PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Universidade Federal do Rio de Janeiro

ESTER MORSOLETTO POEGERE

Um método baseado em padrões ontológicos de colaboração para construção de modelos conceituais de groupware

DISSERTAÇÃO DE MESTRADO

Instituto de Matemática

Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
INSTITUTO TÉRCIO PACITTI DE APLICAÇÕES E PESQUISAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

ESTER MORSOLETTO POEGERE

Um método baseado em padrões ontológicos de colaboração para construção de modelos conceituais de groupware

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Instituto de Matemática e Instituto Tércio Pacciti, Universidade Federal do Rio de Janeiro, como requisito parcial à obtenção do título de Mestre em Informática.

Orientador: Prof. Marcos Roberto da Silva Borges, Ph.D

Co-orientadora: Prof[a]. Claudia Lage Rebello da Motta, D.Sc

Rio de Janeiro
2018

CIP - Catalogação na Publicação

Poegere, Ester Morsoletto
P743m Um método baseado em padrões ontológicos de colaboração para construção de modelos conceituais de groupware / Ester Morsoletto Poegere. -- Rio de Janeiro, 2018.
230 f.

Orientadora: Marcos Roberto da Silva Borges.
Coorientadora: Claudia Lage Rebello da Motta.
Dissertação (mestrado) - Universidade Federal do Rio de Janeiro, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Programa de Pós-Graduação em informática, 2018.

1. Sistemas Colaborativos. 2. Reutilização em Groupware. 3. Modelagem Conceitual Ontológica de Groupware. 4. Reuso de Padrões Ontológicos de Colaboração. I. Borges, Marcos Roberto da Silva, orient. II. Motta, Claudia Lage Rebello da, coorient. III. Título.

Elaborado pelo Sistema de Geração Automática da UFRJ com os dados fornecidos pelo(a) autor(a).

Um método baseado em padrões ontológicos de colaboração para construção de modelos conceituais de groupware

ESTER MORSOLETTO POEGERE

Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática do Instituto de Matemática e do Instituto Tércio Pacitti da Universidade Federal do Rio de Janeiro - UFRJ, como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

Aprovada em 31 de agosto de 2018 por:

_____

Prof. Marcos Roberto da Silva Borges, Ph.D, UFRJ

_____

Prof[a]. Claudia Lage Rebello da Motta, D.Sc, UFRJ

_____

Prof[a]. Adriana Santarosa Vivacqua, D.Sc, UFRJ

_____

Prof[a]. Renata Mendes de Araujo, D.Sc, Membro Externo

_____

Prof[a]. Fernanda Araujo Baião Amorim, D.Sc, Membro Externo

DEDICATÓRIA

Dedico esta dissertação ao meu amado Adonai (**הוהי**) através de Yeshua Hamashiach (**המשיח עישו**) por me incentivar a nunca desistir de um sonho e me apoiar todo o tempo.

AGRADECIMENTOS

Primeiramente, agradeço a Deus (**יהוה**) por meio de Yeshua Hamashiach (**ישוע המשיח**) pela força, coragem e persistência. Obrigada, Senhor, porque até aqui tua mão me guiou, me sustentou em meio às lutas, desafios e, se cheguei até aqui, foi porque o teu amor infinito tem sido o meu sustento em toda esta jornada.

Aos meus amados pais, Calvino Morsoletto e Janete Alves Morsoletto (ambos *in memoriam*), em especial à minha mãe, que não mediu esforços e dedicou sua vida à minha educação e formação. À minha querida irmã Raquel Morsoletto de Oliveira, pelo carinho e companheirismo e aos meus irmãos Marcos Morsoletto e Calvino Morsoletto Filho, que sempre me apoiaram nesta caminhada.

Estas duas pessoas foram fundamentais durante todo o meu mestrado. Por isso sou-lhes muito grata. Obrigada, meu esposo Eduardo Elias Poegere, pela paciência e compreensão. E ao pequeno príncipe, Eduardo Elias Morsoletto Poegere, que, no último ano, esteve tão próximo, literalmente, de mim, que foi tão presente no desenvolvimento deste trabalho e que, agora, me inspira a querer ser mais que fui até hoje!

