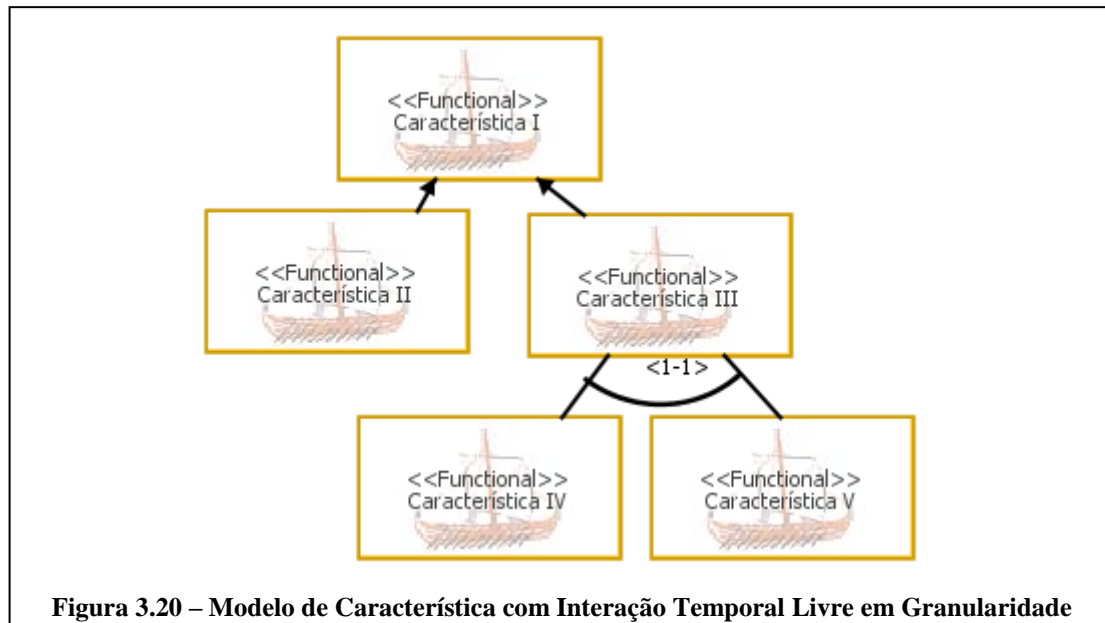


Além disto, é possível observar, na rede apresentada na Figura 3.19, que elementos de granularidades distintas se associam a partir de relacionamentos temporais, como por exemplo, o relacionamento *Fase A*  $\{si\}$  *Característica I*. Esta

capacidade de relacionar elementos de granularidades distintas tem como objetivo o atendimento ao requisito de Interação Temporal Livre em Granularidade.



A disposição em um grafo não hierárquico na qual o Modelo Comportamental se baseia, e o fato da semântica do Modelo de Características não restringir o estabelecimento da rede de relacionamentos temporais, possibilitam que qualquer característica se relacione com outra independentemente de sua disposição na hierarquia do Modelo de Características. Neste sentido, a Interação Temporal Livre em Hierarquia passa a ser observada, conforme o Modelo Comportamental apresentado na Figura 3.19, que é relativo ao Modelo de Características, apresentado na Figura 3.20, prescindindo da disposição hierárquica dos elementos *Característica I*, *Característica II*, *Característica III* e *Característica IV*. Neste exemplo, a *Característica V* não integra o Modelo Comportamental pois não possui relacionamentos temporais com outros elementos, ou seja, não agrega informações acerca do comportamento da linha.



### 3.6 Comentários Finais

Neste capítulo, foram apresentadas as atividades de processo, a notação e o metamodelo que, em conjunto, compõem os elementos conceituais da abordagem TimeFEX. As atividades propostas na Seção 3.3 não buscam definir um processo único ou limitar a abordagem a um único modelo de processo. Os pontos fundamentais desta seção se baseiam, principalmente, na proposta de um dicionário de atividades e no destaque dado ao caráter complementar entre a perspectiva estrutural e a perspectiva temporal ao longo de todo o processo. Neste sentido, em relação às atividades, aspectos como ordem de execução, número de iterações e precedência de execução não são contribuições pretendidas no escopo tratado. Com isto, a utilização da abordagem em conjunto com diferentes metodologias e modelos de processo é viabilizada.

Em relação ao metamodelo apresentado na Seção 3.4, é importante ressaltar que a notação implementada por este não se trata de uma solução única para a abordagem, ou seja, outras notações, mesmo que não formalizadas em nível de metamodelo, são passíveis de serem utilizadas em conjunto com a abordagem proposta. Porém, a utilização de outras notações permanece condicionada unicamente ao uso do Modelo Comportamental e da equivalência entre intervalos e características de software.

# Capítulo 4

## Ambiente de Reutilização de Suporte à Abordagem

### 4.1 Introdução

No Capítulo 3, foram apresentados os elementos conceituais da abordagem TimeFEX para a análise de variabilidade e composição de comportamento. Neste capítulo, serão apresentados detalhes sobre a construção e o funcionamento das ferramentas desenvolvidas para compor um ambiente de reutilização concebido para apoiar as atividades do processo prescritivo apresentado no capítulo anterior. Estas ferramentas compreendem um ambiente de modelagem de características e de comportamentos, e uma API (*Application Programming Interface*) aplicada a este ambiente para o tratamento de leiautes temporais em conformidade ao Modelo de Allen (ALLEN, 1983).

O ambiente de modelagem desenvolvido é interno à plataforma Odyssey Light (ODYSSEY, 2010), ferramenta referida neste trabalho simplesmente como Odyssey. Neste desenvolvimento, foram necessárias extensões sobre o metamodelo da notação Odyssey-FEX (OLIVEIRA, 2006), originalmente utilizada pela plataforma Odyssey para a modelagem de características. A API de tratamento de leiautes temporais foi acoplada ao ambiente de modelagem desta plataforma especificamente para a validação dos leiautes temporais durante a EA.

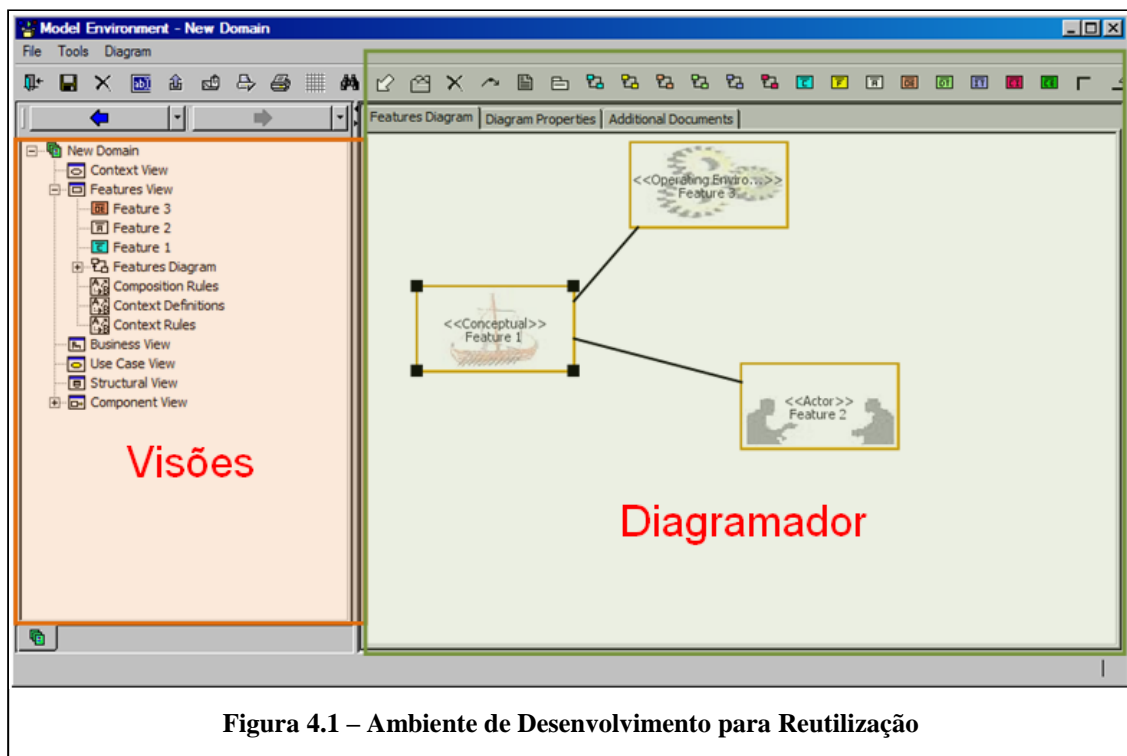
Neste capítulo, a Seção 4.2 apresenta um breve resumo sobre o ambiente Odyssey e como a notação Odyssey-FEX se encontra implementada neste ambiente. Na Seção 4.3, são apresentados os detalhes sobre novas funcionalidades e adequações realizadas sobre a plataforma Odyssey para que esta comporte a Análise de Variabilidade de comportamento orientada a características. Na Seção 4.4, são apresentados detalhes da API de tratamento de leiautes temporais utilizada para a validação dos comportamentos instanciados durante a EA. E finalmente, na Seção 4.5, são apresentados alguns comentários finais sobre os itens apresentados no capítulo.

## 4.2 O Ambiente Odyssey

O ambiente Odyssey (ODYSSEY, 2010) é uma plataforma de desenvolvimento voltada para a reutilização de software baseada em funcionalidades de suporte às atividades de Engenharia de Domínio (desenvolvimento *para* reutilização) e de Engenharia de Aplicação (desenvolvimento *com* reutilização). A sua arquitetura possibilita a utilização de diferentes *plugins* compatíveis com a plataforma que podem ser agregados à mesma de forma dinâmica (FERNANDES *et al.*, 2007). Exemplos destes *plugins* são apresentados em (MURTA *et al.*, 2001), (VERONESE *et al.*, 2002), (DANTAS *et al.*, 2001) e (MAIA *et al.*, 2005).

A customização realizada sobre a plataforma, para a utilização da abordagem TimeFEX, se concentrou especialmente em seu ambiente de ED, prescindindo a apresentação de detalhes sobre outras partes da plataforma no escopo deste trabalho.

O ambiente de ED se baseia em visões acompanhadas por diagramadores específicos, são estas: a visão de Escopo (*Context View*), a visão de Características (*Features View*), a visão de Negócio (*Business View*), a visão de Casos de Uso (*Use Case View*) e a visão de Componentes (*Component View*), conforme o leiaute gráfico apresentado na Figura 4.1.





Uma das principais funcionalidades fornecidas neste ambiente é a modelagem de características, através da visão de características. Esta modelagem é fundamentada na notação Odyssey-FEX (OLIVEIRA, 2006), apresentada em detalhes no Apêndice B. Com isto, a especialização realizada no escopo deste trabalho abrange tanto a plataforma Odyssey, enquanto ferramenta, quanto a notação Odyssey-FEX, enquanto metamodelo, este último apresentado no Capítulo 3.

### 4.3 O Ambiente TimeFEX

A primeira e principal evolução feita sobre a ferramenta foi a criação de uma nova visão e um novo diagramador específicos para a perspectiva comportamental. Esta visão, chamada de Visão Comportamental (*Behavioral View*), e seus elementos são apresentados na Figura 4.2. Nesta visão, durante o desenvolvimento do modelo de domínio, o usuário cria Diagramas de Comportamento (*Temporal Relations Network*) e os demais elementos do Modelo Comportamental descritos em detalhes no Capítulo 3, são estes: as Características Intervalares (*Interval-Based Features*), os Relacionamentos Temporais (*Temporal Relations*) e as regras de composição de comportamento inclusivas e exclusivas (*Layout Rules*).

Além das Características Intervalares, o Diagrama de Comportamento agrega características específicas do Modelo de Características, chamadas neste trabalho de Características Não-Intervalares. No caso do ambiente em questão, estas características são correspondentes às categorias específicas da notação Odyssey-FEX, como por exemplo, a de Características Funcionais.

As categorias de características Não-Intervalares contempladas pelo ambiente de modelagem são apresentadas no Apêndice B. Para adicionar uma Característica Não-Intervalar ao Diagrama de Comportamento, basta arrastar a característica pré-existente do modelo da Visão de Características (*Features View*) para o Diagrama de Comportamento aberto para edição. Com isto, todos os tipos de características, intervalares ou não, passam a se associar a partir de relacionamentos temporais no Diagrama de Comportamento.

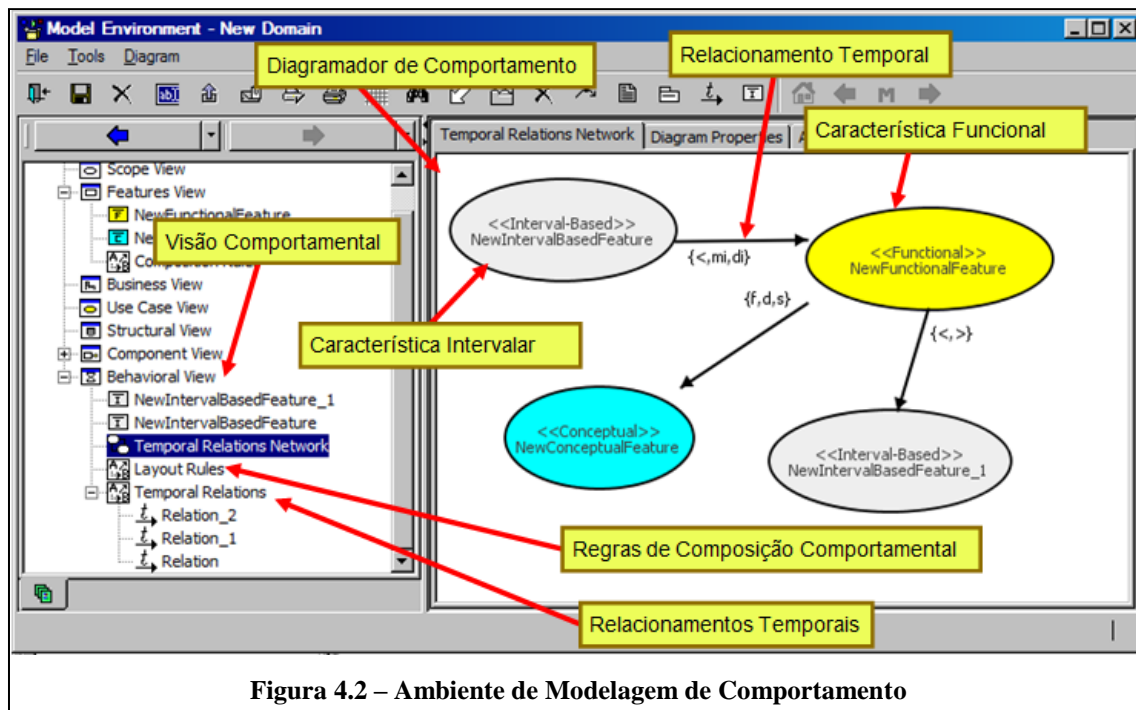

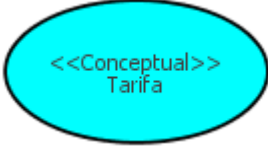


Figura 4.2 – Ambiente de Modelagem de Comportamento

Apesar de todas as categorias de características poderem integrar o Diagrama Comportamental, o mesmo não ocorre em relação ao Diagrama de Características, conforme a restrição da abordagem apontada na Seção 3.3.1.2. Segundo esta restrição, coberta pela ferramenta, a criação de Características Intevalares é restrita ao Modelo Comportamental e a criação de Características Não-Intevalares ao Modelo de Características. Sendo assim, a interrelação entre estas duas categorias de características é restrita aos relacionamentos temporais estabelecidos nos Diagramas Comportamentais. Além disto, características intervalares não participam do Diagrama de Características, pois, como se tratam apenas de referenciais temporais, não afetam a análise de variabilidade e a composição em nível de características.

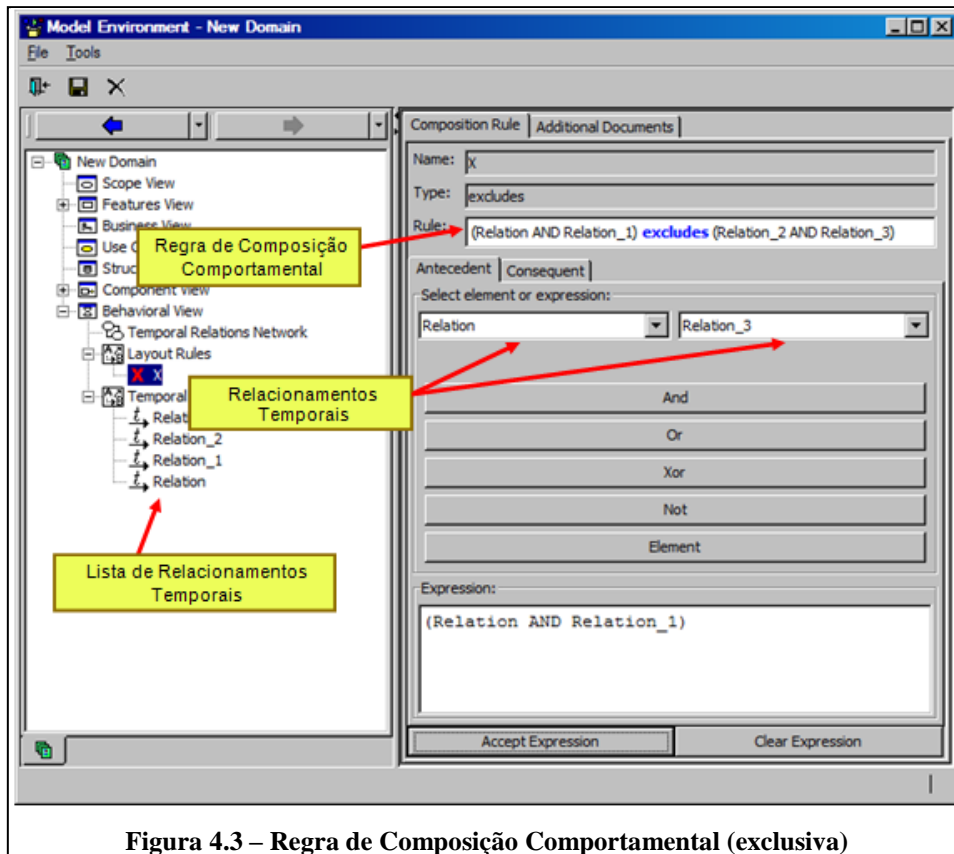
Dentre as evoluções realizadas sobre o ambiente, duas se destacam por seu impacto na implementação sobre a ferramenta anterior. A primeira trata da dissolução de uma restrição da plataforma que originalmente utilizava uma representação gráfica única para cada tipo de elemento para todos os seus diagramas. Esta restrição, nativa da ferramenta, foi tratada uma vez que, na abordagem TimeFEX, características passam a ser representadas de duas formas distintas: no diagrama de características e no diagrama comportamental, podendo figurar simultaneamente nos dois diagramas, conforme apresentado na Tabela 4.1.

