VALTER CASTELHANO DE OLIVEIRA

**MODELAGEM E DESIGN DE SISTEMAS DE
SERVIÇO PARA AUTOMAÇÃO**

Tese apresentada à Escola Politécnica
da Universidade de São Paulo para
obtenção do Título de Doutor em Engenharia.

São Paulo
2013

VALTER CASTELHANO DE OLIVEIRA

**MODELAGEM E DESIGN DE SISTEMAS DE
SERVIÇO PARA AUTOMAÇÃO**

Tese apresentada à Escola Politécnica
da Universidade de São Paulo para
obtenção do Título de Doutor em Engenharia.

Área de Concentração:

Engenharia de Controle e Automação
Mecânica

Orientador:

Prof. Dr. José Reinaldo Silva

São Paulo
2013

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 4 de agosto de 2013.

Assinatura do autor

Assinatura do orientador

FICHA CATALOGRÁFICA

Oliveira, Valter Castelhana de
Modelagem e design de sistemas de serviço para automação/ V.
C. Oliveira. — ed. rev. — São Paulo, 2013.
232 p.

Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e Sistemas Mecânicos.

1. Serviços(Sistemas;Automação) 2. Sistemas de Informação 3. Engenharia(Modelagem;Design) 4. Engenharia de Requisitos I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e Sistemas Mecânicos. II. t.

Para a minha família, Eliana, Letícia e Vitor.

AGRADECIMENTOS

Ao Prof. Dr. José Reinaldo Silva pela diretriz, confiança e inestimáveis sugestões que nortearam este trabalho.

À minha família, Eliana, Letícia e Vitor, pelo apoio constante e compreensão nos momentos que trocamos nossa convivência pela minha dedicação a este trabalho.

A todos os colegas do D-Lab e da UEA pelo trabalho realizado em equipe, colaboração e compartilhamento de informações.

Aos colegas da FATEC pela constante discussão acadêmica, em especial ao Prof. Dr. Luiz Antonio Daniel, diretor da Fatec Indaiatuba.

A toda a equipe da FacTI pelo apoio e auxílio na elaboração de propostas e na realização dos projetos.

Ao Departamento de Engenharia Mecatrônica e Sistemas Mecânicos da escola Politécnica da USP pela oportunidade de realização deste trabalho e aos funcionários deste departamento pela disponibilidade e constante atenção.

RESUMO

O início deste século foi marcado pela mudança de paradigma na economia e nos processos produtivos, migrando de uma orientação a bens materiais para uma orientação a serviço. Ao mesmo tempo, os processos de automação da manufatura e integração de sistemas estão sofrendo alteração, onde modelos clássicos orientados a produto estão sendo substituídos por modelos sustentados por sistemas de informação (eventualmente cognitivos). A tese central deste trabalho é que a abordagem orientada a serviço deve ser baseada na engenharia de sistemas, com sistemas de informação atuando como elementos integradores automatizados do processo de co-criação dos serviços. Neste trabalho são analisadas propostas de formalização e fundamentação (teórica e prática) do processo de design de sistemas de serviço que sigam esta nova tendência, resultando em elementos integradores automatizados. É apresentado um *framework*, chamado SoftDiss, para especificação de sistema de informação de serviço, orientado a modelos, que provê recursos para os processos de eliciação, modelagem e análise de requisitos, baseado em métodos semi-formais (UML e SOMF) e formais (SysML e Petri Nets), visando antecipar a formalização da especificação e contemplar os diversos *viewpoints*. O uso do SoftDiss mostra que a utilização de melhores práticas, ferramentas comerciais e métodos formais, tendo como objetivo co-criação de valor, neste caso, entre desenvolvedores humanos e os sistemas incluídos no processo de design, viabilizam antecipar a formalização e contemplar os diversos *viewpoints* de requisitos. O SoftDiss é aplicado a três casos com estrutura distinta: o primeiro onde a base tecnológica é um sistema *Smart Grid* urbano, o segundo associado a projetos desenvolvidos em laboratórios de pesquisa e desenvolvimento, e o terceiro dedicado aos serviços associados à agricultura de precisão. A diversidade de tipos de serviço deste conjunto mostra a flexibilidade do SoftDiss que é associado ao conceito de serviço e não ao tipo, função ou nicho de aplicação.

Palavras-chave: Serviços (Sistemas; Automação). Sistemas de Informação. Engenharia (Modelagem; Design). Engenharia de Requisitos.

ABSTRACT

The beginning of this century was marked by a paradigm shift in the modeling and design of processes, which moved from goods-dominant to a service-dominant approach. At the same time, manufacturing automation and integration are evolving, opening the possibility for classical models, oriented to products has being replaced by service models, supported by information systems (eventually cognitive). The thesis of this work is that the service oriented approach should be based on systems engineering, with information systems acting to integrate and automate service co-creation. First of all some proposals are considered to formalize and fundament (from a theoretical and practical point of view) the design process of these new service systems and how they turn in key elements for integration and automation. In the following we introduce an framework called SoftDiss for specifying information systems service, model oriented, that provides resources to the processes of elicitation, requirements analysis and modeling, based on semi-formal (SOMF and UML) and formal (SysML and Petri Nets) methods which can anticipate the formal specification, while addressing different viewpoints. The use of SoftDiss shows that using best practices, business tools and formal methods to co-create value - in this case involving human developers and machine systems included in the design process, will lead to the anticipation and a formal representation to requirements viewpoints. SoftDiss is applied to three distinct case studies: the first where the basic technology from an urban Smart Grid, the second associated with projects developed in research laboratories and development, and the third dedicated to services associated with precision agriculture. The diversity of service types shows the flexibility of SoftDiss which is associated with the concept of service and not to the kind, function or application domain.

Keywords: Services (Systems; Automation). Information Systems. Engineering (Modeling; Design). Requirements Engineering.

LISTA DE ILUSTRAÇÕES

1	Metamorfose no nível de abstração do serviço (BELL, 2008) . . .	38
2	<i>Service-Oriented Modeling Framework</i> (SOMF) (BELL, 2008) . . .	39
3	Atividades e artefatos OOSEM (ESTEFAN, 2007)	47
4	Processos ISO/IEC 15288 (HASKINS, 2006)	49
5	Estrutura do SoftDiss	60
6	Visão detalhada do SoftDiss	62
7	Estrutura de pacotes do SoftDiss	63
8	Processo de modelagem dos requisitos	66
9	Diagrama de requisito associado a <i>viewpoint</i>	67
10	<i>Naked Objects</i> associados à indústria agrícola	67
11	Resultado da análise de compatibilidade de <i>viewpoints</i>	69
12	Modelo de atributos de serviço	74
13	Árvore de decisão dos serviços conceituais	75
14	Modelo de associação conceitual	76
15	Modelo de análise contextual	78
16	Modelo de Análise Estrutural	79
17	Modelo de Integração de Negócio	80
18	Exemplo de organização do modelo	83

19	Processo de especificação e design OOSEM (FRIEDENTHAL; MORE; STEINER, 2009)	84
20	Requisitos de domínio (<i>as-is</i>)	85
21	Domínio operacional (<i>as-is</i>)	86
22	Especificação <i>black-box</i> do sistema alvo	87
23	Exemplo de Matriz de relacionamento de alocação	90
24	Modelo Lógico de Transação de Serviço	91
25	Rede GHENeSys construída a partir do Diagrama de Estados	92
26	Implementação do SoftDiss utilizando MDG	95
27	Estrutura de pacotes SoftDiss para <i>Smart Grid</i>	106
28	Pacotes do modelo de requisitos do <i>Smart Grid</i>	107
29	Requisitos associados ao IntelliGrid	108
30	Modelo de atributos do <i>Smart Grid</i>	109
31	Árvore de decisão conceitual do <i>Smart Grid</i>	110
32	Associação Conceitual do <i>Smart Grid</i>	112
33	Análise contextual do serviço RTP	113
34	Análise estrutural dos serviços RTP e AMR	114
35	Integração de negócio dos serviços RTP e AMR	115
36	Diagrama de blocos do SIS DIC	117
37	Diagrama de estado do IDC	117
38	Rede GHENeSys construída a partir do diagrama de estados	118
39	Invariantes lugar (esquerda) e transição (direita)	120

40	Processos de negócio do D-Lab	122
41	Linhas de pesquisa e produtos do D-Lab	123
42	<i>Viewpoints</i> de requisitos do D-Lab	126
43	<i>Naked Objects</i> associados aos membros do D-Lab	126
44	Diagrama de requisitos dos membros internos do D-Lab	127
45	<i>Naked Objects</i> dos membros externos do D-Lab	128
46	Organização do Modelo DesEnv do D-Lab	129
47	Modelo de atributos do D-Lab	130
48	Árvore de decisão dos serviços conceituais do D-Lab	131
49	Serviços resultantes do ConEnv e AnaEnv para o D-Lab	132
50	Análise das Necessidades dos <i>stakeholders</i> (FRIEDENTHAL; MO- ORE; STEINER, 2009)	133
51	Estrutura do domínio operacional <i>as-is</i> do D-Lab	134
52	Diagrama paramétrico de análise de visibilidade do D-Lab	135
53	Requisitos e objetos associados aos <i>stakeholders</i> do D-Lab	135
54	Domínio operacional <i>to-be</i> do SIS D-Lab	136
55	Matriz alocação <i>to-be</i> e serviços do D-Lab	137
56	Casos de uso de negócio do D-Lab	138
57	Atividades da "submissão de artigos" do D-Lab	139
58	Especificação <i>black-box</i> do Site D-Lab	140
59	Requisitos da indústria de equipamentos agrícolas	143
60	<i>Naked Objects</i> da indústria de equipamentos agrícolas	143

61	Requisitos dos proprietários concessionários	144
62	<i>Naked Objects</i> dos proprietários concessionários	144
63	Requisitos dos agricultores	145
64	<i>Naked Objects</i> dos agricultores	145
65	Acoplamento dos <i>viewpoints</i> de requisitos do agronegócio . . .	146
66	Matriz de análise de compatibilidade industria e concessionários	147
67	Modelo do ambiente de desenvolvimento DesEnv do agronegócio	148
68	Modelo de atributos de serviço do agronegócio	149
69	Árvore de decisão dos serviços conceituais do agronegócio . . .	150
70	Modelo de associação conceitual do agronegócio	151
71	Análise contextual do serviço qualificação RH	152
72	Modelo de análise estrutural do serviço formação de talentos . .	153
73	Distribuição geográfica do agronegócio	154
74	Integração do negócio do agronegócio	155
75	Serviços do agronegócio	155
76	Alocação requisitos da industria e serviços	156
77	Estrutura do domínio operacional <i>as-is</i> do agronegócio	157
78	Requisitos de negócio do serviço de manutenção	157
79	Domínio operacional <i>to-be</i> do agronegócio	158
80	Requisitos operacionais do sistema de interesse do agronegócio	158
81	SoftDiss implementado no Enterprise Architect	161
82	<i>Smart Grid</i> - Estrutura de Pacotes	186

83	<i>Smart Grid</i> - Ambiente Requisitos	187
84	<i>Smart Grid</i> - Atores de Negócio	187
85	<i>Smart Grid</i> - Requisitos de Negócio	188
86	<i>Smart Grid</i> - Especificações Intelligrid	189
87	<i>Smart Grid</i> - <i>Naked Objects</i> Baixa Tensão	190
88	<i>Smart Grid</i> - UC Centro de Supervisão	190
89	<i>Smart Grid</i> - UC Unidade Telesupervisão	191
90	<i>Smart Grid</i> - UC Unidade de Distribuição	191
91	<i>Smart Grid</i> - UC Indicador Digital de Consumo	192
92	<i>Smart Grid</i> - Modelo de Atributos de Serviço	193
93	<i>Smart Grid</i> - Árvore de Decisão	193
94	<i>Smart Grid</i> - Associação Conceitual	194
95	<i>Smart Grid</i> - Análise Contextual	195
96	<i>Smart Grid</i> - Análise Estrutural	195
97	<i>Smart Grid</i> - Integração de Negócio	196
98	<i>Smart Grid</i> - Controle do Indicador Digital de Consumo	196
99	<i>Smart Grid</i> - Máquina Estado Centro de Supervisão	197
100	<i>Smart Grid</i> - Máquina Estado Unidade Supervisão	197
101	<i>Smart Grid</i> - Máquina Estado Unidade Distribuição	198
102	<i>Smart Grid</i> - Máquina Estado Indicador Digital Consumo	198
103	D-Lab - Estrutura de Pacotes	199
104	D-Lab - Página Inicial	200

105	D-Lab - <i>Viewpoints</i> Requisitos	200
106	D-Lab - Objetos membros internos	201
107	D-Lab - Requisitos membros internos	201
108	D-Lab - Objetos membros externos	202
109	D-Lab - Requisitos membros externos	202
110	D-Lab - Objetos Colaborador	202
111	D-Lab - Requisitos Colaborador	202
112	D-Lab - Objetos empresa parceira	203
113	D-Lab - Requisitos empresa parceira	203
114	D-Lab - Objetos processos de negócio	203
115	D-Lab - Requisitos processos de negócio	203
116	D-Lab - Organização do DesEnv	204
117	D-Lab - Diagrama de Serviços	205
118	D-Lab - Diagrama de Consumidores	205
119	D-Lab - Diagrama de Atributos	206
120	D-Lab - Árvore Decisão	206
121	D-Lab - Associação Conceitual	207
122	D-Lab - Análise Contextual	207
123	D-Lab - Análise Estrutural	208
124	D-Lab - Integração de Negócio	208
125	D-Lab - Guia: Especificação e Design do Sistema (FRIEDENTHAL; MOORE; STEINER, 2009)	209

126	D-Lab - Guia: Análise das Necessidades dos <i>Stakeholders</i> (FRIEDENTHAL; MOORE; STEINER, 2009)	210
127	D-Lab - Guia: Análise dos Requisitos do Sistema (FRIEDENTHAL; MOORE; STEINER, 2009)	211
128	D-Lab - <i>As-is</i> : Requisitos Operacionais	212
129	D-Lab - <i>As-is</i> : Estrutura Operacional	212
130	D-Lab - <i>To-be</i> : Organização do Modelo	213
131	D-Lab - Ciclo de vida do SIS D-Lab	213
132	D-Lab - <i>To-be</i> : Requisitos Operacionais	214
133	D-Lab - <i>To-be</i> : Estrutura Operacional	214
134	D-Lab - <i>To-be</i> : Casos de Uso Operacionais	215
135	D-Lab - <i>To-be</i> : Atividade UC Programa Pós-Graduação	216
136	D-Lab - <i>To-be</i> : Atividade UC Relatório Técnico	217
137	D-Lab - <i>To-be</i> : Atividade UC Submissão Artigo	218
138	D-Lab - <i>To-be</i> : Paramétrico Análise de Viabilidade	219
139	D-Lab - <i>To-be</i> : Especificação Caixa-Preta do Sistema	219
140	Agro - Estrutura de pacotes	221
141	Agro - Página Inicial	222
142	Agro - <i>Viewpoints</i> Requisitos	222
143	Agro - Objetos Proprietário Concessionário	222
144	Agro - Requisitos Proprietário Concessionário	223
145	Agro - Objetos Indústria	223

146	Agro - Requisitos Indústria	223
147	Agro - Objetos Agricultor	224
148	Agro - Requisitos Agricultor	224
149	Agro - Compatibilidade dos <i>Viewpoints</i> de Requisitos	224
150	Agro - Organização do DesEnv	225
151	Agro - Diagrama de Serviços	226
152	Agro - Diagrama Consumidores	226
153	Agro - Diagrama de Atributos	227
154	Agro - Arvore Decisão	227
155	Agro - Associação Conceitual	228
156	Agro - Análise Contextual	228
157	Agro - Análise Estrutural	229
158	Agro - Integração Negócio Geográfica	229
159	Agro - Integração Negócio Estrutural	230
160	Agro - <i>As-is</i> : Requisitos Operacionais	230
161	Agro - <i>As-is</i> : Estrutura Operacional	230
162	Agro - <i>As-is</i> : Requisitos Operacionais	231
163	Agro - <i>To-be</i> : Organização do Modelo	231
164	Agro - <i>To-be</i> : Requisitos Operacionais	232
165	Agro - <i>To-be</i> : Estrutura Operacional	232

LISTA DE TABELAS

1	Características dos serviços (LOVELOCK; WRIGHT, 2002)	30
2	Fundamentos conceituais da lógica S-D (LUSCH; VARGO; WESSELS, 2008)	33
3	Premissas da concepção do SoftDiss	58
4	Atendimento às premissas de concepção do SoftDiss	97

LISTA DE ABREVIATURAS E SIGLAS

AnaEnv	<i>Analysis Environment</i>
ANEEL	Agência Nacional de Energia Elétrica
CIM	<i>Computation Independent Model</i>
ConEnv	<i>Conceptual Environment</i>
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
CORE	<i>Controlled Requirement Expression</i>
D-Lab	DesignLab - Universidade de São Paulo
DesEnv	<i>Design Environment</i>
EA	<i>Enterprise Architect</i>
EIA	<i>U.S. Energy Information Administration</i>
FACTI	Fundação de Apoio à Capacitação em Tecnologia da Informação
ForEnv	<i>Formal Environment</i>
G-D	<i>Good dominant logic</i>
IEC	<i>International Electrotechnical Commission</i>
INCOSE	<i>The International Council on Systems Engineering</i>
ISO	<i>International Organization for Standardization</i>
LogEnv	<i>Logical Environment</i>
ManEnv	<i>Managerial Environment</i>
MBSE	<i>Model Based Systems Engineering</i>

MDA	<i>Model-driven architecture</i>
MDG	<i>Model Driven Generation</i>
MOF	<i>MetaObject Facility</i>
OMG	<i>Object Management Group</i>
OOSEM	<i>Object-Oriented Systems Engineering Method</i>
PIM	<i>Platform Independent Model</i>
PN	Processos de Negócio
PSM	<i>Platform Specific Model</i>
ReqEnv	<i>Requirement Environment</i>
S-D	<i>Service dominant logic</i>
SADT	<i>Structured Analysis and Design Technique</i>
SBS	<i>Service breakdown structure</i>
SoftDiss	<i>Service-Oriented Framework to the Design of Information System Service</i>
SOMF	<i>Service Object Modeling Framework</i>
SI	Sistemas de informação
SIS	Sistema de informação de serviço
SSME	<i>Service Science, Management and Engineering</i>
SysML	<i>System Modeling Language</i>
TI	Tecnologia da Informação
TIC	Tecnologia da Informação e Comunicação
UML	<i>Unified Modeling Language</i>

UEA	Universidade Estadual do Amazonas
USP	Universidade de São Paulo
VORD	<i>Viewpoint-oriented System Denition</i>
VORV	<i>Viewpoint-oriented Requirements Validation</i>
VOSE	<i>Viewpoint-oriented System Engineering</i>
VSA	<i>Viable System Approach</i>

SUMÁRIO

1	Introdução	20
1.1	Motivação	21
1.2	Resumo das contribuições	25
1.3	Metodologia	26
1.4	Organização do trabalho	27
2	Revisão da Literatura	29
2.1	Desenvolvimento de Sistemas de Serviço	29
2.2	Sistema de Informação de Serviço	40
2.3	Especificação Formal de Sistemas	50
2.4	Síntese do Capítulo	54
3	SoftDiss	55
3.1	Premissas da Fase de Requisitos	55
3.2	Estrutura do SoftDiss	57
3.3	<i>Requirement Environment</i> - ReqEnv	64
3.4	<i>Design Environment</i> - DesEnv	69
3.4.1	<i>Conceptual Environment</i> - ConEnv	72
3.4.2	<i>Analysis Environment</i> - AnaEnv	76
3.4.3	<i>Logical Environment</i> - LogEnv	81

3.5	<i>Formal Environment - ForEnv</i>	88
3.6	<i>Managerial Environment - ManEnv</i>	92
3.7	Síntese do Capítulo	96
4	Resultados Obtidos	99
4.1	Projeto de SIS <i>Smart Grid</i>	101
4.1.1	SoftDiss aplicado ao <i>Smart Grid</i> Urbano	105
4.2	Projeto Design Lab	121
4.2.1	SoftDiss aplicado ao D-Lab	125
4.3	Projeto Agronegócio	140
4.3.1	SoftDiss aplicado ao Serviço de Manutenção	142
4.4	Implementação SoftDiss	159
4.5	Síntese do Capítulo	161
5	Considerações Finais	165
5.1	Contribuições	167
5.2	Trabalhos futuros	168
	Referências	171
	Apêndice A – Modelos SoftDiss	179
A.1	SoftDiss - <i>Smart Grid</i>	179
A.1.1	<i>Smart Grid</i> de Baixa Tensão no Brasil	182
A.2	SoftDiss - SIS D-Lab	199

A.3 SoftDiss - Agronegócio	220
--------------------------------------	-----

1 INTRODUÇÃO

A tese central deste trabalho é que a abordagem orientada a serviço deve ser sustentada pela engenharia de sistemas, com sistemas de informação atuando como elementos integradores automatizados do processo de co-criação dos serviços. Neste trabalho são analisados propostas de fundamentação teórica para a ciência de serviço, o processo de engenharia de requisitos, a influência dos *viewpoints* no processo de especificação de sistema de informação de serviço (SIS), técnicas de gerenciamento do processo de desenvolvimento de sistemas e técnicas de especificação formal. Um *framework* orientado a modelos para especificação de SIS é apresentado, visando antecipar a formalização da especificação e contemplar os diversos *viewpoints* de requisitos.

O *framework* de especificação demonstra que a utilização de melhores práticas, ferramentas comerciais e métodos formais tendo como objetivo co-criação de valor, neste caso, entre desenvolvedores humanos e os sistemas incluídos no processo de design, viabilizam antecipar a formalização e contemplar os diversos *viewpoints* de requisitos. São apresentados os resultados da aplicação do *framework* na especificação de três diferentes tipos de sistemas de serviço, mostrando a abrangência do *framework* de especificação.

Este trabalho é único como proposta de processo de especificação, sustentação, manutenção e evolução de SIS, viabilizado pela orientação a mo-

delos, gerenciamento dos múltiplos *viewpoints* de requisitos e antecipação da formalização, que favorece o sucesso das etapas subsequentes do ciclo de vida do desenvolvimento de sistemas.

1.1 Motivação

A distribuição da mão de obra economicamente ativa (e a própria divisão da economia) pode ser dividida em três setores (e em três fases distintas, a idade média, o período depois da revolução industrial e a era moderna): o primário, onde desde a idade média a mão de obra se concentra predominantemente na agricultura; o secundário, que poderia ser identificado como sendo o das artes manuais, ligadas à produção de teares rústicos, moinhos, destiladores, assim como mobiliário e equipamento de guerra; e o terciário, que na idade média abrangia o artesanato, as artes liberais incluindo a medicina rudimentar. Após a revolução industrial o setor secundário passou a ser identificado como o setor industrial e o terceiro setor assumiu o papel de “*front end*” desse setor, encarregado das vendas e do convencimento dos potenciais consumidores (“*marketing*”). A evolução contínua destes setores levaram a um crescimento no setor de serviço (especialmente após o final do século XIX), atraindo pessoas e investimentos especialmente para o comércio, o que acabou por fazer com que o setor terciário ganhasse uma dinâmica própria, crescendo de forma independente do setor secundário.

Na era moderna, o processo de automação desencadeado tanto no setor primário (agrícola) como no setor secundário (industrial), e fortemente explorado nos dias atuais, resultou na alavancagem do setor terciário (serviço), embora os processos automatizados tenham marcado mais presença no setor secundário (no século XX). A chegada da automação no setor terciário (final do século XX e período atual) ativa novos processos de negócio que exploram

o serviço - não só como uma função complementar dos bens tangíveis - estão tomando vulto e surgem derivações do setor de serviço: o setor quaternário associado aos serviços de informação e conhecimento, e o setor quinquário, com os serviços sem uma relação de troca direta ou associação com lucro ((KELLERMAN, 1985)). Assim, os serviços se tornam uma tendência em todo o mundo, sendo que a economia dos países desenvolvidos está fortemente direcionada para serviços, chegando, em alguns casos, a responder por 80% do PIB destes países ((BITNER; BROWN, 2008; MOUSSA; TOUZANI, 2010)).

Os processos associados aos serviços ainda demandam a modernização dos processos de automação, visando o consequente aumento de escala e qualidade dos serviços mas dependem de um avanço na modelagem e design destes mesmos sistemas.

O início deste século foi marcado por uma potencial mudança de paradigma na modelagem e design dos processos produtivos, migrando de uma abordagem dominada por bens materiais (*goods-dominant*) para uma abordagem dominada por serviços (*service-dominant*) (LUSCH; VARGO; WESSELS, 2008; MOUSSA; TOUZANI, 2010). Na verdade isto foi o reflexo de uma mudança na própria economia e nos processos de negócio (e evidentemente nos sistemas de apoio e de automação), dado que estava acontecendo paralelamente a um aumento substancial dos serviços como base do negócio (transporte, entretenimento, educação, informação, segurança, etc.) (KIM; NAM, 2009).

A entrega de produtos passa a ser substituída pela busca de soluções, o prestígio em possuir um bem tangível se transforma no prazer de utilizá-lo, e a venda de produtos dá lugar à construção de relacionamentos, ou seja, temos uma migração de negócios dominados por bens materiais (*goods dominant logic* ou lógica G-D) para negócios dominados por serviço (*service dominant logic* ou lógica S-D), através da substituição dos meios de produção

de bens materiais para a produção generalizada de serviço (SPOHRER et al., 2008; LUSCH; VARGO; WESSELS, 2008; VARGO; AKAKA, 2009).

Os sistemas de serviço passam a ser o meio de sustentação da lógica S-D, sendo caracterizados pela configuração dinâmica de pessoas, de tecnologias, de organizações e, também, das informações compartilhadas entre estes elementos, que permite aos fornecedores criar e entregar valor aos seus clientes. O conceito chave associado a sistemas de serviço é que estes integram para a co-criação de valor envolvendo fornecedores e clientes em um mesmo processo (SPOHRER et al., 2007; MAGLIO et al., 2009).

Considere-se que os sistemas de serviço vêm se tornando foco de atenção pelo seu aspecto integrador de inovação tecnológica, automação e controle de forma ubíqua. Entretanto, a automação e abstração que estes sistemas provêm dependem tanto da inovação tecnológica quanto das soluções adotadas para o gerenciamento de recursos, da informação, e das restrições inerentes à aplicação. Portanto, as técnicas clássicas aplicadas ao desenvolvimento de sistemas de informação (SI) devem ser revistas para se adequar aos novos desafios propostos para o desenvolvimento destes novos sistemas.

A convergência entre sistemas de serviço e SI ocorre justamente em um período crítico para estes últimos, cuja flexibilidade e capacidade de integração – fundamentais em processos de inovação e automação – estão diretamente ligadas à identificação da sua funcionalidade em um ambiente heterogêneo, atendendo a múltiplas funções e a uma gama variada de usuários e *stakeholders* (STAIR; REYNOLDS, 2010).

O futuro aponta para uma fase de mudança de paradigma, passando do foco em bens para o foco em serviços e a engenharia é a base para esta mudança de paradigma. Sendo assim, tem-se como consequência uma demanda por novas técnicas de design de processo e do design de sistemas

inteiramente voltados à funcionalidade e não para as áreas do conhecimento necessárias à sua concepção. Assim, o design mecatrônico, de base nitidamente interdisciplinar, é também o modelo de referência para a engenharia de serviços. Este design associa várias técnicas e métodos tendo sempre como base a funcionalidade.

A crescente importância dos serviços e da tecnologia da informação (TI) na economia globalizada favorece o aumento de pesquisas na área de SI, com especial atenção aos fundamentos do conhecimento gerencial e técnico nesta área emergente do conhecimento (BARDHAN et al., 2010). O desenvolvimento de SI requer a participação de pessoas com diversos perfis e expectativas, onde o usuário final espera um resultado que atenda às suas necessidades; os financiadores esperam um retorno adequado para o seu investimento; e a equipe de desenvolvimento espera desenvolver um artefato de qualidade, no prazo e custo estipulados e que atenda às reais necessidades dos usuários e financiadores.

Portanto, o equilíbrio e harmonização dos *viewpoints* são a base para se ter sistemas de serviço bem sucedidos e bem integrados no processo produtivo. Outro aspecto igualmente importante é a harmonia entre os sistemas artificiais, as pessoas, as máquinas e os sistemas automatizados. Entretanto, estes dois aspectos não podem ser inseridos nos sistemas modernos sem métodos de design que consigam orientar o processo e garantir o objetivo final.