Aos meus orientadores, Marcos Roberto da Silva Borges e Claudia Lage Rebello da Motta, por acreditarem na minha capacidade e compartilharem comigo o grande conhecimento que possuem. Agradeço a possibilidade de fazer parte dessa comunidade acadêmica tão valorosa e reconhecida mundialmente. Agradeço os momentos de orientação, as revisões cuidadosas e o incentivo constante.

Os meus mais sinceros agradecimentos ao Professor Dr. Carlo Emanuel T. de Oliveira por sua parceria e por seu companheirismo construtivista, que me serviram, por vezes, de farol. À Professora Dra. Maria Luiza Machado Campos, por compartilhar seus conhecimentos comigo. À Professora Dra. Renata Silva Souza Guizzardi, pelo Editor OLED estendido e matérias de apoio. Ao Professor Dr. Claudio Micele de Farias, por suas valiosas contribuições.

Quero agradecer em especial ao Dr. André Luis Nogueira Campos, à Dra. Luciana da Silva Madeira e ao Dr. Jobson Luiz Massolar da Silva, por terem me aberto as portas do estudo acadêmico, levando-me pelas mãos durante todo o início da caminhada.

Agradeço ao Maurício S. Silva e ao Alexandre S. Wolf, pelo COLLABCODE e pelos matérias de apoio. À Fiocruz, por todo o apoio, direto e indireto, ao longo de todo o curso.

Aos meus colegas de curso, pelo apoio, especialmente a Cristiane de Moura Cruz Oliveira, Diana Carolina Arce Cuesta, Angélica Fonseca da Silva Dias, Juliana Baptista dos Santos França, Ivison da Costa Rubim, Bruno Barboza Lage, Josy Corrêa Kopke, Flavio Duarte Pinto, Patricia Merlim Lima Scheidegger e Denilson dos Santos Guimarães.

Estas três pessoas extraordinárias me inspiraram, suportaram minhas dores e me aconselharam: Tia Amélia Monteiro Torres, Dra. Maria Cecília Monteiro Alves Ferreira e Márcia Boell. Por isso, sou-lhes muito grata: muito obrigada.

Aos meus irmãos de coração, Jaqueline Alves Garrido, Márcia Sóliva, Michelli Almeida Bastos, Marcus Vinicius Soares Bastos, Emília Carolina Silva dos Santos e Anderson dos Santos Nepomuceno, pela força e companheirismo.

Aos projetistas que participaram da avaliação deste trabalho e a todos os que estiveram, de forma direta ou indireta envolvidos, nesta caminhada que hoje se encerra para que outras possam ser iniciadas.

À CAPES, pelo suporte financeiro por meio da bolsa de estudos concedida durante o curso.

Aos membros da banca examinadora, pela leitura e contribuições nesta dissertação.

Enfim, a todos os que sempre me incentivaram, com palavras amigas, a percorrer este caminho e tornar possível realizar esta dissertação.

Meu sincero MUITO OBRIGADA!

“Acredite, é hora de vencer
Essa força vem de dentro de você
Você pode até tocar o céu se crer

Acredite que nenhum de nós
Já nasceu com jeito pra super herói
Nossos sonhos a gente é quem constrói
É vencendo os limites, escalando as fortalezas
Conquistando o impossível pela fé

Campeão, vencedor
Deus dá asas, faz teu voo
Campeão, vencedor
Essa fé que te torna imbatível
Mostra seu valor

Acredite que nenhum de nós
Já nasceu com jeito pra super herói
Nossos sonhos a gente é quem constrói
É vencendo os limites
Escalando as fortalezas
Conquistando o impossível pela fé

Campeão, vencedor
Deus dá asas, faz teu voo
Campeão, vencedor
Essa fé que te torna imbatível
Mostra seu valor