 <p data-bbox="456 389 770 465">Representação de Característica Conceitual no Diagrama de Características</p>	 <p data-bbox="815 389 1129 465">Representação de Característica Conceitual no Diagrama de Comportamento</p>
--	--

**Tabela 4.1 – Representações distintas entre diagramas**

Outra importante adequação fundamental para a abordagem foi a valorização de associações entre os elementos, mais especificamente dos relacionamentos temporais presentes na visão comportamental. Na versão da plataforma Odyssey sem suporte a abordagem TimeFEX, as diferentes formas de associação presentes nos modelos não eram tratadas como entidades de primeira-classe. Isto se caracterizava pelo fato das associações serem visualizadas exclusivamente nos diagramas, não figurando nas árvores de elementos dos modelos nas diferentes visões (Figura 4.1). Além disto, estas associações não podiam integrar os parâmetros de regras de composição. Este último impedimento, em especial, conflita com a necessidade da abordagem de utilizar regras de composição de comportamento para a instanciação do leiaute temporal através da escolha consistente de relacionamentos temporais, que por sua vez se tratam, essencialmente, de associações.

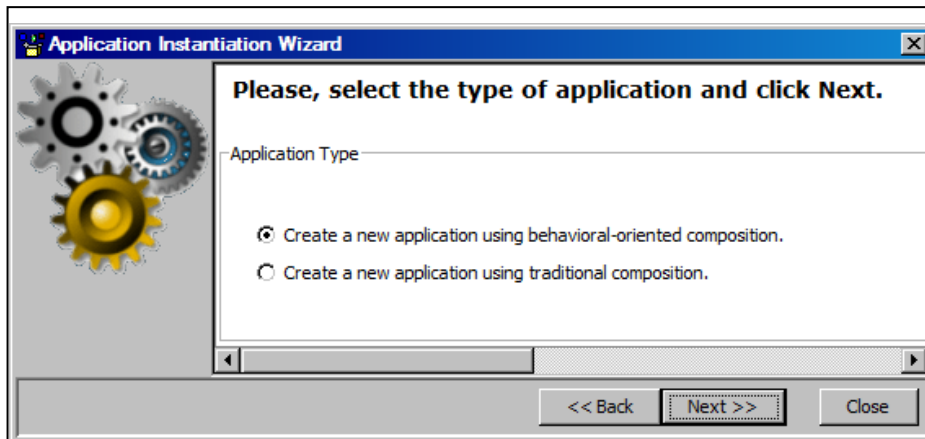
Sendo assim, para a utilização adequada de relacionamentos temporais presentes no modelo de domínio, estes foram integrados à árvore da visão comportamental através de uma listagem específica (*Temporal Relations*). Além disto, a plataforma foi adequada para possibilitar que este tipo de elemento integre os parâmetros utilizados na formulação das regras de composição específicas para o comportamento, em uma listagem própria (*Layout Rules*) do Modelo Comportamental. Estas funcionalidades são apresentadas na Figura 4.3.



**Figura 4.3 – Regra de Composição Comportamental (exclusiva)**

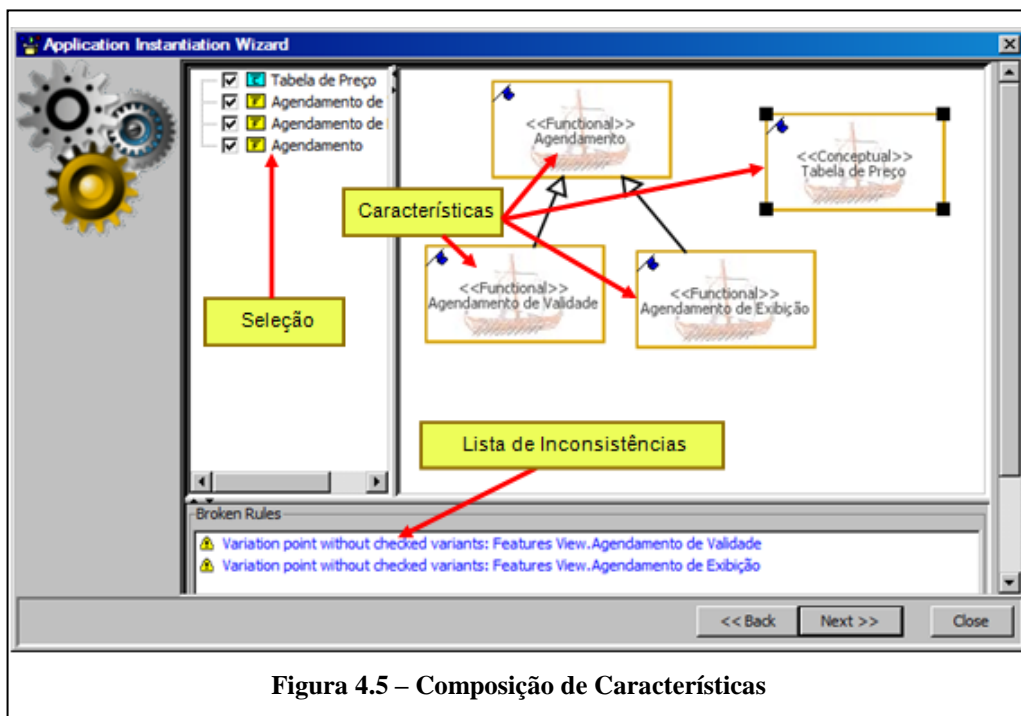
Durante a EA, após a utilização do ambiente de ED, o usuário da plataforma Odyssey realiza a instanciação de aplicações com as informações presentes no modelo de domínio desenvolvido. Este processo é composto por etapas disponibilizadas na forma de um guia de instanciação (*wizard*) que auxilia na configuração da uma nova aplicação, através da seleção dos elementos do modelo de domínio que irão integrá-la, e na verificação desta seleção em relação às regras de boa formação e de composição. Para que este auxílio contemple a instanciação do comportamento, este guia foi acrescido de dois novos passos.

O primeiro passo inserido se trata de um questionamento direcionado ao usuário, para indicar se este deseja utilizar o conhecimento temporal presente no modelo de domínio para auxiliar na instanciação do produto, conforme apresentado na Figura 4.4. Mesmo que este tipo de conhecimento esteja presente no modelo de domínio, a instanciação pode ser realizada de forma transparente, com o descarte deste conhecimento e a utilização exclusiva do modelo de características de acordo com a escolha do usuário. Isto é possível devido a independência entre as perspectivas.



**Figura 4.4 – Decisão sobre uso de conhecimento temporal na EA.**

Após esta escolha, o usuário realiza a seleção de características do Modelo de Características que irão compor o novo produto, conforme apresentado na Figura 4.5. Esta seleção não apresenta Características Intervalares, pois estas não integram este modelo. Esta funcionalidade é anterior às adequações realizadas sobre a ferramenta, mas é análoga e oferece suporte às atividades de Seleção de Características e Verificação de Consistência previstas pelo processo apresentado no Capítulo 3.



**Figura 4.5 – Composição de Características**

Caso o usuário tenha optado anteriormente pela utilização do conhecimento temporal na instanciação, é inserido ao processo um passo específico envolvendo a

seleção dos elementos da visão comportamental. Neste novo passo, é apresentada ao usuário uma lista de relacionamentos temporais passíveis de integrar o comportamento a ser instanciado, conforme apresentado na Figura 4.6 e exemplificado pelos elementos *Relation\_1*, *Relation\_2*, *Relation\_3* e *Relation\_4*. A lista de relacionamentos que é apresentada é restrita aos relacionamentos que envolvem as características escolhidas no passo anterior, obedecendo ao recorte indicado na atividade de Composição de Comportamento, detalhada no Capítulo 3.

Nesta etapa da EA, as duas verificações previstas pela abordagem estão implementadas pelo ambiente de reutilização. A Verificação de Consistência verifica a consistência da seleção de relacionamentos realizada pelo usuário em relação às regras de composição de comportamento e é exemplificada pelo primeiro aviso da caixa de verificação (*Broken Rules*) posicionada no inferior da Figura 4.6. Este aviso trata especificamente de uma regra de composição de mútua exclusividade que não foi obedecida pela seleção.

Já a Validação de Leiaute Temporal valida a consistência do comportamento, da mesma seleção, em conformidade com o Modelo de Allen e é exemplificado pelo segundo aviso da mesma caixa de verificação.

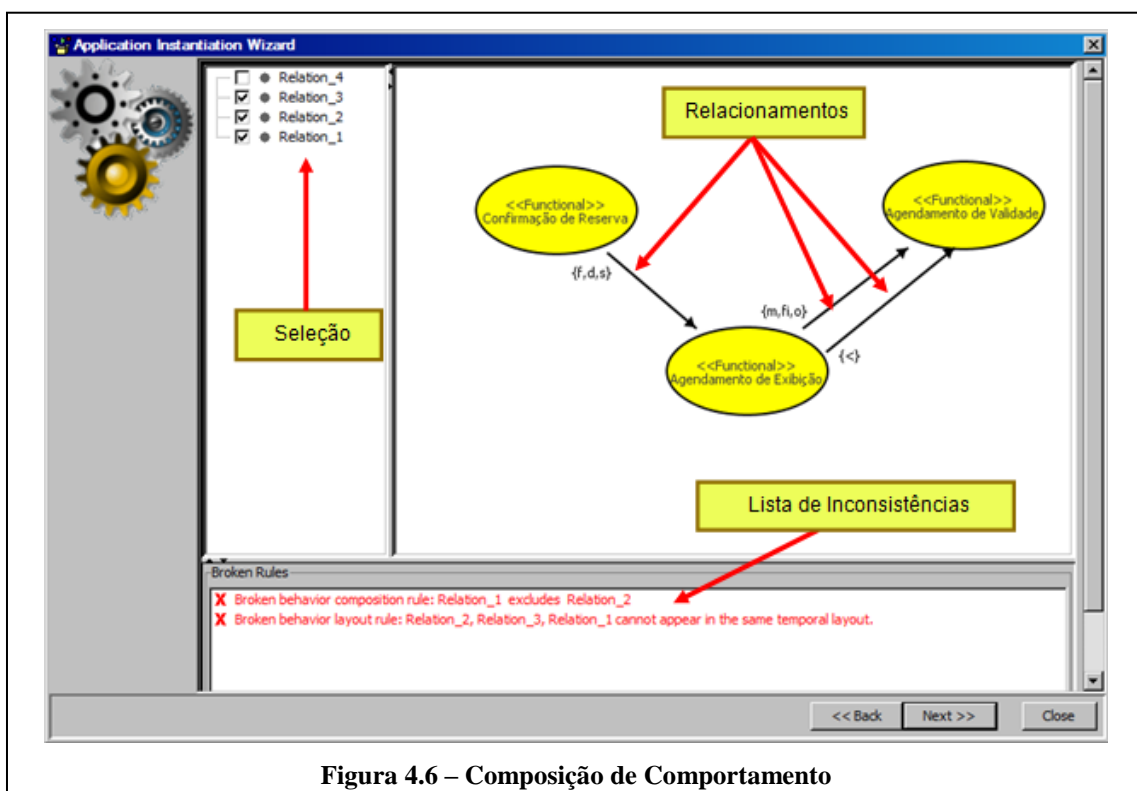


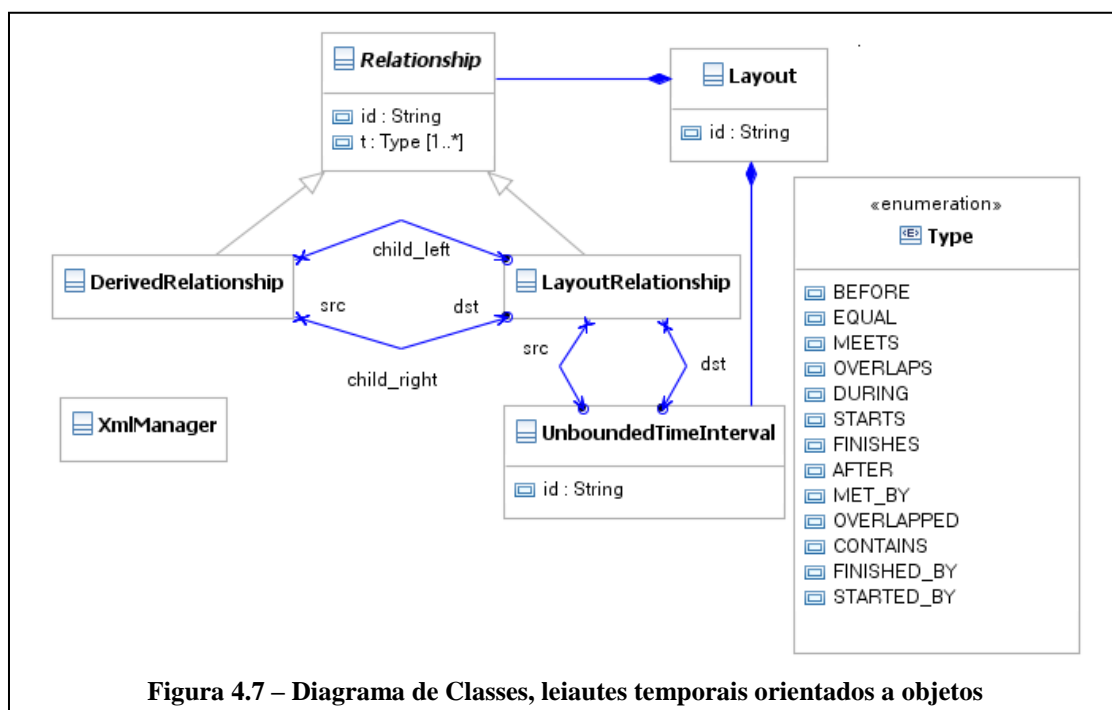
Figura 4.6 – Composição de Comportamento

Inconsistências geradas pela Validação do Leiaute Temporal, em especial, são geradas a partir de uma API agregada ao ambiente Odyssey, detalhada na seção a seguir.

#### 4.4 API de tratamento de leiautes temporais

A ferramenta desenvolvida para o tratamento de leiautes temporais possui três objetivos específicos: representar leiautes temporais em uma estrutura de dados orientada a objetos, promover a interoperabilidade de leiautes temporais entre aplicações e fornecer mecanismos para a validação de leiautes.

Para a representação de leiautes temporais em um ambiente de desenvolvimento e execução orientado a objetos, foram desenvolvidas classes na linguagem Java (ORACLE, 2010) capazes de expressar estruturas de dados referentes a leiautes temporais compatíveis com o Modelo de Allen. Estas classes estão detalhadas e representadas no diagrama da Figura 4.7.



A classe *UnboundedTimeInterval* representa intervalos de tempo especificados de forma relativa, ou seja, posicionados no tempo unicamente através das restrições do Modelo de Allen envolvendo outros intervalos. Sendo assim, não possuem valores

absolutos de posicionamento no tempo e, conseqüentemente, não possuem especificação de duração.

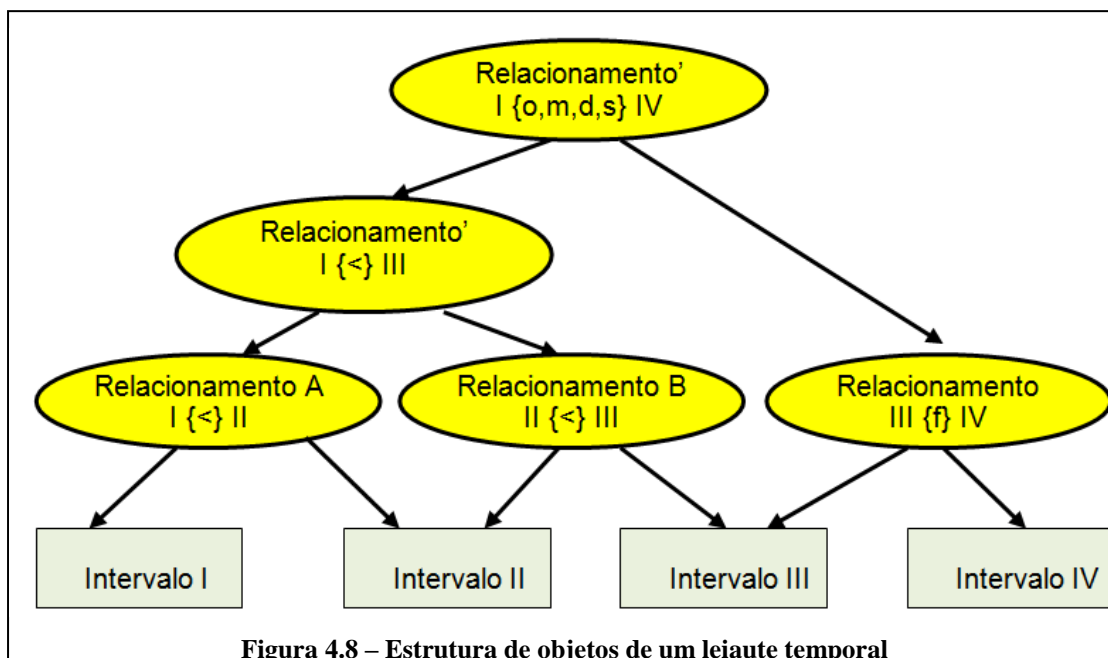
A classe abstrata *Relationship* representa um relacionamento temporal identificado unicamente e que possui um conjunto de restrições obtido a partir do dicionário de restrições da enumeração *Type*.

A classe *LayoutRelationship* representa uma especialização da classe *Relationship* e corresponde a um relacionamento temporal definido explicitamente para o leiaute, ou seja, especificado pelo código do usuário da API em tempo de execução. Neste sentido, não é derivado de outros relacionamentos temporais. Ao definir um relacionamento deste tipo, o usuário indica os parâmetros fundamentais da classe-pai *Relationship* e quais os intervalos de origem e destino que este relaciona.

A classe *DerivedRelationship* representa uma especialização da classe *Relationship* e corresponde a um relacionamento temporal inferido a partir de regras de transição apresentadas no Apêndice C. Este tipo de relacionamento, por ser derivado, é baseado obrigatoriamente em outros dois relacionamentos do tipo *Relationship*, ou seja, se baseia tanto em outros relacionamentos derivados quanto em relacionamentos definidos pelo usuário. O usuário não instancia diretamente este tipo de objeto, já que este é gerado pela API durante a análise do leiaute. Sendo assim, o seu identificador é gerado dinamicamente e os relacionamentos-filho envolvidos na inferência são atribuídos automaticamente pela ferramenta. Esta derivação pode ser exemplificada pelo cenário onde dois relacionamentos do tipo *LayoutRelationship* definem que “Intervalo A {<} Intervalo B” e “Intervalo B {<} Intervalo C”, o que leva a ferramenta a gerar um novo relacionamento do tipo *DerivedRelationship* que define a relação “Intervalo A {<} Intervalo C”. E finalmente, a classe *Layout* agrega todos os intervalos e relacionamentos temporais do leiaute temporal.

A seguir é apresentada na Figura 4.8 uma estrutura de relacionamentos temporais e suas respectivas inferências. Esta estrutura de natureza hierárquica integra o leiaute temporal orientado a objetos e tem como base as classes do metamodelo tratado. Relacionamentos derivados estão posicionados em níveis superiores da hierarquia. Já os relacionamentos de leiaute envolvem exclusivamente intervalos e se encontram em posições inferiores.





Para promover a interoperabilidade com outras aplicações, foi desenvolvido um mecanismo para a geração de representações de leiautes temporais em formato XML (W3C, 2010), do inglês *eXtensible Markup Language*. Tanto a exportação quanto a importação do leiaute ocorrem segundo um esquema XSD (W3C, 2010b), do inglês *XML Schema Definition*, apresentado no Apêndice E. Cada objeto instanciado a partir do metamodelo proposto apresenta um método de exportação em XML que resume as suas informações e as de seus elementos-filho. Na Tabela 4.2, é apresentado um documento referente a um leiaute de entrada obedecendo ao esquema XSD proposto.