A proposta do presente trabalho é propor um método para design de sistemas de serviço na perspectiva de que estes sistemas sejam automatizados, e que resulte em um *framework* de uso prático.

1.2 Resumo das contribuições

As contribuições deste trabalho são as seguintes:

1. Uma visão geral sobre as recentes iniciativas para compor o processo de especificação de sistemas de serviço, base para a ciência de serviço, contribuindo para a elaboração de conhecimento específico sobre sistemas de serviço e para a elaboração de *racionales* sobre projetos nesta área.
2. Criação de uma disciplina de projeto orientada a serviço e destinada à especificação de SIS, suportando as fases iniciais do desenvolvimento, incluindo a eliciação, modelagem e análise de requisitos, através da configuração dinâmica e cognitiva de tecnologias, pessoas, sistemas externos e das informações compartilhadas entre estes elementos. A premissa adicional é que estes SIS (Sistemas de Informação de Serviço) são a base de vários processos de automação que envolvem não apenas máquinas mas também pessoas.
3. Aplicação da disciplina de projeto criada (e implementada em um sistema comercial) ao processo de especificação de três SIS, devidamente escolhidos por terem características diferentes. Um deles é representante do setor quaternário, cujo serviços básicos incluem alto nível de automação como base do seu funcionamento, o outro tem alto nível de co-criação, mas ainda pertencente aos setores quaternário e quinário, e um último pertence ao setor primário (pelo menos teoricamente) embora com alto grau de automação, fugindo portanto dos sistemas primários clássicos.

1.3 Metodologia

Considerando os objetivos do trabalho, o delineamento da pesquisa é baseado em pesquisa-ação (TRIPP, 2005), cujo ciclo de pesquisa aprimora a prática através da oscilação sistêmica entre ação prática e a investigação teórica. A pesquisa-ação conta com processos cíclicos de planejamento, implementação, descrição e avaliação das mudanças para melhoria da prática.

No início do desenvolvimento do trabalho foi realizada a coleta de dados associada a projetos de design de sistemas de serviço, tendo como base projetos envolvendo o fornecimento automatizado de energia (*Smart Grid*), e seus modelos de referência, foi feito um levantamento de dados e uma avaliação de propostas com pesquisadores e profissionais de diversas instituições (D-Lab, FacTI e UEA), e, finalmente, uma pesquisa bibliográfica quanto às propostas de fundamentação teórica para a ciência de serviços.

O tratamento e análise dos dados iniciais resultaram na primeira versão do *framework* para desenvolvimento de SIS (SoftDiss) e a sua aplicação na especificação de SIS para um *Smart Grid* de baixa tensão, especificamente, para o controle de perdas, utilizando o modelo de referência Intelligrid.

Os dados obtidos neste primeiro ciclo da pesquisa foram analisados e um novo ciclo realizado, resultando no aprimoramento do SoftDiss, com a integração de novas técnicas, métodos e ferramentas já conhecidas (Volere, *Naked Objects* e OOSEM), concedendo ao próprio SoftDiss características de um sistema de serviço, fornecendo aos desenvolvedores suporte na tarefa de modelagem e design de SIS. Esta nova versão foi aplicada a um ambiente acadêmico de serviço, especificamente ao D-Lab: trata-se de um sistema de serviço corporativo, dedicado ao suporte das atividades de pesquisa e desenvolvimento realizadas pelo D-Lab. O novo modelo vem sendo implementado

no D-Lab. Este segundo nível tem como referência sistemas com uma base formal (ou pelo menos com restrições já estabelecidas), mas nenhum subsídio, experiência prática, ou modelo de referência (ao contrário do caso anterior).

Um terceiro ciclo de pesquisa passa a ser tratado buscando a validação do SoftDiss através da sua aplicação na modelagem e especificação de SIS associado ao agronegócio, tendo como foco a agricultura de precisão especificamente na área de concessionárias de equipamentos agrícolas, permitindo a aplicação do SoftDiss a uma situação real, resultando em um conjunto de dados que permite aferir a aplicabilidade desta pesquisa na eliciação, modelagem e análise de requisitos de SIS. Neste último caso a aplicação é um sistema de serviço sem formalização, sem modelo de referencia mas com uma base significativa de casos práticos, e que inclui o setor terciário tradicional, usualmente identificado com serviço (e que não contém automação, além do uso reflexo de alguns dispositivos).

1.4 Organização do trabalho

Este primeiro capítulo contém as motivações, as contribuições e a metodologia aplicada a este trabalho de pesquisa.

O segundo capítulo apresenta a revisão dos principais conceitos sobre ciência de serviço, as recentes iniciativas de desenvolvimento de fundamentos acadêmicos e métodos científicos destinados a oferecer melhorias na eficiência e na qualidade dos sistemas de serviço, a tendência dos novos SI como agentes integradores da inovação tecnológica, e finalmente, os conceitos associados à antecipação da formalização no processo de especificação dos SI e várias possibilidades de representação.

No terceiro capítulo é apresentado o *Framework* Orientado a Serviço para Especificação de Sistemas de Informação de Serviço - SoftDiss (*Service-Oriented Framework to the Design of Information System Service*).

O quarto capítulo apresenta a abrangência do SoftDiss e sua aplicabilidade prática a diferentes tipos de sistemas de serviço.

O quinto e último capítulo apresenta as considerações finais do trabalho, as conclusões e propostas de trabalhos futuros.

2 REVISÃO DA LITERATURA

Esta capítulo apresenta os principais conceitos sobre Ciência de Serviço, as recentes iniciativas para a fundamentação conceitual e desenvolvimento de métodos científicos melhorar a eficiência e a qualidade dos sistemas de serviço, especialmente as propostas baseadas no uso de sistemas de informação (SI) como agentes integradores da inovação tecnológica. Esta demanda remete à melhoria do processo de modelagem e design, e portanto também será discutido neste capítulo os conceitos associados à representação formal e validação de modelos de SIS.

2.1 Desenvolvimento de Sistemas de Serviço

Existem diversos conceitos e significados associados ao termo serviço, dependendo da perspectiva, da época e do objetivo considerado. De acordo com o Dicionário Aurélio (HOLANDA, 2010), serviço é uma “atividade econômica que não resulta em produto tangível, em contraste com a produção de mercadorias”, o que destaca a visão clássica de serviço, ou seja, serviço englobando tudo que não é tangível. Na perspectiva do cliente e com um enfoque a processos, Grönroos (2004) define serviço como uma atividade ou um conjunto sequencial de atividades, que geralmente ocorre durante a interação entre os clientes e fornecedores de serviço, visando fornecer uma solução para um problema do cliente. De acordo com Vargo e Lusch (2006), serviço é

a aplicação de recursos visando o benefício de terceiros e, finalmente, considerando especificamente a lógica dominante de serviço (SPOHRER et al., 2008; LUSCH; VARGO; WESSELS, 2008; VARGO; AKAKA, 2009), serviço é a aplicação de competências, incluindo conhecimento e proficiência, visando o benefício de terceiros, conceito de serviço utilizado neste trabalho de pesquisa.

De acordo com Lovelock e Wright (2002) os serviços podem ser claramente diferenciados dos bens tangíveis, através das características listadas na tabela 1.

Tabela 1: Características dos serviços (LOVELOCK; WRIGHT, 2002)

Característica	Descrição
Intangibilidade	Os serviços não podem ser provados, sentidos, ouvidos ou cheirados antes da compra.
Inseparabilidade	Os serviços são produzidos, entregues e consumidos simultaneamente.
Variabilidade	O mesmo serviço, prestado por pessoas diferentes e/ou para clientes diferentes, apresenta variações quanto ao seu resultado.
Perecibilidade	Os serviços não podem ser estocados.

O termo “serviços” no plural indica a forma tradicional de tratamento de serviços, onde os recursos são estáticos e para que gerem valor precisam ser operados, já a utilização do termo “serviço” no singular indica a nova visão, onde os recursos são ativos, dinâmicos, intangíveis e possuem a capacidade de criar valor (VARGO; LUSCH, 2006; VARGO; AKAKA, 2009).

De acordo com Lovelock e Wright (2002), as características marcantes dos serviços (intangibilidade, inseparabilidade, variabilidade e perecibilidade) que os diferem significativamente dos bens materiais, implicam em um maior

grau de exposição a riscos para os clientes, que após consumirem um serviço comparam a qualidade esperada com a recebida, avaliando-o como de superior qualidade, se o desempenho do serviço estiver acima dos seus níveis, como adequado, se a entrega do serviço estiver dentro da zona de tolerância, e como inadequado, se estiver abaixo.

Considerando as características associadas a serviço, bem como, sua definição, pode-se constatar a dificuldade em se obter uma definição precisa e exata de requisitos a serem atendidos por sistemas dedicados a suportar serviço (sistemas de serviço), pois além de atender as necessidades de diversos envolvidos, o sucesso na realização de um serviço só é aferido após a sua realização, limitando a possibilidade de definição de critérios de aceitação mais precisos e exatos.

Um sistema de serviço é caracterizado pela produção de valor baseado na configuração adequada de pessoas, tecnologias, e outros sistemas de serviço, compartilhando informações e, geralmente, associado à troca econômica (MAGLIO et al., 2009; SPOHRER et al., 2007; IFM; IBM, 2008). Diversos sistemas podem ser considerados sistemas de serviço, como por exemplo, famílias, corporações, fundações, ONGs, órgãos do governo, departamentos nas corporações, cidades e até nações.

Spohrer et al. (2008) define formalmente um sistema de serviço como um sistema aberto que compartilha ou aplica recursos para melhorar o estado de outro sistema e, também, adquire recursos externos para melhorar o seu próprio estado. O conceito chave associado a sistemas de serviço é que estes interagem para a co-criação de valor (SPOHRER et al., 2007).

No princípio de co-criação de valor a ideia de em um negócio um lado produzir valor e o outro lado consumir este valor não faz sentido, pois é enfatizada a ideia de criação conjunta de valor envolvendo duas entidades, onde

uma entidade aplica suas competências e a outra integra estas competências com os demais recursos através da co-criação (MAGLIO et al., 2009).

Na lógica dominante de serviço (lógica S-D) as competências especializadas são aplicadas em benefício do cliente, permitindo a criação de valor através da colaboração entre todas as partes envolvidas incluindo o fornecedor e o cliente. Os bens continuam importantes, mas passam a ser tratados como veículos para a transmissão de recursos dentro dos processos que compõem o negócio (MERZ; HE; VARGO, 2009). De acordo com Lusch, Vargo e Wessels (2008), a lógica S-D está focada nos fundamentos conceituais da tabela 2.

Tabela 2: Fundamentos conceituais da lógica S-D (LUSCH; VARGO; WESSELS, 2008)

Fundamento	Descrição
Recursos ativos	Recursos ativos produzem resultados através da atuação e transformação de outros recursos, e são muitas vezes intangíveis, como conhecimento e competências.
Alocação de Recursos	Criação de valor através da transformação de um potencial recurso em benefícios específicos, contando com três aspectos essenciais: a criação de recursos, a integração de recursos e a remoção de resistências entre eles.
Experiência do cliente	Foco na experiência do cliente direciona o serviço, para que suas necessidades sejam atendidas.
Proposição de valor	Cliente é visto como um integrador de diversos recursos objetivando a criação de valor para a sua organização.
Diálogo	Implica em desenvolver uma comunicação efetiva, baseada em confiança, aprendizado conjunto e adaptabilidade.
Rede de criação de valor	Requer uma redefinição da cadeia de suprimentos buscando obter uma rede de sistemas de serviços.
Aprendizado com as trocas	Enfatizada a valorização da utilidade do serviço e da criação conjunta.
Marketing colaborativo	Cliente é tratado como um parceiro colaborador dentro do processo de marketing.

neering) está associada ao estudo dos sistemas de serviço (LUSCH; VARGO; WESSELS, 2008; SPOHRER et al., 2008; IFM; IBM, 2008). Sendo uma disciplina em formação (NG; MAULL, 2009; ZHAO; PERROS, 2009; LI et al., 2007), a Ciência de Serviço necessita de um estudo sistemático e aprofundado que viabilize a criação de um arcabouço de sustentação, fundamentado teoricamente, adequado para ser aplicado nas mais diversas áreas do conhecimento.

A lógica dominante de serviço ou lógica S-D (LUSCH; VARGO; WESSELS, 2008), mesmo que ainda incipiente, oferece uma base para a teorização, confirmação e refinamento do fundamento teórico da ciência de serviço (VARGO; AKAKA, 2009; MAGLIO et al., 2009; LUSCH; VARGO; WESSELS, 2008).

No trabalho de Barile e Polese (2010) a busca de uma formalização para a Ciência de Serviço se dá através da teoria geral dos sistemas (BERTALANFFY, 1950; BOULDING, 1956), explorando a relação entre a proposta baseada na abordagem de sistemas viáveis (VSA – *Viable System Approach*) proposta por Beer (1984) e o recente avanço na área de Ciência de Serviço, denominado Sistema de Serviço Inteligente (*Smart Service Systems*). VSA é uma teoria interdisciplinar aplicada à observação de fenômenos complexos, com base na teoria dos sistemas, com foco na análise das relações entre entidades socioeconômicas, buscando condições viáveis de interação entre elas. O Sistema de Serviço Inteligente é uma proposta baseada na tecnologia da informação e comunicação (TIC). O principal resultado deste é que o VSA fornece informações valiosas para a concepção e gestão dos sistemas de serviços inteligentes, especialmente quanto a harmonização, governança de sistemas e processos de co-criação bem sucedidos.

Kim e Nam (2009) propõem uma teoria para sistemas de serviço baseada em um procedimento sistemático para entender a natureza dos serviços e construir uma teoria integrada de sistemas de serviço que viabilize a ino-

vação e estimule a produtividade de serviço, provendo fundamentos para o design, produção, entrega, operação, manutenção, monitoramento e aprimoramento dos sistemas de serviço. Os componentes do sistema de serviço são identificados e incluem: cliente, produto, processos de negócio, participantes, informação e tecnologia. A especificação da interação entre os componentes do sistema de serviço é bastante complicada e envolve desde o tipo de conhecimento transmitido, a capacitação dos envolvidos, a troca de conhecimento intangível, até a complexidade advinda de uma rede integrada de recursos, participando para criar valor ao cliente. Finalmente, os objetivos dos sistemas de serviço devem ser categorizados em níveis de qualidade percebidos pelo cliente, produtividade atribuída aos provedores e a inovação do serviço resultante, tanto para clientes como para provedores. A contribuição deste trabalho está na criação de uma base para estudos nas áreas de qualidade, produtividade e inovação de serviço, estabelecendo os componentes e a natureza das interações entre eles, destacando a necessidade de definição clara dos termos e definições na criação da base de uma teoria.

Bardhan et al. (2010), propõem um *framework* para avaliar as principais linhas de pesquisa em SSME, com uma análise multidisciplinar, incluindo SI, ciência da computação, economia, finanças, marketing, gestão de operações e gerenciamento de cadeia de suprimentos. Como resultado da análise estes autores obtêm uma cobertura abrangente, com interpretação das principais questões levantadas; perspectivas teóricas iniciais para pesquisas mais profundas; aplicações para entender melhor as inovações orientadas a serviço; e a ciência de serviço como uma área fundamental para as novas pesquisas da área de SI.

O trabalho de Ostrom et al. (2010), que contou com a participação de acadêmicos atuantes em várias disciplinas e ligados a instituições espalhadas

pelo globo, e com a colaboração de executivos ligados a pelo menos 1000 empresas de pequeno, médio e grande porte, permitiu organizar uma lista contendo as dez maiores prioridades de pesquisa para o desenvolvimento da Ciência de Serviço, agrupadas em três aspectos de negócio:

- Prioridades Estratégicas: fomentar a disseminação e o crescimento dos serviços; melhorar a qualidade de vida através de serviços; criar e manter uma cultura de serviço.
- Prioridades de Desenvolvimento: estimular a inovação dos serviços; melhorar o design de serviços; aperfeiçoar as redes de serviço e cadeias de valor.
- Prioridades de execução: efetivar marcas associadas à venda de serviços; melhorar a experiência do serviço através de criação conjunta com o cliente; medir e aperfeiçoar o valor do serviço.
- Permeando todas as prioridades, utilizar a tecnologia para alavancar o serviço.

Stanicek e Winkler (2010) apresentam um arcabouço para modelar as informações associadas a um sistema de serviço, baseado em modelagem conceitual semântica que pode ser combinado a métodos dirigidos a objetivos (*goal-driven*). A modelagem de sistemas de serviço deve ser concentrada nos relacionamentos e não nos objetos que compõem o sistema, registrando a necessidade (*category #goal*) ou o problema a ser resolvido pelo sistema, os requisitos que identificam os atributos, capacidades, características ou qualidades do sistema que originam valor e utilidade ao usuário. O refinamento dos objetivos de um sistema de serviço pode ser registrado em um instrumento gráfico denominado estrutura analítica de serviço ou *service breakdown structure* (SBS), capaz de indicar como um sub-serviço participa da entrega de um

serviço. Esta proposta de modelagem conceitual semântica oferece uma linguagem de comunicação comum tanto para o provedor de serviço como para o cliente, consumidor de serviço.

De acordo com Michael Bell (BELL, 2008), SOMF – *Service Oriented Modeling Framework* é um método de desenvolvimento de *software* que apresenta uma linguagem de modelagem antropomórfica e holística, aplicável aos vários níveis de abstração requeridos pelas etapas do ciclo de vida de desenvolvimento, incluindo desde a visão de negócio até a realização do design do *software*. SOMF oferece aos analistas, arquitetos, desenvolvedores, modeladores e gerentes duas linhas de atuação durante o processo de desenvolvimento de *software*, um baseado no que deve ser feito (*what to do*) e outro no como deve ser feito (*how to do*), e ambos dirigidos por uma linguagem comum de modelagem orientada a serviço, com as seguintes características:

- Plataforma de desenvolvimento de *software* holística a antropomórfica, independente de linguagem particular de programação ou restrições de tecnologia.
- Práticas de desenvolvimento de *software* com disciplina de modelagem e linguagem que propiciam uma visão holística de todas as entidades de *software* da organização, como sistemas legados, serviços, infraestrutura ou processos de negócio.
- Análise, design e arquitetura dirigida por modelos que fomentam a reusabilidade.
- Práticas para gerenciamento do portfólio de serviços e do ciclo de vida do *software*.
- Notação de fácil aprendizado e utilização.

- Conjunto definido de *viewpoints* de modelagem: conceitual, análise, design, integração de negócio e arquitetura.
- Métodos para fortalecer os vínculos entre negócio e tecnologia da informação (TI).
- "Boas" práticas para promover agilidade nos negócios, reuso de ativos e padronização de linguagem de modelagem.
- Transparência na modelagem quanto aos processos de negócio e requisitos tecnológicos viabilizando decisões de investimento e justificativa para arquiteturas.

Os serviços devem evoluir e proporcionar valor a seus consumidores, contribuindo significativamente para os objetivos organizacionais, entretanto estes serviços estão sujeitos a alterações e evoluções, ou até mesmo extinção, sendo assim os serviços podem necessitar serem revisados e retornar para um novo design. Este processo repetitivo e cíclico no qual um serviço vai e volta do estado de design (desenvolvimento do serviço) para o de execução (produção do serviço) é o ciclo de vida do serviço. A Figura 1 representa a metamorfose ocorrida durante o ciclo de vida do serviço quanto ao seu nível de abstração (BELL, 2008).

Figura 1: Metamorfose no nível de abstração do serviço (BELL, 2008)

SOMF (Figura 2) é um *framework* para desenvolvimento de *software* que oferece um conjunto de disciplinas e práticas orientadas a serviço que contribuem para o sucesso do ciclo de vida do desenvolvimento e modelagem do serviço. São seis disciplinas de modelagem orientadas a serviço agrupadas em três ambientes de trabalho, que oferecem artefatos específicos para representar o resultado da modelagem:

1. Ambiente Conceitual, contendo as disciplinas Conceituação de Serviço e Arquitetura Conceitual;
2. Ambiente de Análise, com as disciplinas Análise e Descoberta de Serviço e Integração de Serviço;
3. Ambiente Lógico, agrupando as disciplinas Design de Serviço e Arquitetura Lógica.

Figura 2: *Service-Oriented Modeling Framework* (SOMF) (BELL, 2008)

Nesta seção foram consideradas as propostas de fundamentação teórica

para a Ciência de Serviço. Barile e Polese (2010) abordam a formalização para a Ciência de Serviço através do domínio da teoria geral dos sistemas. Kim e Nam (2009) propõem um procedimento sistemático para entender a natureza dos serviços e para construir uma teoria integrada de sistemas de serviço que viabilize a inovação e estimule a produtividade de serviço, destacando a necessidade de definição clara dos termos e definições na criação da base de uma teoria.

Os trabalhos de Bardhan et al. (2010) e Ostrom et al. (2010), sustentam a utilização de SI como elemento viável para a implementação de sistemas de serviço e a importância do design no processo de desenvolvimento. Stanicek e Winkler (2010) apresentam um arcabouço para modelar as informações associadas a um sistema de serviço, baseado em modelagem conceitual semântica e *goal-driven*. Finalmente, Bell (2008) fornece um *framework* adequado para desenvolvimento de *software*, aplicável aos vários níveis de abstração requeridos pelas etapas do ciclo de vida de desenvolvimento, incluindo desde a visão de negócio até a realização do design do sistema, contando com elementos de rastreabilidade e tratamento de *viewpoints*.

2.2 Sistema de Informação de Serviço

Os sistemas de serviço vêm se tornando foco de atenção pelo seu aspecto integrador de inovação tecnológica, automação e controle de forma ubíqua. Entretanto, a automação e abstração que estes sistemas provêm dependem tanto da inovação tecnológica prevista quanto das soluções adotadas para o gerenciamento de recursos, da informação, e das restrições inerentes à aplicação. Portanto, as técnicas clássicas aplicadas ao desenvolvimento de SI devem ser revistas para se adequar aos novos desafios propostos pela nova Ciência de Serviço (SPOHRER et al., 2008; CHESBROUGH; SPOHRER, 2006).

Os novos SI devem atender requisitos mais sofisticados que permitam a integração dos vários elementos que compõem a empresa, passando a ser um elemento integrador, com funções muito mais elaboradas, exigindo um enquadramento perfeito às necessidades dos sistemas de serviço, coletando informações em várias partes dos processos de negócio (DAVENPORT, 1993; MARSHALL, 2000; AALST, 1998; AALST; HEE, 2002), sempre com o foco no cliente, e entregando as informações nos locais corretos e às pessoas que necessitam da informação.

Os novos SI devem convergir com as necessidades dos sistemas de serviço, entretanto esta convergência ocorre em um momento crítico para os SI, cuja flexibilidade e capacidade de integração, fundamentais em processos de inovação, estão diretamente ligadas à identificação da sua funcionalidade em um ambiente heterogêneo, atendendo a múltiplas funções e a uma gama variada de usuários e *stakeholders* (STAIR; REYNOLDS, 2010).

Sendo assim chega-se a uma situação ambígua, onde a inovação depende intrinsecamente dos SI por um lado, e por outro este mesmo processo de inovação tem sua funcionalidade restrita pela baixa eficiência dos projetos de SI vigente, onde de acordo com *CHAOS Report* (The Standish Group International, 2011) em 2011 apenas 37% dos projetos de desenvolvimento de *software*, incluindo SI, foram bem-sucedidos quanto às premissas de prazo e custo, 21% foram simplesmente abortados, e o restante, 42%, foram aproveitados mas terminaram fora do prazo ou do orçamento ou ambos.

No processo de desenvolvimento de SI são geralmente as etapas iniciais que geram a maior parte dos problemas, mesmo quando as consequências são detectadas em outras fases do desenvolvimento. Paradoxalmente, as etapas iniciais são as menos dispendiosas, por não exigirem grande aquisição de bens e serviços. Estas etapas compreendem a fase de engenharia de

requisitos reconhecida como uma das mais importantes do processo de desenvolvimento (KOTONYA; SOMMERVILLE, 1998). Segundo Carr (2000), o custo para reparar um problema de requisitos quando o sistema já está em produção pode ser até 500 vezes maior do que se o problema fosse detectado e tratado durante a fase de requisitos.

No desenvolvimento de SI, o objetivo da engenharia de requisitos (KOTONYA; SOMMERVILLE, 1998) é a validação dos requisitos pelos *stakeholders*, sendo uma atividade longa, envolvendo um grupo heterogêneo de pessoas que buscam problemas, omissões e ambiguidades no documento de requisitos e sem uma referência documental para ser utilizada como base na validação dos requisitos.

Apesar de não existir um método ideal para a engenharia de requisitos (KOTONYA; SOMMERVILLE, 1998), neste trabalho está sendo utilizado o método Volere, proposto por Robertson e Robertson (2006) e a disciplina de modelagem *Naked Objects* (PAWSON, 2002). *Naked Object* encoraja a especificação de sistemas a partir de objetos comportamentalmente completos, mantendo sempre os dados e o comportamento dos objetos agrupados, e buscando identificar os principais objetos de negócio. O processo de engenharia de requisitos Volere fornece as atividades a serem realizadas para obter a especificação de requisitos. Os requisitos são passíveis de validação e teste, devem possuir um critério de aceitação e são classificados de acordo com os seguintes tipos: requisitos funcionais, que descrevem o que o produto tem de fazer ou que ações processuais devem tomar; requisitos não funcionais, são as propriedades que os sistemas devem ter para atender as funções desejadas; restrições do projeto, são as limitações sobre como o produto deve ser especificado, em geral derivadas da relação do produto com o seu entorno; diretivas do projeto, são as forças associadas ao negócio; e domínio do projeto,

que definem as condições nas quais o projeto será realizado.

Os sistemas de informação de serviço (SIS) devem convergir com características e atributos dos serviços, sendo que o processo de desenvolvimento e manutenção está diretamente ligado à identificação, manutenção e rastreabilidade dos requisitos do sistema em um ambiente heterogêneo, atendendo a múltiplas funções e a uma gama variada de usuários e *stakeholders*, que podem ser representados por pessoas, instituições e até outros sistemas, todos interagindo de forma específica e diferenciada com o sistema, ou seja, cada um possui um *viewpoint* sobre o sistema. Sendo assim no desenvolvimento de SIS deve-se considerar a diversidade dos *viewpoints* (LEITE, 1996; KOTONYA; SOMMERVILLE, 1998; SOMMERVILLE; SAWYER; VILLER, 1997) e a necessidade de garantir a rastreabilidade dos requisitos (LETELIER, 2002; GOTEL; FINKELSTEIN, 1994; TANG; JIN; HAN, 2007) durante todo o processo de desenvolvimento.

De acordo com Kotonya e Sommerville (1996), *viewpoint* é uma coleção de informações apresentada de uma perspectiva particular sobre um determinado sistema, problema, ambiente ou domínio. Cada uma destas perspectivas podem ser associadas aos usuários do sistema, a outros sistemas, aos *stakeholders* e, também, à equipe de desenvolvimento do sistema. Em geral, as informações associadas a cada *viewpoint* são incompletas, sendo os requisitos do sistema obtidos da integração e associação dos diversos *viewpoints*, levando a necessidade de uma atividade destinada a resolução de eventuais conflitos e inconsistências entre as informações oriundas dos diversos *viewpoints*.

Para Leite (1996) existem três áreas principais para a aplicação de *viewpoints* que contribuem para uma efetiva melhoria da produção de SI: reconhecimento de que várias opiniões diferentes devem ser consideradas durante todo o processo de desenvolvimento do sistema; conceito importante para a

representação de especificações, pois oferece um importante arcabouço para implementar análise de consistência e viabilizar um modelo de verificação; e utilização de *viewpoints* como serviços a serem prestados pelo sistema e viabilizando uma estratégia para a especificação e operação do sistema.

Segundo Kotonya e Sommerville (1996) as principais técnicas desenvolvidas para o tratamento de requisitos de sistemas baseadas em *viewpoints* são: SADT (*Structured Analysis and Design Technique*) (Ross & Schoman, 1977), CORE (*Controlled Requirement Expression*) (Mullery, 1979), VOSE (*Viewpoint-oriented System Engineering*) (Finkelstein et al., 1992), VORD (*Viewpoint-oriented System Definition*) (KOTONYA; SOMMERVILLE, 1996) e VORV (*Viewpoint-oriented Requirements Validation*) (Leite & Freeman, 1991).