Tantos recordes você pode quebrar
As barreiras você pode ultrapassar
E vencer

Campeão, vencedor
Deus dá asas, faz teu voo
Campeão, vencedor
Essa fé que te torna imbatível
Mostra seu valor

Letra de música de Luan Santana
“Conquistando o impossível‟

RESUMO

O objetivo deste trabalho é apresentar o método ÔMEGA e seu respectivo modelo de processo de uso ALFA (AP-OPL) para auxiliar na construção de modelos conceituais de *groupware* (ou Sistemas Colaborativos) baseados em Padrões Ontológicos de Colaboração (CONTO), utilizando o módulo OLED Estendido do Editor OLED e os paradigmas da estruturação de requisitos orientados a objetivos, a fim de facilitar o desenvolvimento de modelagem conceitual.

A pesquisa caracteriza-se dentro de uma abordagem quali-quantitativa e faz uso de dois estudos de caso no domínio de colaboração: um, de complexidade simples (*Brainstorming online)* e outro, de maior complexidade (*CollabCode* - Ferramenta para apoio ao desenvolvimento distribuído e colaborativo de software). Neste contexto, verificou-se que o método ÔMEGA é aplicável e eficiente para equipes (pequenas, médias e de complexidade maiores).

A partir desses resultados, pode-se concluir que o método, com o seu modelo de processo de uso ALFA (AP-OPL) é viável, podendo ser usado tanto por analistas de sistemas quanto por analistas de dados de forma desacoplada da implementação dos Sistemas da Informação (SI). Também concluímos que o método é aderente ao ciclo proposto pelas metodologias tradicionais e ágeis, o que só aumenta a sua aplicabilidade.

Palavras-chave: Sistemas Colaborativos. Reutilização em Groupware. Modelagem Conceitual Ontológica de Groupware. Reuso de Padrões Ontológicos de Colaboração.

ABSTRACT

The objective of this work is to present the ÔMEGA method and its respective ALFA (AP-OPL) process model to assist in the construction of conceptual groupware models (or Collaborative Systems) based on Ontological Collaboration Standards (CONTO), using the module OLED Extended OLED Editor and goal-oriented requirements structuring paradigms in order to facilitate the development of conceptual modeling.

The research is characterized within a qualitative-quantitative approach and makes use of two case studies in the domain of collaboration: one, of simple complexity (Brainstorming online) and another, of more complexity (CollabCode - Tool to support distributed and collaborative software development). In this context, it was verified that the ÔMEGA method is applicable and efficient for teams (small, medium and of greater complexity).

Based on these results, it can be concluded that the method with its ALFA (AP-OPL) process model is feasible and can be used by both systems analysts and data analysts in a decoupled way of Information (SI). We also conclude that the method is adherent to the cycle proposed by traditional and agile methodologies, which only increases its applicability.

Keywords: Collaborative Systems. Groupware Reuse. Groupware Ontological Conceptual Modeling. Reuse of Ontological Collaboration Patterns.

LISTA DE ILUSTRAÇÕES

FIGURA 2 : Evolução das pesquisas;

FIGURA 2.1 : Distribuição das pesquisas por categoria;

FIGURA 3 : Matriz espaço e tempo de groupware;

FIGURA 3.1 : Posicionamento dos sistemas colaborativos no espaço em função do Modelo 3C;

FIGURA 3.2 : Elementos para comunicação interativa numa videoconferência;

FIGURA 3.3 : Modelo 3C de Colaboração;

FIGURA 4 : Conceitualização, Modelo, Linguagem e Especificação;

FIGURA 4.1 : Ontologia e linguagem de representação;

FIGURA 4.2 : Diferentes tipos de ontologias;

FIGURA 4.3 : Parte da ontologia funcional unificada A (UFO-A);

FIGURA 4.4 : Relacional universal;

FIGURA 4.5: Fases do processo de desenvolvimento de ontologias do OLED;

FIGURA 4.6 : Tipos de padrões ontológicos;

FIGURA 4.7 : Níveis de generalidade e padrões ontológicos;