O mecanismo de validação é responsável por derivar novos relacionamentos a partir dos relacionamentos de leiaute especificados pelo usuário e por verificar possíveis inconsistências envolvendo estes relacionamentos. A identificação de inconsistências ocorre através do mapeamento de: (i) relacionamentos que resultam em uma restrição envolvendo um intervalo a ele mesmo, e (ii) relacionamentos com restrições distintas envolvendo os mesmos intervalos de origem e destino. O resultado da verificação do exemplo apresentado na Tabela 4.2 é descrito em formato de depuração no Apêndice F. O formato de depuração por sua vez, para que possa ser utilizado por outras e pela própria aplicação, também obedece a um esquema XSD, este apresentado no Apêndice G.

```

<layout id="60">

    <intervalos>
        <intervalo id="A" />
        <intervalo id="B" />
        <intervalo id="C" />
        <intervalo id="D" />
    </intervalos>

    <relacionamentos>
        <relacionamento id="relAparaB" origem="A" destino="B">
            <tipo>CONTEM</tipo>
        </relacionamento>
        <relacionamento id="relBparaC" origem="B" destino="C">
            <tipo>ANTECEDE</tipo>
        </relacionamento>
        <relacionamento id="relDparaC" origem="D" destino="C">
            <tipo>ANTECEDE</tipo>
        </relacionamento>
        <relacionamento id="relDparaB" origem="D" destino="B">
            <tipo>SUCEDE</tipo>
        </relacionamento>
        <relacionamento id="relCparaA" origem="A" destino="B">
            <tipo>FINALIZA_COM</tipo>
        </relacionamento>
    </relacionamentos>
</layout>

```

**Tabela 4.2 – Exemplo de leiaute de entrada**

## 4.5 Comentários Finais

Neste capítulo, foram apresentadas as contribuições da abordagem TimeFEX aplicadas a um ambiente de apoio a reutilização. A ferramenta de modelagem apresentada não se trata de uma forma única ou mandatória para a aplicação da abordagem. A utilização da plataforma Odyssey como base foi condicionada pela predisposição do seu ambiente em fornecer abstrações adequadas a modelagem de características e pela sua estruturação em uma arquitetura extensível.

Em relação à API para tratamento de leiautes temporais, somente o seu potencial de validação foi explorado através da sua utilização pelo ambiente de reutilização, mais especificamente, no suporte à Composição de Comportamento oferecido pelo guia de instanciação. No entanto, a exportação e importação de leiautes em um formato inteligível facilitaram o desenvolvimento e os testes realizados sobre a ferramenta de modelagem. E finalmente, as funcionalidades de representação orientada a objetos e validação de leiautes temporais foram concebidas sob a forma de uma biblioteca, cujo uso pelo ambiente de reutilização tratado por este capítulo constitui uma prova de conceito.

# Capítulo 5

## Experimentação

### 5.1 Introdução

Nos capítulos anteriores foram apresentados os elementos teóricos e as ferramentas de apoio implementadas para a aplicação da abordagem, com a finalidade de utilizar intervalos de tempo em conjunto com a modelagem orientada a características para aprimorar a representação de comportamentos de linhas de produtos de software. Para a verificação do aprimoramento pretendido, torna-se necessário o levantamento de indícios e evidências envolvendo os elementos da abordagem proposta.

Diversas estratégias buscam melhorar a confiabilidade do software através da adoção de “melhores” práticas e procedimentos para o seu desenvolvimento (KITCHENHAM *et al.*, 2004), porém poucas são as evidências objetivas que confirmam aspectos como adequabilidade, limites, riscos, qualidades e custos (DYBA *et al.*, 2005). O levantamento de evidências de forma experimental na engenharia de software é tratado desde a década de 80, sendo verificado em um trabalho seminal apresentado em (BASILI *et al.*, 1986) que derivou abordagens de experimentação nas quais esta parte do estudo é baseada.

A Engenharia de Software Baseada em Evidências (ESBE) por sua vez busca prover mecanismos para que as melhores evidências de pesquisa sejam integradas com a experiência prática e com os valores humanos envolvidos no processo de decisão no ambiente de desenvolvimento e manutenção de software (KITCHENHAM *et al.*, 2004). Sob esta perspectiva, este experimento tem como objetivo fornecer indícios sobre a utilização da abstração de intervalos de tempo na engenharia de linha de produtos de software no contexto de utilização da abordagem proposta.

Na Seção 5.2, é apresentado o planejamento do experimento composto por uma descrição formal da sua natureza e do seu objetivo, além da apresentação dos indicadores observados durante o estudo. Na Seção 5.3, é descrita a execução do experimento que foi dividida em duas etapas nas quais os participantes realizaram um conjunto de atividades solicitadas. Na Seção 5.4, é apresentada a análise de dados quantitativos e qualitativos obtidos durante o estudo de observação, sendo esta apresentação disposta de acordo com as etapas de execução. Nesta análise são

formuladas constatações a partir dos artefatos gerados pelos participantes e de relatos colhidos durante a execução. E finalmente, nas seções 5.5 e 5.6, são apresentados respectivamente, fatores de ameaça à validade do experimento e comentários gerais do estudo.

## 5.2 Planejamento

O experimento realizado é baseado em um estudo de observação. Neste tipo de estudo, o experimentador observa participantes enquanto estes realizam atividades específicas com o objetivo de coletar informações sobre como estas atividades são desempenhadas (SHULL *et al.*, 2001). Os principais indicadores observados neste experimento são: a influência da abordagem sobre a compreensão de domínios de aplicação em sua dimensão temporal e a prevenção de inconsistências temporais no modelo de domínio dada pela identificação de requisitos incompletos e de conflito entre requisitos de software, mais especificamente entre requisitos que envolvem temporalidade. Estes indicadores estão em conformidade com o objetivo do estudo que pode ser resumido em: avaliar a utilização de intervalos de tempo na modelagem de características através da abordagem TimeFEX. Este objetivo, em conjunto com estes indicadores, pode ser definido segundo os parâmetros para estudos experimentais em ES fornecidos em (WOHLIN *et al.*, 2000) e apresentados na Tabela 5.1.

<b>Analisar</b>	a utilização de intervalos de tempo na modelagem de características
<b>Com a finalidade de</b>	caracterizar o seu impacto na análise de variabilidade
<b>Com respeito a</b>	identificação de imprecisão e inconsistências temporais
<b>Do ponto de vista</b>	do engenheiro de domínio
<b>No contexto</b>	de utilização da abordagem TimeFEX

**Tabela 5.1 – Definição do estudo experimental**

Neste estudo de observação, os participantes são expostos a necessidades de alterações sobre dois modelos de domínio pré-existentes baseados em cenários fictícios e pertencentes a domínios de aplicação distintos. Estes modelos tratam basicamente de linhas de produtos de software que devem ser modificadas pelo participante para que esta contemple novas funcionalidades pretendidas para a linha. Estas alterações são apresentadas no experimento como solicitações realizadas por uma empresa fictícia que é proprietária da linha e na qual o participante trabalha como engenheiro de domínio. O

impacto da abordagem sobre a análise de variabilidade, que compõe o objetivo do experimento, é observado a partir de dois indicadores: facilidade de compreensão do domínio em sua dimensão temporal e consistência do modelo de domínio após a realização de adequações pelo participante.

A facilidade de compreensão dos requisitos e das características dos cenários é avaliada mediante a opinião dos participantes sobre dificuldades enfrentadas durante o entendimento do modelo de domínio e sobre a presteza da utilização de um modelo complementar, mais especificamente o Modelo Comportamental, no processo de assimilação. Já em nível de consistência do modelo de domínio, em especial a identificação de requisitos incompletos e de conflitos entre requisitos, pretende-se avaliar a redução da incidência destes fatores nos artefatos gerados por cada participante. Esta verificação é realizada de forma comparativa entre o primeiro cenário, onde os participantes não utilizam um método específico para o tratamento da temporalidade, e um segundo cenário onde o processo e os artefatos propostos pela abordagem TimeFEX são utilizados. Como resultado final do estudo, espera-se verificar de que forma os modelos de características e comportamental desenvolvidos prevêm produtos com comportamentos inconsistentes ou se estes restringem corretamente a linha à instanciação de produtos de comportamento consistente.

### **5.3 Execução**

O estudo foi realizado com seis participantes que inicialmente assinaram uma declaração de consentimento, apresentada no Apêndice H, através da qual concordam com os termos para participar do presente experimento. Além deste documento, os participantes preencheram um formulário de caracterização, apresentado no Apêndice I, que buscou descrever cada participante segundo a sua formação acadêmica, o seu histórico profissional e o seu nível de conhecimento sobre determinados domínios de aplicação. Os domínios questionados neste formulário são caracterizados por agregar complexidade temporal e, entre estes, estão os dois domínios tratados pelos cenários utilizados como exemplo durante a execução do experimento: uma linha de produtos de software para sistemas de compra coletiva na web, e uma linha de sistemas para realização de reservas online para redes hoteleiras.

### 5.3.1 Primeira Etapa

A execução da primeira etapa do experimento consistiu na realização de um treinamento sobre Engenharia de Linhas de Produtos de Software (ELPS) e sobre a modelagem de características com a notação Odyssey-FEX. Após este treinamento, foram apresentados aos participantes artefatos que integram um suposto modelo de domínio baseado em um dos cenários apresentados no Apêndice D. Ou seja, este trata de compras coletivas ou de reservas online. A apresentação destes domínios se deu de forma alternada entre os participantes para que suas especificidades ou uma possível familiaridade do participante não influenciem no resultado do experimento.

O conjunto de artefatos que compõem o modelo de domínio consiste em: (i) um modelo de características, desenvolvido a partir da notação Odyssey-FEX, (ii) um documento contendo uma descrição textual para cada característica do modelo, (iii) uma descrição do comportamento de um exemplo de produto da linha através de um esquema de barras, e (iv) uma lista de requisitos de software envolvendo as características apresentadas. Dentre os requisitos desta lista, foram dispersos requisitos que envolvem aspectos comportamentais e requisitos específicos do negócio que condicionam as configurações dos produtos da linha, mas que não abordam aspectos temporais.

O modelo de domínio apresentado é inicialmente consistente, atende aos requisitos da lista apresentada e não possui elementos específicos da abordagem, com exceção da descrição de comportamento através de intervalos de tempo. Sendo assim, após a leitura dos artefatos, cada participante foi indagado sobre possíveis dúvidas e dificuldades com o intuito de observar o processo de compreensão deste material, mais especificamente dos requisitos que envolvem temporalidade sem um artefato que auxiliasse na sua visualização.

Apesar de não haver conflito entre requisitos na lista inicialmente apresentada, esta agrega requisitos incompletos. Estes requisitos tratam do posicionamento de eventos no tempo utilizando-se, como referencial, características que são opcionais no modelo. Ou seja, indicam que um evento A, por exemplo, ocorre após uma característica B. Porém, a característica B, por ser opcional, pode não estar presente em produtos da linha, trazendo imprecisão sobre quando o evento A deve ocorrer nestes

produtos. Neste sentido, a ausência de características opcionais, utilizadas como referências temporais nos requisitos, incorre na imprecisão sobre o comportamento de determinados produtos.

Outro exemplo desta ocorrência pode ser observado em um requisito utilizado no exemplo que trata de sistemas de compras coletivas (Cenário I). Este requisito determina que uma característica intitulada “Consolidador de Vendas” deve ser iniciada imediatamente após o término da execução da característica de “Monitoramento de Consumo”. No entanto, esta inicialização se torna imprecisa no tempo em produtos onde o “Monitoramento de Consumo” não está presente. Sendo assim, a observância, por parte dos participantes, sobre a falta de exatidão na disposição temporal dada por características opcionais foi inserida nesta etapa de execução como um indicador observado pelo instrutor do experimento.

A seguir, foi dado prosseguimento ao cenário hipotético onde o participante recebe uma solicitação de adequação da linha que é expressa pelo modelo de domínio em mãos. Esta solicitação se baseia em uma nova lista de requisitos que devem ser contemplados pela linha após as adequações, porém preservando o atendimento aos requisitos da lista inicialmente apresentada. Após o término da descrição da evolução pretendida, foi solicitado ao participante que este realizasse as adequações que achasse necessárias sobre o modelo de características apresentado. Neste momento, foi observado se o participante relatou de forma espontânea a necessidade de intervir em outros artefatos do modelo de domínio inicial, além do modelo de características.

Assim como a lista inicial de requisitos, apenas parte dos requisitos da nova lista contempla aspectos comportamentais. Além disto, esta agrega dois problemas: requisitos incompletos, assim como na lista inicial, e é acrescida de requisitos que, em conjunto com os requisitos iniciais, agregam conflitos na especificação. Estes conflitos são evidenciados por inconsistências temporais presentes em produtos da linha que possuem simultaneamente determinadas características envolvidas em requisitos conflitantes.

Uma vez finalizadas as modificações livres, o participante foi indagado, através de um questionário, apresentado no Apêndice J, sobre questões sobre a experiência e as dificuldades enfrentadas tanto no entendimento do domínio quanto na realização das adequações.

### **5.3.2 Segunda Etapa**

Na segunda etapa do experimento, foi realizado um treinamento sobre a abordagem TimeFEX, no qual foram abordadas as atividades de Engenharia de Domínio do processo prescritivo e a notação utilizada para o desenvolvimento do Modelo Comportamental.

Após este treinamento, foram apresentados a cada participante, artefatos de um segundo modelo de domínio, baseado em um cenário e domínio diferentes dos utilizados na primeira etapa. Este segundo cenário também se encontra descrito no Apêndice D. Este modelo de domínio consiste no mesmo conjunto de artefatos apresentados no primeiro cenário, acrescido de um Modelo Comportamental, que por sua vez expressa os aspectos temporais do modelo de domínio inicial.

Este segundo modelo de domínio também é inicialmente consistente e, devido à inserção do Modelo Comportamental, passa a agregar elementos específicos da abordagem, tais como relacionamentos temporais e características intervalares. Após a leitura de todos os artefatos, o participante foi indagado sobre possíveis dúvidas para averiguar como o segundo modelo de domínio, ao ser acrescido de elementos específicos da abordagem, teve o seu entedimento comprometido ou aprimorado em comparação com a primeira etapa, esta realizada sem a utilização da abordagem.

A seguir, foi descrita uma solicitação de adequação para o novo cenário e a sua respectiva lista de novos requisitos a serem contemplados. Após o término da descrição da solicitação, os participantes, utilizando-se dos mecanismos da abordagem, realizaram adequações que julgaram necessárias sobre o modelo de característica e sobre o modelo comportamental. Uma vez finalizadas estas modificações livres, os participantes foram indagados, através de um segundo questionário sobre a experiência e dificuldades enfrentadas com a utilização da abordagem. Este questionário é apresentado no Apêndice K.

## **5.4 Análise dos Resultados**

A primeira análise dos dados cuja aquisição antecede a execução da primeira etapa, trata da caracterização dos participantes. Por motivos de confidencialidade estes são identificados a partir de letras. O resultado da caracterização dos participantes é



apresentado na Tabela 5.2, e reflete o objetivo de diversificar a população do estudo em nível de formação acadêmica, experiência em desenvolvimento de software e conhecimento sobre domínios de aplicação com complexidade temporal. O grupo de participantes foi selecionado por conveniência não havendo qualquer tipo de compensação neste processo.

	<b>Participantes</b>					
	<b>(a)</b>	<b>(b)</b>	<b>(c)</b>	<b>(d)</b>	<b>(e)</b>	<b>(f)</b>
Formação Acadêmica	Sup. Inc.	MSc. Inc.	Sup. Comp.	DSc. Inc.	MSc. Inc.	MSc. Inc.
Nível de experiência na análise e desenvolvimento de sistemas de software (0-3)	1	2	3	3	1	1
Nível de experiência em Engenharia de Linhas de Produtos de Software (0-2)	1	1	1	1	0	1
Nível de experiência sobre Compras Coletivas	Médio	Alto	Alto	Baixo	Baixo	Nenhum
Nível de experiência sobre Mercado Financeiro	Baixo	Baixo	Baixo	Baixo	Nenhum	Médio
Nível de experiência sobre Hotelaria	Nenhum	Baixo	Médio	Alto	Baixo	Nenhum
Nível de experiência sobre Telecomunicações	Baixo	Baixo	Médio	Médio	Alto	Baixo
Nível de experiência sobre TV Digital	Médio	Baixo	Médio	Médio	Alto	Baixo

**Tabela 5.2 – Resultado da caracterização dos participantes.**

O resultado da caracterização aponta vantagens de dois participantes em relação aos demais. O participante (d), por sua formação acadêmica atrelada à experiência na indústria e ao seu nível de conhecimento no setor de hotéis. E o participante (e) se destaca pelo seu alto nível de conhecimento nas áreas de telecomunicações e TV Digital, ambas fortemente caracterizadas pelo aspecto de temporalidade. Além disto, a área de telecomunicações lida intensivamente com problemas de interação entre características. E finalmente, a formação acadêmica do participante (e) é baseada em um mestrado em andamento e que se dá em domínios de aplicação que agregam complexidade temporal.

### 5.4.1 Primeira Etapa

A análise da primeira etapa trata da forma como os participantes utilizaram o modelo de domínio antes e durante as adequações sem o uso da abordagem. O modelo de domínio apresentado nesta etapa é composto pelos artefatos apresentados na Seção 5.3.1. O resultado desta etapa, sob o ponto de vista dos participantes, é apresentado na Tabela 5.3.