De acordo com Santos (2002) os métodos acima indicam que os agentes associados aos *viewpoints* devem ser identificados, estruturados, classificados e priorizados, permitindo relacionar os principais agentes e seus respectivos *viewpoints* no processo de levantamento e tratamento dos requisitos do sistema. A validação dos requisitos é apresentada explicitamente apenas no método VORV e integrada ao processo de levantamento dos requisitos.

De acordo com Gotel e Finkelstein (1994) rastreabilidade de requisitos é a capacidade de descrever e seguir a existência de um requisito do sistema, desde a sua origem até a desativação do sistema, passando pelas diversas etapas de desenvolvimento, especificação, implantação e manutenção. Dois tipos de rastreabilidade de requisitos são fundamentais: rastreabilidade da pré especificação de requisitos (pré-RS), associada aos aspectos da produção do requisito; rastreabilidade da pós especificação de requisitos (post-RS), associada aos aspectos de implantação do requisito.

A rastreabilidade de requisitos garante a concordância entre as necessidades dos *stakeholders* e os artefatos produzidos ao longo do processo de

desenvolvimento de *software*, permitindo descrever e seguir estes artefatos em ambas as direções, da eliciação à implementação, passando por todos os níveis de especificação relacionados (LETELIER, 2002).

A rastreabilidade auxilia a compreender e gerenciar como as informações fornecidas sobre os requisitos, como regras de negócios e as solicitações dos *stakeholders*, são convertidas em um conjunto de necessidades e características do sistema, representadas em especificações no documento de requisitos. A rastreabilidade também permite acompanhar como essas especificações são traduzidas no design, como elas são testadas e como elas são documentadas para o usuário.

Entretanto, a efetividade das práticas de rastreabilidade pode ser questionada nos projetos reais de desenvolvimento de sistemas, pois não há orientações detalhadas sobre os tipos de informações que devem ser reunidas, nem o contexto em que tal informação deve ser utilizada e, também, falta consenso sobre a semântica para as ligações entre os vários níveis de especificação e implementação (LETELIER, 2002).

Os avanços da computação permitiram uma mudança de abordagem no design de sistemas, passando de uma abordagem baseada em documentos para uma abordagem baseada em modelos. A aplicação de modelos possibilita melhorar a qualidade das especificações do sistema através da captura das informações como elementos e relações em um modelo, além de permitir a reutilização destes elementos em múltiplos diagramas. Um elevado nível de precisão e consistência pode ser obtido em virtude da introdução de informações num modelo de computador. Além disso, a rastreabilidade entre os níveis de abstração no projeto pode ser definida explicitamente no modelo como relações entre os elementos (PEARCE; HAUSE, 2012).

A MDA - *Model Driven Architecture*¹ é baseada em OMG MOF² e incorpora a importância da utilização de modelos no processo de desenvolvimento de SI. O processo de modelagem do sistema no nível conceitual deve ser independente de plataforma de implementação, sendo que através de transformações sucessivas sobre o modelo conceitual serão obtidos novos modelos com nível de abstração cada vez mais específico e detalhado, chegando ao nível de implementação.

Estes novos modelos possuem um grau de formalização cada vez maior, reduzindo a ambiguidade. Os modelos gerados no processo são: *Computation Independent Model* (CIM), com maior grau de abstração e descreve o domínio e os requisitos no sistema; *Platform Independent Model* (PIM), definido a partir do CIM e independente de tecnologia e plataforma, mas representando o negócio onde o sistema de enquadra; *Platform Specific Model* (PSM), são modelos específicos que levam em conta a tecnologia empregada na implementação, podendo um PIM originar diversos PSM (KENT, 2002).

De acordo com Estefan (2007), um método aderente ao *Model Based Systems Engineering* (MBSE) é um conjunto de processos, técnicas e ferramentas utilizados para sustentar engenharia de sistemas em um contexto baseado em modelos. O OOSEM (*Object-Oriented Systems Engineering Method*) (FRIEDENTHAL; MOORE; STEINER, 2009) é um método orientado a objetos (RAUMBACH et al., 1991) para a engenharia de sistemas, desenvolvido pelo INCOSE³, que em conjunto com SysML (OMGSYSML, 2012) sustenta MBSE (PEARCE; HAUSE, 2012).

¹MDA - *Model Driven Architecture* (<http://www.omg.org/mda/>)

²MOF - *MetaObject Facility* (<http://www.omg.org/mof/>) é um ambiente básico proposto pela OMG - *Object Management Group* (<http://www.omg.org>) onde os modelos podem ser exportados e importados de diferentes aplicações, transportados através de uma rede, armazenados e recuperados em um repositório e processado em diferentes formatos, incluindo OMG XML *Metadata Interchange* – XMI (<http://www.omg.org/spec/XMI/>)

³INCOSE - *The International Council on Systems Engineering* (<http://www.incose.org/>)

O OOSEM (figura 3) integra uma abordagem baseada em modelos e SysML para suportar a especificação, análise, design e verificação de sistemas (ESTEFAN, 2007). O OOSEM utiliza conceitos de orientação a objeto em conjunto com os métodos de engenharia de sistemas tradicionais (*top-down*), auxiliando a especificação de sistemas flexíveis e extensíveis, e permitindo acomodar tecnologia em evolução e mudanças de requisitos. OOSEM também se destina a facilitar a integração com desenvolvimento orientado a objeto do *software*, com o desenvolvimento do hardware e o teste integrado.

Figura 3: Atividades e artefatos OOSEM (ESTEFAN, 2007)

A figura 3 apresenta as atividades do processo de modelagem baseado no OOSEM, descritas a seguir:

- Analisar as necessidades dos *stakeholders*: captura o estado atual (as-

-is) dos sistemas corporativos, suas limitações e possibilidades de melhorias. O resultado desta análise é utilizado para obter os requisitos corporativos do sistema (*to-be*).

- Definir os requisitos do sistema: especificar os requisitos dos sistemas que suportam as necessidades da missão. O sistema é modelado como uma caixa preta (*black box*), que interage com os sistemas externos e usuários representados no *enterprise model*.
- Definir arquitetura lógica: decomposição e particionamento do sistema em componentes lógicos que interagem para satisfazer os requisitos do sistema.
- Alocar os componentes na arquitetura: descreve o relacionamento entre os componentes físicos do sistema, incluindo *hardware*, *software*, dados e procedimentos.
- Otimizar e avaliar alternativas: invocada nas demais atividades OOSEM para otimizar e aperfeiçoar as arquiteturas candidatas.
- Validar e verificar o sistema: verificar se a especificação do sistema satisfaz os requisitos e validar que os requisitos atendem às necessidades dos *viewpoints*.

A ISO/IEC 15288 (figura 4) é uma norma mundial para o ciclo de vida dos processos de engenharia de sistemas e *software* e define um *framework* de processos a ser aplicado a um sistema durante todo o seu ciclo de vida, incluindo definição e análise de requisitos, design da arquitetura, implementação e verificação (HASKINS, 2006).

Figura 4: Processos ISO/IEC 15288 (HASKINS, 2006)

A ISO/IEC 15288 está organizada em 4 grupos de processos (figura 4): os processos empresariais são focados na capacidade da organização em realizar e sustentar o sistema; os processos contratuais são utilizados para estabelecer os acordos entre empresas na aquisição de bens e serviços; os processos técnicos são utilizados para estabelecer os requisitos para a criação do sistema e para sustentar sua operação durante todo o ciclo de vida; e os processos de projeto incluem planejamento, alocação de recursos, controle, tratamento de riscos e gerenciamento de comunicação do projeto criação e evolução do sistema.

2.3 Especificação Formal de Sistemas

A formalização dentro do processo de desenvolvimento de SI está associada a vários níveis de representação da especificação: i) nas etapas iniciais, favorecendo a comunicação com os envolvidos na eliciação dos requisitos, mas com características de representação imprecisa e ambígua, e ii) nas etapas posteriores, sintetizando uma representação com menor ambiguidade e maior precisão. Por outro lado, tem-se um paradoxo, pois antecipar a formalização pode limitar o levantamento dos requisitos e a comunicação com os *stakeholders*. Sendo assim, uma solução é a adoção de uma formalização gradativa, onde a especificação do sistema se inicia com a modelagem baseada em representações semi-formais, por exemplo, na linguagem UML, e através de processos de transferência passa para uma linguagem como SysML (que é formal mas tem limitações para a simulação e a validação) e finalmente para redes de Petri, onde as características e a execução do modelo e a análise de propriedades são utilizadas para uma rigorosa verificação e validação das especificações do SI.

A SysML (*System Modeling Language*) é uma das principais evoluções da UML versão 2 (KOBRYN, 2004; OMG SysML, 2012), resultado de uma iniciativa compartilhada entre OMG⁴ e INCOSE para criar uma linguagem unificada, formal e semanticamente fundamentada de modelagem que atenda as necessidades da engenharia de sistemas. SysML possui um conjunto de diagramas, de fácil interpretação e sintaxe e semântica rigorosas, adequada para a especificação formal de modelos. Entretanto, SysML possui limitações quanto a execução e análise matemática de seus modelos, limitando a capacidade de análise e verificação das especificações.

⁴OMG - *Object Management Group* (<http://www.omg.org/>)

Através de uma representação gráfica, de fácil aprendizado e representação, a rede de Petri oferece um grande potencial como linguagem de comunicação entre os profissionais das mais diversas áreas da organização, além de oferecer o formalismo matemático adequado para os métodos de análise de processos (AALST, 1998).

Em 1962, Carl Adam Petri apresentou a proposta original da rede de Petri em sua tese de doutorado. A rede de Petri é utilizada como ferramenta de modelagem de sistemas permitindo representação matemática, análise dos modelos e informações sobre a estrutura e o comportamento dinâmico dos sistemas modelados. A rede de Petri admite alterações e extensões podendo ser aplicada a diversas áreas do conhecimento.

A rede de Petri é um grafo bipartido, não nulo, direcionado, formada por dois tipos de componentes, as transições e os lugares (MURATA, 1989). A realização de uma ação está associada a uma pré-condições habilitada, ou seja, as condições das variáveis de estado do sistema podem ser identificadas por meio de marcas nos lugares. A realização desta ação está associada ao disparo de uma transição. Após o disparo da transição, pós-condições são alteradas e as informações alteradas nos lugares seguintes. Lugares são elementos gráficos representados por círculos e transições são representados por retângulos ou barras. Os lugares e as transições são os vértices do grafo associado a rede de Petri e são interligados por arcos direcionados.

De acordo com Murata (1989), a rede de Petri é formalmente representada pela quintupla $RP = (P, T, F, W, M_0)$:

- $P = \{p_1, p_2, \dots, p_m\}$, conjunto finito de lugares;
- $T = \{t_1, t_2, \dots, t_m\}$, conjunto finito de transições;
- $F \subseteq (P \times T) \cup (T \times P)$, conjunto de arcos;

- $W : F\{1, 2, 3, \dots\}$, função de pesos;
- $M_0 : P \Rightarrow N$, marcação inicial;
- $P \cap T = \emptyset$ e $P \cup T \neq \emptyset$.

A rede de Petri clássica permite a modelagem de estados, eventos, condições de sincronização, paralelismo, seleção e iteração, mas não oferece recursos como modelagem de dados e tratamento de tempos, para descrever problemas mais complexos e extensos. Para resolver estes problemas várias extensões foram propostas, sendo que as extensões para variáveis associadas às marcas, tempo de disparo das transações e hierarquia entre os elementos e estruturas do grafo caracterizam as redes de Petri de alto nível (AALST, 1998), ou *High-Level Petri Nets* (SMITH, 1998):

- Quanto as variáveis associadas às marcas: pode-se considerar cores a estas marcas onde as marcações podem representar objetos, como recursos, bens e pessoas (JENSEN, 1998).
- Quanto ao tempo de disparo das transações: pode-se representar sistemas tempo real, onde é importante descrever o comportamento temporal do sistema.
- Quanto a hierarquia entre os elementos e estruturas do grafo: a especificação precisa dos processos de negócio tende a ser relativamente grande e complexa, implicando na utilização de construções hierárquicas no formato de sub-redes.

As redes de Petri vêm sendo amplamente estudadas (MURATA, 1989; ROZENBERG; ENGELFRIET, 1998; DESEL; REISIG, 1998) e aplicadas na modelagem e análise de sistemas a eventos discretos, onde sequências de atividades, paralelas e concorrentes, são reunidas para compor processos.

O ambiente aberto de desenvolvimento GHENeSys, destinado a análise e design de requisitos, *workflow*, fluxo de informações e sistemas de produção, foi concebido e desenvolvido no D-Lab⁵ da Escola Politécnica da Universidade de São Paulo a partir das ideias apresentadas em (MIYAGI, 1988; SILVA; MIYAGI, 1995; SILVA; MIYAGI, 1996) e finalmente formalizadas em (FOYO, 2001). Concebida como uma rede de Petri estendida, com conceitos de orientação a objetos, mecanismos de abstração baseado em hierarquias e síntese com abordagem estruturada e em consonância com o *Petri Net Standard* IEC/ISO 15909 (HILLAH; KORDON; PETRUCCI, 2006). GHENeSys tem potencial para se tornar uma ferramenta unificada para representação da rede de Petri clássica e todas as suas extensões, bem como as redes de alto nível. Pode ser definida como a tupla $G = (P, T, F, K, M_0)$, onde:

$P = \{p_1, p_2, \dots, p_m\}$ conjunto finito de lugares;

$T = \{t_1, t_2, \dots, t_m\}$ conjunto finito de transições;

$F \subseteq (P \times T) \cup (T \times P)$ conjunto de arcos;

$K : P \Rightarrow N^+$, função de pesos;

$M_0 : P \Rightarrow N$, marcação inicial.

Os invariantes são características algébricas fundamentais da rede de Petri e são utilizadas em diversas situações, tais como a verificação de propriedades como: vivacidade, limitação, existência de laços, e outros (MURATA, 1989). Os invariantes representam os componentes repetitivos e conservativos de uma rede, permitindo a identificação de conjuntos de lugares e transições que não podem ser atingidos em função da estrutura da rede, constatação importante para analisar o comportamento do sistema representado pela rede.

⁵<http://dlab.poli.usp.br>

2.4 Síntese do Capítulo

Este capítulo apresentou os principais conceitos sobre ciência de serviço, as recentes iniciativas de desenvolvimento de fundamentos acadêmicos e métodos científicos destinados a oferecer melhorias na eficiência e na qualidade dos sistemas de serviço, a tendência dos novos SI como agentes integradores da inovação tecnológica, e finalmente, os conceitos associados à formalização no processo de especificação dos SI e possibilidades de representação.

3 SOFTDISS

Neste capítulo é apresentado o *Framework* Orientado a Serviço para Especificação de Sistemas de Informação de Serviço - SoftDiss (*Service-Oriented Framework to the Design of Information System Service*). O SoftDiss pode ser adicionado a ambientes de desenvolvimento de software já existentes, especificamente neste trabalho o SoftDiss é implementado no Enterprise Architect (SPARX, 2013). A ideia principal é oferecer um *framework* para suporte às fases iniciais do desenvolvimento de sistema de informação de serviço (SIS), incluindo a eliciação, modelagem e análise de requisitos, através da configuração dinâmica e cognitiva de tecnologias, pessoas, sistemas externos e das informações compartilhadas entre estes elementos.

3.1 Premissas da Fase de Requisitos

No projeto de desenvolvimento de SIS a fase inicial tem um impacto ainda maior que nos demais projetos. As razões para isso são: a necessidade de uma fase de interação e co-criação com os atores e com os beneficiários do serviço, a necessidade de compatibilização de interesses entre *stakeholders* e (as diversas classes de) usuários (com a integração dos *viewpoints* de cada uma das classes de *stakeholders* e usuários), e, finalmente, a necessidade de compatibilizar requisitos e restrições e processos de negócio.

Assim, a fase de requisitos pode ser dividida em uma fase externa onde

os requisitos são eliciados e servem de entrada para o processo de desenvolvimento, e uma fase interna, onde estes requisitos são documentados, dispostos em *viewpoints*, modelados preliminarmente (de modo que se pode ter uma ideia de como cada ator vê o sistema), e analisados para compatibilizar os requisitos e eliminar contradições até que estes virem especificação. No decorrer deste processo, negociações com atores ou classes de atores se fazem necessárias para adaptar o conjunto de requisitos e torná-lo um conjunto convergente, isto é, capaz de atingir uma fase avançada de compatibilização, onde se poderá de fato vislumbrar o modelo do SIS. Esta negociação também constitui subprocessos externos, de natureza semelhante à eliciação básica.

O foco está na proposta de um *framework* para o desenvolvimento de sistema de informação de serviço (SIS), onde, os subprocessos externos não serão considerados, dado que estes são basicamente dependentes da equipe de desenvolvimento e não dos insumos e sistemas de apoio.

Outro aspecto importante é que, dada a importância da fase de requisitos no desenvolvimento de SIS a proposta é antecipar a formalização e tratá-la já nesta fase de análise de requisitos. Portanto, o processo de projeto proposto trabalha com a modelagem semi-formal dos requisitos para, na medida do possível (e da completeza das informações levantadas) formalizar o modelo ainda nesta fase (ou pelo menos no final da fase de análise de requisitos). Consequentemente o sucesso nesta disposição esvazia as fases subsequentes, na medida em que é possível prever que para um bom número de casos se termine a fase de modelagem com o SoftDiss com uma especificação formal e com um modelo (em SysML ou redes de Petri) também formal do SIS. Neste caso a fase de design subsequente envolveria somente o detalhamento do SIS visando a fase de implementação.

Naturalmente a abordagem utilizada para a eliciação e organização dos

requisitos é de fundamental importância. O uso de sistemas de análise como o KAOS (LAMSWEEERDE, 2001) ou o i* (YU, 1995; YU, 1997) pode trazer uma nova dimensão para a discussão e para a contribuição deste trabalho. Entretanto optou-se por tratar o problema pelos aspectos gerais e propor um *framework* que possa se acoplar a qualquer método de eliciação e análise. Portanto, foram utilizados métodos convencionais e bem conhecidos na comunidade de tecnologia da informação (TI) como orientação básica para o levantamento dos requisitos.

A seguir a proposta do SofDiss é apresentada destacando os elementos da sua estrutura básica e sua funcionalidade. Exemplos refletem a aplicação destes mesmos elementos nos estudos de caso que foram utilizados para validar e gerar *feedback* para o processo de construção do SoftDiss. Evidentemente não se espera um entendimento completo claro destes flashes até que cada um deles seja apresentado em detalhe, mas por outro lado a repetição e a referência cruzada poderá ser ainda melhor do que apresentar exemplos superficiais e pouco convincentes.

3.2 Estrutura do SoftDiss

A proposta de um *framework* tem como objetivo direcionar o processo de modelagem onde a entrada são especificações (funcionais, não-funcionais, processos de negócio e restrições) e, com base em "boas" práticas e no formalismo de esquemas trabalhar a passagem destes requisitos (após análise) em especificação (formal) de SIS. Portanto, a concepção do SoftDiss é baseada nas cinco premissas listadas na tabela 3.

Tabela 3: Premissas da concepção do SoftDiss

#	Premissas
P1	Um procedimento baseado em modelos é a melhor forma de conduzir o processo, e que é possível acelerar o processo de formalização.
P2	A eliciação e análise de requisitos é a principal etapa no ciclo de vida do desenvolvimento de SIS, visando a sustentação do processo de co-criação com o cliente do serviço.
P3	A antecipação da formalização dos requisitos é indispensável no ciclo de vida do desenvolvimento de SIS, viabilizando a obtenção de requisitos consistentes e completos o mais cedo possível e considerando a diversidade de variáveis e <i>viewpoints</i> que acometem um serviço.
P4	A gestão sobre os requisitos é indispensável para garantir o processo de negociação e validação com os representantes dos diversos <i>viewpoints</i> .
P5	O processo de desenvolvimento de SIS deve ser baseado em "boas" práticas sustentadas por normas, procedimentos, técnicas, métodos, ferramentas e ambientes de apoio.

O SoftDiss é um *framework*¹ de eliciação, documentação, análise, modelagem de requisitos de SIS. Portanto, o SoftDiss serve a desenvolvedores de SIS na tarefa de modelagem e design até chegar a uma especificação formal do sistema-alvo (*system of interest*).

Entre as premissas básicas, o SoftDiss parte do pressuposto de que os requisitos se dividem em processos de negócio (PN) e requisitos livres (OLI-

¹*Framework* é definido como uma sequencia de atividades de projeto coordenadas e assistidas por computador, e baseadas em métodos que agregam formalismo, técnicas de modelagem e design de sistemas e "boas" práticas extraídas da experiência. Assim, um *framework* difere de um ambiente completo e pode ser inserido em um ou mais destes ambientes.

VEIRA; SILVA, 2011), contemplando os diversos *viewpoints*, e segue um processo gradativo de formalização da representação orientada a modelos que vão da especificação informal e semi-formal das etapas iniciais, até uma especificação formal dos requisitos eliciados. Os PN são associados estritamente aos *stakeholders*, enquanto os demais requisitos podem atender aos *viewpoints* de várias classes de agentes, incluindo usuários, clientes, gerentes, e outros.

A figura 5 apresenta a estrutura do SoftDiss, onde a entrada é a solicitação para solução de um determinado problema e a saída é a especificação formal do SIS (sistema-alvo).

O SoftDiss possui quatro ambientes: o Ambiente de Requisitos (*Requirement Environment*), o Ambiente de Especificação (*Design Environment*), o Ambiente de Formalização (*Formal Environment*) e o Ambiente Gerencial (*Managerial Environment*), todos orientados a serviço e sustentados pela configuração dinâmica e cognitiva de tecnologias, sistemas externos, pessoas, processos e informações compartilhadas.

Figura 5: Estrutura do SoftDiss

O Ambiente de Requisitos - ReqEnv (*Requirement Environment*) oferece os serviços de organização e registro dos diversos *viewpoints* a serem considerados na especificação dos SIS. No ReqEnv são tratadas as atividades que sustentam a representação dos diversos *viewpoints*, oriundos dos processos de negócio, requisitos dos usuários e modelos de referência, através de disciplinas e métodos que permitam manter a rastreabilidade e padronização das diversas fontes de requisitos. Métodos básicos de análise são aplicados neste ambiente, visando o acoplamento entre a PN e os requisitos básicos (OLIVEIRA; SILVA, 2011).

O Ambiente de Especificação - DesEnv (*Design Environment*) oferece os recursos para identificar e analisar os serviços necessários para solucionar o problema fornecido na entrada do SoftDiss e resultar na especificação formal de SIS, contando com vários níveis de abstração da representação orientada a modelos, sendo informal e semi-formal nas etapas iniciais, favorecendo a co-

municação com os envolvidos, entretanto imprecisa e ambígua, e nas etapas finais com representação formal precisa e sem ambiguidades.

O Ambiente de Formalização - ForEnv (*Formal Environment*) fornece os serviços associados ao tratamento da validação dos *viewpoints* de requisitos, através da verificação estrutural dos modelos, verificação formal matemática e simulação de modelos.

O Ambiente Gerencial - ManEnv (*Managerial Environment*) oferece os serviços de gerenciamento das atividades e trocas de informação do SoftDiss, sustentando o planejamento e o controle das atividades, alocando os recursos e viabilizando o fluxo de informação entre os diversos ambientes. O ManEnv concentra as atividades de gestão do processo de especificação e modelagem do SIS atuando na sincronização entre as atividades dos demais ambientes. O ManEnv atua diretamente na manutenção dos fundamentos da orientação a serviço onde os recursos são ativos, integrados, alocados visando criação de valor e focados no cliente, que por sua vez, também, atua como um recurso ativo participando do processo de co-criação. O ManEnv trata da manutenção do ciclo de vida dos SIS, desde a identificação até a verificação formal, incluindo também as requisições de revisão e melhorias nos SIS já em operação, resultando em nova especificação ou até mesmo a extinção dos SIS quando estes deixam de cumprir sua função principal.

A figura 6 apresenta o SoftDiss, com os detalhes de cada ambiente, os fluxos de informação e controle, e a identificação das técnicas, sistemas externos, ferramentas e normas que sustentam o ambiente orientado a serviço para especificação de SIS.

Figura 6: Visão detalhada do SoftDiss

O resultado obtido pelo SoftDiss é um conjunto de modelos representando os diversos níveis de abstração resultantes, tanto do processo gradativo de formalização dos requisitos, como da harmonização entre os *viewpoints* associados ao SIS (sistema-alvo). MDA² (*Model Driven Architecture*) e MOF³ (*Meta-Object Facility Specification*), ambas propostas e sustentadas pela OMG (*Object Management Group*), suportam a manutenção e processamento dos vários modelos do SoftDiss. A ferramenta de software Enterprise Architect - EA (SPARX, 2013) provê os recursos para desenvolvimento, registro e documentação dos modelos.

²<http://www.omg.org/mda/>

³<http://www.omg.org/mof/>

A figura 7 apresenta a estrutura de pacotes do SoftDiss implementada no EA. Os pacotes agrupam os modelos resultantes dos ambientes ReqEnv e DesEnv. Nos pacotes *Services* e *Customer* os serviços e consumidores identificados são documentados e são origem para a rastreabilidade destes elementos. A figura 7 apresenta os pacotes do estudo de caso do Agronegócio (seção 4.3). A seção 3.6 - *Managerial Environment* (ManEnv) descreve em detalhes a organização dos pacotes do SoftDiss e a implementação no EA.

Figura 7: Estrutura de pacotes do SoftDiss

Nas próximas seções são abordados os detalhes de cada um dos ambientes do SoftDiss e os relacionamentos existentes entre eles, bem como exemplos das representações e modelos utilizados.

3.3 Requirement Environment - ReqEnv

O Ambiente de Requisitos - ReqEnv (*Requeriment Environment*) oferece os serviços de organização, evolução, manutenção e registro dos diversos *viewpoints* de requisitos, tais como os processos de negócio, os diversos requisitos dos usuários e os modelos de referência, a serem considerados durante todo o processo de obtenção da especificação dos SIS. Os serviços do ReqEnv são gerenciados pelo ManEnv e utilizados pelos demais ambientes do SoftDiss como fonte de requisitos para os diversos níveis de abstração da representação.

Os *viewpoints* de requisitos são fornecidos como entrada para o ReqEnv em diversos tipos de representação e linguagens de modelagem. No ReqEnv cada *viewpoint* de requisitos é transformado para uma representação orientada a objetos em UML, considerando a disciplina de modelagem *Naked Objects* (PAWSON, 2002), o método o Volere (ROBERTSON; ROBERTSON, 2006) como processo e como ferramenta de software o Enterprise Architect.

Na disciplina proposta na abordagem Naked Object é encorajada a especificação de sistemas a partir de objetos comportamentalmente completos, mantendo sempre os dados e o comportamento dos objetos agrupados, e buscando identificar os principais objetos de negócio. O processo de engenharia de requisitos Volere fornece as atividades a serem realizadas para obter a modelagem orientada a objetos, representada em UML, dos diversos *viewpoints* de requisitos. Os requisitos são passíveis de validação, devem possuir um critério de aceitação e são classificados de acordo com os seguintes tipos:

- Requisitos Funcionais: descrevem o que o produto tem de fazer ou que ações processuais deve tomar.

- Requisitos Não Funcionais são as propriedades que as funções devem ter.
- Restrições do Projeto: são as limitações de como o produto deve ser especificado.
- Diretivas do Projeto: são as forças associadas ao negócio.
- Domínio do Projeto: definem as condições nas quais o projeto será realizado.

A figura 8 apresenta o processo de modelagem de requisitos empregado no ReqEnv, onde os diversos *viewpoints* de requisitos são identificados e registrados como diagramas UML, distinguindo preliminarmente requisitos básicos e PN, para que possam posteriormente ser analisados e acoplados utilizando verificação de consistência básica (OLIVEIRA; SILVA, 2009; OLIVEIRA, 2008).

Figura 8: Processo de modelagem dos requisitos

De acordo com o processo de modelagem dos requisitos (figura 8), primeiramente são definidos os direcionadores do projeto (objetivo do projeto, clientes, consumidores, usuários do produto e outros *stakeholders*) e as restrições do projeto (requisitos, glossário, fatos relevantes e premissas). Em seguida, para cada *viewpoint* identificado - incluindo processos de negócio, requisitos dos usuários e modelos de referência - são criados pacotes (*packages*) separados no EA, determinados os requisitos funcionais, os requisitos não funcionais e as questões relevantes, sendo o resultado registrado no diagrama de requisitos do EA (figura 9).