FIGURA 4.8 : Três subontologias CONTO e sua relações;

FIGURA 4.9 : Subontologia de Cooperação;

FIGURA 4.10 : Subontologia de Comunicação;

FIGURA 4.11 : Subontologia de Coordenação dependência de Fluxo (Flow);

FIGURA 4.12 : Subontologia de Coordenação dependência de Usabilidade (*Usability*);

FIGURA 4.13 : Subontologia de Coordenação dependência de Transferência (*Transfer*);

FIGURA 4.14 : Subontologia de Coordenação dependência de Encaixe (*Fit*);

FIGURA 4.15 : Fragmento da Subontologia de Coordenação;

FIGURA 4.16 : Fragmento da Ontologia de Coordenação;

FIGURA 4.17 : Modelo do processo da C-OPL;

FIGURA 4.18 : Processo da metodologia SABÍO;

FIGURA 4.19 : Importação catálogo DROPs de colaboração no OLED;

FIGURA 4.20 : Configuração das questões de competência;

FIGURA 4.21 : Criação do Padrão variante CSS;

FIGURA 5 : Processo de definição das etapas de trabalho do método ÔMEGA;

FIGURA 5.1 : Esboço da atividade de obtenção dos artefatos de colaboração;

FIGURA 5.2 : Diagrama da “Lista de itens de verificação dos requisitos”;

FIGURA 5.3: Lista de elementos contidos Doc. de informações necessárias.

FIGURA 5.4 : Lista de itens de estruturação dos requisitos;

FIGURA 5.5 : Elementos para um bom entendimento do problema;

FIGURA 5.6 : Oito passos para construção de modelo conceitual ONTOUML;

FIGURA 5.7 : Os passos avançados para construção de modelo conceitual ONTOUML;

FIGURA 5.8 : Modelo de processo de uso ALFA (AP-OPL) do método ÔMEGA;

FIGURA 5.9 : Caminhos de entrada do modelo de processo de uso do método ÔMEGA;

FIGURA 5.10 : Os POs das áreas problemáticas de CONTO;

FIGURA 5.11 : Visão detalhada da OPL ALFA (AP-OPL);

FIGURA 6 : Modelo conceitual específico do Brainstorming online;

FIGURA 6.1 : Verificação sintática do groupware de Brainstorming online;

FIGURA 6.2 : Níveis da teoria da atividade usada no CollabCode;

FIGURA 6.3 : Ferramentas disponíveis no CollabCode;

FIGURA 6.4 : Interfaces do CollabCode;

FIGURA 6.5 : Modelo de Entidade Relacionamento (MER) do *groupware CollabCode;*

FIGURA 6.6 : Modelo físico da base de dados do *groupware CollabCode;*

FIGURA 6.7 : Modelo conceitual específico do CollabCode;

FIGURA 6.8 : Verificação sintática do groupware CollabCode;

FIGURA 7 : Situação atual de formação acadêmica;

FIGURA 7.1 : Grau de formação acadêmica;

FIGURA 7.2 : Grau de experiência em modelagem conceitual;

FIGURA 7.3 : De que forma a experiência em modelagem conceitual foi adquirida;

FIGURA 7.4 : Grau de experiência em desenvolvimento de ontologias;

FIGURA 7.5 : Experiência anterior em desenvolvimento de ontologias;

FIGURA 7.6 : Grau de experiência em linguagem de padrões ontológicos (OPL);

FIGURA 7.7 : De que forma foi adquirida a experiência em linguagem de padrões ontológicos (OPL);

FIGURA 7.8 : Quão difícil/fácil você considera entender o mecanismo de funcionamento do ÔMEGA;

FIGURA 7.9 : Quão difícil você considera entender o mecanismo de funcionamento da OPL;

FIGURA 7.10 : Quão difícil/fácil você considera entender os elementos da OPL;

FIGURA 7.11 : Essa versão ALFA (OP_OPL) tornou o processo mais claro;

FIGURA 7.12 : Facilidade de entendimento do processo da OPL;

FIGURA 7.13 : Avaliação geral do método ÔMEGA.