	<b>Sim</b>	<b>Não</b>	<b>Parcialmente</b>
<b>Os requisitos que envolvem temporalidade foram de fácil compreensão?</b>	4 (a,c,d,e)	0	2 (b,f)
<b>Os requisitos que envolvem temporalidade foram suficientes para descrever satisfatoriamente o comportamento de todos os produtos da linha?</b>	3 (a,c,e)	1 (f)	2 (b,d)
<b>Em algum momento você considerou mais de um comportamento que os produtos da linha podem adquirir?</b>	5 (a,b,c,e,f)	1 (d)	0
<b>Os produtos previstos pela linha apresentaram algum comportamento inconsistente?</b>	2 (f,e)	2 (a,c)	2 (b,d)

**Tabela 5.3 – Formulário de Avaliação da Primeira Etapa**

O principal aspecto observado na primeira etapa foi a forma como os participantes realizaram as adequações solicitadas sem a utilização de um método específico para o tratamento dos requisitos temporais. Neste processo, durante a análise de variabilidades, todos os requisitos que não envolvem temporalidade, mas que impactaram na variabilidade da linha foram tratados sem maiores dificuldades. Porém, durante o tratamento *ad hoc* dos requisitos de comportamento, emergiram alguns padrões de execução e constatações:

**A.1)** Durante o entendimento inicial do domínio apresentado, antes da realização de adequações, o participante (a) relatou que interpretou o domínio de forma errada, e que confundiu a linha de sistema de compras coletivas (Cenário I) com uma linha de sistema de vendas comuns pela web. No entanto, o mesmo indicou que a leitura dos requisitos que tratam dos aspectos temporais do negócio foi fundamental para a identificação do equívoco;

**A.2)** Também em relação ao entendimento geral do domínio, em diferentes momentos, o participante (b) indicou a descrição de características como sendo confusa, vaga, conflitante e incompleta e que os requisitos são vagos em relação ao que cada característica faz. O cenário apresentado ao participante (b) na primeira etapa foi o Cenário II;

**A.3)** Ao considerarem os requisitos temporais, os participantes (b),(c),(e) e (f) perguntaram sobre, ou realizaram de forma espontânea, mudanças no esquema de barras, apresentado como exemplo de comportamento e que se refere a um produto específico da linha. Esta verificação refletiu uma dificuldade em separar o entendimento do comportamento de um produto, sem variabilidade e resultante da Engenharia de Aplicação, e de uma família de comportamentos, esta com variabilidade e resultante da Engenharia de Domínio. Este contraste é reforçado pelos participantes (c) e (d) que não realizaram mudanças no esquema de barras e relataram espontaneamente o entendimento correto, de que o esquema não possui variabilidade e que é pertinente à análise de apenas um produto da linha. Neste processo, o participante (c), ao indicar que o exemplo apresentado deixaria de compor a linha após as adequações, referenciou qual novo requisito o produto, ao qual o comportamento de exemplo se refere, não atenderia;

**A.4)** O participante (e), integrante com maior conhecimento sobre domínios que envolvem temporalidade, identificou um suposto conflito temporal entre os requisitos 5, 7 e 14 do Cenário I. Porém, esta identificação partiu de uma interpretação dos requisitos divergente da disposição dos intervalos que integram o gabarito;

**A.5)** Os participantes (b) e (c), afirmaram incisivamente que os requisitos temporais não são pertinentes ao modelo de características, porém não identificaram que algumas configurações dadas por este modelo, em conjunto com estes requisitos, prevêm produtos com comportamentos inconsistentes;

**A.6)** No Cenário I, utilizado pelos participantes (a), (c) e (e), nenhum destes identificou a inconsistência temporal existente no cenário e afirmaram, ativamente,

com exceção do participante (e), que este cenário não possuía inconsistências. A inconsistência correta é determinada pelos requisitos 3, 11 e 13;

**A.7)** No Cenário II, utilizado pelos participantes (b), (d) e (f), apenas o participante (f) conseguiu identificar a inconsistência temporal do cenário. O restante dos participantes não identificou inconsistências, porém relatou que não tinha formas de garantir que o cenário não possuía inconsistências. A inconsistência correta deste cenário é determinada pelos requisitos 6, 9 e 11;

**A.8)** O único participante que identificou que alguns requisitos são incompletos, ou seja, que são baseados em características opcionais, foi o participante (e). Neste sentido, o participante deliberadamente inseriu novas regras de composição inclusivas no modelo de características condicionando a existência de características tratadas pelo requisito à existência da característica opcional utilizada como referência.

## 5.4.2 Segunda Etapa

A análise da segunda etapa trata da forma como os participantes utilizaram o modelo de domínio antes e durante as adequações com o uso da abordagem TimeFEX. O modelo de domínio apresentado nesta etapa é composto pelos artefatos apresentados na Seção 5.3.2. O resultado desta etapa, sob o ponto de vista dos participantes, é apresentado na Tabela 5.4.

	<b>Sim</b>	<b>Não</b>	<b>Parcialmente</b>
<b>Os diferentes tipos de relacionamentos temporais entre intervalos utilizados na técnica são de fácil entendimento?</b>	4 (a,b,c,e)	0	2 (d,f)
<b>A utilização do modelo comportamental impactou no trabalho realizado?</b>	5 (a,b,d,e,f)	0	1 (c)
<b>A transcrição de requisitos de software para relacionamentos temporais impactou na realização das tarefas?</b>	3 (a,d,f)	3 (b,c,e)	0
<b>A validação do modelo comportamental através do suporte computacional impactou na realização das tarefas?</b>	5 (a,c,d,e,f)	1 (b)	0

**Tabela 5.4 – Formulário de Avaliação - Segunda Etapa**

O principal aspecto observado na segunda etapa foi a forma como os participantes realizaram as adequações solicitadas, utilizando o processo e a notação para o tratamento dos requisitos temporais da abordagem TimeFEX. Nesta etapa, assim como na primeira, todos os requisitos que não envolvem temporalidade, mas que impactaram na variabilidade da linha foram tratados sem maiores dificuldades durante a análise de variabilidade. Durante o tratamento dos requisitos de comportamento com o uso da abordagem, emergiram novos padrões de execução e constatações:

**B.1)** Ao considerarem os requisitos temporais, nenhum dos participantes recorreu a mudanças no esquema de barras apresentado como exemplo de comportamento e que se refere a um único produto da linha. Esta verificação refletiu a separação adequada do comportamento de um produto, sem variabilidade e resultante da EA, e de uma família de comportamentos, com variabilidade e resultante da ED;

**B.2)** Os participantes (a) e (c) questionaram como características poderiam ser identificadas como opcionais através da visualização do modelo comportamental, pois esta propriedade não possui uma notação específica para o modelo comportamental na abordagem TimeFEX;

**B.3)** Em relação a identificação de relacionamentos do modelo comportamental, os participantes (a), (b), (c) e (e) não nomearam os relacionamentos temporais. Já os participantes (d) e (f) nomearam relacionamentos com o número dos requisitos que os geraram. Neste processo, o participante (d) restringiu a identificação aos novos relacionamentos gerados durante a adequação;

**B.4)** Os participantes (a), (c) e (d) realizaram apontamentos pertinentes sobre a necessidade ou não de se expressar determinadas características no modelo comportamental. Neste sentido, a ausência da representação se deu sobre características que não figuram como origem ou destino de nenhum relacionamento temporal, ou cujos relacionamentos não agregam conhecimento relevante ao modelo de domínio;

**B.5)** Todos os participantes relataram impactos positivos na utilização do modelo comportamental (Tabela 5.4, questão 2). Dentre estes impactos foram relatados:

facilidade na compreensão do problema, do cenário e dos requisitos, relatada pelos participantes (a) e (b); a possibilidade de utilizar os requisitos temporais na análise de variabilidade, relatada pelos participantes (c) e (d), que na primeira etapa desconsideraram os mesmos; a possibilidade de identificar inconsistências, relatada pelo participante (e); e a utilização do modelo comportamental em detrimento do uso de exemplos de comportamento específicos a um único produto, relatada pelo participante (f);

**B.6)** Os participantes (b), (c) e (e) verificaram a imprecisão sobre o comportamento dada por requisitos temporais incompletos, ou seja, que determinam relacionamentos temporais dependentes de características opcionais;

**B.7)** Em relação ao suporte computacional, todos os participantes avaliaram positivamente a sua utilização (Tabela 5.4). Dentre os relatos, destacam-se: a independência do desenvolvimento do modelo de domínio em relação a experiência do engenheiro de software, redução do esforço, e a possibilidade de identificar inconsistências em cadeias maiores e mais complexas de relacionamentos que poderiam não ser identificadas manualmente;

**B.8)** Os participantes (a) e (b) relataram que o segundo domínio apresentado possui menor complexidade, ou que era mais bem escrito. Os cenários relatados por estes participantes tratam respectivamente dos Cenários II e I;

**B.9)** Os participantes (a), (c), (d) e (f) experimentaram ou relataram dificuldades na utilização dos elementos específicos da abordagem. Entre estes elementos destacam-se: orientação correta de relacionamentos, utilização das restrições temporais e seus inversos; e a utilização de regras de composição entre relacionamentos temporais.

### **5.4.3 Considerações sobre os resultados**

A partir das constatações apresentadas, é possível constituir indícios sobre aspectos positivos e negativos da abordagem proposta. Estes indícios são baseados essencialmente na conjugação de constatações da primeira e da segunda etapa, ou seja, inicialmente sem e após com o uso da técnica proposta. Ao realizar estas conjugações

foi tomado o cuidado de se utilizar constatações baseadas em ambos os cenários que foram apresentados alternadamente entre um participante e outro, a fim de prevenir o viés dado pelos modelos de domínio utilizados como exemplo.

Nas primeiras observações acerca da compreensão do modelo, na constatação A.2, o participante (b) relatou incisivamente acerca da falta de clareza do modelo de domínio, mais especificamente o modelo determinado pelo Cenário II. No entanto, este apontamento foi considerado como *outlier* dada a falta de sua confirmação no restante dos relatos de utilização deste cenário acrescido do fato do participante (d), com maior nível de experiência de desenvolvimento e do domínio deste cenário, ter relatado a descrição destas mesmas características como sendo sucinta e bem escrita. Sendo assim, resume-se aos relatos apresentados na constatação B.8, que apresentam indícios sobre a facilitação na compreensão do domínio durante a segunda etapa, em ambos os cenários, dada pelo uso da abordagem.

De acordo com a percepção da maioria dos participantes, os requisitos utilizados na primeira etapa foram inicialmente de fácil compreensão (Tabela 5.3), no entanto a sua análise de forma *ad hoc* não contribuiu para a identificação de conflitos de forma intuitiva, conforme apresentado nas constatações A.6 e A.7. De acordo com a resposta fornecida pelo participante (b), um esquema gráfico seria necessário para que os requisitos fossem compreendidos e entendidos adequadamente. Além disto, o participante (f) apontou de forma discursiva que, na sua percepção, requisitos desta natureza introduziriam mais fatos incorretos.

Outro aspecto sobre o entendimento envolve a diferenciação da representação do comportamento específico de um produto, e da representação do comportamento variável referente a toda a linha de produtos. A utilização da técnica propiciou aos participantes, compreenderem e diferenciarem a representação do comportamento específico de um produto e de um artefato adequado a representação de um comportamento com variabilidade para toda a linha. Indícios que baseiam esta afirmação podem ser observados a partir das constatações descritas nos itens A.3 e B.1, onde quatro dos participantes passaram a modelar a variabilidade de comportamento no artefato adequado (Modelo Comportamental) prescindindo do uso do gráfico de barras.

Através da conjugação das constatações A.8 e B.6, foi possível observar que o uso da abordagem trouxe maior observância acerca da imprecisão do comportamento dada pela identificação de requisitos incompletos. Na segunda constatação, o único participante a identificar imprecisão na primeira etapa (participante “e”), após o uso da

abordagem, deixou de inserir deliberadamente novas regras de composição inclusivas na tentativa de resolver a imprecisão. Em lugar a esta solução, este participante, assim como os demais, durante a segunda etapa, realizou questionamentos pertinentes para que os requisitos fossem completados e trouxessem precisão ao comportamento expresso pelo modelo de domínio através de informações levantadas de análise, e não de suposições.

Em relação à validação do modelo comportamental, uma vez identificadas suas inconsistências, ao contrário do que era esperado durante o planejamento, os participantes não realizaram alterações no modelo de características deliberadamente e por conta própria. O participante (e), por exemplo, introduziu uma nota auxiliar indicando o conflito observado no modelo. Já os participantes (d) e (f) optaram por observar a origem das inconsistências, mais especificamente os requisitos que as originaram, e renegociar os mesmos, a fim de obter uma especificação sem conflitos. Neste sentido, os participantes não criaram novas regras de composição entre características ou entre relacionamentos temporais. Ou seja, não inseriram informações estranhas e que não foram elicitadas no modelo de domínio.

No relato B.2, é possível constatar uma deficiência da notação TimeFEX sobre a visualização de opcionalidade de características sob a perspectiva de comportamento (Modelo Comportamental) que, segundo o questionamento realizado pelos participantes (a) e (e), foi necessária, mas que não é provida pela notação.

Em um maior detalhamento da constatação B.4, os participantes refletiram ativamente sobre a ausência e a necessidade de que algumas características figurassem no modelo comportamental. Individualmente, os participantes (a), (c) e (d) concluíram por conta própria e de forma pertinente sobre a necessidade de determinadas características figurarem ou não neste modelo, reforçando o caráter intuitivo da complementaridade entre as duas perspectivas, defendido ao longo do Capítulo 3.

Dificuldades na utilização de elementos específicos da abordagem foram observadas na constatação B.9. Sem relacionarem a este apontamento, os participantes (d) e (f) indicaram o uso de formas mais eficazes e parametrizadas de se montar relacionamentos temporais a partir de requisitos escritos através de linguagem natural, auxiliando, por exemplo, na determinação da orientação do relacionamento e na escolha de suas restrições.

E finalmente, a ausência de impacto na realização das tarefas, apontadas pelos participantes (b) e (c) na Tabela 5.4, derivou da avaliação destes de que a atividade de



adequação da linha se resume na intervenção sobre o modelo de características. Como este não foi modificado pelos participantes, estes entenderam que a atividade como um todo não foi impactada.

O indicativo fornecido pelo participante (e), de que os requisitos foram suficientes para a descrição do comportamento, foi realizado em conjunto com uma escolha própria de atrelar regras de composição inclusivas para cada relacionamento temporal direcionado para características intervalares.

Não há garantias de que os relacionamentos temporais e suas restrições transcritas sejam fiéis aos requisitos. Uma transcrição errônea acabou por resultar em um Modelo Comportamental futuramente validado em relação a sua própria estrutura, mas que não foi verificado em relação aos requisitos. Neste sentido, os participantes (d) e (f) sugeriram a utilização de ferramental para a escrita do relacionamento a partir do requisito, utilizando-se termos de linguagem natural. O participante (f) sugeriu que o termo obtido em linguagem natural seja expresso no modelo comportamental em detrimento das restrições de Allen (Tabela 3.2).

## **5.5 Ameaças à Validade**

Alguns fatores do planejamento e da execução do experimento constituem questões que podem impactar ou limitar a validade dos resultados deste experimento, fatores estes comumente chamados de ameaças a validade (WOHLIN *et al.*, 2000). Apesar de estas ameaças serem evitadas ao longo do planejamento e execução do estudo, a sua presença é inerente ao processo de experimentação. Neste sentido, os seguintes elementos representam ameaças a validade do estudo apresentado:

- Os cenários descritos nos modelo de domínio apresentados aos participantes tratam de situações hipotéticas. Este fator possibilitou que situações mais simplificadas e que propiciassem o tratamento da temporalidade fossem tratadas. Sendo assim, os resultados apresentados são restritos às proporções dos cenários apresentados;
- A atividade executada pelos participantes se restringiu à adequação de um modelo de domínio pré-existente. Sendo assim, a situação onde um novo modelo de domínio é concebido com o uso da abordagem não foi abordada pelo experimento. Esta ameaça destaca uma ênfase do estudo à análise de

variabilidade, porém com restrições acerca da engenharia de domínio como um todo;

- Apenas atividades da fase de Engenharia de Domínio foram tratadas, ou seja, o processo de instanciação de um novo produto (Engenharia de Aplicação) com o uso da abordagem não foi tratado. Apesar de esta etapa estar fora do escopo do objetivo, o apontamento é válido para o planejamento de experimentos posteriores mais completos e abrangentes sobre a técnica;
- Apesar de o experimento verificar um aprimoramento dado pelo uso da abordagem em nível de facilidade de compreensão, redução de inconsistências e identificação de requisitos incompletos, o mesmo não levanta indícios de forma comparativa em relação a abordagens existentes;
- A atividade de identificação de características não foi realizada pelos participantes. Neste sentido, são necessários indícios para avaliar o impacto de diferentes resultados desta atividade sobre o uso da abordagem;
- Apesar de a abordagem prever a possibilidade de utilizar diferentes notações compatíveis com a notação FODA (KANG *et al.*, 1990) para a modelagem de características, o estudo se restringiu ao uso da notação OdysseyFEX (OLIVEIRA, 2006);
- Ao analisar o contexto de utilização da abordagem TimeFEX, foi escolhida a utilização da notação OdysseyFEX para a modelagem de características. Apesar de todos os elementos da notação terem sido abordados no treinamento inicial, apenas parte do vocabulário desta notação foi utilizada nos cenários.

## 5.6 Comentários Finais

O planejamento, execução e análise dos resultados do estudo de observação agregaram informações importantes sobre benefícios, potencialidades, limitações e pontos de melhoria para a técnica.

Entre os benefícios encontrados, destacou-se uma melhor compreensão do problema e do domínio através do uso das ferramentas de representação e análise propostas pela abordagem TimeFEX. O processo de análise proposto pela técnica possibilitou representar, de forma complementar, a variabilidade de comportamento entre os produtos da linha, fornecendo um artefato adequado para esta representação e inserindo a dimensão temporal entre os aspectos pela análise de variabilidade.

Além disto, foram observados resultados convergentes sobre a forma como a abordagem contribuiu para a identificação de inconsistências em relacionamentos temporais, estas derivadas de requisitos incompletos ou de conflitos entre requisitos. Neste sentido, o suporte computacional proposto auxiliou na identificação destes elementos, além de ter sido utilizado para confirmar ocorrências encontradas manualmente.

Entre os principais problemas encontrados no uso da abordagem está a ausência de representação de opcionalidade de características de software no modelo comportamental. Além desta carência, foram constatadas dificuldades no desenvolvimento do modelo comportamental devido à falta de suporte para a transcrição de requisitos de software, representados textualmente, para elementos específicos da abordagem, mais especificamente relacionamentos temporais e restrições entre intervalos.

Estes problemas possibilitaram a melhor compreensão de pontos de melhoria. Entre estes se encontra a capacidade de representação de opcionalidade, não somente entre características de software, mas a possibilidade de agregar opcionalidade também a relacionamentos temporais, mantendo o caráter adotado pela técnica de tratar relacionamentos temporais como entidades de primeira ordem.

Outro ponto de melhoria observado foi o desenvolvimento de ferramentas com as quais o engenheiro de domínio possa, através do uso de elementos parametrizados e representados através de linhagem natural, derivar relacionamentos temporais, mas especificamente conjunto de restrições do Modelo de Allen, de forma facilitada e intuitiva.

E finalmente, os problemas e os pontos de melhoria identificados foram utilizados, respectivamente, como insumo para a descrição de limitações da abordagem e oportunidades para trabalhos futuros, elementos estes descritos a seguir no Capítulo 6.

# Capítulo 6

## Conclusão

### 6.1 Epílogo

A Engenharia de Linhas de Produtos de Software (ELPS) promove o reuso em larga escala com a finalidade de atingir e maximizar os benefícios esperados pela reutilização de software. Entre estes benefícios estão: a amortização de custos de desenvolvimento e testes, facilitação na definição de padrões e normas, redução do *time-to-market* e a promoção da interoperabilidade entre sistemas.

Este processo de engenharia é auxiliado pela modelagem de características. Com esta modelagem é possível analisar a variabilidade dos produtos de uma linha a fim de identificar as diferentes configurações obtidas a partir de conjugações de características de software, levando-se em consideração a semântica existente entre estes elementos. Nesta análise de variabilidade, é observado como os produtos se diferenciam e quais os pontos de convergência e divergência em nível de características. Além disto, a modelagem orientada a características proporciona uma forma facilitada de análise dado o seu alto nível de abstração, promovendo com isso o entendimento entre os diversos envolvidos neste processo (*stakeholders*).

Diversos domínios de aplicação são caracterizados por sua complexidade temporal. Assim, os sistemas que integram estes domínios são passíveis de serem obtidos através da ELPS agregando, portanto, a temporalidade e sua semântica observada entre as características durante a análise de variabilidade. Como consequência, persiste uma complexidade complementar dada pela interação entre estas características.

Neste contexto, foi apresentada a abordagem TimeFEX, que tem como objetivo utilizar a abstração de intervalos de tempo para auxiliar na análise de variabilidade e consistência de LPS em sua dimensão temporal. Esta abordagem foi descrita através de um processo prescritivo para a sua aplicação, uma notação para o desenvolvimento de Modelos Comportamentais, um ambiente de reutilização de apoio à criação e utilização destes modelos e uma ferramenta de validação dos leitantes temporais.