Figura 9: Diagrama de requisito associado a *viewpoint*

A figura 9 apresenta um dos *viewpoints* de requisitos eliciados no estudo de caso do agronegócio (seção 4.3), neste caso os requisitos eliciados para a classe de agentes da indústria.

Os requisitos são analisados de acordo com a disciplina *Naked Objects* (PAWSON, 2002) e registrados no diagrama de objetos do EA. A figura 10 apresenta os objetos associados ao mesmo *viewpoint* do agronegócio, a indústria de equipamentos agrícolas.

Figura 10: *Naked Objects* associados à indústria agrícola

O ReqEnv fornece aos demais ambientes o serviço de suporte ao tratamento de requisitos, contemplando os diversos *viewpoints*, padronizados na

representação orientada a objetos em UML, considerando a disciplina de modelagem *Naked Objects*. Sendo um ambiente orientado a serviço, o ReqEnv participa de uma cadeia de prestação de serviços, contando com a participação de recursos ativos, alocados visando criação de valor e focados no cliente. Os recursos ativos são resultantes da aplicação dos métodos, processos e técnicas do *Naked Objects* e do Volere. Os clientes do ReqEnv são os demais ambientes do SoftDiss, que atuam dentro de processo de co-criação fornecendo realimentação para os requisitos e direcionamento a negociação e convergência dos requisitos junto aos diversos *viewpoints*.

A atividade final do processo de modelagem dos requisitos (figura 8), que trata o acoplamento entre os *viewpoints*, é realizada de acordo com os seguintes passos:

1. Primeiramente, todos os requisitos são agrupados e mapeados em classes de agentes. Nenhum grupo de requisitos pode estar dissociado do respectivo agente de *viewpoint*.
2. Em seguida, um conjunto especial de *viewpoints* é identificado como processos de negócio (PN) e geralmente mapeado aos *stakeholders*.
3. Uma primeira análise de compatibilidade é realizada utilizando verificação de consistência básica visando o acoplamento entre os *viewpoints* de requisitos básicos e de PN, onde os mesmos requisitos podem ser encontrados em diferentes *viewpoints*, sendo necessário um mapeamento metafórico de termos (OLIVEIRA; SILVA, 2011).

A figura 11 apresenta o resultado da análise de compatibilidade entre os três *viewpoints* (Indústria, Concessionário e Agricultor) do estudo de caso do Agro-negócio (seção 4.3). O recurso de análise de acoplamento entre *viewpoints*,

de acordo com a metodologia de pesquisa (seção 1.3), foi incorporado ao SoftDiss como resultado do estudo de caso Agronegócio.

Figura 11: Resultado da análise de compatibilidade de *viewpoints*

O resultado da análise de compatibilidade documenta o acoplamento entre os *viewpoints* buscando eliminar contradições e duplicações, através de negociação com os atores e tratando a rastreabilidade. Observar que cada requisito da figura 11 possui uma indicação de origem (*from*), resultado do recurso de importação do EA.

3.4 Design Environment - DesEnv

O Ambiente de Especificação - DesEnv (*Design Environment*) é basicamente inspirado no SOMF (BELL, 2008) e constituído por um grupo de ambi-

entes internos que, em conjunto, permitem identificar, analisar, especificar, registrar e documentar os serviços associados à solução do problema fornecido na entrada do SoftDiss, atender os *viewpoints* de requisitos e resultar na especificação formal dos SIS. O DesEnv conta com vários níveis de abstração da representação orientada a modelos, sendo informal e semi-formal nas etapas iniciais, favorecendo a comunicação com os envolvidos, entretanto imprecisa e ambígua, e nas etapas finais com representação formal precisa buscando eliminar ambiguidades.

O DesEnv é composto por três ambientes internos (figura 6): Ambiente Conceitual - ConEnv (*Conceptual Environment*); Ambiente de Análise - AnaEnv (*Analysis Environment*); e Ambiente Lógico - LogEnv (*Logical Environment*).

No ConEnv os requisitos eliciados dos diversos *viewpoints* fornecidos pelo ReqEnv são utilizados como entrada para a definição da arquitetura conceitual dos serviços básicos identificados. Uma combinação adequada destes serviços básicos resultará na especificação do SIS, que é a solução do problema de design.

O AnaEnv transforma os serviços conceituais em serviços analíticos, utilizando o conhecimento do domínio do problema para validar a arquitetura dos serviços básicos que podem fornecer soluções conceituais para o design. Caso não seja encontrado um serviço simples de composição, o AnaEnv deve fornecer a melhor solução aproximada e relatar ao ManEnv, permitindo o tratamento gerencial da eventual diferença encontrada. Neste trabalho não está sendo considerada uma solução inteligente para identificar a natureza e dimensionamento destas diferenças. Tal ferramenta poderia ser integrada no ManEnv para rastrear os requisitos (ReqEnv) e identificar os *viewpoints* não contemplados, ou ao menos, obter uma boa estimativa sobre estas diferenças.

O LogEnv trata a transformação dos serviços analíticos - compatíveis e harmônicos com os serviços básicos - em elementos mais tangíveis permitindo retratar os relacionamentos, a estrutura e o comportamento dos serviços básicos viabilizando a obtenção de uma arquitetura lógica que sustente uma solução de serviço otimizado, ou seja, uma solução sem duplicações ou dispêndio desnecessário de recursos. Também no LogEnv, a solução geral pode ser transferida para uma representação formal, onde a solução candidata poderá ser verificada ou simulada.

O *Service Oriented Modeling Framework* (SOMF) (BELL, 2008) fornece as práticas, processos e disciplinas das etapas iniciais do DesEnv, representadas pelos ambientes internos: ConEnv e AnaEnv. Estas etapas iniciais utilizam para a representação da especificação a linguagem semi-formal SOMF, estendida da UML e que facilita e estimula a comunicação com os *stakeholders*, fator importante nas etapas iniciais de captura das necessidades e identificação dos serviços. Nesta etapa os serviços são identificados, conceituados, analisados e integrados com as necessidades de negócio.

O SoftDiss agrega contribuição aos ambientes ConEnv e AnaEnv originalmente propostos pelo SOMF através da organização, evolução, manutenção, rastreabilidade e registro dos diversos *viewpoints* de requisitos oferecidos pelo ReqEnv e dos serviços oferecidos pelo ManEnv que viabilizam o controle de versão da documentação, visão integrada dos demais ambientes, gerenciamento das atividades e alocação de recursos.

O LogEnv, etapa final do *Design Environment*, a representação semi-formal SOMF é transformada para uma representação em SysML, com características de representação precisa e livre de ambiguidades. A representação em SysML permite a verificação estrutural da especificação e, também, utilizando-se os recursos de execução dos modelos dinâmicos, torna a especifi-

cação passível de análise por simulação.

3.4.1 Conceptual Environment - ConEnv

O Ambiente Conceitual - ConEnv (*Conceptual Environment*) oferece os recursos que viabilizam a identificação, definição e estruturação dos serviços conceituais associados à solução do problema de entrada do SoftDiss. O ConEnv utiliza o método proposto no SOMF, composto pela disciplina Conceitualização do Serviço (*Service Conceptualization*), que oferece os recursos para a atribuição, identificação e classificação dos serviços. e pela disciplina Arquitetura Conceitual (*Conceptual Architecture*), que oferece os recursos para a formação de uma arquitetura de associação entre os serviços. Estas duas disciplinas em conjunto visam a obtenção do modelo conceitual orientado a serviço, representado por cinco elementos principais, cada um contribuindo para a fundamentação conceitual dos serviços e estabelecimento de uma taxonomia de negócios e de tecnologia:

- Atribuição: processo de definição dos atributos que caracterizem os serviços conceituais em termos de características, qualidade e propriedades.
- Identificação e descoberta: processo de identificação que permite a descoberta de serviços conceituais.
- Classificação: processo de classificação dos serviços conceituais identificados visando produzir uma taxonomia organizacional.
- Associação: processo que revela a relação entre os serviços conceituais.
- Formação: processo que viabiliza a formação de estruturas conceituais de serviços.

O Ambiente Conceitual do SoftDiss agrega contribuição ao ambiente original do SOMF, através do tratamento explícito dos *viewpoints* de requisitos do ReqEnv, onde cada *viewpoint* é endereçado e a representação utilizada é padronizada pela orientação a objetos em UML, e do serviço oferecido pelo ManEnv de rastreabilidade dos requisitos, controle de versão da documentação e visão integrada dos demais ambientes.

O processo de atribuição determina os atributos essenciais através da análise dos *viewpoints* de requisitos, obtidos através do serviço prestado pelo ReqEnv, visando determinar os atributos essenciais e prover justificativas e motivações para o desenvolvimento de serviços.

O modelo de atributos da figura 12 é obtido a partir da combinação dos atributos essenciais no primeiro nível e a combinação destes atributos nos níveis imediatamente inferiores, sendo o último nível a convergência de todos os atributos. Este modelo fornece uma visão da associação dos atributos essenciais e permite a identificação priorizada e agrupada dos serviços, sendo um instrumento essencial para a rastreabilidade e priorização dos serviços nas etapas subjacentes do SoftDiss, pois mesmo sendo uma etapa onde os modelos possuem uma representação semi-formal (SOMF) já permite uma associação entre os requisitos dos *viewpoints* e os serviços identificados. Os atributos essenciais visibilidade, comunicação, integração e contribuição são oriundos do estudo de caso do D-Lab e resultado dos serviços prestados pelos ReqEnv de manutenção e rastreabilidade dos *viewpoints* de requisitos e do processo de negociação com os atores sustentado pelo ambiente gerencial - ManEnv do SoftDiss.

Figura 13: Árvore de decisão dos serviços conceituais

Os serviços conceituais ligados às folhas das diversas árvores de decisão fornecem a taxonomia dos serviços associados à solução do problema de entrada do SoftDiss e viabilizam a priorização dos serviços conceituais de acordo com a importância de cada atributo essencial. A importância dos atributos é determinada pelos envolvidos na especificação do SIS e a negociação para esta ordenação dos atributos é um serviço oferecido pelo ManEnv e será utilizada para determinar os serviços essenciais prioritários.

A próxima etapa do ConEnv é a determinação dos consumidores dos serviços essenciais. Os recursos do ReqEnv fornecem os consumidores, extraídos dos *viewpoints* de requisitos.

Os serviços conceituais identificados através do processo de associação, podem ser classificados e agrupados através dos métodos de associação conceitual organizando e relacionando os serviços com seus respectivos consumidores. A figura 14 apresenta o modelo conceitual de associação para os

serviços identificados na figura 13, associando os serviços conceituais com os devidos consumidores dos serviços. Novamente o exemplo da figura 14 é extraído do caso do D-Lab e apresenta a associação entre os consumidores e os serviços identificados.

Figura 14: Modelo de associação conceitual

Os modelos de atributos, árvore de decisão e associação representam o modelo conceitual orientado a serviço, resultado obtido no ConEnv através da aplicação das disciplinas Conceituação do Serviço e Arquitetura Conceitual. O modelo conceitual orientado a serviço é utilizado como entrada para o Ambiente de Análise - AnaEnv.

3.4.2 Analysis Environment - AnaEnv

O Ambiente de Análise - AnaEnv (*Analysis Environment*) oferece o serviço de transformação do serviço conceitual inicial em um serviço que propicie cate-

gorização mais aprofundada analisando o domínio do problema, entendendo as soluções conceituais existentes na organização, avaliando se o serviço proposto resolve o problema de entrada do SoftDiss, e, caso não resolva, qual a lacuna existente, analisando os sistemas legados e modelando uma proposta de solução. O AnaEnv é composto pelas disciplinas Descoberta e Análise de Serviço (*Service Discovery & Analysis*) e Integração de Negócio (*Business Integration*).

A disciplina Descoberta e Análise de Serviço oferece práticas para auxiliar na caracterização dos sistemas organizacionais, identificação de novos serviços, avaliação da contribuição do serviço para a solução do problema e estabelecimento de propostas de solução. As práticas incluem a classificação de serviços e análises de granularidade, agregação, decomposição, unificação e intersecção dos serviços.

O resultado da aplicação das práticas da disciplina de Descoberta e Análise de Serviço é o modelo de análise contextual representado na figura 15, onde o serviço conceitual “Validação textos” do caso D-Lab que agrega o serviço “Gestão de Documentos”, que por sua vez, generaliza quatro serviços atômicos e agrega o serviço “Acompanhamento de Tese”.

Figura 15: Modelo de análise contextual

O modelo de análise estrutural representa a manipulação estrutural da análise dos serviços, considerando a relação entre os serviços e seus serviços atômicos. A figura 16 representa dois serviços conceituais ("Gestão de Documentos" e "Acompanhamento de Tese" do D-Lab) ligados por um conector de acoplamento e expandidos com seus serviços atômicos através de um conector de composição. O modelo de análise estrutural permite detalhar a estrutura de relacionamento entre os serviços e o tipo de ligação com os serviços atômicos, documentando e gerenciando a rastreabilidade.

Figura 16: Modelo de Análise Estrutural

A disciplina Integração de Negócio permite integrar os serviços analisados com os domínios de negócio, através de práticas e de uma linguagem de modelagem que incentive a integração entre os serviços e o modelo de negócio da organização. O termo domínio de negócio identifica uma organização empresarial, linha de negócio, ocupação de negócio, departamento, divisão ou parceiro de negócios que planeja utilizar serviços para automatizar operações de negócios associadas ao problema de entrada do SoftDiss. As perspectivas de negócio são baseados nos *viewpoints* de requisitos (ReqEnv) e negociadas com os atores no ManEnv.

A figura 17 apresenta o modelo de integração de negócio, contendo as perspectivas operacional e regulatória do caso D-Lab e a associação de integração com os serviços.

Figura 17: Modelo de Integração de Negócio

No AnaEnv que se dá a transição entre as práticas de abstração e de realização. Nas práticas de abstração os serviços são descobertos, identificados e organizados em uma arquitetura, já nas práticas de realização os aspectos abstratos dos serviços passam a ser formalizados e estabelecidos como conceitos organizacionais e tendem a serem aspectos tangíveis da lógica dos serviços, contando com padrões de mensagem, funcionalidades técnicas e de negócio, estruturação dos serviços e organização dos processos organizacionais e *workflow*.

O resultado obtido no AnaEnv é um conjunto de informações que representa os serviços identificados, conceituados, priorizados, analisados e integrados com as necessidades de negócio, com "boas" práticas de rastreabilidade e atendendo aos diversos *viewpoints* de requisitos, mas é uma modelagem representada em linguagem semi-formal SOMF.

3.4.3 Logical Environment - LogEnv

O Ambiente Lógico - LogEnv (*Logical Environment*) oferece os serviços de transformação dos serviços analíticos em elementos mais tangíveis denominados serviços lógicos - representados pelo modelo de especificação formal orientado a serviço - permitindo retratar os relacionamentos, a estrutura e o comportamento dos serviços viabilizando a criação de uma arquitetura lógica para os serviços. Tem-se aqui a antecipação da formalização da representação da especificação que passa a ser formal em SysML em substituição a representação semi-formal em SOMF utilizada nas etapas anteriores.

O resultado do AnaEnv é a lista de serviços identificados, conceituados, analisados, integrados e, principalmente, priorizados em função dos atributos essenciais definidos no ConEnv e com rastreabilidade em relação aos níveis de abstração anteriores (ConEnv e AnaEnv). O serviço de transformação do LogEnv utiliza esta lista priorizada de serviços como ponto de partida para selecionar o serviço mais prioritário e que será indicado como o SIS de interesse (*system of interest*) para iniciar a especificação formal e através de um processo iterativo em espiral, atendendo a lista de prioridade, os demais serviços são integrados à especificação.

A transformação dos serviços analíticos nos serviços lógicos, representados pelo modelo de especificação formal orientado a serviço, é suportada por um conjunto de processos baseados em MBSE (*Model Based Systems Engineering*), contemplando a norma ISO/IEC 15288 ⁴, dirigidos pelo método OOSEM⁵ e resultando em um modelo representado em SysML.

O resultado obtido no AnaEnv é a representação dos serviços identifica-

⁴ISO/IEC 15288 - *Systems Engineering: system life cycle processes* (HASKINS, 2006).

⁵OOSEM - *Object-Oriented Systems Engineering Method* (FRIEDENTHAL; MOORE; STEINER, 2009)

dos, conceituados, analisados, integrados, e priorizados e rastreáveis em função dos atributos essenciais, atendendo aos diversos *viewpoints* de requisitos, mas é uma modelagem representada em linguagem semi-formal SOMF. Esta representação em SOMF é transformada no modelo de especificação formal orientado a serviço, representado em SysML, através da aplicação do OOSEM, atendendo a norma ISO-15288 e com uma modelagem orientada a modelos (MBSE).

O OOSEM integra uma abordagem baseada em modelos e SysML para suportar a especificação, análise, design e verificação dos sistemas. OOSEM utiliza conceitos de orientação a objeto em conjunto com os métodos de engenharia de sistemas tradicionais (*top-down*), auxiliando a especificação de sistemas flexíveis e extensíveis, permitindo acomodar tecnologia em evolução e mudanças de requisitos. OOSEM também se destina a facilitar a integração com desenvolvimento orientado a objeto de software, desenvolvimento de hardware e teste.

No OOSEM a organização dos modelos é padronizada através de uma estrutura de pacotes. A organização do modelo (*Model Organization*) é uma estrutura recursiva de pacotes representando a hierarquia sistêmica, onde um pacote é definido para cada bloco, que por sua vez é decomposto em pacotes que formam o próximo nível de detalhamento ou decomposição. A figura 18 apresenta o diagrama de pacotes (*Model Organization*), associado ao caso Agronegócio (seção 4.3), contendo a organização dos pacotes que compõem a especificação do projeto (FRIEDENTHAL; MOORE; STEINER, 2009).

Figura 18: Exemplo de organização do modelo

No LogEnv do Softdiss a metodologia OOSEM é aplicada para capturar a especificação formal em SysML dos requisitos dos SIS associados à solução

do problema de entrada do SoftDiss. No LogEnv são aplicadas as seguintes atividades propostas na OOSEM: análise das necessidades dos *stakeholders*, análise dos requisitos do sistema, definição da arquitetura lógica, gerenciamento da rastreabilidade dos requisitos, otimização e avaliação de alternativas e síntese das arquiteturas candidatas (figura 19).

Figura 19: Processo de especificação e design OOSEM (FRIEDENTHAL; MOORE; STEINER, 2009)

A atividade análise das necessidades dos *stakeholders* provê o entendimento das necessidades dos envolvidos com o sistema quanto ao problema a ser solucionado pelo SIS e a especificação dos requisitos de negócio. Nesta atividade é capturado o estado atual (*as-is*) dos sistemas corporativos, suas limitações e possibilidades de melhorias e o ReqEnv fornece os requisitos

dos diversos *viewpoints*. Como resultado da atividade de análise das necessidades dos *stakeholders* são produzidos modelos para representar o domínio operacional (*as-is*), os requisitos de domínio (*as-is*), os parâmetros de eficiência, casos de uso de negócio e, finalmente, o modelo de domínio operacional do sistema alvo (*system of interest*). Visando um melhor entendimento da atividade de análise das necessidades dos *stakeholders*, será utilizado o atendimento ao serviço “Contrato de manutenção” do caso Agronegócio, priorizado em função dos resultados anteriores (ConEnv e LogEnv) e negociado pelo ManEnv com os atores. A figura 20 apresenta os requisitos de domínio associados ao serviço “Contrato de manutenção”, com destaque para a rastreabilidade dos requisitos desde os estágios iniciais com os diversos *viewpoints* de requisitos.

Figura 20: Requisitos de domínio (*as-is*)

O ReqEnv fornece os requisitos dos *viewpoints*, especificamente dos processos de negócio, e o recurso de alocação (*Allocation*) do SysML é utilizado para relacionar os requisitos dos *viewpoints* com os elementos de modelagem, permitindo através da verificação estrutural da modelagem em SysML, avaliar se os requisitos estão alocados aos modelos. Informações adicionais

de relacionamento são registradas utilizando o recurso racional (*Racionale*) do SysML, que permite indicar as razões para uma decisão específica de modelagem (figura 21).

Figura 21: Domínio operacional (*as-is*)

A atividade de análise dos requisitos do sistema trata o sistema de interesse como caixa-preta (*black-box*), considerando as características externas e o comportamento das entradas e saídas do sistema. Nesta atividade são definidos e modelados os cenários dos casos de uso, identificando as interações entre o sistema de interesse e os sistemas externos. O contexto do sistema de interesse é definido, contendo as interfaces com todos os sistemas externos e usuários participantes dos cenários. O ReqEnv fornece os requisitos dos *viewpoints*, especificamente dos processos de negócio, e o recurso de alocação (*Allocation*) do SysML é utilizado para relacionar os requisitos dos *viewpoints* com os elementos de modelagem, permitindo através da verificação estrutural da modelagem em SysML, avaliar se os requisitos estão alocados aos modelos. As restrições e propriedades críticas do sistema de interesse são identificados, incluindo requisitos de desempenho, alocações de tempo necessárias para satisfazer os parâmetros de eficiência. A figura 22 está as-

sociada ao caso do D-Lab e apresenta a especificação caixa-preta do sistema de interesse “Site DLab”.

Figura 22: Especificação *black-box* do sistema alvo

Os requisitos analisados do sistema de interesse são especificados como caixa-preta, o comportamento e as características físicas são definidas, incluindo requisitos funcionais, interfaces externas, necessidades de controle e armazenamento. As ações a serem desempenhadas pelo sistema de interesse definidas nos cenários são especificados em máquinas de estado, onde condições de guarda especificam valores de entrada e disponibilidade de recursos. A variação e a mutação dos requisitos são inerentes aos SIS, implicando na análise da variação dos requisitos do sistema de interesse e na especificação do potencial de variabilidade das interfaces externas, número de usuários, novas funcionalidades e alteração de funcionalidades. As restrições de especificação do sistema de interesse, limitam o espaço de solução e são impostas pelos diversos *viewpoints*, estas restrições são fornecidas pelo ReqEnv e alocadas aos diversos modelos viabilizando a verificação estrutural

e consistência dos modelos.

A ferramenta Enterprise Architect (EA), adotada para a implementação do SoftDiss, provê verificação estrutural dos modelos, capacidade de simulação e implementação de modelos de comportamento formalmente representados em SysML.

3.5 Formal Environment - ForEnv

O Ambiente de Formalização - ForEnv (*Formal Environment*) fornece os serviços associados à evolução dos requisitos para a especificação formal e a preparação para o processo de verificação - que pode eventualmente ser feito por verificação de modelos.

A última etapa do DesEnv traduz o modelo de design orientado a serviço de UML para SysML e transfere este modelo para o ForEnv. O ManEnv coordena a transferência dos modelos e fornece o *feedback* sobre os resultados deste processo.

O modelo resultante do LogEnv (etapa final do DesEnv) representado em SysML, é transferido para o ForEnv utilizando os mapeamentos de transferência MDA oferecidos pelo EA. O ForEnv trata a verificação estrutural do modelo, através da consistência entre os elementos do modelo e a metalinguagem de SysML, e do comportamento dinâmico da especificação, que pode ser realizado diretamente no EA através de simulação, ou através da transformação dos diagramas de estado em SysML para rede GHENeSys, possibilitando a verificação das especificações através da análise das propriedades e análise de invariantes da rede de Petri.

A verificação estrutural dos modelos é realizada com recursos oferecidos pelo EA para verificação de integridade de modelo e análise de relaciona-

mento entre elementos do modelo. A verificação de integridade de modelo indica as falhas estruturais em relação aos metamodelos de SysML. A análise do relacionamento entre os elementos dos modelos é obtida com o recurso matriz de relacionamento do EA (figura 23), que permite estabelecer o relacionamento entre elementos do modelo e indicar os pontos e os tipos de ligação existentes, inclusive o relacionamento de alocação, utilizado para associar os *viewpoints* de requisitos aos elementos de especificação e modelagem. No caso da figura 23, extraída do caso Agronegócio (seção 4.3), a matriz mostra a relação de alocação entre os requisitos da indústria e os serviços identificados.

	Requisitos - Industria::Consolidação da Marca	Requisitos - Industria::Desenvolvimento de talentos	Requisitos - Industria::Evolução dos produtos	Requisitos - Industria::Expansão do market share	Requisitos - Industria::Formação equipe atendimento	Requisitos - Industria::Nova cultura de serviços	Requisitos - Industria::Padronização concessionários	Requisitos - Industria::Padronização operação	Requisitos - Industria::Padronização visual	Requisitos - Industria::Processos de treinamento	Requisitos - Industria::Utilização da experiência do cliente
Services::Acompanhamento pós-venda	↑			↑							↑
Services::Capacitação		↑			↑			↑	↑	↑	
Services::Capacitação EAD		↑			↑		↑	↑	↑	↑	
Services::Contrato de manutenção			↑			↑					↑
Services::Controle estoque de peças							↑	↑			
Services::Diagnóstico remoto GPRS			↑			↑					
Services::Informações Contábeis Concessionário							↑	↑			
Services::Informações Gerenciais Concessionário							↑	↑			↑
Services::Modelo referência arquitetura loja	↑			↑			↑	↑	↑		
Services::Modelo referência processos de negócio	↑			↑		↑	↑	↑			↑
Services::Processo de qualificação de RH		↑	↑	↑	↑		↑	↑		↑	
Services::Processo de venda	↑			↑			↑	↑			↑

Figura 23: Exemplo de Matriz de relacionamento de alocação

Existem várias propostas partindo de uma representação em UML, utilizando modelos em rede de Petri e suas extensões como uma via para simular e verificar os modelos. Muitos destes trabalhos oferecem métodos para construir redes de Petri que representem o comportamento dos sistemas partindo dos diagramas de sequência, diagramas de atividades, diagramas de estados, diagramas de caso de uso, ou diagramas de atividades (STAINES, 2010; WANG; DAGLI, 2008). No SoftDiss é utilizado o método proposto em Baresi e Pezzè

(2001) que transforma os diagramas de estado de UML (mesmo diagrama de estado utilizado em SysML) em redes de Petri. A figura 24 apresenta o diagrama de estado do indicador digital de consumo associado ao caso *Smart Grid* (seção 4.1).

Figura 24: Modelo Lógico de Transação de Serviço

O ForEnv define disciplinas para a verificação formal do comportamento da especificação, utilizando rede de Petri GHENeSys. A rede GHENeSys é obtida com a aplicação do método proposto em Baresi e Pezzè (2001), que permite a transformação de um diagrama de estado em uma rede de Petri. A figura 25 apresenta a rede GHENeSys resultante da transformação do diagrama de estados da figura 24.

Figura 25: Rede GHENeSys construída a partir do Diagrama de Estados

A análise das propriedades da rede e de seus invariantes permite verificar se o comportamento especificado no design do serviço atende os requisitos estabelecidos. A seção 4.1.1 apresenta o resultado desta análise para a aplicação *Smart Grid*.

3.6 Managerial Environment - ManEnv

O Ambiente Gerencial - ManEnv (*Managerial Environment*) oferece os serviços de gerenciamento das atividades e trocas de informação do SoftDiss, sustentando o planejamento e o controle das atividades, alocando os recursos e viabilizando o fluxo de informação entre os diversos ambientes.

O ManEnv concentra as atividades de gestão do processo de especificação e modelagem do SIS atuando na sincronização entre as atividades dos

demais ambientes. O ManEnv atua diretamente na manutenção dos fundamentos da orientação a serviço onde os recursos são ativos, integrados, alocados visando criação de valor e focados no cliente, que por sua vez, também, atua como um recurso ativo participando do processo de co-criação.

O ManEnv trata da manutenção do ciclo de vida dos SIS, desde a identificação até a verificação formal, incluindo também as requisições de revisão e melhorias nos SIS já em operação, resultando em nova especificação ou até mesmo a extinção dos SIS quando estes deixam de cumprir sua função principal.

A norma ISO/IEC 15288 provê ao ManEnv uma estrutura comum, integrada e compreensível para descrever e gerenciar todo o ciclo de vida de sistemas, desenvolvidos pelos mais variados tipos e tamanhos de organizações. Define um conjunto de processos e, também uma terminologia, que podem ser aplicados a todos os níveis hierárquicos associados ao desenvolvimento de sistemas. O ManEnv é sustentado especificamente pelos processos de planejamento, avaliação e controle, tratamento de risco e gerenciamento de configuração e informação.

O processo de planejamento estabelece o escopo, os recursos (materiais, humanos e financeiros) e o agendamento de tarefas do projeto suportado por SoftDiss, indicando o direcionamento e a infraestrutura necessárias para avaliação e controle do projeto.