LISTA DE TABELAS

TABELA 1 : String de busca;

TABELA 2 : Retorno das bibliotecas digitais;

TABELA 2.1 : Critérios de inclusão e de exclusão;

TABELA 2.2 : Categorias das pesquisas;

TABELA 3 : Requisitos de Groupware do Usuário (RGU);

TABELA 3.1 : Requisitos de Groupware do Desenvolvedor (RGD);

TABELA 4: Postulado e definições relacionados aos Sortais;

TABELA 4.1: Definições das categorias de Sortals e MixinUniversal (ou Não-Sortals);

TABELA 4.2: Categorias ontológicas de tipos de Classes Todo-Parte (*Meronymic Relations*);

TABELA 4.3: Relações de dependência (Dependency Relationship);

TABELA 4.4: Relações de Todo-Parte (*Meronymic Relations*);

TABELA 4.5: Relacionamentos em Diagramas de Classes (*Relationships in Class Diagrams*);

TABELA 4.6: Dicionário de Termos da Subontologia de Cooperação;

TABELA 4.7: Dicionário de Termos da Subontologia de Comunicação;

TABELA 4.8: Dicionário de Termos da Subontologia de Coordenação;

TABELA 4.9: Especificação parcial do padrão (CSL);

TABELA 4.10: Especificação parcial do padrão (LT);

TABELA 4.11: Especificação parcial do padrão (CSS);

TABELA 4.12: Especificação parcial do padrão (CSAC);

TABELA 4.13: Especificação parcial do padrão (CSAD);

TABELA 4.14: Especificação parcial do padrão (CSA);

TABELA 4.15: Especificação parcial do padrão (CA);

TABELA 4.16: Especificação parcial do padrão (CAD);

TABELA 4.17: Especificação parcial do padrão (CAC);

TABELA 4.18: Especificação parcial do padrão (CCAC);

TABELA 4.19: Especificação parcial do padrão (CCIC);

TABELA 4.20: Especificação parcial do padrão (ACC);

TABELA 4.21: Especificação parcial do padrão (ACD);

TABELA 4.22: Especificação parcial do padrão (ACPD);

TABELA 4.23: Especificação parcial do padrão (ACPPS);

TABELA 4.24: Especificação parcial do padrão (APAC);

TABELA 4.25: Especificação parcial do padrão (GPCCC);

TABELA 4.26: Especificação parcial do padrão (GC);

TABELA 4.27: Especificação parcial do padrão (GD);

TABELA 4.28: Especificação parcial do padrão (CSRS);

TABELA 4.29: Especificação parcial do padrão (CSR);

TABELA 4.30: Especificação parcial do padrão (RC);

TABELA 4.31: Especificação parcial do padrão (RN);

TABELA 4.32: Especificação parcial do padrão (CIP);

TABELA 4.34: Especificação parcial do padrão (CIPS);

TABELA 5 : Elementos visuais utilizados na legenda da OPL ALFA (AP-OPL) visão geral;

TABELA 5.1 : Catálogo de DROPs de Colaboração Consolidado;

TABELA 5.2 : Elementos visuais utilizados na legenda da OPL ALFA (AP-OPL) visão detalhada;

TABELA 6 : Organize seu conhecimento do domínio;

TABELA 6.1 : Conceitos fundamentais;

TABELA 6.2 : Escolha preliminar dos POs que serão usados;

TABELA 6.3 : Características funcionais do contexto;

TABELA 6.4 : Estereótipos das Classes identificados;

TABELA 6.5 : Dimensões da colaboração disponíveis no *CollabCode;*

TABELA 6.6 : Papéis, permissões e responsabilidade do CollabCode;

TABELA 6.7 : Características funcionais do contexto;

TABELA 6.8 : Estereótipos das Classes identificados;

TABELA 6.9 : Mapeamento entre Classe e diagrama MER;

TABELA 6.10 : Comparação entre entidades.