E finalmente, elementos da solução proposta foram avaliados em um experimento concebido e executado sob a forma de um estudo de observação, que por sua vez derivou potencialidades, problemas e limitações inerentes ao desenvolvimento de uma nova técnica.

Neste último capítulo, são descritas na Seção 6.2 as principais contribuições deste trabalho, através das quais é possível observar e derivar tanto limitações da técnica quanto oportunidades para trabalhos futuros, elementos estes também apresentados no capítulo, respectivamente, nas Seções 6.3 e 6.4.

## **6.2 Contribuições**

O trabalho apresentado nesta dissertação teve como principal objetivo o desenvolvimento dos elementos de uma proposta para o tratamento da dimensão temporal pela Engenharia de Linhas de Produtos de Software (ELPS). Entre as contribuições observadas neste trabalho, estão:

- Concepção de atividades de um processo prescritivo para o tratamento do comportamento de LPS desde a Engenharia de Domínio à Engenharia de Aplicação. Além das atividades do processo em si, a contribuição envolve aspectos genéricos e inerentes ao tratamento da temporalidade no contexto da ELPS, entre estes, a complementaridade no tratamento da estrutura e do comportamento, a divisão destas perspectivas em visões e artefatos distintos e correspondentes, e o impacto da variabilidade do comportamento sobre a consistência de produtos de uma linha;
- Concepção da notação TimeFEX utilizada especificamente para agregar semântica temporal entre características de software. Esta notação, formalizada através de um metamodelo, é apoiada por um ambiente de reutilização para a construção e validação de modelos desenvolvidos a partir da mesma; e
- Realização de um estudo de observação sobre o tratamento da variabilidade de comportamentos em LPS. Este estudo foi realizado no escopo da abordagem proposta e com uso da orientação a intervalos de tempo. Os resultados obtidos resultaram na caracterização do uso e dos impactos da abordagem TimeFEX sobre a compreensão de domínio, a identificação de conflitos entre requisitos, e sobre a identificação de inconsistências temporais presentes em modelos de domínio.

As principais contribuições observadas, e listadas acima, derivaram outras contribuições e subprodutos de trabalho. Estes listados a seguir:

- Análise dos requisitos necessários para a representação da dimensão temporal em LPS orientada a características. Esta análise foi realizada através do estudo de outras abordagens pesquisadas da literatura. Os requisitos levantados envolvem principalmente a expressividade de um modelo específico para a análise de variabilidade de comportamentos que é adotado por cada abordagem. Neste sentido, foram identificados diferentes níveis de independência entre as variabilidades de comportamento e de estrutura, ou seja, de acoplamento entre configurações de comportamentos e configurações de características; e
- Desenvolvimento de uma API para a representação e validação de leiautes temporais, definidos a partir de intervalos e restrições de Allen (ALLEN, 1983), especificamente para ambientes de desenvolvimento orientados a objetos.

### **6.3 Limitações**

O desenvolvimento dos elementos constituintes da abordagem e, principalmente, a análise dos resultados do estudo de observação, possibilitaram a identificação de limitações, estas listadas a seguir:

- A abordagem TimeFEX não possui mecanismos de verificação de relacionamentos temporais (e seus conjuntos de restrições) em relação a outros elementos do modelo de domínio. Ou seja, modelos comportamentais distintos podem ser obtidos a partir de um mesmo modelo de domínio. Esta divergência é baseada nas diferentes interpretações que os Engenheiros de Domínio podem ter sobre um mesmo elemento, ou conjunto de elementos, do modelo de domínio. O reflexo desta limitação pôde ser observado durante o experimento, quando diferentes modelos comportamentais foram desenvolvidos a partir dos mesmos requisitos textuais do modelo de domínio;
- A abordagem TimeFEX é centrada na orientação a intervalos de tempo. Ou seja, os comportamentos expressos na variabilidade do Modelo Comportamental devem

ser passíveis de serem representados e entendidos a partir de intervalos de tempo, o que restringe a sua utilização em determinados domínios de aplicação. Outros modelos mais genéricos e de maior cobertura para representação de tempo, como por exemplo a lógica modal, não são cobertos pela abordagem;

- A abordagem TimeFEX prevê a utilização de Modelos de Características obtidos a partir de notações que agreguem os elementos básicos da notação FODA. Neste sentido, elementos semânticos de maior expressividade, tais como, multiplicidade, generalização, agregação e composição, não são explorados pela técnica;
- O estudo de observação apresentado levou em consideração a aplicação da abordagem proposta, prescindindo de comparação experimental com abordagens existentes. Além disto, o estudo e suas constatações se limitaram ao cenário de utilização da notação Odyssey-FEX para o desenvolvimento do modelo de características;
- As ferramentas de apoio ao desenvolvimento e validação de Modelos Comportamentais não foram utilizadas pelos participantes. Sendo assim, não constituem objeto de avaliação do estudo de observação e não tiveram a sua presteza confirmada quando utilizada pelo engenheiro de domínio;

## **6.4 Trabalhos Futuros**

Em diferentes momentos do desenvolvimento e da avaliação da abordagem, foram identificadas oportunidades para trabalhos futuros fundamentados em pontos de melhoria e limitações. A seguir são listadas as principais oportunidades observadas:

- Desenvolvimento de mecanismos para a verificação de conformidade entre elementos do modelo de domínio e os relacionamentos temporais. Mais especificamente em relação à orientação e ao conjunto de restrições destes relacionamentos. Este indicativo foi obtido durante a avaliação da abordagem, quando foi descrita a necessidade de ferramental para a construção de novos

relacionamentos a partir de termos parametrizados e representados em linguagem natural;

- O mapeamento entre o modelo de características e o modelo comportamental é realizado sobre a semântica primordial existente na notação FODA (KANG *et al.*, 1990). Neste sentido, se faz possível a extensão deste mapeamento a outros elementos cuja semântica se faz presente em trabalhos derivados desta notação seminal, tais como: generalização/especialização entre características, composição, agregação, etc. Porém, vale lembrar que esta extensão tende a ser inversamente proporcional à generalidade da técnica, em relação a quais notações de modelagem de características podem ser utilizadas em conjunto com a mesma;
- A utilização de intervalos de tempo através do modelo proposto em (ALLEN, 1983), e que se trata de um modelo baseado em restrições, foi motivada pela intuitividade inerente ao conceito de intervalo de tempo. Porém, a utilização de um modelo baseado em eventos, tal como o modelo NCM (CASANOVA *et al.*, 1991), não foi abordada neste trabalho. Neste sentido, a aplicação da técnica com o modelo comportamental e metamodelo adaptados a este outro modelo, representa um caminho de alternativo ao que foi proposto neste trabalho. Este caminho de solução conta, inclusive, com a possibilidade de utilização da linguagem NCL (NCL, 2009) como ferramenta para geração de código durante a Engenharia de Aplicação; e
- Aplicação da abordagem TimeFEX em projetos reais e de maior porte e a utilização do ferramental de apoio à modelagem e validação por engenheiros de domínio. Com isto, será possível identificar melhorias e evoluções tanto para o processo proposto quanto para a notação proposta e o seu ferramental de apoio.



# Referências Bibliográficas

- ABOWD, G., BASS, L., CLEMENTS, P.C., *et al.*, 1997, *Recommended Best Industrial Practice for Software Architecture Evaluation*, Software Engineering Institute (SEI), CMU/SEI-96-TR-025.
- ABRAHAM, T., RODDICK, J.F., 1999, "Survey of Spatio-Temporal Databases", *Geoinformatica*, v. 3, n. 1, pp. 61-99.
- ALFERT, K., 2005, "Requirements, Features and Aspects for Software Product Lines". In: *6th Workshop on Early Aspects*, Rennes, France.
- ALFÉREZ, M., MOREIRA, A., KULESZA, U., *et al.*, 2009, "Detecting feature interactions in SPL requirements analysis models", In: *Proceedings of the First International Workshop on Feature-Oriented Software Development*, pp. 117-123, Denver, Colorado, USA.
- ALLEN, J.F., 1983, "Maintaining Knowledge about Temporal Intervals", *Communications of the ACM*, v. 26, n. 11 (November), pp. 832-843.
- AMYOT, D., LOGRIPPO, L., BUHR, R.J.A., GRAY, T., 1999, "Use case maps for the capture and validation of distributed systems requirements", In: *Proceedings of the International Symposium on Requirements Engineering*, pp. 44-53, Limerick, Ireland.
- AMYOT, D., 2003, "Introduction to the User Requirements Notation: learning by example", *Computer Networks*, v. 42, n. 3, pp. 285-301.
- ANDERSON, T.L., 1982, "Modeling Time at the Conceptual Level", In: *Proceedings of the Second International Conference on Databases: Improving Database Usability and Responsiveness*, pp. 273-297, Israel.
- BASIL, V.R., SELBY, R.W., HUTCHENS, D.H., 1986, "Experimentation in software engineering", *IEEE Transactions on Software Engineering*, v. 12, n. 7 (July), pp. 733-743.
- BENTHEM, J.V., 1991, *The Logic of Time-A Model-Theoretic Investigation into the Varieties of Temporal Ontology and Temporal Discourse*, Kluwer Academic Publishers.

- BERTINO, E., FERRARI, E., 1998, "Temporal Synchronization Models for Multimedia Data", *IEEE Transactions on Knowledge and Data Engineering*, v. 10, n. 4 (July/August), pp. 612-531.
- BOLOGNESI, T., BRINKSMA, E., 1987, "Introduction to the ISO specification language LOTOS", *Computer Networks and ISDN Systems*, v. 14, n. 1, pp 25-59.
- BOLOUR, A., ANDERSON, T. L., DEKEYSER, L. J., *et al.*, 1982, "The role of time in information processing: A Survey", *SIGMOD Rec.*, v. 12, n. 3, pp. 27-50.
- BOSCH, J., 2000, "*Design and use of software architectures: adopting and evolving a product-line approach*", ACM Press/Addison-Wesley Publishing Co.
- BROWN, T.J., GAWLEY, R., BASHROUSH, R., *et al.*, 2006, "Weaving behavior into feature models for embedded system families", In: *10th International Software Product Line Conference*, pp. 52-61, Baltimore, Maryland, USA.
- BUCHANAN, M.C., ZELLWEGER, P.T., 1993, "Automatic Temporal Layout Mechanisms". In: *MULTIMEDIA '93 Proceedings of the first ACM international conference on Multimedia*, pp. 341-350, New York, USA.
- BÜHNE, S., LAUENROTH, K., POHL, K., 2004, "Why is it not Sufficient to Model Requirements Variability with Feature Models?", *Proceedings of Workshop: Automotive Requirements Engineering*, pp. 5-12, Nagoya, Japan.
- BUHR, R. J. A., 1999, "Making behavior a concrete architectural concept", In: *Proceedings of the Thirty-second Annual Hawaii International Conference on System Sciences*, v. 8, pp. 1-5, Maui, Hawaii, USA.
- BUSCHMANN, F., MEUNIER, R., ROHNERT, H., *et al.*, 2006, "Pattern-oriented software architecture: a system of patterns", John Wiley & Sons, Inc.
- CALDER, M., KOLBERG, M., MAGILL, E. H., *et al.*, 2003, "Feature interaction: a critical review and considered forecast", *Computer Networks: The International Journal of Computer and Telecommunications Networking*, v. 41, n. 1 (January), pp. 115-141.
- CARROLL, J. M., THOMAS, J.C., MILLER, L.A., *et al.*, 1980, "Aspects of solution structure in design problem solving", *American Journal of Psychology*, v. 93, n. 2, pp. 269-284.
- CASANOVA, M. A., TUCHERMAN, L., LIMA, M. J. D., *et al.*, 1991, "The Nested Context Model for Hyperdocuments", In: *Proceedings of the third annual ACM conference on Hypertext*, pp. 193-201, San Antonio, Texas, United States.

- CURTIS, B., 1984, "Fifteen years of psychology in software engineering: Individual differences". In: *Proceedings of the seventh ACM international conference on Software Engineering*, pp. 97-106, Piscataway, NJ, USA.
- CZARNECKI, K., EISENECKER, U.W., 2000, "*Generative programming: methods, tools, and applications*", ACM Press/Addison-Wesley Publishing Co.
- CZARNECKI, K., HELSEN, S., EISENECKER, U., 2004, "Staged configuration using feature models". In: *Software Product Lines: Third International Conference, SPLC 2004, Proceedings*, v. 3154, pp. 266–283, Boston, MA, USA, August 30-September 2.
- DANTAS, A.R., CORREA, A.L., WERNER, C., 2001, "Oráculo: Um Sistema de Críticas para a UML". In: *XV Simposio Brasileiro de Engenharia de Software - SBES, Caderno de Ferramentas*, pp. 398-403, Rio de Janeiro, Rio de Janeiro, Brasil, Outubro.
- DYBA, T., KITCHENHAM, B.A., JORGENSEN, M., 2005, "Evidence-based software engineering for practitioners", *IEEE Software*, v. 22, n. 1 (Jan/Feb), pp. 58-64.
- ERIKSSON, M., BÖRSTLER, J., BORG, K., 2005, "The PLUSS Approach - Domain Modeling with Features, Use Cases and Use Case Realizations", *Software Product Lines*, v. 3714, *Lecture Notes in Computer Science*, Springer Berlin-Heidelberg, pp. 33-44, 2005.
- ERIKSSON, M., BÖRSTLER, J., BORG, K., 2006, "Software product line modeling made practical", *Communications of the ACM*, v. 49, n. 12 (December), pp. 49-54.
- FERNANDES, P., PRUDÊNCIO, J.G., MARINHO, A., *et al.*, 2007, "Carga Dinâmica de Componentes via Biblioteca Brechó". In: *Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software (SBCARS 2007), Sessão de Ferramentas*, pp. 1-8, Campinas, São Paulo, Brasil, Agosto.
- FRAKES, W.B., KANG, K., 2005, "Software reuse research: Status and future", *IEEE Transactions on Software Engineering*, v. 31, n. 2, pp. 529-536.
- FREKSA, C., 1992, "Temporal reasoning based on semi-intervals", *Artificial Intelligence*, v. 54, pp. 199-227.
- GONZALEZ, A., LUNA, C., 2008, "Behavior Specification of Product Lines via Feature Models and UML Statecharts with Variabilities", In: *Proceedings of the 2008 International Conference of the Chilean Computer Science Society*, pp. 32-41, Punta Arenas, Chile.

- GRISS, M.L., FAVARO, J., WALTON, P., 1994, "Managerial and organizational issues-starting and running a software reuse program", In: Schaefer, W., Prieto-Diaz, R. e Matsumoto, M. (eds.), *Software Reusability*, pp. 51-78, New York, USA, Ellis Horwood.
- GRISS, M.L., 1997, "Software reuse architecture, process, and organization for business success", In: *Proceedings of the Eighth Israeli Conference on Computer Systems and Software Engineering*, pp. 86-89, Herzliya, Israel.
- GRISS, M.L., FAVARO, J., D'ALESSANDRO, M., 1998, "Integrating feature modelling with the RSEB". In: *Proceedings of the fifth International Conference on Software Reuse – ICSR5*, pp. 76-85, Victoria, British Columbia, Canada.
- GRISS, M. L., 2000, "Implementing Product-Line Features with Component Reuse", In: *Proceedings of the 6th International Conference on Software Reuse: Advances in Software Reusability*, pp. 137-152, Viena, Austria.
- GUESGEN, H.W., 1989, *Spatial Reasoning Based on Allen's Temporal Logic*, International Computer Science Institute, Berkeley University, ICSI Technical Report TR-89-049.
- van GURP, J., BOSCH, J., SVAHNBERG, M., 2001, "On the Notion of Variability in Software Product Lines", In: *Proceedings of the Working IEEE/IFIP Conference on Software Architecture*, pp. 45-54, Amsterdam, Netherlands.
- HAREL, D., 1987, Statecharts: A visual formalism for complex systems, *Science of Computer Programming*, v. 8, pp. 231-274.
- HARSU, M., 2002, *A Survey on Domain Engineering*, Institute of Software Systems, Tampere University of Technology. In: <http://practise2.cs.tut.fi/pub/papers/domeng.pdf>, acessado em: 16/11/2010.
- ITU, 2008, "User requirements notation (URN) – Language definition". In: <http://www.itu.int/rec/T-REC-Z.151-200811-I/en>, acessado em 04/12/2010.
- JACOBSON, I., 1994, *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison Wesley Longman Publishing Co., Inc.
- KANG, K.C., COHEN, S.G., HESS, J.A., *et al.*, 1990, *Feature-Oriented Domain Analysis (FODA) - Feasibility Study*, Software Engineering Institute (SEI), CMU/SEI-90-TR-21.
- KANG, K.C., KIM, S., LEE, J., 1998, "FORM: A feature-oriented reuse method with domain-pecific reference architectures" *Annals of Software Engineering*, v. 5, pp. 143-168.

- KANG, K.C., LEE, J., DONOHOE, P., 2002, "Feature-oriented product line engineering", *IEEE Software*, v. 19, n. 4 (July/August), pp. 58-65.
- KIRNER, T.G., DAVIS, A.M., 1996, "Requirements specification of real-time systems: temporal parameters and timing-constraints", *Information and Software Technology*, v. 38, pp. 735-741.
- KITCHENHAM, B. A., DYBA, T., JORGENSEN, M., 2004, "Evidence-Based Software Engineering", In: *Proceedings of the 26th International Conference on Software Engineering*, pp. 273-281, Edinburgh, Scotland.
- KNAUFF, M., 1999, "The cognitive adequacy of Allen's interval calculus for qualitative spatial representation and reasoning", *Spatial Cognition and Computation*, v. 1, n. 3, pp. 261-290.
- KRUCHTEN, P.B., 1995, "The 4+1 View Model of Architecture", *IEEE Software*, v.12, n.6 (November), pp. 42-50.
- LADKIN, P.B., 1990, *Constraint Reasoning With Intervals: A Tutorial, Survey and Bibliography*, International Computer Science Institute, Berkeley University, ICSI Technical Report TR-90-059.
- LEE, J., MUTHIG, D., 2006, "Feature-oriented variability management in product line engineering", *Communications of the ACM*, v. 49, pp. 55-59.
- LINDEN, F.J., SCHMID, K., ROMMES, E., 2007, *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*, Springer-Verlag New York, Inc.
- LITTLE, T.D.C., GHAFOR, A., 1993, "Interval-based conceptual models for time-dependent multimedia data", *IEEE Transactions on Knowledge and Data Engineering*, v. 5, n. 4 (August), pp. 551-563.
- MAIA, N.E.N., BLOIS, A.P.B., WERNER, C., 2005, "Odyssey-MDA: Uma Ferramenta para Transformação de Modelos UML". In: *XIX Simpósio Brasileiro de Engenharia de Software, Sessão de Ferramentas*, pp. 51-56, Uberlândia, Minas Gerais, Brasil, Outubro.
- MOON, M., HONG, M., YEOM, K., 2008, "Two-Level Variability Analysis for Business Process with Reusability and Extensibility", In: *Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference*, pp. 263-270, Turku, Finland.
- MURTA, L.G.P., BARROS, M.O., WERNER, C.M.L., 2001, "FrameDoc: Um Framework para a Documentação de Componentes Reutilizáveis". In: *IV*

- International Symposium on Knowledge Management/Document Management*, pp. 241-259, Curitiba, Paraná, Brasil, Agosto.
- NCL, 2009, *Nested Context Language (NCL) and Ginga-NCL for IPTV services*, International Telecommunication Union, T-REC-H.761-200904-P.
- NORTHROP, L., 2002, "SEI's software product line tenets", *IEEE Software*, v. 19, n. 4 (July/August), pp. 37-40.
- ODYSSEY, 2010, "Odyssey Project Homepage". In: <http://reuse.cos.ufrj.br/odyssey>. Acessado em 24/11/2010.
- OLIVEIRA, R., BLOIS, A.P., VASCONCELOS, A., WERNER, C.M.L., 2005, *Metamodelo de Características da Notação Odyssey-FEX: Descrição de Classes*, COPPE/UFRJ, Projeto Reuse - Técnicas e Ferramentas de apoio à Reutilização de Software - Relatório Técnico 2/2005. In: [http://reuse.cos.ufrj.br/files/publicacoes/relatorioTecnico/RT\\_OLIVEIRA\\_OdysseyFEX.pdf](http://reuse.cos.ufrj.br/files/publicacoes/relatorioTecnico/RT_OLIVEIRA_OdysseyFEX.pdf).
- OLIVEIRA, R., 2006, *Formalização e Verificação de Consistência na Representação de Variabilidades*. Dissertação de Mestrado. Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil.
- OMG, 2010, "Unified Modeling Language Infrastructure and Supersctructure Specification Version 2.3". In: <http://www.omg.org/spec/UML/2.3/>, acessado em 04/12/2010.
- ORACLE, 2010, "java.com: Java + You". In: <http://www.java.com/>. Acessado em 24/11/2010.
- PARNAS, D.L., 1976, "On the Design and Development of Program Families", *IEEE Transactions on Software Engineering*, v. 2, n. 7 (January), pp. 1-9.
- SHIRI, M., HASSINE, J., RILLING, J., 2007, "Feature interaction analysis: a maintenance perspective", In: *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering*, pp. 437-440, Atlanta, Georgia, USA.
- SHULL, F., CARVER, J., TRAVASSOS, G.H., 2001, "An empirical methodology for introducing software processes", *SIGSOFT Software Engineering Notes*, v. 26, Sept., pp. 288-296.
- SZASZ, N., VILANOVA, P., 2008, "Statecharts and Variabilities", *Second International Workshop on Variability Modelling of Software-intensive Systems*, pp. 131-140, Essen, Alemanha.