O processo de avaliação coleta e avalia dados sobre o projeto que permitem o acompanhamento do projeto, comparando os resultados obtidos com as linhas de base definidas no planejamento, permitindo o monitoramento e avaliação de maturidade do projeto.

O processo de controle usa a avaliação para dirigir os esforços do projeto,

direcionando recursos para manter o progresso do projeto e corrigir eventuais desvios do plano.

A gestão de riscos lida com as incertezas no ciclo de vida do sistema. Este processo é usado para entender e evitar impactos potenciais para custo, cronograma e escopo do projeto, com uma abordagem proativa e estruturada.

O objetivo da gestão de configuração é garantir uma gestão eficaz da evolução e das diversas versões da especificação do sistema durante o seu ciclo de vida. A gestão de informação garante que os dados estejam devidamente armazenados, seguros, e acessíveis aos elementos da equipe de trabalho, garantindo a criação e manutenção da integridade do sistema de artefatos relevantes do ciclo de vida.

O ManEnv é implementado pela tecnologia MDG⁶ (*Model Driven Generation Technologies*) recurso oferecido pelo EA, que permite estender as capacidades de modelagem do EA para domínios e notações específicas. No caso do SoftDiss, MDG é utilizado para gerar a implementação do processo de gerenciamento, desenvolvimento do projeto e organização dos modelos. A figura 26 apresenta a estrutura de pacotes obtida pelo MDG SoftDiss.

⁶Sparx *Model Driven Generation Technologies* - http://www.sparxsystems.com.au/resources/mdg_tech/

Figura 26: Implementação do SoftDiss utilizando MDG

O MDG SoftDiss fornece um *template* de criação dos pacotes e as recomendações necessárias para a conjugação dos diversos métodos, técnicas, padrões e normas utilizados no SoftDiss. Na figura 18 foi apresentado o mo-

delo de pacotes resultante.

3.7 Síntese do Capítulo

O SoftDiss é um *framework* orientado a modelos para especificação de SIS, que suporta eliciação, modelagem e análise de requisitos, baseado em representações semi-formais (UML e SOMF) e formais (SysML e Petri Nets), visando antecipar a formalização da especificação e contemplar os diversos *viewpoints* de requisitos.

O SoftDiss direciona um processo de modelagem onde a entrada são especificações (funcionais, não-funcionais, processos de negócio e restrições) e, com base em "boas" práticas e no formalismo de esquemas trabalha a passagem destes requisitos (após análise) em especificação (formal) de SIS.

As cinco premissas (tabela 3) para concepção do SoftDiss foram atendidas por construção, conforme explicação na tabela 4.

Tabela 4: Atendimento às premissas de concepção do SoftDiss

#	Atendimento a premissa
P1	O fato do SoftDiss ser um <i>framework</i> orientado a modelos está presente no desenvolvimento de todo o sistema (portanto limita a sua inserção somente em ambientes que tenham a mesma orientação), mas é particularmente reforçada pelo ManEnv que gerencia o desenvolvimento de um SIS.
P2	O ReqEnv fornece ao SoftDiss os serviços de tratamento da eliciação e análise de requisitos, sendo a principal etapa no ciclo de vida do desenvolvimento de SIS, visando a sustentação do processo de co-criação com o cliente do serviço,
P3	O DesEnv e seus ambientes internos ConEnv, AnaEnv e LogEnv, propiciam a antecipação da formalização dos requisitos, viabilizando a obtenção de requisitos consistentes e completos o mais cedo possível e considerando a diversidade de variáveis e <i>viewpoints</i> que acometem um serviço.
P4	A norma ISO/IEC 15288 provê ao ManEnv uma estrutura comum, integrada e compreensível para descrever e gerenciar todo o ciclo de vida de sistemas, inclusive sobre os requisitos, fato indispensável para garantir o processo de negociação e validação com os representantes dos diversos <i>viewpoints</i> . Além disso a rastreabilidade entre os diversos elementos de especificação (incluindo requisitos) é mantida pelos recursos do EA e pelos serviços dos diversos ambientes do SoftDiss.
P5	O processo de desenvolvimento de SIS é baseado em "boas" práticas sustentadas por normas, procedimentos, técnicas, métodos, ferramentas e ambientes de apoio, incluindo, SOMF, EA, Volere, <i>Naked Objects</i> , UML, SysML, OOSEM, GHENeSys e ISO/IEC 15288.

Portanto, SoftDiss é um *framework* de eliciação, documentação, análise, modelagem de requisitos de SIS. O próprio SoftDiss serve assim a desenvolvedores de SIS na tarefa de modelagem e design até chegar a uma especificação formal do sistema-alvo (*system of interest*).

4 RESULTADOS OBTIDOS

O uso de modelos no processo de design de sistemas de informação segue a tendência atual proposta pela OMG e pelo INCOSE ((INCOSE, 2007)). A aderência do design por modelos ("*model driven design*") a sistemas de serviço se justifica pela premissa de que o processo de automação aplicado aos sistemas de serviço seja implementado através dos sistemas de informação - como uma forma interativa e que admite co-criação - e por razões que serão mostradas nos exemplos de aplicação, tais como flexibilidade, aderência a requisitos voláteis, aderência a sistemas com um alto número de *viewpoints*, e facilidade de reutilização. No entanto, nenhuma destas vantagens pode ser formalmente demonstrada, especialmente quando tratadas na fase inicial do processo de *design*, isto é, na eliciação análise de requisitos e compatibilização destes com os processos de negócio. Assim, neste capítulo são apresentados os resultados da aplicação do SoftDiss a três projetos de design de sistemas de informação de serviço (SIS), devidamente escolhidos por terem características diferentes e portanto apresentares desafios diferentes ao processo.

O primeiro projeto tem como base a automação flexível do fornecimento de energia elétrica (*Smart Grid*); o segundo está associado a um ambiente (acadêmico) de projeto, especificamente ao D-Lab (que em essência é similar a inúmeras empresas de consultoria e design de serviço); o terceiro projeto trata a especificação dos SIS associados ao processo de automação no agro-

negócio, tendo como foco a agricultura de precisão, especificamente na escolha e aquisição de equipamentos agrícolas. O apêndice A contém os modelos SoftDiss completos associados a cada um dos três projetos.

Os casos em estudo foram escolhidos de acordo com os seguintes critérios: em primeiro lugar um representante dos sistemas do setor terciário e quaternário¹ ao mesmo tempo, isto é, um sistema dedicado a serviços não-básicos (e não associados a um processo produtivo do setor secundário) e que inclui um alto nível de automação como base do seu funcionamento. Este é o caso de uma gama enorme de sistemas emergentes, e em especial é o caso dos sistemas *Smart Grid*. A particularidade de vários destes sistemas e em especial dos *Smart Grids* é que o design e desenvolvimento conceitual do sistema antecede à criação do serviço. Assim, estes sistemas são desenhados a partir de modelos de referência aos quais se acrescenta os serviços e não criando os serviços a partir dos requisitos.

Como segundo caso, tem-se um sistema pertencente aos setores quaternário e quinário², onde o processo de produção e serviço está inteiramente baseado em conhecimento, já incluindo mecanismos de troca simbólica nos seus processos de negócio.

Finalmente foi considerado um sistema pertencente ao setor primário mas com alto grau de automação, fugindo portanto dos sistemas primários clássicos. A proposta neste caso é analisar a incursão dos serviços e dos processos de automação baseados em SIs e sistemas de serviço nos setores mais tradicionais que seriam o primário e secundário. A falta de um caso do setor secundário (industrial) se justifica porque neste caso (onde se destacam os sistemas *Supply Chain*), tem-se por um lado uma discussão sobre a simbiose

¹Quaternário, setor econômico que inclui os serviços de informação e conhecimento (TI, telecomunicações, educação, P&D, planejamento, consultoria, e outros) ((KELLERMAN, 1985))

²Quinário, setor econômico que inclui os serviços sem ligação direta de lucro (saúde, educação, cultura, ONG's, e outros)((KELLERMAN, 1985))

entre produto e serviço e uma discussão já avançada sobre *manufacturing services* (DUTRA; OLIVEIRA; SILVA, 2013). Estas duas discussões entretanto transcendem ao processo de design e fogem ao foco deste trabalho que é o problema da modelagem e design de serviço.

Assim, este capítulo mostra a abrangência do SoftDiss e sua aplicabilidade prática a diferentes tipos de SIS. Como a proposta original é também propor um *framework* de utilidade prática, o SoftDiss foi inserido no sistema comercial Enterprise Architect (EA) da SparX, como já mencionado na seção 3.6, onde também se encontra o *framework* original do SOMF. Um acordo de cooperação foi assinado entre o D-Lab e a SparX para este fim.

4.1 Projeto de SIS Smart Grid

O *Smart Grid* vem promovendo a automação no setor de energia incorporado a conceitos de sustentabilidade e racionalização da produção e uso de energia. Os benefícios obtidos com o *Smart Grid* podem ser significativos, permitindo aos consumidores, entre outras coisas, exercer um melhor controle do consumo de energia, integrar uma nova geração de aparelhos elétricos como veículos híbridos e eletrodomésticos, e associar diferentes fontes de geração de energia, visando aumentar a eficiência energética e conseguir um melhor aproveitamento vindo de fontes renováveis.

De acordo com EPRI (2008) o termo *Smart Grid* é a sobreposição de uma comunicação unificada e de sistemas de controle sobre a infraestrutura existente de rede de energia para fornecer informação às entidades interessadas. Isso o coloca em um domínio híbrido (terciário/quaternário) na medida em que presta um serviço direto baseado na informação (não exatamente "conhecimento") vinda dos usuários. Conhecimento pode ser gerado pelo uso (deri-

vado da co-criação) e absorvido pelos insumos ditos "inteligentes" do sistema. Na prática, o *Smart Grid* otimiza a distribuição e entrega de energia, minimiza as perdas, faz auto ajustes e viabiliza uma próxima geração de dispositivos e aplicações voltados à eficiência energética.

O *Smart Grid* de baixa tensão (GOMES; PRINTES; RAMOS, 2010) é um caso particular e um componente importante do modelo de referência *Smart Grid* cujo objetivo é gerenciar a distribuição da energia disponível (ainda que predominantemente elétrica) para consumidores individuais, institucionais e comerciais, de forma sustentável, sem desperdícios, na medida exata das suas necessidades, evitando desvios e perdas. Por simplicidade, nesta aplicação será considerado apenas o sistema restrito ao varejo, isto é, são excluídos os usuários que demandam serviços mais sofisticados providos por subestações.

O *Smart Grid* de baixa tensão tem a concessionária de energia elétrica como base de distribuição, *clusters* de transformadores onde é tratada a distribuição da carga e usuários finais conectados a um dos transformadores. Esta arquitetura é relacional e informada, isto é, os usuários podem gerenciar o seu consumo, enviar restrições, interrupções do serviço ou preferências, de modo a prover sustentabilidade, racionalidade no uso da energia e evitar desperdícios. O problema é também simplificado ao supor que estes usuários são só consumidores e não podem se transformar em pequenos fornecedores de energia (que é o caso mais comum no Brasil).

As informações sobre a situação dos transformadores são usadas pelo nível intermediário que estabelece o balanço de carga e a informação sobre os *clusters* é sintetizada pela concessionária. Mesmo com todas estas simplificações tem-se um sistema em três camadas (uma árvore) com informação fluindo nos dois sentidos: das folhas para a raiz e da raiz para as folhas. As simplificações foram feitas exatamente para reduzir a arquitetura do problema

de um grafo mais sofisticado a uma árvore n-aria. Sobre este sistema flui ao mesmo tempo a energia fornecida e a informação sobre parâmetros do sistema e sobre preferências e/ou configurações dos usuários. O objetivo final é o serviço, isto é, o atendimento a diferentes demandas de energia pelos usuários finais, de forma racional (do ponto de vista da distribuidora) e sustentável (do ponto de vista dos reguladores). Tem-se, portanto um intrincado sistema de *viewpoints*, que restringe o fluxo de energia e cuja realização maior se dá pelo atendimento racionalizado de todos os *viewpoints* simultaneamente.

O tratamento proposto é justamente o de um sistema de serviço automatizado, aqui traduzido em um sistema de informação acoplado a um sistema de provimento direto de serviço e a um sistema de gestão e balanço de carga. Note-se que a realização do negócio está ligada justamente à integração entre osistema de informação, o provimento de energia e o balanço de carga, portanto é quase inviável sem a devida harmonização destes componentes.

O gerenciamento de requisitos é determinante para o sucesso dos projetos, quando os ambientes de desenvolvimento e de aplicação são disjuntos, isto é, usuários e *stakeholders* pertencem a um domínio com pouca ou nenhuma interseção com a área da engenharia, informática e automação, os processos de eliciação e análise de requisitos se tornam bastante complicados (QUEIROZ; SILVA, 2009). Uma forma para amenizar a dificuldade de tratar domínios disjuntos é a utilização de modelos ou arquiteturas de referência, que possibilitem a aproximação entre os ambientes de desenvolvimento e de aplicação.

O projeto IntelliGrid³, originalmente denominado projeto IECSA (*Integrated Energy and Communications System Architecture*), é patrocinado pelo EPRI⁴-

³IntelliGrid (<http://intelligrid.epri.com/>)

⁴EPRI - *Electric Power Research Institute* (<http://www.epri.com>)

Electric Power Research Institute com o objetivo de criar uma base técnica para um *Smart Grid*. Teve como principal resultado em 2003, a proposta da *IntelliGrid Architecture*⁵, baseada em padrões abertos, orientada a requisitos e destinada a integração de redes de dados e equipamento, para permitir a interoperabilidade entre produtos e sistemas, fornecendo às concessionárias de energia elétrica métodos, ferramentas e recomendação de padrões e tecnologias para a implementação de sistemas (HUGHES, 2003a).

O projeto IntelliGrid foi desenvolvido por um grupo de especialistas gerenciados pela General Electric com contribuições da *Utility Consulting International* (UCI), SISCO, Lucent, EnerNex, Hypertek e outros. O resultado foi a arquitetura de referência *IntelliGrid Architecture*, uma proposta de organização sistêmica do *Smart Grid*, organizando em seus componentes as relações existentes com outros sistemas, incluindo os princípios orientadores da sua concepção e evolução e, também, a organização de tipos comuns de requisitos.

O projeto foi desenvolvido visando capturar e organizar os requisitos das funções do *Smart Grid* e, também, foi estruturado de forma a integrar as necessidades de negócio de operação dos sistemas de energia, as visões estratégica e tática, e permitindo aplicar diversas tecnologias para acomodar sistemas legados, multiplicidade de fornecedores e tecnologias, padrões, tecnologias e melhores práticas. Foram identificadas e registradas cerca de 400 funções. Consideração de critérios de configuração, qualidade de serviço, segurança e gerenciamento de dados, originaram um subconjunto de funções chave, que foram detalhadas, capturadas em narrativas e diagramas UML (HUGHES, 2003b). Um resultado relevante da importância e da aplicabilidade da *IntelliGrid Architecture* no Brasil, encontra-se na CEMIG (Companhia Ener-

⁵*IntelliGrid Architecture* (<http://www.intelligrid.info/>)

gética de Minas Gerais), que, de acordo com McGranaghan e Dollen (2008), calçou o desenvolvimento de um projeto *Smart Grid* nos resultados desta arquitetura.

Considerando as semelhanças entre o Brasil e os EUA, quando considerada a rede de baixa tensão, nesta aplicação a *Intelligrid Architecture* é utilizada como arquitetura de referência, permitindo uma abordagem baseada em domínios próximos para tratamento de requisitos e especificação de um novo sistema de informação de serviço (SIS) para *Smart Grid* de baixa tensão (OLIVEIRA et al., 2012; OLIVEIRA; SILVA, 2011).

4.1.1 SoftDiss aplicado ao Smart Grid Urbano

O problema de entrada para o SoftDiss é a especificação de um SIS destinado a controlar o Indicador Digital de Consumo – IDC (GOMES; PRINTES; RAMOS, 2010), denominado sistema alvo. O ReqEnv do SoftDiss é suprido com as informações da *IntelliGrid Architect*, visando a especificação do sistema alvo baseado nas seguintes funções:

- Solicitar autenticação na rede de distribuição: deve solicitar autenticação, esperar pela resposta e depois armazenar a autenticação.
- Mostrar o consumo normal ao usuário: deve receber uma mensagem, que será tratada para sua identificação, para processar os dados de medição e mostrar na tela o consumo.
- Permitir ao usuário solicitar a potência instantânea utilizada: o usuário pode receber a potência instantânea (PI) pressionando um botão, onde o IDC solicita à Unidade de Distribuição (UD) a PI, espera pela resposta e exibe a PI na tela.

- Mostrar mensagens da Central de Supervisão (CS) para o usuário final: semelhante à exibição da PI, entretanto as mensagens de CS, consumo normal e PI, não podem ser exibidos ao mesmo tempo.

A figura 27 apresenta a estrutura de pacotes do SoftDiss. Observar que esta estrutura de pacotes ainda é uma versão preliminar do SoftDiss, com diferenças significativas no ReqEnv e no LogEnv em relação a versão atual do SofDiss (figura 26 da seção 3.6).

Figura 27: Estrutura de pacotes SoftDiss para *Smart Grid*

O ReqEnv (seção 3.3) utiliza o modelo de referência *IntelliGrid Architect* para capturar os principais *viewpoints* de requisitos. Nas versões iniciais do SoftDiss os requisitos dos diferentes *viewpoints* ainda eram modelados em diversos tipos de diagramas (requisitos, casos de uso, objetos, e outros), com evolução para o diagrama de requisitos e *Naked Objects* nas versões subsequentes do SoftDiss. A figura 28 apresenta a estrutura de pacotes da modelagem dos requisitos representando os diversos *viewpoints*, mostrando a capacidade do SoftDiss para tratar problemas com larga escala de *viewpoints*, requisitos e sistemas interligados e legados.

Figura 28: Pacotes do modelo de requisitos do *Smart Grid*

A figura 29 apresenta o detalhe do pacote *IntelliGrid Specification* com os principais requisitos extraídos do modelo de referência *IntelliGrid*. As características do projeto IntelliGrid oferecem a este extenso número de requisitos um alto grau de confiabilidade e adequação ao problema.

Figura 29: Requisitos associados ao IntelliGrid

O DesEnv (seção 3.4) oferece os insumos para identificar e analisar os serviços necessários para resolver o problema de entrada do SoftDiss e resulta na especificação formal do SIS (sistema alvo), através da realização dos três ambientes internos do DesEnv: ConEnv (seção 3.4.1), AnaEnv (seção 3.4.2) e LogEnv (seção 3.4.3).

O ConEnv utiliza os serviços do ReqEnv para determinar os atributos essenciais para *Smart Grid*: sustentabilidade (*sustainability*), manutenção (*maintenance*), queda de energia (*disruption*), perdas (*losses*), segurança (*security*) e escolhas do consumidor (*choice*). O modelo de atributos da figura 30 é obtido a partir da combinação dos atributos essenciais no primeiro nível e a combinação destes atributos nos níveis imediatamente inferiores, sendo o último nível a convergência de todos os atributos. Mesmo contemplando apenas os dois principais *viewpoints* (consumidor e provedor de energia) e procurando restringir o número de atributos essenciais, a complexidade e a abrangência do *Smart Grid* está refletida no modelo de atributos.

Figura 30: Modelo de atributos do *Smart Grid*

O modelo de atributos oferece informações para a identificação e descoberta de serviços, através da análise das combinações obtidas e do caminho percorrido na rede, permitindo a determinação de árvores de decisão que orientam na identificação, descoberta e, também, oferecem um instrumento de rastreabilidade entre os atributos e os serviços que serão identificados no processo de identificação dos serviços conceituais, obtendo como resultado a árvore de decisão da figura 31.

Figura 31: Árvore de decisão conceitual do *Smart Grid*

A figura 31 é uma árvore de decisão obtida a partir dos atributos essenciais - perda, manutenção e queda de energia. A árvore de decisão possui em sua raiz o atributo perda e a cada nível a decisão associada ao atendimento dos demais atributos, manutenção e queda. Nas folhas da árvore estão os serviços conceituais identificados ligados aos devidos atributos e agrupados em *cluster*. Os requisitos necessários para a descoberta e identificação dos serviços conceituais representados na figura 31 são obtidos através do controle do

ManEnv (seção 3.6) e da negociação com os diversos *viewpoints* fornecidos pelo ReqEnv.

Os serviços conceituais identificados podem ser classificados e agrupados através dos métodos de associação conceitual, organizando os serviços em diversas possibilidades de arquitetura: hierárquica, estrela, rede e circular. A figura 32 apresenta o modelo conceitual de associação para os serviços identificados na figura 31. Os serviços conceituais da figura 32 estão conectados aos seus respectivos consumidores:

- *Consumer*, representando os consumidores urbanos (residenciais e comerciais, excluindo consumo industrial intensivo) de energia elétrica.
- *System Operator*, representando o órgão regulamentador de energia nacional, no caso a ANEEL.
- *Energy Service Provider*, representando a concessionária prestadoras dos serviços associados à energia elétrica.

Figura 32: Associação Conceitual do *Smart Grid*

Os modelos de atributos, árvore de decisão e modelo de associação representam o modelo conceitual orientado a serviço, resultado obtido no ConEnv através da aplicação das disciplinas Conceitualização de Serviço (*Service Conceptualization*) e Arquitetura Conceitual (*Conceptual Architecture*) do SOMF (BELL, 2008), com rastreabilidade entre os serviços, atributos e associações sustentadas pela ferramenta EA e pela atuação do ManEnv, gerenciando e

negociando o atendimento às demandas dos diversos *viewpoints* de requisitos mantidos pelo ReqEnv.

O AnaEnv trata a transformação do serviço conceitual inicial em um serviço que propicie categorização mais aprofundada analisando o domínio do problema, entendendo as soluções conceituais existentes na organização, avaliando se o serviço proposto resolve o problema e, caso não resolva, qual a lacuna existente, analisando os sistemas legados e modelando uma proposta de solução.

O resultado da etapa de Análise e Descoberta de Serviço do AnaEnv é um modelo de análise contextual representado na figura 33, onde o serviço RTP – *Real Time Pricing*, após as práticas de análise, sempre com a atuação do ManEenv e suporte do ReqEnv, é refinado em diversos serviços atômicos agrupados no espaço de contenção de serviço.

Figura 33: Análise contextual do serviço RTP

O modelo de análise estrutural representa a manipulação estrutural da análise dos serviços, considerando a relação entre um serviço e seus serviços atômicos. No caso da figura 34 os serviços RTP e AMR (*Automatic Meter Reading*) estão ligados por um conector de acoplamento e cada um dos ser-

viços RTP e AMR estão expandidos e associados a seus serviços atômicos através de um conector de composição.

Figura 34: Análise estrutural dos serviços RTP e AMR

A disciplina Integração de Negócios do AnaEnv é o último estágio do SoftDiss sustentado pelo SOMF e permite representar a integração dos serviços analisados com os domínios de negócio. A figura 35 apresenta o modelo de integração de negócio, onde os serviços RTP e AMR estão ligados pelo mesmo conector de acoplamento, os domínios de negócio - Operação de Distribuição, Operação de Mercado, Serviços do Consumidor e Recursos Distribuídos - estão integrados a perspectivas de Operação e Manutenção, que por sua vez integram os serviços RTP e AMR, ora por operação, ora por manutenção ou por ambas. Os domínios e perspectivas foram eliciados pelo ReqEnv tendo como referência o modelo de referência IntelliGrid.

Figura 35: Integração de negócio dos serviços RTP e AMR

O resultado obtido no AnaEnv é um vasto conjunto de informações que representa os serviços identificados, conceituados, analisados e integrados com as necessidades de negócio, com boas práticas de rastreabilidade e atendendo aos diversos *viewpoints* de requisitos, mas é uma modelagem representada em linguagem semi-formal SOMF.

O LogEnv trata a transformação dos serviços analíticos em elementos mais tangíveis denominados serviços lógicos - representados pelo Modelo de Design Orientado a Serviço - permitindo retratar os relacionamentos, a estrutura e o comportamento dos serviços viabilizando a criação de uma arquitetura lógica para os serviços. A representação da especificação passa a ser formal em SysML e oferecendo recursos de verificação e validação da especificação

resultante.

No ciclo inicial de desenvolvimento do SoftDiss, a transformação dos serviços analíticos nos serviços lógicos ainda não eram aplicados o método OO-SEM e a norma ISO/IEC 15288, apenas um conjunto de processos baseados em MBSE eram utilizados, sustentados pelos recursos da ferramenta EA e a representação em SysML.

Os serviços identificados e analisados no ConEnv e AnaEnv, priorizados pela atuação do ManEnv, indicaram como um dos sistemas alvo o SIS destinado a controlar o IDC⁶ (Indicador Digital de Consumo).

No estudo do *Smart Grid* um dos objetivos foi determinar a viabilidade do SoftDiss para suprir uma verificação formal ao atendimento de conjunto de requisitos oriundos de diversos *viewpoints*, sendo assim, os serviços prestados pelo LogEnv objetivaram obter um modelo em SysML que viabilizasse esta verificação.

A figura 36 apresenta a especificação caixa-preta do SIS de controle do DIC representando, em um diagrama de blocos da SysML, os componentes do sistemas.

⁶IDC ou em inglês DIC (*Digital Consumption Indicator*)

Figura 36: Diagrama de blocos do SIS DIC

O diagrama da figura 36 sustenta a criação do diagrama de estado (figura 37) principal do funcionamento do SIS de controle do DIC, elemento básico para a próxima etapa do SoftDiss, a aplicação dos serviços do ForEnv (seção 3.5).

Figura 37: Diagrama de estado do IDC

Como apresentado na seção 3.5, o modelo resultante do LogEnv, representado em SysML, é transferido para o ForEnv utilizando os mapeamentos de transferência MDA oferecidos pelo EA. No estudo do *Smart Grid*, os serviços do ForEnv permitem que o diagrama de estado da figura 37 seja transformado em uma rede de Petri utilizando-se o método proposto em Baresi e Pezzè (2001). A figura 38 apresenta a rede GHENeSys resultante da transformação do diagrama de estados da figura 37.

Figura 38: Rede GHENeSys construída a partir do diagrama de estados

O sistema alvo no estudo do *Smart Grid* é a especificação de um SIS destinado a controlar o Indicador Digital de Consumo – IDC (GOMES; PRINTES; RAMOS, 2010) e que deve satisfazer aos requisitos dos diversos *viewpoints*, mantidos pelo ReqEnv e oriundos do modelo de referência *IntelliGrid Architecture*. Visando demonstrar os serviços de verificação formal do ForEnv, o IDC deve atender aos seguintes requisitos:

- [Req1] Solicitar autenticação na rede de distribuição: deve solicitar autenticação, esperar pela resposta e depois armazenar a autenticação.
- [Req2] Mostrar o consumo normal ao usuário: deve receber uma mensagem, que será tratada para sua identificação, processar os dados de medição e mostrar na tela o consumo normal.
- [Req3] Permitir ao usuário solicitar sua potencia instantânea: o usuário pode receber a potência instantânea (PI) pressionando um botão, onde o IDC solicita à UD a PI, espera pela resposta e exibe a PI na tela.
- [Req4] Mostrar mensagens da CS para o usuário final: semelhante à exibição da PI, entretanto as mensagens de CS, consumo normal e PI, não podem ser exibidos ao mesmo tempo.

A verificação formal dos requisitos é realizada através dos serviços de análise de propriedades das redes de Petri, especificamente a análise de invariantes (seção 2.3) de uma rede GHENeSys.

No estudo do *Smart Grid* foi considerado um caso simplificado, mas que pode ser expandido para situações reais de maior volume, devido a características específicas da rede GHENeSys concebida como uma rede de Petri estendida, com conceitos de orientação a objetos, mecanismos de abstração baseado em hierarquias e síntese com abordagem estruturada. Além disso, o DLab mantém o projeto GHENeSys, responsável pelo desenvolvimento e suporte de uma ferramenta de software aberto destinada ao tratamento de redes GHENeSys.

A figura 38 apresenta a rede de Petri GHENeSys resultante da transformação do diagrama de estados da figura 37. A figura 39 mostra o resultado do cálculo dos invariantes, obtidos pelo Sistema GHENeSys, para esta rede.