LISTA DE ABREVIATURAS E SIGLAS

SE	Engenharia de Software;
SI	Sistemas da Informação;
EG	Engenharia de Groupware;
EO	Engenharia de Ontologias;
BD	Banco de Dados;
UML	Linguagem de modelagem unificada (*Unified Modeling Language*);
ONTOUML	Linguagem de modelagem unificada ontológica (*Ontological Unified Modelling Language*);
UFO	Ontologia de fundamentação unificada (*Unified Foundational Ontology*);
POs	Padrões ontológicos (*Ontology Patterns*);
LP ou OPL	Linguagem de padrões (*Pattern Language*);
OPLs (ou LPOs)	Linguagem de padrões ontológicos (*Ontology Pattern Language*);
COPs	Padrões ontológicos de conteúdo (*COPs - Content Ontology Patterns)*;
FOPs	Padrões ontológicos fundacionais (FOPs - *Foundational Ontology Patterns*);
DROPs	Padrões ontológicos relacionados ao domínio (*Domain-Related Ontology Patterns*);
C-OPL	Linguagem de padrões ontológicos de colaboração (*Collaboration Ontology Pattern Language);*
OLED	*OntoUML Lightweight Editor*;
CONTO	Ontologia de colaboração (*Collaboration Ontology*);
OMG	Linguagem de especificação de modelagem unificada *(Unified Modeling Language Specification);*
NEMO	Núcleo de Estudos em Modelagem Conceitual e Ontologias;
COLLABCODE	Ferramenta para apoio ao desenvolvimento distribuído e colaborativo de software;
ÔMEGA	Um método baseado em padrões ontológicos de colaboração para auxiliar na construção de modelos conceituais de *groupware*;
ALFA (AP-OPL)	O modelo de processo de uso ALFA (AP-OPL) tem por objetivo nortear o uso dos POs de domínio do catálogo de DROPs colaboração consolidado. A AP-OPL baseia-se em UFO-A, uma ontologia de duradouros (*Endurants*);
MDD	Desenvolvimento orientado por modelos;
MFDS	Modelos Físicos de Dados;
RUP	Processo Unificado da Rational (*Rational Unified Process*);
QCs	Questões de competência;
SC	Sessão Colaborativa;
OPL-ML	Linguagem de Modelagem da Linguagem de Padrões (*Ontology Pattern Laguage Modeling Language*);
SABIO	Abordagem Sistemática para a Construção de Ontologias (*Systematic Approach for Building Ontologies*).

SUMÁRIO

CAPÍTULO 1. INTRODUÇÃO 20
1.1 MOTIVAÇÃO 23
1.2 CARACTERIZAÇÃO DO PROBLEMA 24
1.3 OBJETIVOS DA PESQUISA 25
1.4 ABORDAGEM DE SOLUÇÃO 25
1.5 MÉTODO DE PESQUISA 27
1.6 ESTRUTURA DA DISSERTAÇÃO 29
CAPÍTULO 2. REVISÃO SISTEMÁTICA DA LITERATURA 31
2.1 PROCEDIMENTOS METODOLÓGICOS 31
2.1.1 Etapa 1 – Definição do escopo da revisão sistemática 31
2.1.2 Etapa 2 – Busca e seleção de estudos primários 31
2.1.3 Etapa 3 – Extração de dados 32
2.1.4 Etapa 4 – Análise e síntese 33
2.2 CONSIDERAÇÕES FINAIS DA REVISÃO SISTEMÁTICA 34
CAPÍTULO 3. ASPECTOS RELEVANTES SOBRE OS SISTEMAS COLABORATIVOS 38
3.1 SISTEMAS COLABORATIVOS OU GROUPWARE 38
3.2 CLASSIFICAÇÃO DE GROUPWARE 38
3.3 ENGENHARIA DE GROUPWARE 41
3.4 TIPOS DE GROUPWARE 41
3.5 REQUISITOS DE GROUPWARE 43
3.6 REÚSO EM COLABORAÇÃO 47
3.7 COLABORAÇÃO EM GROUPWARE 48
3.8 O MODELO 3C DE COLABORAÇÃO 49
3.9 MODELOS PARA SISTEMAS DE INFORMAÇÃO 50
3.9.1 Modelagem ontológica no apoio à modelagem conceitual 51
3.9.2 Modelos de domínio e ontologias 53
CAPÍTULO 4. ONTOLOGIAS E ASPECTOS RELEVANTES SOBRE O DOMÍNIO DA COLABORAÇÃO 54
4.1 ONTOLOGIAS 54
4.2 ONTOLOGIAS DE FUNDAMENTAÇÃO 56
4.3 UFO: UNIFIED FOUNDATIONAL ONTOLOGY 58
4.3.1 Pressuposto téorico: a linguagem ontouml 62
4.3.2 Análise ontológica 67
4.3.3 Ferramenta oled para modelagem conceitual ontológica 70
4.3.4 Ferramenta oled estendido 72
4.4 PADRÕES ONTOLÓGICOS 72
4.5 LINGUAGEM DE PADRÕES ONTOLÓGICOS 76
4.6 CONTO: ONTOLOGIA DE COLABORAÇÃO 78