- TRACZ, W., 1994, "Domain-specific software architecture (DSSA) frequently asked questions (FAQ)", *SIGSOFT Software Engineering Notes*, v. 19, pp. 52-56.
- UMAIDEIA, 2010, "Sistema de Ajuda da Plataforma I-Reserve". In: <http://ajuda.ireserve.com.br/>, acessado em 12/12/2010.
- UMAIDEIA, 2010b, "Seja bem vindo ao I-Hotelaria". In: <http://ajuda.ihotelaria.com.br/>, acessado em 12/12/2010.
- UMAIDEIA, 2010c, "UmaIdeia, transformando idéias em realidade". In: [http://www.umaideia.com.br/quem\\_somos/index.asp](http://www.umaideia.com.br/quem_somos/index.asp), acessado em 12/12/2010.
- VELTHUIJSEN, H., BOUMA, L.G., 1994, "*Feature Interactions in Telecommunications Systems*", IOS Press.
- VERONESE, G., CORREA, A., WERNER, C., *et al.*, 2002, "ARES: Uma Ferramenta de Engenharia Reversa Java-UML". In: *VI Simpósio Brasileiro de Engenharia de Software, Sessão de Ferramentas*, pp. 347-352, Gramado, Rio Grande do Sul, Brasil, Outubro.
- WOHLIN, C., RUNESON, P., HÖST, M., *et al.*, 2000, *Experimentation in Software Engineering: an introduction*, USA, Kluwer Academic Press.
- W3C, 2010, "Extensible Markup Language (XML)". In: <http://www.w3.org/XML/>. Acessado em 24/11/2010.
- W3C, 2010b, "XML Schema Part 1: Structures Second Edition". In: <http://www.w3.org/TR/xmlschema-1/>. Acessado em 24/11/2010.

# Apêndice A

## Especificação do Domínio

O domínio analisado trata de um conjunto de sistemas utilizados por redes de hotelaria para disponibilização de tarifas na web. A arquitetura dos produtos desta plataforma é dividida em dois subsistemas: um ambiente administrativo na web para cadastro e edição de tarifas, utilizado exclusivamente por funcionários do hotel mediante autenticação, e um website institucional onde as tarifas são divulgadas ao público. A seguir são descritos em detalhes os principais elementos envolvidos na linha de produtos e como estes estão relacionados.

- **Tarifa:** as redes de hotéis que utilizam o sistema realizam o lançamento de diferentes tarifas ao longo do ano. Estas tarifas se referem à organização dos diferentes períodos que regem a hospedagem, tais como: disponibilidade de reserva dos quartos, preço praticado no período, o período de exibição dos quartos nos sites de web e condições especiais de contratação. Exemplos de tarifa são pacotes especiais para o dia dos namorados, natal, ano novo e condições especiais para eventos corporativos;
- **Unidade Habitacional:** nome dado a cada apartamento de uma construção com fins hoteleiros, mas também aplicável a edifícios residenciais. Na indústria hoteleira, o número de unidades habitacionais é um dos principais indicadores do tamanho de um hotel ou flat (UMAIDEIA, 2010);
- **Agendamento de Validade:** este agendamento se refere ao período de dias em que unidades habitacionais serão reservadas pelo cliente para a tarifa praticada. O agendamento de validade de uma tarifa é realizado pelos funcionários do hotel no ambiente administrativo da plataforma. O agendamento de validade representa as datas em que uma determinada tarifa poderá ser reservada. Este agendamento está relacionado ao período desejado pelo visitante (futuro hóspede) (UMAIDEIA, 2010). Ao



preencher o filtro de check in e check out no site (Figura 3.4), o mesmo retornará as tarifas que estiverem com validade agendada para o período desejado, possibilitando efetivar a reserva;

- **Solicitação de Reserva:** esta ação corresponde ao pedido enviado pelo cliente ao hotel pela web. Uma solicitação é vinculada: a uma tarifa que é praticada, ao período de dias em que o usuário deseja permanecer no hotel e a unidade habitacional na qual deseja se hospedar. Após o envio da solicitação ao hotel, este confirma o seu recebimento através de um Email ou SMS;
- **Agendamento de Exibição:** este agendamento corresponde ao período de dias em que uma tarifa será exibida na Web e no sistema interno de TV. As solicitações de reserva são realizadas exclusivamente neste período. O agendamento de exibição de uma tarifa é realizado por funcionários do hotel no ambiente administrativo da plataforma;
- **Confirmação de Reserva:** após a verificação de possíveis pendências sobre a solicitação enviada pelo hóspede, o hotel realiza a sua confirmação enviando um Email ou SMS, e informa que a solicitação foi atendida com sucesso mediante um código referente a reserva efetuada;
- **Tabela de Preços:** uma tabela indica o preço praticado para uma unidade habitacional em um determinado período de dias da tarifa. O cadastro de uma tabela de preços para uma tarifa é realizado por funcionários do hotel no ambiente administrativo da plataforma; e
- **Blackout Date:** serve para indisponibilizar uma determinada unidade habitacional em um intervalo de datas desejado. Desta forma, ela não aparecerá na lista de habitações possíveis para reserva numa determinada tarifa, mesmo que possua um valor (R\$) cadastrado.

## Requisitos

**Req.1:** A exibição da tarifa deve iniciar antes ou de forma simultânea a sua validade, ou seja, às datas em que as diárias podem ser utilizadas. Neste sentido, é desejável que os hóspedes possam ter conhecimento da Tarifa antes dos seus dias de hospedagem;

**Req.2:** Todo o período de validade de uma tarifa deve ter uma tabela de preço. Ou seja, não pode haver dias de validade sem um preço estipulado; e


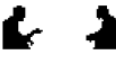



**Req.3:** Uma tarifa só será retornada numa consulta ao site se a mesma estiver com o agendamento de exibição cadastrado para a data em que ocorreu a consulta (a data atual ao acesso do visitante) e o agendamento de validade cadastrado, contemplando o período preenchido nos filtros do site (UMAIDEIA, 2010).

# Apêndice B

## Odyssey-FEX

A notação Odyssey-FEX tem como objetivo representar características de software e sua variabilidade em modelos de características e é utilizada pelo ambiente de modelagem da plataforma Odyssey (ODYSSEY, 2010). A semântica desta notação é formalizada em um metamodelo e um conjunto de regras de boa formação apresentado em (OLIVEIRA *et al.*, 2005) e (OLIVEIRA, 2006).

Durante o desenvolvimento do modelo de características nesta notação, cada característica é classificada segundo a sua categoria, variabilidade e opcionalidade. Estas três dimensões de variação ocorrem separadamente, ou seja, são escolhidas de forma independente e ortogonal. A classificação nestas categorias é realizada entre os tipos apresentados na Figura B.1. Cada característica pode pertencer a somente uma única categoria.

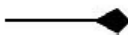




<u>Ícone</u>	<u>Tipo de Característica</u>	
	<b>Características de Domínio</b> – Características intimamente ligadas à essência do domínio. Representam as funcionalidades e/ou os conceitos do modelo e correspondem a casos de uso e componentes estruturais concretos.	Características de Análise
	<b>Características de Entidade</b> – São os atores do modelo. Entidades do mundo real que atuam sobre o domínio. Podem, por exemplo, expor a necessidade de uma interface com o usuário ou de procedimentos de controle.	
	<b>Características de Ambiente Operacional</b> - Características que representam atributos de um ambiente que uma aplicação do domínio pode usar e operar. Ex: tipo de terminal, sistemas operacionais, bibliotecas etc.	Características Tecnológicas
	<b>Características de Tecnologia de Domínio</b> - Características que representam detalhes de implementação de mais baixo nível, específicos para o contexto de um domínio. Ex: métodos de navegação em um domínio de aviões.	
	<b>Características de Técnicas de Implementação</b> – Características que representam detalhes de implementação de mais baixo nível, contudo de cunho mais genérico que as relativas à camada de tecnologia de domínio. Ex: técnicas de sincronização.	

**Figura B.1 – Categorias de Características da Notação Odyssey-FEX (OLIVEIRA, 2006)**

Em relação à variabilidade, características são classificadas entre pontos de variação, quando se tratam de aspectos parametrizáveis do domínio, e características variantes, que constituem alternativas possíveis para estes pontos de variação. Além disto, é possível a utilização de características invariantes, ou seja, que são fixas entre os diversos produtos da linha.

A terceira classificação ocorre sobre a opcionalidade. Características são identificadas como opcionais quando não integram todos os produtos da linha, e obrigatórias, quando são comuns a todos estes. Nesta classificação, características são representadas, respectivamente, através de caixas de contorno pontilhado e caixas de contorno contínuo.

As características definidas nesta notação se associam a partir das ligações apresentadas na Figura B.2. Esta diversificação de interações possibilita agregar uma semântica mais rica ao modelo de características e facilitar o entendimento do modelo de domínio.

<u>Representação</u>	<u>Descrição</u>
	<b>Composição</b> – Relacionamento em que uma característica é composta de várias outras. Denota relação na qual uma característica é parte fundamental de outra, de forma que a primeira não existe sem a segunda.
	<b>Agregação</b> – Relacionamento em que uma característica representa o todo, e as outras as partes. Similar à composição, porém as características envolvidas existem independentemente uma da outra.
	<b>Herança</b> – Relacionamento em que há uma generalização/especialização das características. Este tipo de relacionamento indica que as características mais especializadas (filhas) herdam as peculiaridades de características mais generalizadas (antecessores).
	<b>Associação</b> – Relacionamento simples entre duas características. Denota algum tipo de ligação entre seus membros. Pode ser nomeada, indicando um tipo específico de ligação.
	<b>Alternativo</b> ( <i>Alternative</i> ) - Relacionamento entre um ponto de variação e suas variantes, denota a pertinência de uma variante a um determinado ponto de variação.
<u>&lt;&lt;Implemented By&gt;&gt;</u>	<b>Implementado por</b> ( <i>Implemented By</i> ) - Relacionamento entre Características de Domínio e Características Tecnológicas, ou entre Características Tecnológicas que se encontrem em camadas diferentes. Indicam que uma determinada tecnologia é utilizada para implementar a característica relacionada.
<u>&lt;&lt;Communication Link&gt;&gt;</u>	<b>Ligação de Comunicação</b> ( <i>Communication Link</i> ) - Relacionamento existente entre Características de Entidade e Características de Domínio. Cumpre o mesmo papel do relacionamento de associação entre atores e casos de uso na UML

**Figura B.2 – Ligações entre Características da Notação Odyssey-FEX (OLIVEIRA, 2006)**

E finalmente, assim como em outras notações, o Odyssey-FEX prevê a utilização de regras de composição inclusivas e exclusivas. Estas são representadas no modelo de características através das letras “R” e “X”, respectivamente, que acompanham as características envolvidas na regra, mais especificamente em seu canto inferior.

A formalização do metamodelo, apresentada em (OLIVEIRA *et al.*, 2005), envolve a definição de classes que são distribuídas entre três pacotes: “Principal”, que define a taxonomia das características, “Relacionamento”, que especifica as propriedades dos relacionamentos entre características, e “Regras de Composição”, que define como as regras de dependência e exclusividade são formadas.

## Apêndice C

### Tabela de Transições do Modelo de Allen

R1/R2	<	>	d	di	o	oi	m	mi	s	si	f	fi
“antes de” (<)	<		< o m d s	<	<	< o m d s	<	< o m d s	<	<	< o m d s	<
“depois de” (>)		>	> oi mi d f	>	> oi mi d f	>	> oi mi d f	>	> oi mi d f	>	>	>
“durante” (d)	<	>	d		< o m d s	> oi mi d f	<	>	d	> oi mi d f	d	< o m d s
“contém” (di)	< o m di fi	> oi di mi si	o oi dur con =	di	o di fi	oi di si	o di fi	oi di si	di fi o	di	di si oi	di
“sobrepõe início de” (o)	<	> oi di mi si	o d s	< o m di fi	< o m	o oi dur con =	<	oi di si	o	di fi o	d s o	< o m
“sobrepõe fim de” (oi)	< o m di fi	>	oi d f	> oi mi di si	o oi dur con =	> oi mi	o di fi	>	oi d f	oi > mi	oi	oi di si
“antecede” (m)	<	> oi di si mi	o d s	<	<	o d s	<	f fi =	m	m	d s o	<
“sucede” (mi)	< o m di fi	>	oi d f	>	oi d f	>	s si =	>	d f oi	>	mi	mi
“inicia com” (s)	<	>	d	< o m di fi	< o m	oi d f	<	mi	s	s si =	d	< m o
“iniciado por” (si)	< o m di fi	>	oi d f	di	o di fi	oi	o di fi	mi	s si =	si	oi	di
“finaliza com” (f)	<	>	d	> oi mi di si	o d s	> oi mi	m	>	d	> oi mi	f	f fi =
“finalizado por” (fi)	<	> oi mi di si	o d s	di	o	oi di si	m	si di oi	o	di	f fi =	fi

**Tabela C.1 – Relações do Modelo de Allen <sup>12</sup>**

<sup>1</sup> Os relacionamentos *dur* e *con* se tratam de contrações dos conjuntos {d, s, f} e {di, si, fi}

<sup>2</sup> A transitividade do relacionamento de equivalência (=) foi suprimida.

# Apêndice D

## Cenários de Experimentação

### Cenário I - Sistemas de Compra Coletiva

#### Informações do Domínio

Uma linha de produtos de software agrega sistemas web de apoio à publicação de ofertas de produtos e serviços para compra coletiva. Nesta modalidade de negócio, preços inferiores à média de mercado são praticados para maiores volumes de venda, para com isso realizar a divulgação em massa dos responsáveis pelos produtos ou serviços ofertados. Cada responsável é comumente chamado de estabelecimento parceiro.

Cada oferta cadastrada é divulgada nos meios de comunicação e é vendida através de um portal da web. No momento em que um cliente realiza uma aquisição de uma oferta pelo site, um cupom é gerado e enviado ao mesmo, contendo um código exclusivo que identifica a operação de compra. Imediatamente antes da aquisição do produto ou consumo do serviço, o cliente apresenta o código do cupom de compra no estabelecimento parceiro. Com este código, o estabelecimento é capaz de identificar a venda e concluir a entrega do produto ou prestação do serviço.

Esta linha de produtos de software prevê sistemas com as características descritas a seguir e dispostas no Modelo de Características apresentado na Figura D.1:

#### Oferta

- Esta característica trata essencialmente das ofertas cadastradas no sistema e se refere ao subsistema responsável pelo seu cadastro, pela manutenção de suas informações e pela geração de cupons referentes à mesma. Além disto, trata de estipular e controlar o período de utilização, ou seja, a data de início e fim referentes ao período em que os cupons de uma determinada oferta poderão ser utilizados no respectivo estabelecimento parceiro;

### Controlador de Exibição na TV

- Módulo responsável pelo controle da exibição das ofertas na TV. Toda oferta anunciada nos sistemas com esta característica possui, obrigatoriamente, um período de dias, delimitado por uma data de início e uma data de término, no qual será exibida na rede aberta de televisão;

### Controlador de Exibição na Web

- Módulo responsável pelo controle da exibição das ofertas na web. Toda oferta anunciada nos sistemas com esta característica possui, obrigatoriamente, um período de dias, delimitado por uma data de início e uma data de término, no qual será exibida na web;

### Monitoramento de Consumo

- Subsistema responsável pela monitoração do consumo dos cupons que dão direito às ofertas. Este monitor serve para capturar e registrar pela web o consumo dos cupons comunicado pelos estabelecimentos parceiros no momento em que estes são utilizados;

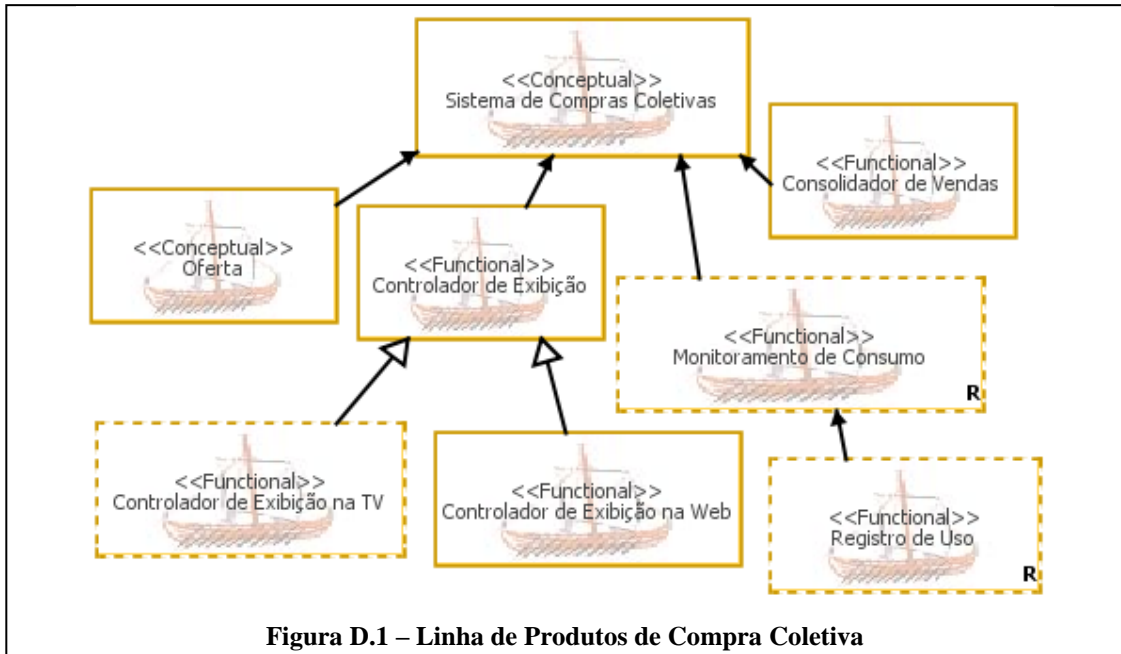
### Registrador de Uso

- Subsistema responsável por tratar interfaces diferenciadas de comunicação (SMS, email, telefone, etc.) através das quais os estabelecimentos parceiros registram o uso de cupons; e