Place	Transition
Inicio	1.0
Sol_Aut	1.0
EsperaAut	1.0
ArmacenaAut	1.0
EsperaAção	1.0
TrataMens	1.0
ConsNormal	1.0
MensCS	1.0
SolicitaIP	1.0
EsperaPI	1.0
TrataPI	1.0
Fim	1.0
MostrarLCD	1.0

Place	Transition
a1	1.0 0.0 0.0 0.0
a2	1.0 0.0 0.0 0.0
a3	1.0 0.0 0.0 0.0
a4	1.0 0.0 0.0 0.0
a5	1.0 1.0 1.0 0.0
a6	1.0 1.0 0.0 0.0
a7	1.0 0.0 1.0 0.0
a8	1.0 0.0 0.0 1.0
a9	1.0 1.0 0.0 0.0
a10	1.0 0.0 1.0 0.0
a11	1.0 0.0 0.0 1.0
a12	1.0 0.0 0.0 1.0
a13	1.0 0.0 0.0 1.0

Figura 39: Invariantes lugar (esquerda) e transição (direita)

Os invariantes de lugar obtidos e exibidos na figura 39 permitem definir a seguinte equação:

$$\begin{aligned}
 &M(\text{Inicio}) + M(\text{SolAut}) + M(\text{EsperaResp}) + M(\text{TrataPI}) + M(\text{Fim}) + \\
 &M(\text{EsperaAut}) + M(\text{ArmazenaAut}) + M(\text{EsperaAcao}) + M(\text{TrataMensagem}) + \\
 &M(\text{ConsumoNormal}) + M(\text{MostrarLCD}) + M(\text{MensagemCS}) + M(\text{SolicitaIP}) = 1
 \end{aligned}$$

Analisando os invariantes de lugar e a equação obtida conclui-se que os processos que mostram os dados da medição e mostram os dados da potência instantânea (PI) não são realizados ao mesmo tempo, portanto na tela (LCD) só poderá ser exibida uma das mensagens, que é o comportamento

esperado para os requisitos [Req2], [Req3] e [Req4]. Considerando os invariantes de transição obtidos pode-se concluir que as transições a1, a2, a3 e a4, são disparadas pelo menos uma vez, o que significa que a autenticação sempre se realiza [Req1], atendendo requisitos fundamentais do IDC.

4.2 Projeto Design Lab

O Design Lab⁷ (D-lab) pertence ao grupo de pesquisa do CNPq⁸ com foco no design, modelagem de sistemas, controle e auxílio a tomada de decisões. O D-Lab está fisicamente localizado no campus da Universidade de São Paulo - USP, Departamento de Engenharia Mecatrônica e Sistemas Mecânicos da Escola Politécnica.

O D-Lab é eminentemente um laboratório de pesquisa em *Engineering Design*, co-criando valor através da relação de produção de conhecimento a partir da interação de entidades e aplicando competências especializadas (conhecimento e habilidades) para benefício do próprio D-Lab e da comunidade (alunos, empresas, instituições de pesquisa e colaboradores externos), através da titulação de alunos, colaboração com outros grupos de pesquisa, e/ou com instituições e pesquisadores externos.

Os objetivos gerais do D-Lab estão relacionados com o avanço do conhecimento acadêmico, com o desenvolvimento de projetos com foco na criação de objetos artificiais, sempre em torno do tema *Engineering Design*. Alguns objetivos específicos são: formação de pessoal (pesquisadores) nos diversos níveis; associação com inovação acadêmica e aplicações; criação de uma cultura multidisciplinar; conexão com entidades internacionais; e contribuição

⁷D-Lab - Design Lab (<http://dlab.poli.usp.br>)

⁸CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico (<http://www.cnpq.br>)

(serviço) a programas de pós-graduação.

Os processos de negócio do D-Lab, sintetizados na figura 40, permitem que se atinjam os objetivos gerais e específicos. Na entrada estão os diversos alunos (iniciação científica, mestrado e doutorado), os pós-doutores, os engenheiros atuantes em projetos, os colaboradores externos e as empresas parceiras. Os insumos para os processos vem dos recursos físicos e econômicos (fomentos), da infraestrutura fornecida pelo PPGEM⁹ e dos insumos dos projetos externos de desenvolvimento. O objetivo específico, formação de pessoal, é alcançado com as titulações obtidas pelos alunos. Entretanto, deve se observar que diversos tipos de produtos são entregues como saída do processo: artigos, livros, relatórios técnicos e inovação na forma de projetos de desenvolvimento.

Figura 40: Processos de negócio do D-Lab

O D-Lab tem seus interesses divididos em dois ramos: o primeiro vol-

⁹PPGEM - Programa de Pós-Graduação em Engenharia Mecânica da Escola Politécnica da USP

tado ao processo de projeto; e o segundo voltado aos sistemas inteligentes e sistemas especiais (pela complexidade, natureza crítica, etc.). A figura 41 apresenta as linhas de pesquisa do D-Lab. No topo, ligado ao tema central, *Engineering Design*, tem-se a linha de análise de requisitos e métodos de design. À esquerda os métodos formais de modelagem, verificação, *model checking* e análise, e à direita os métodos inteligentes, começando pelos métodos de *AI Planning*, passando pela inteligência de máquina e terminando com os métodos colaborativos, baseados em SIS. A parte inferior mostra os produtos, resultado de projetos de desenvolvimento interno, tendo como base dois sistemas de software, Sistema GHENeSys e itSIMPLE.

Figura 41: Linhas de pesquisa e produtos do D-Lab

O SIS D-Lab é um sistema de informação de serviço proposto para auxiliar o D-Lab a monitorar a coerência e ligação entre os conceitos, o conhecimento e o grau de maturidade de cada integrante, de modo a permitir um planejamento do investimento acadêmico e dos resultados. O sistema deve servir também de ligação entre os trabalhos em desenvolvimento e entre os membros. O SIS D-Lab deve ser o elo de ligação entre os membros do D-Lab e

entre o D-Lab e o público em geral. O próprio SIS DLab é um projeto de sistema de informação inteligente e baseado no conceito de ambiente cognitivo de design. O desenvolvimento deste sistema faz uso da própria teoria de design desenvolvida no D-Lab. O SIS D-Lab deve ser um ambiente colaborativo e institucional.

A proposta visa dar ao D-Lab uma estrutura corporativa, de modo a integrar as atividades inerentes a este, sejam acadêmicas, de difusão, formação direta (graduação, pós-graduação e extensão), suporte e consultoria a empresas ou órgãos de governo, ou o desenvolvimento de sistemas e software - que podem ser de domínio público, para suporte acadêmico ou para fins comerciais.

O ponto chave para esta atuação é a integração dos processos, de modo que as técnicas estudadas sejam aplicadas (ganhando inclusive *feedback* para o trabalho acadêmico) e sejam virtualizadas. Deve ainda permitir uma maior sinergia entre os participantes, membros do D-Lab, colaboradores, etc., permitindo a todos ter uma experiência de trabalho em equipe.

O projeto de desenvolvimento do SIS D-Lab pretende criar uma "cultura D-Lab" melhorando a infraestrutura e compartilhando o mesmo conjunto de ferramentas de modo a agilizar a colaboração e reduzir os custos operacionais. O sucesso do projeto passa pela distribuição em diferentes grupos de *stakeholders* e contribuição para atendimento aos respectivos *viewpoints*. No final espera-se um D-Lab que seja uma corporação para prestação de serviço integrada por um sistema de informação cognitivo, tendo como atributos:

- visibilidade acadêmica (denotado pelo número de acessos e consultas da comunidade acadêmica)
- visibilidade técnica (denotado pelo número de acessos e consultas vin-

das do mercado e de empresas de base tecnológica

- referencia em formação (denotado pelo número de consultas por parte de alunos de graduação, pós-graduação e de extensão)
- reciprocidade (denotado pelo número de participações em projetos de desenvolvimento e consultoria)
- visibilidade institucional (denotado pelo número de consultas da própria USP, órgãos de governo, etc.)

A aplicação do SoftDiss objetiva a especificação do SIS D-Lab, considerando os requisitos dos diversos *viewpoints* envolvidos e viabilizando o processo dinâmico de co-criação entre fornecedor e consumidor, inerente a um sistema de serviço e fundamental para um ambiente de pesquisa e inovação.

4.2.1 SoftDiss aplicado ao D-Lab

O problema de entrada para o SoftDiss é a especificação do SIS D-Lab. A figura 42 apresenta o diagrama de pacotes com os diversos *viewpoints* dos requisitos eliciados. Nesta aplicação do SoftDiss, diferentemente da aplicação *Smart Grid*, os *viewpoints* de requisitos passam a ser tratados no ReqEnv considerando a disciplina de modelagem *Naked Objects*, o Volere como recurso de documentação de processo e recursos MBSE associados ao EA.

Figura 42: *Viewpoints* de requisitos do D-Lab

O grau de prioridade dado pelo D-Lab aos diversos *viewpoints* é distinto, bem como a capacidade de prover recursos a estes *viewpoints*. A figura 43 apresenta *Naked Objects* associados aos membros internos do D-Lab (pesquisadores que estão alocados dentro do D-Lab), que representam a maioria dos agentes e com alta demanda de recursos.

Figura 43: *Naked Objects* associados aos membros do D-Lab

A figura 44 apresenta o diagrama de requisitos dos membros internos do D-Lab. Estes requisitos podem ser aplicados, com alguma adaptação, a empresas de consultoria e grandes centros de pesquisa e desenvolvimento, con-

cedendo ao SoftDiss do D-Lab grande potencial para sustentar a modelagem de serviços destes dois tipos de entidade.

Figura 44: Diagrama de requisitos dos membros internos do D-Lab

Contrastando com os membros internos do D-Lab, a figura 45 apresenta *Naked Objects* associados aos membros externos do D-lab (pesquisadores que trabalham em projetos do D-Lab alocados em instituições espalhadas pelo mundo) e que possuem demandas de recursos do D-Lab, mas incorrem em restrições de acesso e segurança. Portanto a especificação dos serviços prestados pelo SIS D-Lab deve contemplar estes contrastes.

Figura 45: *Naked Objects* dos membros externos do D-Lab

O DesEnv oferece os recursos para identificar e analisar os serviços necessários para solucionar o problema da especificação do SIS D-Lab e resultar na especificação formal, contando com vários níveis de abstração da representação orientada a modelos. A figura 46 apresenta o diagrama de pacotes com a organização do DesEnv para o D-Lab.

Figura 46: Organização do Modelo DesEnv do D-Lab

Os requisitos do ReqEnv são a referência para a negociação junto aos *stakeholders* do D-Lab para selecionar os atributos essenciais (seção 3.4.1) e

obter modelo de atributos de serviço (figura 47). O modelo de atributos orienta a identificação, priorização e conceituação dos serviços.

Figura 47: Modelo de atributos do D-Lab

O modelo de atributos oferece a base para o processo de identificação e descoberta de serviços, pois a análise das combinações obtidas entre os atributos permite construir árvores de decisão, onde as ramificações são as condições de aderência aos atributos. A árvore de decisão da figura 48 apresenta o conjunto de serviços identificados e associados aos atributos visibilidade e comunicação do caso D-Lab.

Figura 48: Árvore de decisão dos serviços conceituais do D-Lab

No caso do D-Lab foi elaborada apenas uma árvore de decisão dos serviços conceituais, suficiente para identificar os principais serviços. Entretanto, em outras aplicações poderiam ser elaboradas várias árvores considerando outros atributos essenciais.

Os serviços identificados e associados no ConEnv (seção 3.4.1) são tratados pelo AnaEnv (seção 3.4.2) em busca de uma categorização mais aprofundada analisando o domínio do problema, entendendo as soluções conceituais existentes na organização e avaliando se o serviço proposto resolve o problema de especificação do SIS D-Lab. Os detalhes desta etapa da modelagem foram utilizados como exemplo na apresentação do SoftDiss (seções 3.4.1 e 3.4.2) e, também, estão detalhadas no apêndice A.2.

O resultado obtido é o conjunto de serviços identificados, conceituados, analisados, integrados e, principalmente, priorizados em função dos atributos essenciais (figura 49).

Figura 49: Serviços resultantes do ConEnv e AnaEnv para o D-Lab

No LogEnv, etapa final do DesEnv, a representação semi-formal SOMF dos serviços é transformada para uma representação em SysML, com características de representação precisa e livre de ambiguidades.

O processo Especificação e Design OOSEM (figura 19 da seção 3.4.3) define as atividades a serem realizadas. A primeira atividade é a “Análise das Necessidades dos *Stakeholders*” que captura o estado atual (*as-is*) dos sistemas corporativos, suas limitações e possibilidades de melhorias. A figura 50 apresenta as sub atividades definidas para a análise das necessidades dos *stakeholders*.

Figura 50: Análise das Necessidades dos *stakeholders* (FRIEDENTHAL; MOORE; STEINER, 2009)

Como resultado da análise das necessidades dos *stakeholders* são produzidos modelos para representar o domínio operacional (*as-is*), os requisitos de domínio (*as-is*), os parâmetros de eficiência, casos de uso de negócio e, finalmente, o modelo de domínio operacional do sistema alvo (*system of interest*).

O D-Lab possui uma série de sistemas e processos já implantados e em produção e que necessitam ser avaliados quanto a continuidade e evolução. A figura 51 apresenta o diagrama de blocos com a estrutura do modelo operacional *as-is* do D-Lab (atividade “Caracterização *as-is* do sistema e da organização” da figura 50). São considerados vários sistemas legados, o Drupal¹⁰,

¹⁰Drupal - plataforma de gerenciamento de conteúdo (<http://drupal.org/>)

o Lyx¹¹, SVN¹² e ClockingIT¹³ e o próprio site do D-Lab na internet, um dos focos de interesse para melhoria e evolução.

Figura 51: Estrutura do domínio operacional *as-is* do D-Lab

A atividade “Realizar análise causal” resulta no modelo de análise de visibilidade (figura 52), que realiza o requisito Visibilidade (incluindo os requisitos aninhados definidos no ReqEnv de visibilidade institucional, acadêmica e técnica) e especifica as estatísticas de acesso a serem realizadas para aferir a visibilidade do SIS D-Lab.

¹¹Lyx - processador de documentos Latex (<http://www.lyx.org/>)

¹²SVN - Subversion, controle de versão (<http://subversion.apache.org/>)

¹³ClockinIT - gerenciamento de projetos online (<http://www.clockingit.com/>)

Figura 52: Diagrama paramétrico de análise de visibilidade do D-Lab

A figura 53 apresenta os requisitos de domínio transferidos do ReqEnv e associados aos processos de negócio, resultado da atividade “Especificar requisitos” (figura 50).

Figura 53: Requisitos e objetos associados aos *stakeholders* do D-Lab

A atividade “Definir BDD de domínio operacional *to-be*” resulta no dia-

grama de blocos do novo SIS D-Lab, resultado da transformação dos serviços analisados (figura 49). A figura 54 apresenta o diagrama de blocos do domínio operacional *to-be* do SIS D-Lab, onde os blocos Drupal, Lyx e SVN são reaproveitados integralmente da estrutura do modelo *as-is*. Duas novas ferramentas são incorporadas ao SIS D-Lab, o Mendeley¹⁴ e o RedMine¹⁵.

Figura 54: Domínio operacional *to-be* do SIS D-Lab

A figura 55 apresenta a matriz relacionando (rastreadibilidade) o domínio operacional *to-be* do SIS D-Lab e os serviços, indicando os blocos alocados aos serviços.

¹⁴Mendeley - rede social acadêmica e gerenciador de referências (<http://www.mendeley.com/>)

¹⁵RedMine - ferramenta de apoio ao gerenciamento de projetos (<http://www.redmine.org/>)

	2-Structure::Acesso interno	2-Structure::Acesso publico	2-Structure::Diab Project	2-Structure::GHENeSys	2-Structure::Impressão	2-Structure::Infraestrutura D-Lab	2-Structure::itsIMPLE	2-Structure::Mendeley	2-Structure::Operational Domain to-be	2-Structure::Projetos	2-Structure::RedMine	2-Structure::Servidor BD	2-Structure::SIS Diab
Services::Acompanhamento Projetos Desenvolvimento										↑	↑		
Services::Acompanhamento Tese							↑			↑			
Services::Divulgação de trabalhos													↑
Services::Gestão de Documentos													↑
Services::Informações corporativas													↑
Services::Informações Individuais													↑
Services::Informações projetos pesquisa													↑
Services::Validação textos							↑						

Figura 55: Matriz alocação *to-be* e serviços do D-Lab

A figura 56 apresenta os casos de uso de negócio, resultado da atividade “Definir casos de usos de negócio” (figura 52), atendidos pelo SIS D-Lab. Vários dos casos de uso são implementados com a definição de processos e sustentados pelas ferramentas oferecidas pelo SIS D-Lab, incluindo o site na internet.

Figura 56: Casos de uso de negócio do D-Lab

A figura 57 apresenta as atividades associadas ao caso de uso Submissão de Artigo, fornecendo a especificação do processo a ser utilizado pelos integrantes do D-Lab para submissão de artigos.

Figura 57: Atividades da "submissão de artigos" do D-Lab

A nova versão do site do D-Lab na internet é o sistema de interesse a ser representado em um grau de abstração mais detalhado. A figura 58 apresenta o diagrama de blocos da especificação caixa-preta do site do D-Lab na

internet.

Figura 58: Especificação *black-box* do Site D-Lab

O SoftDiss oferece recursos para a representação, e principalmente gerenciamento evolutivo, dos serviços prestados por um ambiente representativo dos setores quaternário e quinquário. Sendo um processo “vivo” e contínuo de evolução dos serviços, durante o desenvolvimento do trabalho diversos requisitos foram inseridos, como por exemplo a necessidade de gerenciamento de projetos externos ao D-Lab e que resultou no novo serviço prestado pelo SIS D-Lab, alcançado pela substituição do ClockingIT pelo RedMine, que exigiu esforços de configuração e treinamento na nova ferramenta.

4.3 Projeto Agronegócio

O setor agrícola sempre foi considerado como integrante do setor primário da economia, e caracterizado por empregar parcelas pequenas da mão de obra economicamente ativa. A evolução levou a integração deste setor

com o setor industrial gerando a agricultura industrial (agroindústria) que ampliou muito a empregabilidade ao tempo em que trouxe a automação em uma combinação no mínimo discrepante.

Em um movimento ainda mais recente foi lançada a agricultura de precisão (VVA Consulting, 2012; MAPA, 2011), onde o problema principal é a regulação de insumos e recursos, uma vez modelado o processo principal de produção agrícola. Esta nova tendência combina o setor agrícola com o setor de serviços, não na sua versão convencional mas na linha da engenharia de serviço. Assim, de modo semelhante ao setor industrial, em que, eminentemente baseado em sistemas discretos, se assiste o surgimento dos *manufacturing services* (ETTLIE; ROSENTHAL, 2011), a agricultura de precisão tem como tendência marcante o “agroserviço”, como estratégia de negócios e também como estratégia de controle e racionalização de recursos.

O agronegócio, incluindo a agricultura de precisão, está em alta no Brasil e fomentando o interesse da indústria mundial de máquinas agrícolas na consolidação e expansão de suas marcas em território nacional. Um ambiente relacionando os principais atores do agronegócio - indústria, rede de concessionários e agricultores - já está estabelecido. A nova orientação para serviço (agroserviço), sustentada pela ciência de serviço, oferece perspectivas inovadoras ao agronegócio, onde novos processos, sustentados por SIS, fortalecem e expandem o relacionamento entre estes atores.

A otimização do desempenho destes sistemas passa por questões que requerem o entendimento das relações existentes entre os diversos *stakeholders* ou envolvidos no negócio, incluindo aspectos técnicos, comerciais, sociais, legais e econômicos, formando um conjunto intrincado de pontos de vista que determinam o sucesso do agronegócio.

Nesta aplicação, o SoftDiss é utilizado para modelagem dos serviços exis-

tentes entre os atores do agronegócio e para a especificação do SIS de sustentação do contrato de manutenção de equipamentos agrícolas.

4.3.1 SoftDiss aplicado ao Serviço de Manutenção

Nesta aplicação busca-se a validação do SoftDiss através da aplicação a uma situação real, e conseqüentemente, resultando em um conjunto de dados que permitam aferir a aplicabilidade do *framework* proposto na eliciação, modelagem e análise de requisitos de SIS, sem modelo de referencia mas com uma base de casos práticos significativo, e que inclui a inserção de serviços no setor primário da economia, agora associado a automação dos processos agrícolas e em especial a automação do processo de inserção de recursos (equipamentos agrícolas) e o serviço de manutenção destes insumos.

Em relação às aplicações anteriores (*Smart Grid* e D-lab) destaca-se o serviço oferecido pelo ManEnv de implementação MDG SoftDiss e sintetizada na figura 26 da seção 3.6. Destaque também para análise de compatibilidade entre *viewpoints* como serviço do ReqEnv (seção 3.3).

A modelagem dos requisitos foi realizada de acordo com o processo empregado no ReqEnv (figura 8 da seção 3.3). Os direcionadores (objetivo do projeto, clientes, consumidores, usuários do produto e *stakeholders*) e as restrições (requisitos, glossário, fatos relevantes e premissas) foram definidos, resultando como objetivo o desenvolvimento do SIS para serviço de manutenção e a seleção dos *viewpoints* de requisitos. Os requisitos foram eliciados considerando-se três *viewpoints*, a indústria montadora de equipamentos agrícolas (representada por consultores ligados a empresa), os concessionários (neste caso representados por 2 concessionários do estado de São Paulo) e os agricultores (representados por uma equipe de consultoria especializada).

A figura 59 apresenta os requisitos eliciados para a indústria e a figura 60 os respectivos *Naked Objects*. Neste caso, os requisitos da indústria representam o *viewpoint* dos processos de negócio, pois direcionam a estratégia básica de toda a cadeia de negócio, incluindo os concessionários e mantendo o foco no cliente consumidor (agricultor).

Figura 59: Requisitos da indústria de equipamentos agrícolas

Figura 60: *Naked Objects* da indústria de equipamentos agrícolas

As figuras 61 e 62 apresentam os requisitos e *Naked Objects* associados aos proprietários dos concessionários de equipamentos agrícola sob contrato de concessão da indústria.

Figura 61: Requisitos dos proprietários concessionários

Figura 62: *Naked Objects* dos proprietários concessionários

Os requisitos e *Naked Objects* associados aos agricultores clientes dos concessionários e consumidores de equipamentos agrícolas são apresentados nas figuras 63 e 64, respectivamente.

Figura 63: Requisitos dos agricultores

Figura 64: *Naked Objects* dos agricultores

A última etapa é a análise de compatibilidade entre os três *viewpoints* de requisitos. A análise foi realizada através de verificação de consistência básica visando o acoplamento entre os *viewpoints* da indústria (requisitos dos processos de negócio) e os demais requisitos (concessionários e agricultores), em busca requisitos equivalentes.

A figura 65 apresenta o resultado da análise de compatibilidade entre os três *viewpoints* (Indústria, Concessionário e Agricultor). Este modelo foi obtido depois de várias iterações (incluindo as etapas subsequentes do DesEnv), o texto de vários requisitos foram ajustados buscando a padronização de termos. O resultado foi validado com os representantes da indústria e dos concessionários.

Figura 65: Acoplamento dos *viewpoints* de requisitos do agronegócio

No modelo de requisitos da figura 65 todos os requisitos são indexados com o *viewpoint* de origem (clausula *from* abaixo de cada um). requisitos equivalentes encontrados em dois *viewpoints* são relacionados pelo conector *copy* e a relação de aninhamento (ou contenção) entre requisitos de *viewpoints* diferentes são realizados com o conector de contenção diretamente no diagrama de acoplamento. Os *Naked Objects* dos três *viewpoints* foram utilizados como referência de validação e orientação de possíveis acoplamentos. A figura 66 apresenta a matriz de associação entre os requisitos da indústria e dos concessionários considerando a conexão de aninhamento entre os requisitos destes *viewpoints*.

	Requisitos - Indústria::Consolidação da Marca	Requisitos - Indústria::Desenvolvimento de talentos	Requisitos - Indústria::Evolução dos produtos	Requisitos - Indústria::Expansão do market share	Requisitos - Indústria::Formação equipe atendimento	Requisitos - Indústria::Nova cultura de serviços	Requisitos - Indústria::Padronização concessionários	Requisitos - Indústria::Padronização operação	Requisitos - Indústria::Padronização visual	Requisitos - Indústria::Processos de treinamento	Requisitos - Indústria::Utilização da experiência do cliente
Requisitos - Proprietário Concessionário::Atendimento fiscal											
Requisitos - Proprietário Concessionário::Desenvolvimento de talentos											
Requisitos - Proprietário Concessionário::Desenvolvimento industria complementar			↑			↑					
Requisitos - Proprietário Concessionário::Expansão comercial											
Requisitos - Proprietário Concessionário::Expansão do market share											
Requisitos - Proprietário Concessionário::Expansão serviço											
Requisitos - Proprietário Concessionário::Fidelização cliente											
Requisitos - Proprietário Concessionário::Inovação em serviços	↑										
Requisitos - Proprietário Concessionário::Integração administração, vendas, peças e serviços						↑					
Requisitos - Proprietário Concessionário::Lucratividade											
Requisitos - Proprietário Concessionário::Política de treinamento											
Requisitos - Proprietário Concessionário::Processo de venda, cliente promotor											↑
Requisitos - Proprietário Concessionário::Retenção de talentos		↑									

Figura 66: Matriz de análise de compatibilidade industria e concessionários

O DesEnv oferece os recursos para identificar e analisar os serviços necessários para a especificação do SIS de manutenção. A figura 67 apresenta o diagrama de pacotes com a organização do DesEnv.

Figura 67: Modelo do ambiente de desenvolvimento DesEnv do agronegócio

Foram negociados com os *stakeholders* os atributos essenciais (padronização, rentabilidade, formação de talentos e fidelização) apresentados no

modelo de atributos de serviço (figura 68).

Figura 68: Modelo de atributos de serviço do agronegócio

A árvore de decisão da figura 69 apresenta o conjunto de serviços identificados e associados aos atributos formação de talentos e padronização. Para o agronegócio o atributo rentabilidade foi definido como o mais importante pelos *stakeholders*, entretanto por ser inerente ao negócio deve sempre ser considerado para as decisões. Apenas uma árvore de decisão foi necessária para identificar os serviços.

Figura 69: Árvore de decisão dos serviços conceituais do agronegócio

A figura 70 apresenta o modelo conceitual de associação para os serviços identificados, associando os serviços conceituais com os devidos consumidores dos serviços. Neste caso, os consumidores se identificam com os agentes dos *viewpoints* adicionado do governo.

Figura 70: Modelo de associação conceitual do agronegócio

A figura 71 apresenta o resultado da análise contextual dos principais serviços associados ao atributo formação de talentos. A análise resulta na descoberta dos serviços atômicos capacitação em vendas, manutenção, sistemas e peças (capacitação nos principais setores que compõem uma concessionária).

Figura 71: Análise contextual do serviço qualificação RH

A figura 72 apresenta o modelo resultante da análise estrutural dos serviços associados ao atributo formação de talentos, detalhando a estrutura de relacionamento entre os serviços e o tipo de ligação com os serviços atômicos. No caso, o serviço capacitação generaliza o serviço de capacitação EAD, agrega o processo de qualificação de RH e é composto pelos serviços atômicos de capacitação em vendas, manutenção, sistemas e peças.

Figura 72: Modelo de análise estrutural do serviço formação de talentos

A distribuição geográfica do agronegócio esta apresentada na figura 73. Novos consumidores são identificados (*holding* multinacional e funcionários das concessionárias) como consequência da análise de integração de negócio.

Figura 73: Distribuição geográfica do agronegócio

A figura 74 relaciona os serviços com as perspectivas de negócio (padronização, operacional e inovação), indicando quais os serviços devem ser considerados para cada uma das perspectivas.

Figura 74: Integração do negócio do agronegócio

O resultado obtido é o conjunto de serviços identificados, conceituados, analisados, integrados e, principalmente, priorizados em função dos atributos essenciais (figura 75).

Figura 75: Serviços do agronegócio

A matriz de associação entre os requisitos da indústria e os serviços, considerando a relação de alocação, está na figura 76 e permite verificar a rastreabilidade entre requisitos e serviços.