4.6.1 Subontologia de Cooperação 79
4.6.2 Subontologia de Comunicação 85
4.6.3 Subontologia de Coordenação 88
4.7 PADRÕES ONTOLÓGICOS DE COLABORAÇÃO 94
4.7.1 Padrões da subontologia de cooperação 95
4.7.1.1 CSL– Localização da sessão colaborativa 95
4.7.1.2 LT – Tipo de localização 96
4.7.1.3 CSS - Tipo de sessão colaborativa 96
4.7.1.4 CSAC – Ações de contribuição da sessão colaborativa 97
4.7.1.5 CSAD – Descrição do acordo de sessão colaborativa 97
4.7.1.6 CSA – Agentes de sessão colaborativa 98
4.7.1.7 CA – Acordo colaborativo 98
4.7.1.8 CAD – Descrição do acordo colaborativo 99
4.7.1.9 CAC – Composição do acordo colaborativo 99
4.7.1.10 CCAC – Compromisso colaborativo para ações de contribuição 100
4.7.1.11 CCIC – Compromisso colaborativo do colaborador 100
4.7.1.12 ACC – Composição da ação de contribuição 101
4.7.1.13 ACD – Dependência da ação de contribuição 102
4.7.1.14 ACPD – Descrição do plano da ação de contribuição 102
4.7.1.15 ACPPS – Situação antes e depois da ação de contribuição 103
4.7.1.16 APAC – Agente realiza ação de contribuição 103
4.7.1.17 GPCCC – Objetivo como conteúdo proposicional de um compromisso colaborativo 104
4.7.1.18 GC – Conflito entre objetivos 104
4.7.1.19 GD – Dependência entre dois objetivos 105
4.7.2 Padrões da subontologia de coordenação 105
4.7.2.1 CSRS – Recursos simples de sessão colaborativa 105
4.7.2.2 CSR – Collaborative session resource 106
4.7.2.3 RC – Composição de recurso 106
4.7.2.4 RN – Natureza de recurso 107
4.7.3 Padrões da subontologia de comunicação 107
4.7.3.1 CIP – Padrão de interação de comunicação 107
4.7.3.2 CIPS – Padrão de interação de comunicação simples 108
4.7.3.3 IL – Padrão de linguagem idiomática 109
4.8 AVALIAÇÃO DOS MODELOS VIA OLED 109
4.9 OPL DE COLABORAÇÃO 110
4.10 CATÁLOGO DROPs DE COLABORAÇÃO 112
4.11 CARREGANDO O CATÁLOGO DROPs DE COLABORAÇÃO NO OLED 114
4.12 QUESTÕES DE COMPETÊNCIA DOS DROPs DE COLABORAÇÃO NO OLED 115
4.13 PADRÕES ALTERNATIVOS OU VARIANTES NO OLED 116
CAPÍTULO 5. PROCEDIMENTOS PARA CONSTRUÇÃO DO MÉTODO E PROCESSO DE USO 118