### Consolidador de Vendas

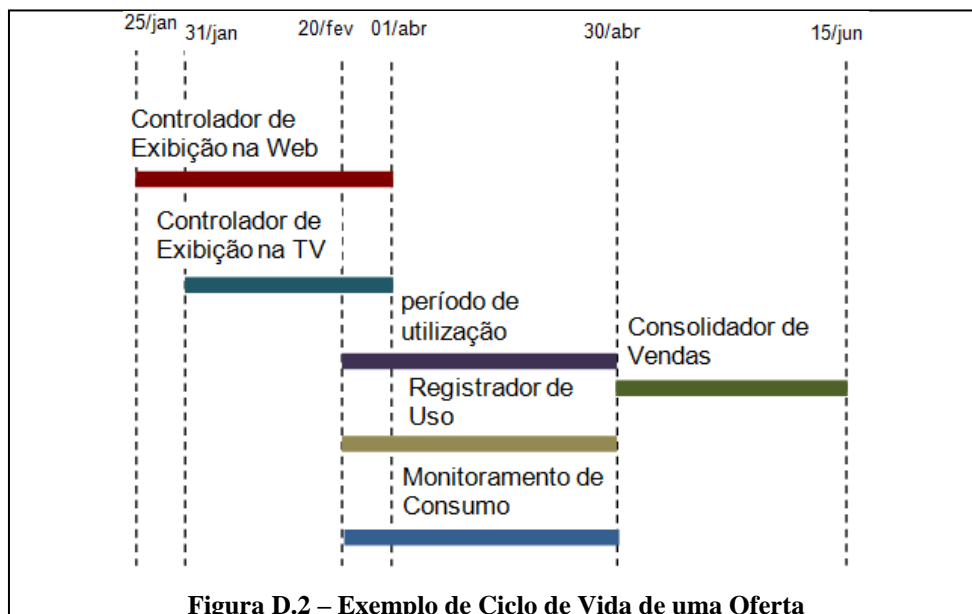
- Esta característica é responsável pela geração de relatórios sobre o comportamento de compra de ofertas e de consumo de cupons, e fornecem indicadores como horários, datas e antecedências médias em que as ofertas foram compradas e consumidas com maior frequência. E finalmente, sistemas com a característica *Monitoramento de Consumo* incluem na consolidação de uma determinada oferta, o comportamento do consumo de seus cupons. Já em sistemas onde a característica *Monitoramento de Consumo* é ausente, o relatório se resume às informações de compra e geração de cupons.





A marcação com a letra “R” no modelo de características é referente a regra de composição inclusiva que determina que a característica *Registro de Uso* requer a característica *Monitoramento de Consumo*.

Um aspecto importante sobre esta linha de produtos é que a escolha das características que constituirão um sistema determina o ciclo de vida que é seguido por todas as ofertas cadastradas no sistema final. A seguir, é apresentado na Figura D.2, um exemplo de ciclo de vida contendo as etapas seguidas pelas ofertas cadastradas em um sistema que contém todas as características apresentadas.



Uma vez que cada sistema é caracterizado pelo ciclo de vida padrão seguido por todas as suas ofertas, os seguintes requisitos regem o funcionamento destes sistemas e o arranjo temporal deste ciclo:

**Req.1:** Entre as informações fornecidas no cadastro de uma oferta, está o período de utilização dos seus cupons. Esta informação não pode ser alterada;

**Req.2:** O período de *Monitoramento de Consumo* é simultâneo ao período de utilização de cupons;

**Req.3:** O *Consolidador de Vendas* inicia a geração do relatório consolidado imediatamente após o término da captura das informações pelo *Monitoramento de Consumo*;

**Req.4:** O estabelecimento parceiro é responsável por estipular quais informações constam na consolidação;

**Req.5:** Os controladores de exibição de ofertas são iniciados obrigatoriamente antes do *Monitoramento de Consumo* ser iniciado, havendo um espaço de tempo entre estes dois eventos. Além disto, controladores de exibição de ofertas são finalizados antes que este monitoramento tenha terminado, havendo um espaço de tempo entre estes dois eventos;

**Req.6:** Em sistemas onde o *Monitoramento de Consumo* é ausente, os controladores de exibição devem terminar imediatamente antes do início do período de utilização de cupons;

**Req.7:** O período de atividade do *Registrador de Uso* é simultâneo ao período de *Monitoramento de Consumo*;

## **Solicitação de Adequações**

Você foi contratado como engenheiro de software pela empresa especializada no ramo de compras coletivas e que detém os direitos da linha de produtos apresentada. Foi solicitado que você realizasse a evolução da linha para que esta contemple produtos com um novo subsistema responsável pelo recebimento de avaliações sobre a experiência de consumo das ofertas. Esta evolução agrega novos requisitos, apresentados a seguir, que devem ser atendidos por este trabalho:

**Req.8:** A funcionalidade de recebimento de avaliações estará presente em apenas uma parcela dos novos sistemas pretendidos para a linha;

**Req.9:** As ofertas cadastradas nos sistemas podem ser especializadas em Artigos Esportivos, Gastronomia, Beleza e Viagens. Quando isto ocorre, todas as ofertas do sistema tratam de uma mesma especialidade;

**Req.10:** O recebimento de avaliações dos clientes ocorre durante um determinado período interno ao ciclo de vida da oferta;

**Req.11:** O período em que as avaliações são recebidas é iniciado imediatamente após o término do *Monitoramento de Consumo*;

**Req.12:** Sistemas de Compra Coletiva especializados em ofertas de gastronomia não possuem exibição na TV;

**Req.13:** O *Consolidador de Vendas* termina imediatamente antes do início do período de recebimento de avaliações. Isto ocorre porque o formato da solicitação que é enviada aos clientes é condicionado pelas informações obtidas desta consolidação; e

**Req.14:** O *Controlador de Exibição na TV* deve iniciar após o *Registrador de Uso* ter sido iniciado, havendo um espaço de tempo estes dois eventos.

Com base nestas informações, realize adequações que você julgar necessárias sobre os artefatos indicados pelo instrutor do experimento, para que a linha de produtos, após as alterações, passe a contemplar o recebimento de avaliações de clientes.

## Cenário II - Sistemas Reservas em Hotéis

### **Informações do Domínio**

Uma linha de produtos de software agrega sistemas web para a administração de tarifas de hospedagem em uma rede de hotéis. Cada hotel desta rede possui um sistema no qual suas tarifas são cadastradas, para que possam ser exibidas pela web por onde hóspedes solicitam reservas. Esta linha de produtos prevê sistemas com as características descritas a seguir e dispostas no Modelo de Característica apresentado na Figura D.3:

#### Tarifa

- Esta característica trata essencialmente das tarifas cadastradas no sistema e se refere ao subsistema responsável pelo seu cadastro e pela manutenção de suas informações, como por exemplo, condições especiais de aquisição. Exemplos de tarifa são pacotes para o dia dos namorados, natal, ano novo e condições especiais para eventos corporativos;

#### Agendamento de Exibição

- Este módulo é responsável pelo agendamento de exibição realizado para cada tarifa que é cadastrada no sistema. Este agendamento corresponde ao período de dias, delimitado por uma data de início e uma data de fim, em que uma tarifa será exibida na Web. Os sistemas que não possuem esta característica têm suas reservas realizadas diretamente por agências de viagens, ou seja, suas tarifas não são divulgadas para os usuários da web;

#### Agendamento de Validade

- Este módulo é responsável pelo agendamento de validade realizado para cada tarifa que é cadastrada no sistema. Este agendamento corresponde ao período de dias, delimitado por uma data de início e uma data de fim, em que os quartos serão reservados e futuramente ocupados pelo hóspede;

### Solicitador de Reserva

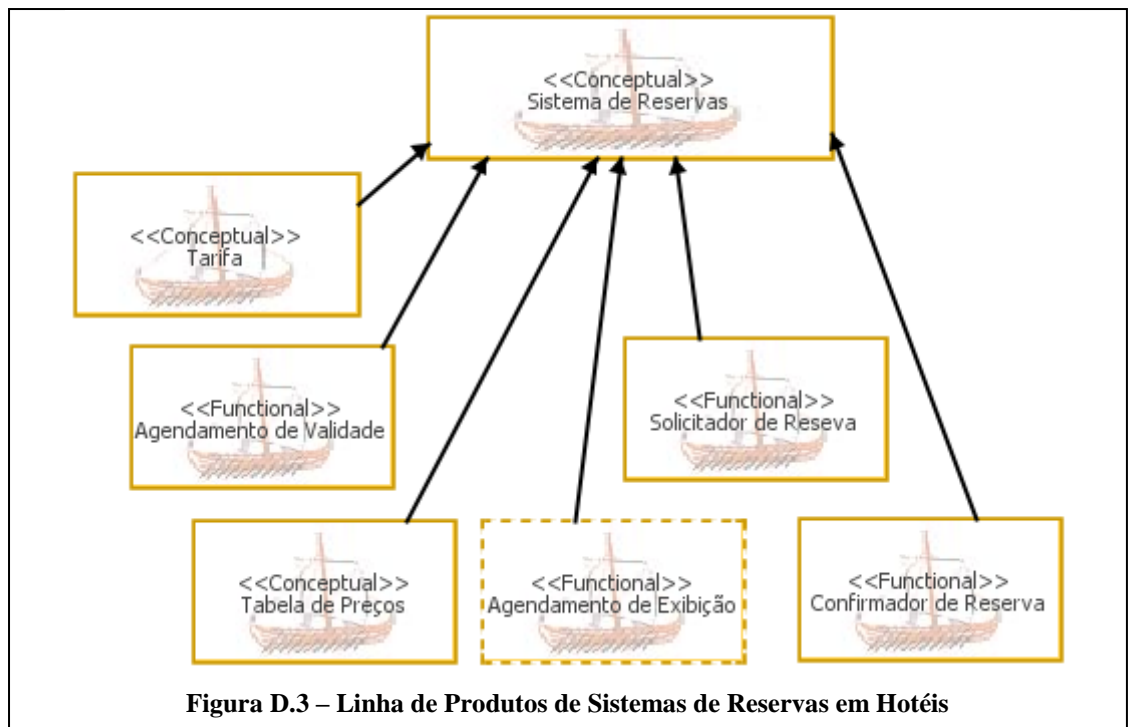
- Módulo responsável pelo envio de solicitações de reserva para um respectivo hotel da rede. Este envio é realizado pelos hóspedes a partir da web. Cada solicitação é relacionada a uma tarifa cadastrada no sistema do hotel para o qual esta é enviada. No momento em que a solicitação é enviada, esta permanece pendente de confirmação;

### Confirmador de Reserva

- Módulo responsável pelo envio de confirmações de reserva aos hóspedes por SMS e/ou email. Este envio é realizado após a verificação de pendências realizada pelos funcionários do hotel. Cada confirmação é relacionada a uma solicitação que antes permanecia pendente; e

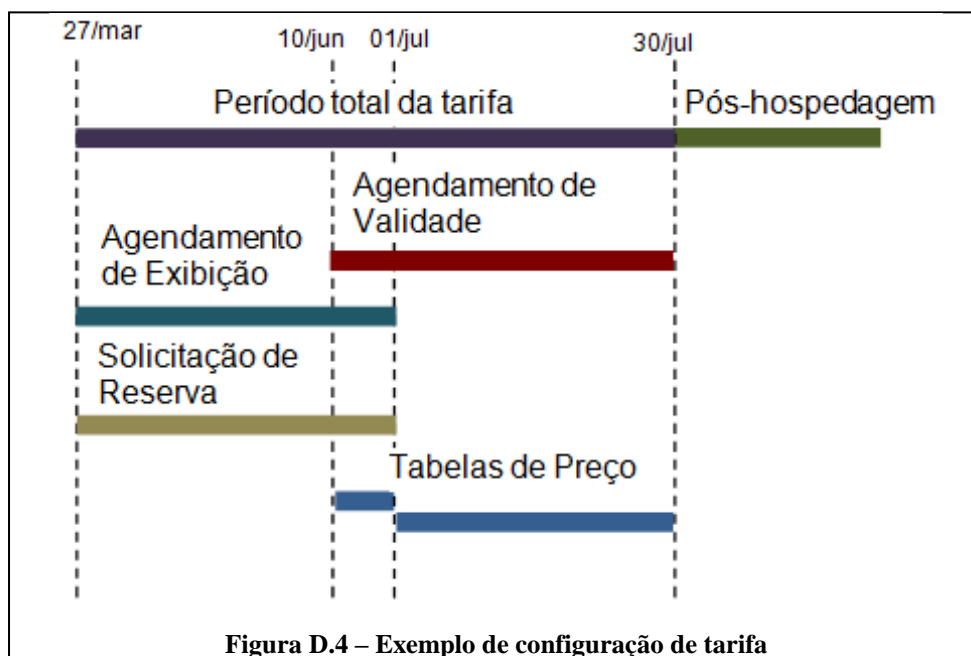
### Tabela de Preços

- Este módulo é responsável pelo cadastro das tabelas de preço da tarifa. Uma tabela de preço determina o valor que é pago pelos hóspedes nos dias em que estes estiveram hospedados no hotel.



**Figura D.3 – Linha de Produtos de Sistemas de Reservas em Hotéis**

Um aspecto importante sobre esta linha de produtos é que a escolha das características que constituirão um sistema determina os agendamentos e configurações necessárias para cada tarifa, que é cadastrada neste sistema final. A seguir, é apresentado na Figura D.4, um exemplo de configuração contendo os agendamentos necessários às tarifas cadastradas em um sistema que contém algumas das características apresentadas.



Uma vez que cada sistema é caracterizado por configurações que devem ser feitas para cada tarifa cadastrada, os seguintes requisitos determinam o funcionamento do sistema e o arranjo temporal destas configurações:

**Req.1:** Os módulos de *Agendamento de Exibição* e o *Agendamento de Validade*, utilizados no cadastro de uma tarifa, são utilizados exclusivamente pelos funcionários em um ambiente administrativo interno do hotel;

**Req.2:** O período de envio de solicitações de reserva é simultâneo ao período do *Agendamento de Exibição*;

**Req.3:** O período total da tarifa corresponde desde o início do período do *Agendamento de Exibição* ao término do período do *Agendamento de Validade*;

**Req.4:** O período imediatamente posterior ao *Agendamento de Validade* é chamado pós-hospedagem;

**Req.5:** Em todos os sistemas onde o *Agendamento de Exibição* está presente, este deve iniciar antes do *Agendamento de Validade* ter iniciado;

**Req.6:** Em uma parte dos sistemas, o período determinado pelo *Agendamento de Exibição* termina durante o período do *Agendamento de Validade* ou junto com o término deste agendamento. Já na outra parte dos sistemas, o período determinado pelo *Agendamento de Exibição* termina imediatamente antes do período determinado pelo *Agendamento de Validade* iniciar;

### **Solicitação de Adequações**

Você foi contratado como engenheiro de software pela empresa fornecedora dos sistemas utilizados pela rede de hotéis. Foi solicitado que você realize a evolução da linha para que esta contemple produtos com um novo subsistema responsável pelo envio de confirmações de reserva em massa. Esta evolução agrega novos requisitos, apresentados a seguir, que devem ser contemplados no seu trabalho:

**Req.7:** A funcionalidade do *Confirmador de Reservas* integra apenas parte dos sistemas da linha;

**Req.8:** Sistemas com *Confirmador de Reservas* não possuem o *Confirmador de Reservas em Massa* e vice-versa;

**Req.9:** O *Confirmador de Reservas em Massa* deve enviar todas as confirmações em um período único que inicia imediatamente após o término do período do *Agendamento de Exibição*;

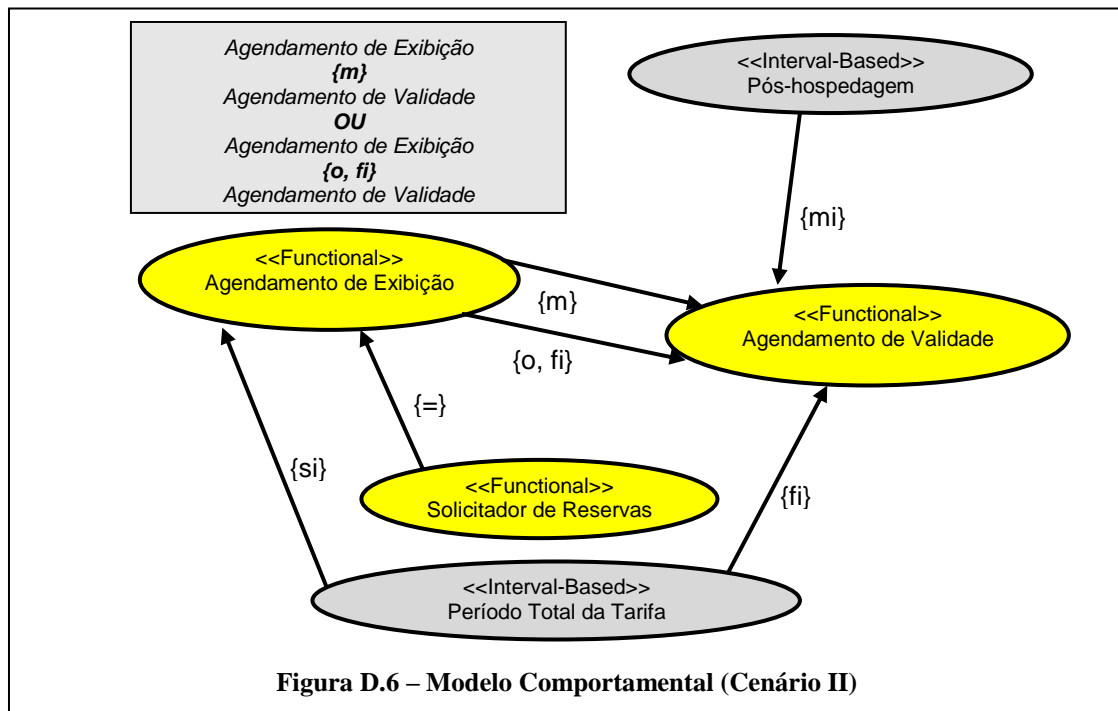
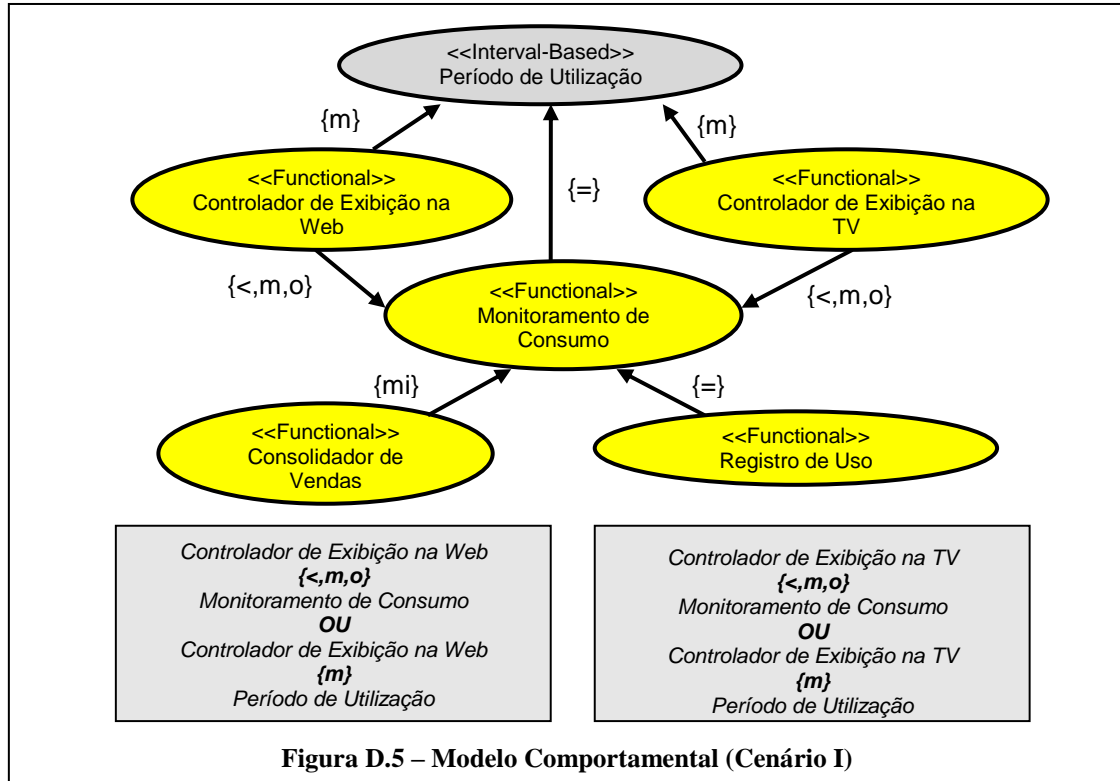
**Req.10:** Período de solicitações de reserva inicia e termina necessariamente antes do período de *Agendamento de Validade*; e

**Req.11:** O período determinado pelo *Agendamento de Validade* deve iniciar imediatamente após o término do envio de todas as confirmações pelo *Confirmador de Reservas em Massa*.