	Requisitos - Industria::Consolidação da Marca	Requisitos - Industria::Desenvolvimento de talentos	Requisitos - Industria::Evolução dos produtos	Requisitos - Industria::Expansão do market share	Requisitos - Industria::Formação equipe atendimento	Requisitos - Industria::Nova cultura de serviços	Requisitos - Industria::Padronização concessionários	Requisitos - Industria::Padronização operação	Requisitos - Industria::Padronização visual	Requisitos - Industria::Processos de treinamento	Requisitos - Industria::Utilização da experiência do cliente
Services::Acompanhamento pós-venda	↑			↑							↑
Services::Capacitação		↑			↑			↑	↑	↑	
Services::Capacitação EAD		↑			↑		↑	↑	↑	↑	
Services::Contrato de manutenção			↑			↑					↑
Services::Controle estoque de peças							↑	↑			
Services::Diagnóstico remoto GPRS			↑			↑					
Services::Informações Contábeis Concessionário							↑	↑			
Services::Informações Gerenciais Concessionário							↑	↑			↑
Services::Modelo referência arquitetura loja	↑			↑			↑	↑	↑		
Services::Modelo referência processos de negócio	↑			↑		↑	↑	↑			↑
Services::Processo de qualificação de RH		↑	↑	↑	↑		↑	↑		↑	
Services::Processo de venda	↑			↑			↑	↑			↑

Figura 76: Alocação requisitos da industria e serviços

O próximo passo definido pelo SoftDiss é a especificação e design do SIS de manutenção utilizando o método OOSEM. A primeira atividade é a análise das necessidades dos *stakeholders* tendo como resultado o modelo de domínio operacional (*as-is*), requisitos de negócio e o modelo de domínio operacional do SIS de manutenção (*system of interest*). A figura 77 apresenta o diagrama de blocos com a estrutura do modelo operacional *as-is*.

Figura 77: Estrutura do domínio operacional *as-is* do agronegócio

Os requisitos de negócio são apresentados na figura 78. O diagrama de requisitos mostra o serviço contrato de manutenção e os relacionamentos de alocação dos requisitos, para explicitar a rastreabilidade entre eles e sustentar a especificação e o design do SIS de manutenção.

Figura 78: Requisitos de negócio do serviço de manutenção

A figura 79 apresenta o diagrama de blocos do domínio operacional *to-be*, com os sistemas legados a serem incorporados, o bloco contrato de manuten-

ção representando o novo tipo de serviço e o SIS de manutenção (*system of interest*) a ser especificado e desenvolvido para implementar o novo serviço.

Figura 79: Domínio operacional *to-be* do agronegócio

A figura 80 apresenta os requisitos (fornecidos pelo ReqEnv) a serem atendidos pela especificação do SIS de manutenção.

Figura 80: Requisitos operacionais do sistema de interesse do agronegócio

O próximo passo seria a especificação caixa preta do sistema de interesse, entretanto este projeto está aguardando o fechamento de acordo com indústria para ter continuidade, com proposta de projetos encaminhadas para a indústria nacional e alguns concessionários.

4.4 Implementação SoftDiss

O SoftDiss é um framework que visa direcionar um processo de modelagem onde a entrada são requisitos funcionais e não-funcionais, processos de negócio e restrições, e, com base em boas práticas e no formalismo de esquemas trabalhar a passagem destes requisitos em especificação (formal) de SIS.

O SoftDiss é um sistema de serviço composto de sistemas (*systems of systems*), que inclui novos e consolidados sistemas diferentes. O Enterprise Architect (EA) é o ambiente concentrador destes sistemas que compõem o SoftDiss, atuando como ambiente de implementação da estrutura do SoftDiss e permitindo a agregação dos demais métodos, técnicas, ferramentas e tecnologias.

O EA é um ambiente orientado a modelos com suporte MDA (atende plenamente MBSE), que sustenta o paradigma da orientação a objetos (inclusive *Naked Objects*), com suporte completo a UML 2.0 e SysML (juntamente com MBSE suporte ao método OOSEM) e, finalmente, o SOMF é um *framework* nativo da implementação do EA.

O processo de engenharia de requisitos Volere fornece as atividades a serem realizadas para obter a especificação de requisitos (funcionais, não funcionais, restrições, diretivas de projeto e domínio do projeto), passível de modelagem em UML ou mesmo SysML e, portanto com suporte do EA.

A norma ISO/IEC 15288 provê ao SoftDiss uma estrutura comum, integrada e compreensível para descrever e gerenciar todo o ciclo de vida de sistemas. A implementação MDG do SoftDiss contempla os processos técnicos da norma, entretanto, os demais processos de gestão são realizados pelo estabelecimento de processos específicos.

A verificação do comportamento dinâmico da especificação é realizada com base no modelo GHENeSys, concebida como uma rede de Petri estendida, com conceitos de orientação a objetos. Na versão atual do SoftDiss a integração com GHENeSys é feita por processo, mas trabalhos futuros indicam a possibilidade de integração, devido a características comuns como orientação a objetos e modelos, fundamento semântico baseado em redes de Petri da UML 2.0 e intercâmbio de modelos baseado em MOF.

Para melhor reproduzir uma situação real de design de SIS o SoftDiss foi inserido no EA da SparX, como já mencionado na seção 3.6, onde também se encontra o *framework* original do SOMF. Um acordo de cooperação foi assinado entre o D-Lab e a SparX para este fim.

O ManEnv é implementado pela tecnologia MDG¹⁶ (*Model Driven Generation Technologies*) recurso oferecido pelo EA, que permite estender as capacidades de modelagem do EA para domínios e notações específicas. No caso do SoftDiss, MDG é utilizado para gerar a implementação do processo de gerenciamento, desenvolvimento do projeto e organização dos modelos. A figura 81 apresenta a estrutura de pacotes obtida pelo MDG SoftDiss e a implementação do SoftDiss na tela de “*Model Wizard*” do EA destinada a criação de projetos suportados por tecnologia MDG.

¹⁶Sparx Model Driven Generation Technologies - http://www.sparxsystems.com.au/resources/mdg_tech/

Figura 81: SoftDiss implementado no Enterprise Architect

O MDG SoftDiss fornece um *template* de criação dos pacotes e as recomendações necessárias para a conjugação dos diversos métodos, técnicas, padrões e normas utilizados no SoftDiss.

A instalação do MDG SoftDiss no EA é realizada de acordo com as indicações da própria ferramenta.

4.5 Síntese do Capítulo

Este capítulo mostra a abrangência do SoftDiss e sua aplicabilidade prática a diferentes tipos de SIS, pode-se constatar que as melhores práticas, fer-

ramentas comerciais e métodos formais propostos pelo SoftDiss favorecem a antecipação da formalização, a efetivação da rastreabilidade e o tratamento de *viewpoints*.

Todos os casos abordados contaram com a participação e cooperação de vários profissionais, entretanto, o trabalho de modelagem foi realizado por apenas um profissional. Uma possível dificuldade é a utilização do SoftDiss por uma equipe extensa de profissionais onde, questões como a capacitação para o uso da ferramenta Enterprise Architect (EA) e do SysML, alocação e divisão de tarefas precisam ser resolvidas.

A norma ISO/IEC 15288 estabelece um *framework* de processos para tratamento de todo o ciclo de vida de um sistema, incluindo definição e análise de requisitos, design da arquitetura, implementação e verificação, e como todo sistema de qualidade, sua aplicação é dependente da estrutura da organização alvo.

O estudo do *Smart Grid* fornece resultados que mostram a capacidade do SoftDiss de atender a resolução de problemas com larga escala de *viewpoints*, requisitos e sistemas interligados e legados. O modelo de referência *IntelliGrid Architecture* apresenta um número elevado de domínios, componentes, requisitos e funcionalidades, o SoftDiss permitiu representar diversos destes elementos nos mais diversos níveis de abstração, considerando os vários *viewpoints* de requisitos (diversos domínios e múltiplas funções fornecidos pelo modelo de referência), além de viabilizar a análise dos requisitos. O SoftDiss permite o tratamento da rastreabilidade dos requisitos, tratada desde o modelo de atributos (figura 30), passando pelo modelo conceitual dos serviços (árvore de decisão conceitual e associação conceitual, figura 31 e 32, respectivamente) e chegando até os níveis mais detalhados de análise dos serviços (análise estrutural dos serviços, figura 34, e integração de negócio,

figura 35) e verificação formal dos requisitos através da análise de propriedades das redes de Petri. O projeto *Smart Grid* em parceria com a UEA está em andamento e deve ser aplicado na especificação dos serviços associados a *Smart Grid* do novo campus da UEA a ser construído em Manaus.

No estudo do D-Lab pôde-se constatar que o SoftDiss oferece recursos para a representação, e principalmente, o gerenciamento evolutivo, dos serviços prestados por um ambiente representativo dos setores quaternário e quinário. O grau de prioridade dado pelo D-Lab aos diversos *viewpoints* é distinto, bem como a capacidade de prover recursos a estes *viewpoints* e o SoftDiss mostrou-se eficiente para tratar este problema, como pode ser observado já no tratamento inicial da modelagem dos requisitos contemplando os contrastes entre os diversos *viewpoints* e refletido na especificação do domínio operacional com os acessos internos e públicos ao SIS D-Lab. O caso do D-Lab é um bom exemplo de que SIS são formados por sistemas, pessoas e processos.

O SoftDiss mostrou-se eficiente ao atender uma característica marcante em serviço, a co-criação de valor, que resulta em grande volatilidade nos requisitos, abordado no caso D-Lab com a alteração da ferramenta de gestão de projetos em função do atendimento a confidencialidade dos dados de projetos externos. Outro resultado importante é que a modelagem do D-Lab, com alguma adaptação, pode ser aplicada a empresas de consultoria e grandes centros de pesquisa e desenvolvimento, concedendo à modelagem SoftDiss do D-Lab grande potencial para sustentar a modelagem de serviços destes dois tipos de entidade.

Os resultados obtidos no estudo do agronegócio mostram a possibilidade de registro, documentação e gerenciamento da rastreabilidade entre os diversos elementos do modelo e os requisitos eliciados, fornecendo uma estrutura

formal para o modelo e permitindo a comprovação do atendimento dos requisitos dos diversos *viewpoints*, resultado observado, por exemplo na figura 76, que apresenta a rastreabilidade entre requisitos e serviços. Este projeto também está em andamento, aguardando o fechamento de acordo com indústria e a posterior aplicação do SoftDiss em diversas concessionárias nacionais.

Finalmente, a incorporação do SoftDiss ao Enterprise Architect (EA) pode favorecer a aplicação da engenharia de serviços a vários projetos e o acordo com a SparX oferece um canal de comunicação com os usuários para realimentar o SoftDiss com críticas e sugestões de melhorias, além de permitir validar a aplicabilidade do SoftDiss a um conjunto maior de problemas.

5 CONSIDERAÇÕES FINAIS

Com o presente trabalho chama-se a atenção para a importância da engenharia de serviço no âmbito da pesquisa em ciência de serviço, motivado pelo fato que pode ser prevista que uma significativa demanda por sistemas de serviço está se aproximando, o que levanta a possibilidade de ocorrência da mesma situação do passado, a chamada "crise do software".

A tese central deste trabalho é que a abordagem orientada a serviço deve ser sustentada pela engenharia de sistemas, com sistemas de informação atuando como elementos integradores automatizados do processo de co-criação do serviço, que foi confirmado pelos resultados e contribuições deste trabalho.

Os novos sistemas de serviço são compostos de sistemas (*systems of systems*), portanto, sistemas de grande porte, seja pelo tamanho seja pela complexidade, e cujo processo de design demanda uma abordagem sistêmica e orientada a serviço. O SoftDiss é um *framework* com estas características e foi aqui apresentado como prova de conceito. Entre suas características destaca-se:

1. abordagem orientada a modelo (MBSE, OOSEM, SysML);
2. a aderência a atual tendência da UML 2.0 na representação e análise de requisitos, inclusive usando rede de Petri para expressar o diagrama de estados;

3. abordagem orientada a sistema de sistema (*system of system*), que integra sistemas já consolidados para compor novos sistemas e suportar novos paradigmas;
4. uso de sistema de grande porte e escaláveis que podem suportar o desenvolvimento de sistemas de sistemas. No caso específico do SoftDiss, o Enterprise Architect (EA) foi o escolhido por integrar os métodos colocados como importantes e adequados para compor os sistemas de informação de serviço (SIS);
5. um tratamento especial para as fases iniciais onde uma atenção especial é dada à análise de *viewpoints*, como base para co-criação;
6. atenção detalhada para a correspondência entre os processos de negócio e os requisitos básicos;
7. ferramenta genérica para integração com o processo de gestão.

A proposta do SoftDiss alia "boas" práticas a ferramentas e métodos já consolidados e abordagens formais, tendo como objetivo a co-criação de valor, no caso, entre desenvolvedores humanos e os sistemas incluídos no processo de design - EA, GHENeSys e o SOMF que foi aprimorado em alguns de seus ambientes para se adequar a proposta atual.

No projeto de desenvolvimento de SIS as fases iniciais (requisitos e design), diretamente abordadas pelo SoftDiss, tem um impacto ainda maior que nos demais projetos. As razões para isso são: a necessidade de uma fase de interação e co-criação com os atores e com os beneficiários do serviço, a necessidade de compatibilização de interesses entre *stakeholders* e (as diversas classes de) usuários (com a integração dos *viewpoints* de cada uma das classes de *stakeholders* e usuários) e, finalmente, a necessidade de compatibilizar requisitos e restrições e processos de negócio.

5.1 Contribuições

As principais contribuições deste trabalho são:

1. Foi apresentada uma visão geral sobre as recentes iniciativas para compor processo de especificação de sistemas de serviço, base para a ciência de serviço. Foram consideradas propostas de fundamentação teórica para a ciência de serviço, complementadas por teorias, técnicas, métodos, normas e práticas para eliciação de requisitos, rastreabilidade, modelagem orientada a modelos e a objetos, tratamento de *viewpoints*, gerenciamento do ciclo de vida dos projetos e representação da especificação.
2. Foi desenvolvido um *framework* orientado a serviço (SoftDiss) destinado a especificação de sistema de informação de serviço (SIS), suportando as fases iniciais do desenvolvimento, incluindo a eliciação, modelagem e análise de requisitos, através da configuração dinâmica e cognitiva de tecnologias, pessoas, sistemas externos e das informações compartilhadas entre estes elementos.
3. O SoftDiss foi incorporado à ferramenta de modelagem comercial EA, utilizando a tecnologia MDG. Um acordo de cooperação foi estabelecido entre o D-Lab e a Sparx, fornecedora da ferramenta EA, visando a utilização do EA no desenvolvimento deste trabalho e nas demais pesquisas do D-Lab.
4. O SoftDiss foi aplicado a três projetos de especificação de SIS. Os resultados obtidos foram utilizados na evolução do próprio SoftDiss e, também, geraram publicação de artigos, proposta de projetos de pesquisa e implementação de SIS.

5.2 Trabalhos futuros

A proposta do SoftDiss alia "boas" práticas a integração com ferramentas consolidadas para tratar problemas de grande porte e o uso de métodos formais de forma harmônica, técnica e economicamente viável. Mais especificamente foi tratada a integração das seguintes técnicas e ferramentas: EA, GHENeSys, SOMF, SysML, UML, OOSEM, norma ISO/IEC 15288, *template* Volere para documentação de requisitos e abordagem *Naked Objects*. A associação destas técnicas, softwares, métodos, representações formais, é a base do *framework* proposto que por sua vez apresenta uma ligação conceitual entre sistemas de informação (cognitivos), teoria de sistemas e ciência de serviço que se reflete na disciplina de projeto inserida no SoftDiss.

Deve ser salientado que alguns aspectos não podem ser dispensados, como a rastreabilidade na análise de *viewpoints* durante a fase de eliciação e análise de requisitos. Além disso, a interação entre os agentes para validar requisitos (geralmente entre desenvolvedores e elementos da classe dos *stakeholders* ou dos clientes) é uma questão-chave na nova engenharia de serviço, uma vez que os sistemas de serviço devem resultar da convergência entre diversas e diferentes expectativas de várias classes de agentes, para gerar valor a partir do processo de automação. Entretanto o processo de análise de requisitos como foi realizado neste trabalho (o suporte mais importante do *framework* proposto), não contemplou a parte cognitiva, e consequentemente os sistemas de informação gerados não são (em nenhum dos estudos de caso) inteligentes. Na verdade, o suporte para este tipo de sistema é bastante restrito, na atual fase do SoftDiss (embora ainda responda por um número razoavelmente grande de aplicações). Em trabalhos futuros pretende-se incluir o tratamento de intenções durante a fase de análise de requisitos, o que aponta para uma abordagem para o tratamento de requisitos *i** (YU, 1997) cri-

ada em 1997 e desenvolvida mais recentemente até atingir estágios que a habilitam a implementações de grande porte.

Dentre os ambientes do SoftDiss a maior demanda de processos cognitivos se localiza no ManEnv, especificamente em dois processos: no tratamento de requisitos, especialmente quando um conjunto de serviços básicos não estiverem completamente acoplados os requisitos gerais; e quando ManEnv precisar planejar ações e intervenções, quando uma abordagem inteligente poderia ser muito útil para realimentar o processo de design - inserindo a reutilização de processos, baseado em objetos e/ou em metáforas (SILVA, 1992). Estes processos cognitivos não foram incluídos neste trabalho por duas razões: i) pelo impacto na ampliação do escopo; ii) porque ainda não existe um ambiente implementado que possa integrar o *framework* proposto (exceto se fosse desenvolvido em paralelo a este trabalho). Uma outra razão, não menos importante, é que seria recomendável inserir processos inteligentes tendo já resolvido as questões básicas e procedurais, que é o objetivo deste trabalho. Aliás, em se tratando de ferramentas inteligentes tem-se a perspectiva de, em futuro próximo, incluir no *framework* proposto mais uma ferramenta desenvolvida no D-lab: o itSIMPLE (VAQUERO; SILVA; BECK, 2011) que é uma ferramenta para planejamento inteligente, e que portanto daria conta de um dos aspectos dos sistemas cognitivos não contemplados neste trabalho.

Finalmente, a engenharia de serviço é muito dependente da comunicação, uma vez que a ideia geral é que um sistema de serviço seja sempre colaborativo e totalmente integrado com os seus atores. Isso poderia ser contemplado estendendo o processo de design atual para incluir um outro processo: reusabilidade. No entanto, para aplicar reusabilidade de forma eficiente é preciso enfrentar uma questão relevante: diferentes projetos utilizam diferentes glosários (palavras e termos), especialmente se houver diferença significativa

no domínio da aplicação do projeto. Uma abordagem baseada em metáforas (aplicada a especificação de requisitos) foi proposta por Silva (1992) para encontrar similaridades na derivação de termos em diferentes conjuntos diferente de discurso de domínio. Em um futuro próximo, esta poderá ser uma forma interessante de enfrentar problemas de comunicação no processo de projeto de SIS. O processo de validação de requisitos pode ser beneficiado por este recurso, já que o mesmo problema pode ocorrer entre os domínios de diferentes classes de agentes.

Outra questão a ser considerada é o gerenciamento de projetos de grande porte com uma alocação numerosa de profissionais responsáveis pelo design. Neste caso, o ManEnv deverá ser estendido para contemplar o gerenciamento do trabalho de vários profissionais. A norma ISO/IEC 15288 define os processos de projeto que incluem o planejamento, alocação de recursos, controle, tratamento de riscos e gerenciamento de comunicação, que devem ser implementados integralmente no ManEnv.

Portanto, qualquer que seja a agenda de pesquisa para o futuro esta deve incluir a discussão sobre a composição de técnicas que viabilizem a engenharia de serviço não só pelo seu impacto acadêmico mas também por razões práticas.

REFERÊNCIAS

AALST, W. Three good reasons for using a Petri-net-based workflow management system. *KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE*, Citeseer, p. 161–182, 1998.

AALST, W.; HEE, V. *Workflow management: models, methods and systems*. Cambridge: MIT Press, 2002.

ANEEL. Projeto Estratégico : Programa Brasileiro de Rede Elétrica Inteligente. 2010. Disponível em: <http://www.aneel.gov.br/arquivos/PDF-/PeD_2008-ChamadaPE11-2010.pdf>.

BARDHAN, I. R. et al. An Interdisciplinary Perspective on IT Services Management and Service Science. *Journal of Management Information Systems*, v. 26, n. 4, p. 13–64, abr. 2010. ISSN 0742-1222.

BARESI, L.; PEZZÈ, M. Improving UML with Petri nets. *Electronic Notes in Theoretical Computer Science*, v. 44, n. 4, p. 107–119, jul. 2001. ISSN 15710661. Disponível em: <[http://linkinghub.elsevier.com/retrieve/pii-S1571066104809472](http://linkinghub.elsevier.com/retrieve/pii/S1571066104809472)>.

BARILE, S.; POLESE, F. Smart Service Systems and Viable Service Systems: Applying Systems Theory to Service Science. *Service Science Journal*, v. 2, n. 1/2, p. 21 – 40, 2010. Disponível em: <<http://www.sersci.com/ServiceScience/upload/12704925720.pdf>>.

BEER, S. The Viable System Model: Its Provenance, Development, Methodology and Pathology. *The Journal of the Operational Research Society*, v. 35, n. 1, p. 7–25, 1984. Disponível em: <<http://www.jstor.org/pss-/2581927>>.

BELL, M. *Service-Oriented Modeling: Service Analysis, Design, and Architecture*. 1. ed. New Jersey: Wiley, 2008. 384 p.

BERTALANFFY, L. An outline of general system theory. *The British Journal for the Philosophy of science*, Br Soc Philosophy Sci, v. 1, n. 2, p. 134, 1950. Disponível em: <<http://bjps.oxfordjournals.org/cgi/reprint/1/2/134.pdf>>.

BITNER, M.; BROWN, S. The service imperative. *Business Horizons*, v. 51, n. 1, p. 39–46, 2008. ISSN 00076813. Disponível em: <<http://linkinghub-elsevier.com/retrieve/pii/S0007681307001097>>.

BOULDING, K. General systems theory-the skeleton of science. *Management science*, v. 6, n. 1, p. 127–139, 1956. Disponível em: <<http://www.jstor.org/stable/2627132>>.

CARR, J. Requirements engineering and management: the key to designing quality complex systems. *The TQM Magazine*, v. 12, n. 6, p. 400–407, 2000.

CHESBROUGH, H.; SPOHRER, J. I. M. A research manifesto for services science. *Communications of the ACM*, v. 7, n. 49, p. 35–40, 2006.

DAVENPORT, T. *Process innovation: reengineering work through information technology*. Boston: Harvard Business School Press, 1993.

DESEL, J.; REISIG, W. Place/transition Petri nets. *Lectures on Petri Nets I: Basic Models*, Springer, p. 122–173, 1998. Disponível em: <<http://www.springerlink.com/index/X6HN592L35866LU8.pdf>>.

DUTRA, D.; OLIVEIRA, V. C. de; SILVA, J. R. Manufacturing as Service : the challenge of Intelligent Manufacturing. In: *11th IFAC Workshop on Intelligent Manufacturing Systems (IMS13)*. São Paulo, SP, Brasil: [s.n.], 2013. p. 6.

EPRI. The Green Grid - Energy Savings and Carbon Emissions Reductions Enabled by a Smart Grid. 2008. Disponível em: <<http://www.smartgridinformation.info/pdf/2458\doc\1.pdf>>.

ESTEFAN, J. Survey of model-based systems engineering (MBSE) methodologies. *In cose MBSE Focus Group*, 2007. Disponível em: <<http://www.omgsysml.org/MBSE\Methodology\Survey\RevB.pdf>>.

ETTLIE, J. E.; ROSENTHAL, S. R. Service versus Manufacturing Innovation*. *Journal of Product Innovation Management*, v. 28, n. 2, p. 285–299, mar. 2011. ISSN 07376782. Disponível em: <<http://doi.wiley.com/10.1111/j.1540-5885-2011.00797.x>>.

FOYO, P. del. *GHENeSys: Uma Rede Estendida Orientada a Objetos para Projeto de Sistemas Discretos*. Tese (Mestrado) — Universidade de São Paulo, 2001.

FRIEDENTHAL, S.; MOORE, A.; STEINER, R. *A Practical Guide to SysML: The Systems Modeling Language*. First edit. San Francisco, CA: Morgan Kaufmann, 2009. 560 p.

GOMES, R.; PRINTES, A.; RAMOS, C. Proposta de Sistema com Arquitetura para Implementação de uma Smart Grid na Rede de Distribuição em Baixa Tensão. In: *SBSE 2010 - Simpósio Brasileiro do Sistema Elétrico*. Belém, PA, Brasil: [s.n.], 2010. p. 1–6. Disponível em: <[http://www.labplan.ufsc.br/congressos/III SBSE - 2010/PDF/SBSE2010-0271.PDF](http://www.labplan.ufsc.br/congressos/III%20SBSE%20-%202010/PDF/SBSE2010-0271.PDF)>.

GOTEL, O.; FINKELSTEIN, C. An analysis of the requirements traceability problem. In: *Proceedings of 1st International Conference on Requirements Engineering*. IEEE Computer Society Press, 1994. p. 94–101. ISBN

0-8186-5480-5. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03-wrapper.htm?arnumber=292398> http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=292398>.

GRÖNROOS, C. *Marketing : gerenciamento e serviços*. Rio de Janeiro: Elsevier/Campus, 2004. 482 p.

HASKINS, C. *Systems Engineering Handbook: a Guide for System Life Cycle Processes and Activities*. 1. ed. San Diego: INCOSE, 2006. 185 p. Disponível em: <<http://smlab.kaist.ac.kr/Course/CC532/2012/LectureNote/2012/INCOSE Systems Engineering Handbook v3.1 5-Sep-2007.pdf>>.

HILLAH, L.; KORDON, F.; PETRUCCI, L. PN standardisation: a survey. *Lecture Notes in Computer Science : Formal Techniques for Networked and Distributed Systems - FORTE*, p. 307–322, 2006. Disponível em: <<http://www.springerlink.com/index/A61183710J14KJ77.pdf>>.

HOLANDA, A. B. *Dicionário Aurélio da Língua Portuguesa*. 5a edição. ed. São Paulo, SP, Brasil: Positivo Editora, 2010. 2222 p. ISBN 9788538541981.

HUGHES, J. *The Integrated Energy and Communication Systems Architecture Volume I: User Guidelines and Recommendations*. 2003. Disponível em: <<http://www.intelligrid.info/>>.

_____. *The Integrated Energy and Communication Systems Architecture Volume II: Functional Requirements*. 2003. Disponível em: <<http://www.intelligrid.info/>>.

IEA. *Key World Energy Statistics 2011*. out. 2011. Disponível em: <<http://www.iea.org>>.

IFM; IBM. *Succeeding through service innovation: A service perspective for education, research, business and government*. 2008. Disponível em: <<http://www.ifm.eng.cam.ac.uk/resources/service/succeeding-through-service-innovation/>>.

INCOSE. *SYSTEMS ENGINEERING VISION 2020*. 2007.

JENSEN, K. An introduction to the practical use of coloured petri nets. *Lecture Notes in Computer Science*, Springer, p. 237–292, 1998. Disponível em: <<http://www.springerlink.com/index/b766176514616731.pdf>>.

KELLERMAN, A. THE EVOLUTION OF SERVICE ECONOMIES : A GEOGRAPHICAL PERSPECTIVE. *The Professional Geographer*, v. 37, n. 2, p. 37–41, 1985.

KENT, S. Model driven engineering. In: *Integrated Formal Methods*. Springer, 2002. v. 1, n. 1-2, p. 286–298. Disponível em: <<http://www.springerlink.com/index/9VUQB4HP8FYG2ADV.pdf>>.

KIM, Y. J.; NAM, K. Service Systems and Service Innovation: Toward the Theory of Service Systems. In: *AMCIS 2009 Proceedings*. [S.l.: s.n.], 2009.

KOBRYN, C. UML 3.0 and the Future of Modeling. *Software and Systems Modeling*, p. 4–8, 2004. Disponível em: <[http://www.springerlink.com/index-F7AQG1WBJB6G2J7F.pdf](http://www.springerlink.com/index/F7AQG1WBJB6G2J7F.pdf)>.

KOTONYA, G.; SOMMERVILLE, I. Requirements engineering with viewpoints. *Software Engineering Journal*, v. 11, n. 1, p. 5, 1996. ISSN 02686961.

_____. *Requirements Engineering: Processes and Techniques*. [S.l.]: John Wiley & Sons Ltd., 1998.

LAMSWEERDE, A. V. Goal-oriented requirements engineering: A guided tour. *Requirements Engineering*, 2001. . . . , n. August, 2001. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=948567>.

LEITE, J. C. S. D. P. Viewpoints on viewpoints. In: *ACM Workshop SIGSOFT*. [S.l.]: ACM Press, 1996. p. 285–288. ISBN 0897918673.

LETELIER, P. A framework for requirements traceability in UML-based projects. In: *1st International Workshop on Traceability in Emerging Forms of Software Engineering. In conjunction with the 17th IEEE International Conference on Automated Software Engineering*. [S.l.: s.n.], 2002.

LI, H.-f. et al. Services Science , Management , and Engineering : A Literature Review in the Perspective of Management Science. In: *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. [S.l.: s.n.], 2007. p. 1438–1441.

LOVELOCK, C.; WRIGHT, L. *Principles of Service Marketing and Management*. Upper Saddle River, NJ: Prentice Hall, 2002. 436 p.

LUSCH, R.; VARGO, S.; WESSELS, G. Towards a conceptual foundation for service science: Contributions from service-dominant logic. *IBM Systems Journal*, p. 1–9, 2008.

MAGLIO, P. P. et al. The service system is the basic abstraction of service science. *Information Systems and e-Business Management*, v. 7, n. 4, p. 395–406, 2009. ISSN 1617-9846. Disponível em: <<http://www.springerlink.com/index/10.1007/s10257-008-0105-1>>.

MAPA. AGRICULTURA DE PRECISÃO. 2011. Disponível em: <<http://www.agriculturadeprecisao.org.br>>.

MARSHALL, C. *Enterprise Modeling with UML*. Upper Saddle River, NJ: Addison-Wesley, 2000.

MCGRANAGHAN, M.; DOLLEN, D. V. Utility experience with developing a smart grid roadmap. *IEEE Power and Energy*, p. 1–5, 2008.

MERZ, M. a.; HE, Y.; VARGO, S. L. The evolving brand logic: a service-dominant logic perspective. *Journal of the Academy of Marketing Science*, v. 37, n. 3, p. 328–344, 2009. ISSN 0092-0703. Disponível em: <<http://www.springerlink.com/index/10.1007/s11747-009-0143-3>>.

MIYAGI, P. E. *Control System Design, Analysis and Implementation of Discrete Event Production Systems by using Mark Flow Graph*. Tese (Ph.D. thesis), 1988.

MOUSSA, S.; TOUZANI, M. A literature review of service research since 1993. *Journal of Service Science*, v. 2, n. 2, p. 173–212, dez. 2010. ISSN 2005-3274. Disponível em: <http://www.springerlink.com/index/10.1007/s12927-010-0008-9>.

MURATA, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, IEEE, v. 77, n. 4, p. 541–580, 1989. ISSN 0018-9219. Disponível em: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=24143.

NG, I.; MAULL, R. Embedding the new Discipline of Service Science. In: Demirkan, H, J.H. Spohrer and V. Krishna Eds (Ed.). *The Science of Service Systems*. Springer, 2009. v. 44, n. 0, cap. Service Sc. Disponível em: http://business-school.exeter.ac.uk/documents/discussion_papers/management/2009/0901.pdf.

NIST. NIST Framework and Roadmap for Smart Grid Interoperability Standards. 2010. Disponível em: http://www.nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf.

OLIVEIRA, V. C. *Proposta de método para gestão de requisitos de sistemas integrando modelagem de negócio e linguagens formais*. 153 p. Tese (Mestrado) — Escola Politécnica da Universidade de São Paulo, 2008. Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3135/tde-27022009-104608/publico/dissertacaoValterOliveirarev.pdf>.

OLIVEIRA, V. C. et al. Automação e ciência de serviço aplicado a smart grid. In: *CBA Congresso Brasileiro de Automática*. [S.l.: s.n.], 2012. p. 589–596. ISBN 9788580010695.

OLIVEIRA, V. C.; SILVA, J. R. Ciência de Serviços na Gestão de Requisitos de Sistemas. In: *4º Workshop de Pós-Graduação e Pesquisa - Centro Paula Souza*. São Paulo, SP, Brasil: [s.n.], 2009. p. 1–10.

_____. Process Oriented Design Service: Case Study for Automated Information System. In: *21st Brazilian Congress of Mechanical Engineering*. Natal, RN, Brasil: [s.n.], 2011. p. 10.

OMGSYSML. OMG Systems Modeling Language (OMG SysML v1.3). 2012. Disponível em: <http://www.omg.org/spec/SysML/1.3/>.

OSTROM, a. L. et al. Moving Forward and Making a Difference: Research Priorities for the Science of Service. *Journal of Service Research*, v. 13, n. 1, p. 4–36, 2010. ISSN 1094-6705.

PAWSON, R. Naked objects. *Software, IEEE*, n. August, p. 81–83, 2002. Disponível em: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1020291.

PEARCE, P.; HAUSE, M. Model-Based Submarine Design. In: *SETE and APCOSE 2012*. Brisbane, Australia: [s.n.], 2012. p. 15.

QUEIROZ, R.; SILVA, J. R. Eliciação e comunicação de requisitos em domínios disjuntos: estudo de caso para automação na área médica. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, v. 20, n. 4, p. 637–651, 2009.

RAUMBAUGH, J. R. et al. *Object-Oriented Modeling and Design*. New Jersey: Prentice-hall, 1991.

ROBERTSON, S.; ROBERTSON, J. *Mastering the requirements process*. Second edi. Upper Saddle River, NJ: Addison-Wesley, 2006.

ROZENBERG, G.; ENGELFRIET, J. Elementary net systems. In: *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets*. London, UK: Springer-Verlag, 1998. p. 12–121. Disponível em: <http://www.springerlink.com/index/k6w5866420031272.pdf>.

SANTOS, E. *Verificação de requisitos de sistemas utilizando redes de petri*. 80 p. Tese (Mestrado) — Universidade de São Paulo, 2002.

SILVA, J. R. *Uma formalização do processo de design baseado em metáforas: Sua aplicação na automatização de Sistemas de Eventos Discretos*. 179 p. Tese (Doutorado) — Universidade de São Paulo, 1992.

SILVA, J. R.; MIYAGI, P. E. PFS/MFG: A High Level Net for the Modeling of Discrete Manufacturing Systems. In: MATOS, L. C.; AFSARMANESH, H. (Ed.). *Balanced Automation Systems, Architectures and Design Methods*. [S.l.]: Chapman & Hall, 1995. p. 349–362.

_____. A formal approach to PFS/MFG: a Petri Net representation of Discrete Manufacturing Systems. In: *Studies in Informatics and Control*. Romania: IC Publications, 1996.

SMITH, E. Principles of high-level net theory. *Lectures on Petri Nets I: Basic Models*, 1998. Disponível em: <http://www.springerlink.com/index/78125201814583v8.pdf>.

SOMMERVILLE, I.; SAWYER, P.; VILLER, S. Viewpoints for requirements elicitation: a practical approach. In: *Proceedings of IEEE International Symposium on Requirements Engineering: RE '98*. [S.l.]: IEEE Comput. Soc, 1997. p. 74–81. ISBN 0-8186-8356-2.

SPARX. *Sparx Systems Pty Ltd*. 2013. Disponível em: <http://www.sparxsystems.com.au/>.

SPOHRER, J. et al. Steps Toward a Service Systems. *IEEE Computer Society*, n. January, p. 71–77, 2007.

_____. The Service System Is the Basic Abstraction of Service Science. In: *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*. [S.I.]: Ieee, 2008. p. 104–104. ISBN 0-7695-3075-8. ISSN 1530-1605.

STAINES, A. Supporting Requirements Engineering With Different Petri Net Classes. *INTERNATIONAL JOURNAL OF COMPUTERS*, v. 4, n. 4, 2010. Disponível em: <<http://www.naun.org/journals/computers/19-438.pdf>>.

STAIR, R. M.; REYNOLDS, G. W. *Information Systems*. Australia: Cengage Learning, 2010.

STANICEK, Z.; WINKLER, M. Service Systems Through The Prism of Conceptual Modeling. *Service Science Journal*, v. 2, p. 112 – 125, 2010.

TANG, A.; JIN, Y.; HAN, J. A rationale-based architecture model for design traceability and reasoning. *Journal of Systems and Software*, v. 80, n. 6, p. 918–934, jun. 2007. ISSN 01641212. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0164121206002287>>.

The Standish Group International. The CHAOS Report. 2011. Disponível em: <<http://www.standishgroup.com/>>.

TRIPP, D. Pesquisa-ação: uma introdução metodológica. *Educação e pesquisa*, v. 31, n. 3, p. 443–466, 2005.

VAQUERO, T.; SILVA, J.; BECK, J. A brief review of tools and methods for knowledge engineering for planning & scheduling. In: *Proceedings of the Knowledge Engineering for Planning and Scheduling (KEPS) workshop. The 21th International Conference on Automated Planning & Scheduling (ICAPS 2011)*. Freiburg. Germany.: [s.n.], 2011.

VARGO, S.; AKAKA, M. Service-Dominant Logic as a Foundation for Service Science: Clarifications. *Service Science Journal*, v. 1, n. 1, p. 32–41, 2009.

VARGO, S. L.; LUSCH, R. F. Service-dominant Logic: What It Is, What It Is Not, What It Might Be. In: VARGO, S. L.; LUSCH, R. F. (Ed.). *The Service-dominant Logic of Marketing: Dialog, Debate, and Directions*. Armonk: M.E. Sharpe, 2006. p. 43–56.

VVA Consulting, V. Gns in agriculture: a market and technological assessment on North America, EU, Australia and Brazil. 2012. Disponível em: <<http://www.vvaconsulting.net/>>.

WANG, R.; DAGLI, C. H. An Executable System Architecture Approach to Discrete Events System Modeling Using SysML in Conjunction with Colored Petri Net. *2008 2nd Annual IEEE Systems Conference*, Ieee, p. 1–8, abr. 2008. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4518997>>.

YU, E. S.-k. *Modelling Strategic Relationships for Process*. 121 p. Tese (PhDThesis) — University of Toronto, 1995.

YU, E. S.-k. Towards modelling and reasoning support for early-phase requirements engineering. *Proceedings of ISRE '97: 3rd IEEE International Symposium on Requirements Engineering*, IEEE Comput. Soc. Press, p. 226–235, 1997. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03-/wrapper.htm?arnumber=566873>>.

ZHAO, G.; PERROS, H. How service science management and engineering (SSME) can evolve to an academic discipline? *International Journal of Web Engineering*, Volume 5, n. Issue 4, 2009. Disponível em: <<http://inderscience-metapress.com/index/G0J1836512J04032.pdf>>.

APÊNDICE A – MODELOS SOFTDISS

Este apêndice contém os modelos SoftDiss completos associados a cada um dos três projetos: *Smart Grid*, automação flexível do fornecimento de energia; SIS D-Lab, ambiente acadêmico de serviço; Agronegócio, especificação dos sistemas de serviço associados a agricultura de precisão especificamente na área de concessionárias de equipamentos agrícolas.

A.1 SoftDiss - Smart Grid

O setor de fornecimento e distribuição de energia elétrica é um dos mais conservadores do mercado. Especificamente, na geração hidroelétrica muitas decisões de implementação foram tomadas há mais de 120 anos, e se caracterizam por serem orientadas a bens materiais (*goods-dominant*), incorrendo em serviços de baixa flexibilidade e que colocam o consumidor em uma posição eminentemente passiva. Assim, este setor se manteve praticamente inalterado por quase cem anos, o que fica visível se comparado a outros setores mais dinâmicos como o de telecomunicações, computação ou eletrônica.

A rede de energia atual não tem acompanhado o ritmo dos desafios modernos, contando com diversos problemas sistêmicos, como: ameaças de segurança para os fornecedores de energia, incluindo o *cyber* ataque; fontes limitadas de geração alternativa de energia; inexistência de soluções para o uso conservativo de energia; incapacidade de garantir o fornecimento cons-

tante de energia; e falta de gerencia da rede de distribuição. Sendo assim, um dos grandes desafios tecnológicos atuais consiste em transformar a rede de energia atual em um sistema dinâmico, flexível e adaptável.

Um processo recente, denominado *Smart Grid*, vem promovendo atualizações no setor de energia, através da incorporação de diversos avanços tecnológicos, tanto em países desenvolvidos como em países em desenvolvimento, incluindo o Brasil. Os benefícios obtidos com o *Smart Grid* podem ser significativos, permitindo aos consumidores, entre outras coisas, exercer um melhor controle do consumo de energia, integrar uma nova geração de aparelhos elétricos como veículos híbridos e eletrodomésticos, e associar diferentes fontes de geração de energia, visando aumentar a eficiência energética e conseguir um melhor aproveitamento vindo de fontes renováveis.

De acordo com EPRI (2008) o termo *Smart Grid* é a sobreposição de uma comunicação unificada e de sistemas de controle sobre a infraestrutura existente de rede de energia para fornecer informação às entidades interessadas. É um sistema que otimiza a distribuição e entrega de energia, minimiza as perdas, faz auto ajustes e viabiliza uma próxima geração de dispositivos e aplicações voltados à eficiência energética.

Os Estados Unidos e a Europa estão atuando de forma mais intensa na implantação do *Smart Grid*. A Europa trabalha na utilização de fontes distribuídas e renováveis de energia, buscando redução de emissões, geração renovável na matriz e eficiência energética. Os Estados Unidos pretendem investir cerca de US\$ 4,5 bilhões em *Smart Grid*, através de um amplo programa envolvendo empresas de distribuição, fabricantes de equipamentos, empresas de telecomunicações e empresas de TI juntamente com as entidades governamentais como DOE – U.S. *Departamento of Energy* e NIST - *National Institute of Standards*, visando definir padrões para a *Smart Grid* que garantam a

interoperabilidade de protocolos e equipamentos em diversas áreas que integram o processo, como a de telecomunicações, tecnologia da informação (TI) e energia. Austrália, China e Coreia possuem motivos muito semelhantes aos dos EUA e da Europa para a implantação de *Smart Grid* (NIST, 2010).

Porém, o sistema elétrico brasileiro é único no mundo, com uma matriz energética baseada principalmente em energias renováveis, com dimensões continentais que constituem um desafio para a integração da geração e da infraestrutura de transmissão de energia, certamente diferenciadas da Europa e, também, dos Estados Unidos, que apesar de sua extensão territorial possui características de geração menos distribuídas. De acordo com “Projeto estratégico: Programa Brasileiro de Rede Elétrica Inteligente”, elaborado pela ANEEL¹, em 2010, a confiabilidade e qualidade da energia elétrica são exigências básicas dos consumidores brasileiros e a implantação de *Smart Grid* busca atender esta demanda, permitindo que a ANEEL estabeleça um mecanismo de aferição da qualidade na prestação de serviços associados ao consumo de energia elétrica, através da padronização de informações e dados (ANEEL, 2010).

O questionamento quanto à necessidade de uma rede de distribuição flexível, acessível, confiável e econômica, como se observa em outras localidades nos Estados Unidos e em países da Europa, antecede os desafios e a implementação de *Smart Grid* no Brasil. O *Smart Grid* é necessário e conveniente para o Brasil, mas com motivações diferentes do resto do mundo, contribuindo para a melhoria da qualidade do serviço prestado ao consumidor final, tendo potencial para melhorar a confiabilidade do sistema elétrico nacional e para reduzir os desperdícios. De acordo com a EIA ²(U.S. Energy

¹ANEEL (Agência Reguladora de Energia Elétrica), órgão regulador do sistema elétrico no Brasil - <http://www.aneel.gov.br>

²EIA U.S. Energy Information Administration, instituição mantida pelo governo dos EUA que coleta, analisa e divulga informações do setor elétrico de forma imparcial e independente

Information Administration), o Brasil apresenta perda de 17%, consequência das extensas linhas de transmissão entre as usinas hidroelétricas geradoras e os grandes centros urbanos, e também, das perdas devido ao furto de energia nos grandes centros urbanos, estimada em 8,7% da energia produzida (IEA, 2011; ANEEL, 2010).

A.1.1 Smart Grid de Baixa Tensão no Brasil

Um bom exemplo de sistema de serviço é o caso de um *Smart Grid* de baixa tensão (GOMES; PRINTES; RAMOS, 2010), cujo objetivo é fornecer energia disponível (ainda que predominantemente elétrica) para consumidores individuais, institucionais e comerciais, de forma sustentável, sem desperdícios, na medida exata das suas necessidades, evitando desvios e perdas. Por simplicidade, o sistema é restrito ao varejo, isto é, são excluídos os usuários que demandam serviços mais sofisticados providos por subestações.

O *Smart Grid* de baixa tensão apresenta a concessionária de energia elétrica como base de distribuição, e *clusters* de transformadores onde é tratada a distribuição da carga, e usuários finais conectados a um dos transformadores. Esta arquitetura é relacional e informada, isto é, os usuários podem gerenciar o seu consumo, enviar restrições, interrupções do serviço ou preferências, de modo a prover sustentabilidade, racionalidade no uso da energia e evitar desperdícios. O problema é também simplificado ao supor que estes usuários são só consumidores e, na modelagem atual não podem se transformar em pequenos fornecedores de energia.

As informações sobre a situação dos transformadores são usadas pelo nível intermediário que estabelece o balanço de carga e a informação sobre os *clusters* é sintetizada pela concessionária. Mesmo com todas estas simplifica-

(<http://www.eia.gov/countries/data.cfm>).

ções temos um sistema em três camadas (uma árvore) com informação fluindo nos dois sentidos: das folhas para a raiz e da raiz para as folhas. As simplificações foram feitas exatamente para reduzir a arquitetura do problema de um grafo mais sofisticado a uma árvore. Sobre este sistema flui ao mesmo tempo a energia fornecida e a informação sobre parâmetros do sistema e sobre preferências dos usuários. O objetivo final é o serviço, isto é, o atendimento a diferentes demandas dos usuários finais, da forma mais racional (do ponto de vista da distribuidora) e de forma sustentável (do ponto de vista dos reguladores). Temos, portanto um intrincado sistema de *viewpoints*, que restringe o fluxo de energia e cuja realização maior se dá pelo atendimento racionalizado de todos os *viewpoints* simultaneamente.

O tratamento proposto é justamente o de um sistema de serviço, aqui traduzido em um sistema de informação acoplado a um sistema de provimento direto de serviço. Note-se que a realização do negócio está ligada justamente ao acoplamento sistema de informação e provimento de energia, portanto é quase inviável sem a harmonização destes.

O gerenciamento de requisitos é determinante para o sucesso dos projetos, quando os ambientes de desenvolvimento e de aplicação são disjuntos, isto é, usuários e *stakeholders* pertencem a um domínio com pouca ou nenhuma interseção com a área da engenharia, informática e automação, os processos de eliciação e análise de requisitos se tornam bastante complicados (QUEIROZ; SILVA, 2009). Uma forma para amenizar a dificuldade de tratar domínios disjuntos é a utilização de modelos ou arquiteturas de referência, que possibilitem a aproximação entre os ambientes de desenvolvimento e de aplicação.

O projeto IntelliGrid³, originalmente denominado projeto IECSA (*Integrated*

³IntelliGrid (<http://intelligrid.epri.com/>)

Energy and Communications System Architecture), é patrocinado pelo EPRI⁴-*Electric Power Research Institute* com o objetivo de criar uma base técnica para um *Smart Grid*. Teve como principal resultado em 2003, a proposta da *IntelliGrid Architecture*⁵, baseada em padrões abertos, orientada a requisitos e destinada a integração de redes de dados e equipamento, e que permita a interoperabilidade entre produtos e sistemas, fornecendo às concessionárias de energia elétrica métodos, ferramentas e recomendação de padrões e tecnologias para a implementação de sistemas (HUGHES, 2003a).

O projeto *IntelliGrid* foi desenvolvido por um grupo de especialistas gerenciados pela General Electric com contribuições da *Utility Consulting International* (UCI), SISCO, Lucent, EnerNex, Hypertek e outros. O resultado foi a arquitetura de referência *IntelliGrid Architecture*, uma proposta de organização sistêmica do *Smart Grid*, organizando em seus componentes as relações existentes com outros sistemas, incluindo os princípios orientadores da sua concepção e evolução, e, também, a organização de tipos comuns de requisitos.

O projeto foi desenvolvido visando capturar e organizar os requisitos das funções do *Smart Grid* e, também, foi estruturado de forma a integrar as necessidades de negócio de operação dos sistemas de energia, as visões estratégica e tática, e permitindo aplicar diversas tecnologias para acomodar sistemas legados, multiplicidade de fornecedores e tecnologias, padrões, tecnologias e melhores práticas. Foram identificadas e registradas cerca de 400 funções. Consideração de critérios de configuração, qualidade de serviço, segurança e gerenciamento de dados, originaram um subconjunto de funções chave, que foram detalhadas, capturadas em narrativas e diagramas UML (HUGHES, 2003b). Um resultado relevante da importância e da aplicabilidade

⁴EPRI - *Electric Power Research Institute* (<http://www.epri.com>)

⁵*IntelliGrid Architecture* (<http://www.intelligrd.info/>)

da *IntelliGrid Architecture* no Brasil, encontra-se na CEMIG (Companhia Energética de Minas Gerais), que, de acordo com McGranaghan e Dollen (2008), calçou o desenvolvimento de um projeto *Smart Grid* nos resultados desta arquitetura.

Considerando as semelhanças entre o Brasil e os EUA, quando considerada a rede de baixa tensão, neste estudo a *Intelligrid Architecture* é utilizada como arquitetura de referência, permitindo uma abordagem baseada em domínios próximos para tratamento de requisitos e especificação de um novo sistema de informação de serviço (SIS) para *Smart Grid* de baixa tensão (OLIVEIRA et al., 2012; OLIVEIRA; SILVA, 2011).

A ANEEL, visando incentivar inovações do setor elétrico, regulamentou o Programa de Pesquisa e Desenvolvimento do Setor de Energia Elétrica - P&D ANEEL, no qual as empresas concessionárias, permissionárias ou autorizadas para fazer a distribuição, transmissão e geração de energia elétrica devem aplicar anualmente um percentual mínimo de sua receita operacional líquida em pesquisa. Enquanto o autor desta pesquisa prestava serviços à Facti⁶, em 2010 e 2011, visando a implementação de um projeto que permitisse a aplicação do SoftDiss em condições reais, uma proposta de projeto de P&D ANEEL foi apresentada à concessionária de energia elétrica Coelce⁷. A proposta foi elaborada em parceria com equipe da UEA⁸. A proposta foi avaliada pela ANEEL, mas não foi priorizada pelo comitê de P&D da Coelce para contratação, sendo assim a aplicação do SoftDiss ficou restrita à especificação dos serviços tendo como base modelo de referência.

A projeto *Smart Grid* está em andamento e neste apêndice está retratado

⁶Facti - Fundação de Apoio à Capacitação em Tecnologia da Informação (<http://www.facti.com.br/>)

⁷Coelce, concessionária de energia elétrica nacional situada em Fortaleza/CE (<https://www.coelce.com.br/>)

⁸UEA - Universidade do Estado do Amazonas (<http://www.uea.edu.br/>)

a última versão da modelagem SoftDiss. A figura 82 apresenta a estrutura de pacotes do SoftDiss e orienta o entendimento das demais figuras.

Figura 82: *Smart Grid* - Estrutura de Pacotes

Figura 83: *Smart Grid* - Ambiente Requisitos

Figura 84: *Smart Grid* - Atores de Negócio

Figura 85: *Smart Grid* - Requisitos de Negócio

Figura 86: *Smart Grid* - Especificações Intelligrid

Figura 87: *Smart Grid* - *Naked Objects* Baixa Tensão

Figura 88: *Smart Grid* - UC Centro de Supervisão

Figura 89: *Smart Grid* - UC Unidade Telesupervisão

Figura 90: *Smart Grid* - UC Unidade de Distribuição

Figura 91: *Smart Grid* - UC Indicador Digital de Consumo

Figura 92: Smart Grid - Modelo de Atributos de Serviço

Figura 93: Smart Grid - Árvore de Decisão

Figura 94: *Smart Grid* - Associação Conceitual

Figura 95: *Smart Grid* - Análise Contextual

Figura 96: *Smart Grid* - Análise Estrutural

Figura 97: *Smart Grid* - Integração de Negócio

Figura 98: *Smart Grid* - Controle do Indicador Digital de Consumo

Figura 99: *Smart Grid* - Máquina Estado Centro de Supervisão

Figura 100: *Smart Grid* - Máquina Estado Unidade Supervisão

Figura 101: *Smart Grid* - Máquina Estado Unidade Distribuição

Figura 102: *Smart Grid* - Máquina Estado Indicador Digital Consumo

A.2 SoftDiss - SIS D-Lab

A projeto SIS D-Lab está em andamento e neste apêndice está retratado a última versão da modelagem SoftDiss. A figura 103 apresenta a estrutura de pacotes do SoftDiss e orienta o entendimento das demais figuras.

Figura 103: D-Lab - Estrutura de Pacotes

Figura 104: D-Lab - Página Inicial

Figura 105: D-Lab - Viewpoints Requisitos

Figura 106: D-Lab - Objetos membros internos

Figura 107: D-Lab - Requisitos membros internos

Figura 108: D-Lab - Objetos membros externos

Figura 109: D-Lab - Requisitos membros externos

Figura 110: D-Lab - Objetos Colaborador

Figura 111: D-Lab - Requisitos Colaborador

Figura 112: D-Lab - Objetos empresa parceira

Figura 113: D-Lab - Requisitos empresa parceira

Figura 114: D-Lab - Objetos processos de negócio

Figura 115: D-Lab - Requisitos processos de negócio

Figura 117: D-Lab - Diagrama de Serviços

Figura 118: D-Lab - Diagrama de Consumidores

Figura 119: D-Lab - Diagrama de Atributos

Figura 120: D-Lab - Árvore Decisão

Figura 121: D-Lab - Associação Conceitual

Figura 122: D-Lab - Análise Contextual

Figura 123: D-Lab - Análise Estrutural

Figura 124: D-Lab - Integração de Negócio

Figura 125: DLab - Guia: Especificação e Design do Sistema (FRIEDENTHAL; MOORE; STEINER, 2009)

Figura 126: DLab - Guia: Análise das Necessidades dos *Stakeholders* (FRIEDENTHAL; MOORE; STEINER, 2009)

Figura 127: DLab - Guia: Análise dos Requisitos do Sistema (FRIEDENTHAL; MOORE; STEINER, 2009)

Figura 128: D-Lab - As-is: Requisitos Operacionais

Figura 129: D-Lab - As-is: Estrutura Operacional

Figura 130: D-Lab - *To-be*: Organização do Modelo

Figura 131: D-Lab - Ciclo de vida do SIS D-Lab

Figura 132: D-Lab - *To-be*: Requisitos Operacionais

Figura 133: D-Lab - *To-be*: Estrutura Operacional

Figura 134: D-Lab - *To-be*: Casos de Uso Operacionais

Figura 135: D-Lab - *To-be*: Atividade UC Programa Pós-Graduação

Figura 136: D-Lab - *To-be*: Atividade UC Relatório Técnico

Figura 137: D-Lab - *To-be*: Atividade UC Submissão Artigo

Figura 138: D-Lab - *To-be*: Paramétrico Análise de Viabilidade

A.3 SoftDiss - Agronegócio

A projeto Agronegócio está em andamento e neste apêndice está retratado a última versão da modelagem SoftDiss. A figura 140 apresenta a estrutura de pacotes do SoftDiss e orienta o entendimento das demais figuras.

Figura 140: Agro - Estrutura de pacotes

Figura 141: Agro - Página Inicial

Figura 142: Agro - Viewpoints Requisitos

Figura 143: Agro - Objetos Proprietário Concessionário

Figura 144: Agro - Requisitos Proprietário Concessionário

Figura 145: Agro - Objetos Indústria

Figura 146: Agro - Requisitos Indústria

Figura 147: Agro - Objetos Agricultor

Figura 148: Agro - Requisitos Agricultor

Figura 149: Agro - Compatibilidade dos Viewpoints de Requisitos

Figura 150: Agro - Organização do DesEnv

Figura 151: Agro - Diagrama de Serviços

Figura 152: Agro - Diagrama Consumidores

Figura 153: Agro - Diagrama de Atributos

Figura 154: Agro - Arvore Decisão

Figura 155: Agro - Associação Conceitual

Figura 156: Agro - Análise Contextual

Figura 157: Agro - Análise Estrutural

Figura 158: Agro - Integração Negócio Geográfica

Figura 159: Agro - Integração Negócio Estrutural

Figura 160: Agro - As-is: Requisitos Operacionais

Figura 161: Agro - As-is: Estrutura Operacional

Figura 162: Agro - As-is: Requisitos Operacionais

Figura 163: Agro - To-be: Organização do Modelo

Figura 164: Agro - *To-be*: Requisitos Operacionais

Figura 165: Agro - *To-be*: Estrutura Operacional