Com base nestas informações, realize adequações que você julgar necessárias sobre os artefatos indicados pelo instrutor do experimento, para que a linha de produtos passe a contemplar a confirmação de reservas em massa.

## Informações Comportamentais

O comportamento dos produtos da linha de cada cenário é expresso nos Modelos Comportamentais apresentados na Figura D.5 e na Figura D.6.





# Apêndice E

## Esquema de representação de Leiautes

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:complexType name="tLayout">
    <xsd:sequence>
      <xsd:element name="intervalos" minOccurs="1" maxOccurs="1" type="tIntervalo" />
      <xsd:element name="relacionamentos" minOccurs="1" maxOccurs="1"
        type="tRelacionamento" />
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" />
  </xsd:complexType>

  <xsd:complexType name="tIntervalo">
    <xsd:sequence>
      <xsd:element name="intervalo" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:attribute name="id" type="xsd:string" use="required" />
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="tRelacionamento">
    <xsd:sequence>
      <xsd:element name="relacionamento" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="tipo" minOccurs="0" maxOccurs="unbounded"
              type="tTipoRelacionamento" />
          </xsd:sequence>
          <xsd:attribute name="id" type="xsd:string" use="required" />
          <xsd:attribute name="origem" type="xsd:string" use="required" />
          <xsd:attribute name="destino" type="xsd:string" use="required" />
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:simpleType name="tTipoRelacionamento">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="ANTECEDE" />
      <xsd:enumeration value="SUCEDE" />
      <xsd:enumeration value="ESTA_CONTIDO" />
      <xsd:enumeration value="CONTEM" />
      <xsd:enumeration value="SOBREPOE_INICIO" />
      <xsd:enumeration value="SOBREPOE_FIM" />
      <xsd:enumeration value="ANTECEDE_IMEDIATAMENTE" />
      <xsd:enumeration value="SUCEDE_IMEDIATAMENTE" />
      <xsd:enumeration value="INICIA_COM" />
      <xsd:enumeration value="INICIADO_POR" />
      <xsd:enumeration value="FINALIZA_COM" />
      <xsd:enumeration value="FINALIZADO_POR" />
      <xsd:enumeration value="EQUIVALE" />
      <xsd:enumeration value="&lt;" />
      <xsd:enumeration value="&gt;" />
      <xsd:enumeration value="m" />
      <xsd:enumeration value="mi" />
      <xsd:enumeration value="d" />
      <xsd:enumeration value="di" />
      <xsd:enumeration value="f" />
      <xsd:enumeration value="fi" />
    </xsd:restriction>
  </xsd:simpleType>

```

```

        <xsd:enumeration value="o" />
        <xsd:enumeration value="oi" />
        <xsd:enumeration value="=" />
        <xsd:enumeration value="si" />
        <xsd:enumeration value="s" />
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="tTipoRelacionamentoAbreviado">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="&lt;" />
        <xsd:enumeration value="&gt;" />
        <xsd:enumeration value="m" />
        <xsd:enumeration value="mi" />
        <xsd:enumeration value="d" />
        <xsd:enumeration value="di" />
        <xsd:enumeration value="f" />
        <xsd:enumeration value="fi" />
        <xsd:enumeration value="o" />
        <xsd:enumeration value="oi" />
        <xsd:enumeration value="=" />
        <xsd:enumeration value="si" />
        <xsd:enumeration value="s" />
    </xsd:restriction>
</xsd:simpleType>

    <xsd:element name="layouts">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="layout" type="tLayout"
maxOccurs="unbounded" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>

```

# Apêndice F

## Saída de Depuração

```
<?xml version="1.0"?>
<debug xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<intervalos>
  <intervalo id="A"/>
  <intervalo id="B"/>
  <intervalo id="C"/>
  <intervalo id="D"/>
</intervalos>

<relacionamentos>
  <RelacionamentoBase id="relDparaB" rep="D :: (&gt; ) SUCEDE :: B"/>
  <RelacionamentoBase id="relBparaC" rep="B :: (&lt; ) ANTECEDE :: C"/>
  <RelacionamentoBase id="relDparaC" rep="D :: (&lt; ) ANTECEDE :: C"/>
  <RelacionamentoBase id="relAparaB" rep="A :: (di ) CONTEM :: B"/>
  <RelacionamentoBase id="relCparaA" rep="A :: (f ) FINALIZA_COM :: B"/>
</relacionamentos>

<derivacoes>
  <RelacionamentoDerivado rep="(D :: (&gt; ) SUCEDE :: A)">
    <RelacionamentoBase id="relDparaB" rep="D :: (&gt; ) SUCEDE :: B"/>
    <RelacionamentoBase id="relCparaA" rep="A :: (f ) FINALIZA_COM :: B"/>
  </RelacionamentoDerivado>
  <RelacionamentoDerivado rep="(B :: (&lt; ) ANTECEDE :: C)">
    <RelacionamentoBase id="relDparaB" rep="D :: (&gt; ) SUCEDE :: B"/>
    <RelacionamentoBase id="relDparaC" rep="D :: (&lt; ) ANTECEDE :: C"/>
  </RelacionamentoDerivado>
  <RelacionamentoDerivado rep="(A :: (&lt; ) ANTECEDE :: C)">
    <RelacionamentoBase id="relAparaB" rep="A :: (di ) CONTEM :: B"/>
    <RelacionamentoBase id="relBparaC" rep="B :: (&lt; ) ANTECEDE :: C"/>
  </RelacionamentoDerivado>
  <RelacionamentoDerivado rep="(B :: (&lt; ) ANTECEDE :: D)">
    <RelacionamentoBase id="relCparaA" rep="A :: (f ) FINALIZA_COM :: B"/>
  </RelacionamentoDerivado>
  <RelacionamentoDerivado rep="(A :: (&lt; ) ANTECEDE :: D)">
    <RelacionamentoBase id="relAparaB" rep="A :: (di ) CONTEM :: B"/>
    <RelacionamentoBase id="relDparaB" rep="D :: (&gt; ) SUCEDE :: B"/>
  </RelacionamentoDerivado>
  <RelacionamentoDerivado rep="(D :: (&gt; ) SUCEDE :: B)">
    <RelacionamentoDerivado rep="(D :: (&gt; ) SUCEDE :: A)">
      <RelacionamentoBase id="relDparaB" rep="D :: (&gt; ) SUCEDE :: B"/>
      <RelacionamentoBase id="relCparaA" rep="A :: (f ) FINALIZA_COM :: B"/>
    </RelacionamentoDerivado>
    <RelacionamentoBase id="relAparaB" rep="A :: (di ) CONTEM :: B"/>
  </RelacionamentoDerivado>
  <RelacionamentoDerivado rep="(C :: (&gt; ) SUCEDE :: B)">
    <RelacionamentoDerivado rep="(A :: (&lt; ) ANTECEDE :: C)">
      <RelacionamentoBase id="relCparaA" rep="A :: (f ) FINALIZA_COM :: B"/>
      <RelacionamentoDerivado rep="(B :: (&lt; ) ANTECEDE :: C)">
        <RelacionamentoBase id="relDparaB" rep="D :: (&gt; ) SUCEDE :: B"/>
        <RelacionamentoBase id="relDparaC" rep="D :: (&lt; ) ANTECEDE :: C"/>
      </RelacionamentoDerivado>
    </RelacionamentoDerivado>
    <RelacionamentoBase id="relAparaB" rep="A :: (di ) CONTEM :: B"/>
  </RelacionamentoDerivado>
  <RelacionamentoDerivado rep="(A :: (&lt; ) ANTECEDE :: D)">
    <RelacionamentoBase id="relAparaB" rep="A :: (di ) CONTEM :: B"/>
  </RelacionamentoDerivado>
  <RelacionamentoDerivado rep="(B :: (&lt; ) ANTECEDE :: D)">
    <RelacionamentoBase id="relCparaA" rep="A :: (f ) FINALIZA_COM :: B"/>
  </RelacionamentoDerivado>
  <RelacionamentoDerivado rep="(A :: (&lt; ) ANTECEDE :: D)">
    <RelacionamentoBase id="relAparaB" rep="A :: (di ) CONTEM :: B"/>
    <RelacionamentoBase id="relDparaB" rep="D :: (&gt; ) SUCEDE :: B"/>
  </RelacionamentoDerivado>
</RelacionamentoDerivado>
<RelacionamentoDerivado rep="(A :: (&lt; ) ANTECEDE :: D)">
```

```

    <RelacionamentoBase id="relAparaB" rep="A :: (di ) CONTEM :: B"/>
    <RelacionamentoBase id="relDparaB" rep="D :: (> ) SUCEDE :: B"/>
  </RelacionamentoDerivado>
  <RelacionamentoDerivado rep="(A :: (< ) ANTECEDE :: C)">
    <RelacionamentoBase id="relCparaA" rep="A :: (f ) FINALIZA_COM :: B"/>
    <RelacionamentoDerivado rep="(B :: (< ) ANTECEDE :: C)">
      <RelacionamentoBase id="relDparaB" rep="D :: (> ) SUCEDE :: B"/>
      <RelacionamentoBase id="relDparaC" rep="D :: (< ) ANTECEDE :: C"/>
    </RelacionamentoDerivado>
  </RelacionamentoDerivado>
</derivacoes>

<inconsistencias>
  <RelacionamentoDerivado rep="(A :: ( ) :: B)">
    <RelacionamentoBase id="relCparaA" rep="A :: (f ) FINALIZA_COM :: B"/>
    <RelacionamentoBase id="relAparaB" rep="A :: (di ) CONTEM :: B"/>
  </RelacionamentoDerivado>
</inconsistencias>

</debug>

```

# Apêndice G

## Esquema de Depuração

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="debug">
    <xsd:complexType>
      <xsd:sequence>

        <xsd:element name="intervalos" minOccurs="1" maxOccurs="1">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="intervalo" minOccurs="0" maxOccurs="unbounded">
                <xsd:complexType>
                  <xsd:attribute name="id" type="xsd:string" use="required" />
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>

        <xsd:element name="relacionamentos" minOccurs="0" maxOccurs="1">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="RelacionamentoBase" type="tRelacionamentoBase"
minOccurs="1" maxOccurs="unbounded"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>

        <xsd:element name="derivacoes" minOccurs="1" maxOccurs="1">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="RelacionamentoDerivado"
type="tRelacionamentoDerivado" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>

        <xsd:element name="inconsistencias" minOccurs="1" maxOccurs="1">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="RelacionamentoDerivado"
type="tRelacionamentoDerivado" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>

      </xsd:sequence>

      <xsd:attribute name="data" type="xsd:string"/>
      <xsd:attribute name="id_layout_origem" type="xsd:string"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="tRelacionamentoBase">
```

```

        <xsd:attribute name="id" type="xsd:string" use="required" />
        <xsd:attribute name="rep" type="xsd:string" use="required" />
    </xsd:complexType>

    <xsd:complexType name="tRelacionamentoDerivado">
        <xsd:sequence>
            <xsd:choice>
                <xsd:element name="RelacionamentoBase" type="tRelacionamentoBase"/>
                <xsd:element name="RelacionamentoDerivado"
type="tRelacionamentoDerivado"/>
            </xsd:choice>
            <xsd:choice>
                <xsd:element name="RelacionamentoBase" type="tRelacionamentoBase"/>
                <xsd:element name="RelacionamentoDerivado"
type="tRelacionamentoDerivado"/>
            </xsd:choice>
        </xsd:sequence>

        <xsd:attribute name="rep" type="xsd:string" use="required" />

    </xsd:complexType>
</xsd:schema>

```

# Apêndice H

## Declaração de Consentimento

Declaro ter mais de 18 anos de idade e concordo em participar do experimento realizado em razão da dissertação de mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação pelo mestrando **Wagner Schau de Castro**.

### OBJETIVO

O presente experimento visa analisar o impacto do uso de intervalos de tempo na descrição de comportamentos de linhas de produtos de software através da abordagem TimeFEX, proposta pela dissertação tratada nesta declaração.

### PROCEDIMENTO

A aplicação do estudo será realizada em duas fases. Em cada uma destas, o participante será solicitado a analisar e realizar modificações sobre um conjunto de artefatos específicos de um domínio de aplicação e área de conhecimento. Eu entendo que, uma vez que o experimento tenha terminado, o trabalho por mim desenvolvido será utilizado no presente estudo que envolve os procedimentos e as técnicas ensinadas durante o experimento.

### CONFIDENCIALIDADE

Toda informação coletada neste estudo é confidencial, e meu nome não será identificado em momento algum. Da mesma forma, me comprometo a não comunicar os meus resultados enquanto não terminar o estudo, bem como manter sigilo sobre as técnicas e os documentos que integram o experimento.

### BENEFÍCIOS, LIBERDADE DE DESISTÊNCIA

Eu entendo que os benefícios que receberei deste estudo são limitados ao aprendizado do material que é distribuído. Eu entendo que sou livre para realizar perguntas a qualquer momento, solicitar que qualquer informação relacionada a minha pessoa não seja incluída no estudo ou comunicar a minha desistência de participação. Eu entendo que participo de livre e espontânea vontade com o único intuito de contribuir para o avanço e desenvolvimento das técnicas e processos tratados pelo estudo.

### **Pesquisador Responsável**

Wagner Schau de Castro

### **Professor Responsável**

Profª. Cláudia Maria Lima Werner

Nome: \_\_\_\_\_

Local e data: \_\_\_\_\_

Assinatura: \_\_\_\_\_

# Apêndice I

## Formulário de Caracterização

Nome do Participante: \_\_\_\_\_

### Formação Acadêmica:

<input type="checkbox"/> Superior Incompleto	<input type="checkbox"/> Mestrado Incompleto	<input type="checkbox"/> Doutorado Incompleto
<input type="checkbox"/> Superior Completo	<input type="checkbox"/> Mestrado Completo	<input type="checkbox"/> Doutorado Completo

### Experiência:

Indique o seu nível de experiência na análise e desenvolvimento de sistemas de software:

<input type="checkbox"/>	<i>Nenhuma experiência</i>
<input type="checkbox"/>	<i>Atuação restrita a projetos acadêmicos</i>
<input type="checkbox"/>	<i>Desenvolvimento de projetos pessoais</i>
<input type="checkbox"/>	<i>Experiência profissional em projetos para a indústria</i>

Indique o seu nível de experiência na Engenharia de Linhas de Produtos de Software (LPS):

<input type="checkbox"/>	<i>Nenhum</i>
<input type="checkbox"/>	<i>Estudei em aula ou em livro</i>
<input type="checkbox"/>	<i>Experiência profissional em projetos para a indústria</i>

Indique o seu nível de conhecimento sobre os domínios de conhecimento listados de acordo com a legenda abaixo:

	Nenhum	Baixo	Médio	Alto
Compras Coletivas				
Mercado Financeiro				
Hotelaria				
Telecomunicações				
TV Digital				



# Apêndice J

## Formulário de Avaliação – Primeira Etapa

Nome do Participante: \_\_\_\_\_

**Responda às perguntas abaixo justificando e especificando sua resposta quando aplicável.**

Os requisitos que envolvem temporalidade foram de fácil compreensão?	<input type="radio"/> Sim <input type="radio"/> Não <input type="radio"/> Parcialmente
--	---

Estes mesmos requisitos foram suficientes para descrever satisfatoriamente os comportamentos de todos os produtos da linha?	<input type="radio"/> Sim <input type="radio"/> Não <input type="radio"/> Parcialmente

Em algum momento você considerou mais de um comportamento que os produtos da linha podem adquirir?	<input type="radio"/> Sim <input type="radio"/> Não <input type="radio"/> Parcialmente

Os produtos previstos pela linha apresentaram algum comportamento inconsistente?	<input type="radio"/> Sim <input type="radio"/> Não <input type="radio"/> Parcialmente

Quais outras dificuldades você experimentou durante a realização das tarefas?

# Apêndice K

## Formulário de Avaliação – Segunda Etapa

Nome do Participante: \_\_\_\_\_

**Responda às perguntas abaixo justificando e especificando sua resposta quando aplicável.**

Os diferentes tipos de relacionamentos temporais entre intervalos utilizados na técnica são de fácil entendimento?	<input type="radio"/> Sim <input type="radio"/> Não <input type="radio"/> Parcialmente
--	---

A utilização do modelo comportamental impactou no trabalho realizado?	<input type="radio"/> Sim <input type="radio"/> Não <input type="radio"/> Parcialmente

A transcrição de requisitos de software para relacionamentos temporais impactou na realização das tarefas?	<input type="radio"/> Sim <input type="radio"/> Não <input type="radio"/> Parcialmente

A validação do modelo comportamental através do suporte computacional impactou na realização das tarefas?	<input type="radio"/> Sim <input type="radio"/> Não <input type="radio"/> Parcialmente

Quais outras dificuldades você experimentou durante a realização das tarefas?