5.1 ETAPAS DO MÉTODO ÔMEGA 118
5.2 OBTER ARTEFATOS DE COLABORAÇÃO 121
5.3 OBTER REQUISITOS DE GROUPWARE 122
5.4 ANALISAR REQUISITOS DE GROUPWARE 123
5.5 SOLICITAR AJUSTES NA ESPECIFICAÇÃO DE REQUSITOS 126
5.6 OBTER AJUSTES SOLICITADOS NA ESPECIFICAÇÃO DE REQUISITOS 127
5.7 LOCALIZAR INFORMAÇÕES NECESSÁRIAS PARA COMPREENSÃO DO DOMÍNIO 127
5.8 ESTRUTURAR OS REQUISITOS DE GROUPWARE OBTIDOS 129
5.9 EXECUTAR O SUBPROCESSO ADEQUAR OS REQUISITOS DE GROUPWARE AOS PADRÕES 137
5.10 CONSTRUIR MODELO ESPECÍFICO NO OLED 138
5.11 GERAR NOVOS ARTEFATOS DE USO 147
CAPÍTULO 6. DOIS EXEMPLOS DE USO 158
6.1 PRIMEIRO EXEMPLO DE USO (*BRAINSTORMING ONLINE* - COMPLEXIDADE SIMPLES) 158
6.2 SEGUNDO EXEMPLO DE USO (*COLLABCODE* - MAIOR COMPLEXIDADE) 172
CAPÍTULO 7. ESTUDO EXPERIMENTAL PARA AVALIAÇÃO DO MÉTODO E MODELO DE PROCESSO DE USO .. 189
7.1 PLANEJAMENTO DA AVALIAÇÃO 189
7.2 RESULTADOS OBTIDOS 191
7.3 PERCEPÇÃO DOS PARTICIPANTES 200
7.4 DISCUSSÕES 201
7.5 CONSIDERAÇÕES FINAIS SOBRE A AVALIAÇÃO 202
CAPÍTULO 8. CONSIDERAÇÕES FINAIS 203
8.1 CONSIDERAÇÕES FINAIS DO TRABALHO 203
8.2 CONTRIBUIÇÕES 204
8.3 DIFICULDADES E LIMITAÇÕES 204
8.4 PERSPECTIVAS FUTURAS E APRIMORAMENTO DO TRABALHO 205
REFERÊNCIAS BIBLIOGRÁFICAS 206
APÊNDICE A LISTA DE PUBLICAÇÕES INCLUÍDAS NA REVISÃO SISTEMÁTICA 216
APÊNDICE B CHECKLIST DE AVALIAÇÃO DOS REQUISITOS DE GROUPWARE 218
APÊNDICE C FORMULÁRIO UTILIZADO NO ESTUDO EXPERIMENTAL 220
APÊNDICE D QUESTIONÁRIO UTILIZADO NAS ENTREVISTAS DO ESTUDO EXPERIMENTAL 231

CAPÍTULO 1. INTRODUÇÃO

A pressão que a economia tem exercido sobre projetistas de *software* tem sido crescente, exigindo a redução de custos em projetos, demandando que produtos e inovações cheguem ao mercado cada vez mais rápido. E é justamente na tentativa de reduzir custos globais, evitar o retrabalho, encurtando assim o tempo gasto, bem como aumentar a robustez dos produtos e a produtividade, que essas práticas que envolvem a reutilização de *software* vêm acompanhando a evolução da Engenharia de *Software* (SE) Colaborativos (ou *groupware).*

Neste contexto, o processo de desenvolvimento visa a facilitar a execução e o controle de tarefas comuns que, em sua maioria, são realizadas de forma distribuída e colaborativa. Para Gamma et al. (1994), é raro encontrarem-se soluções realmente novas, pois, sempre que os problemas se assemelham a outros encontrados antes, há a reutilização de partes das soluções, ou melhor, antigas soluções tornam-se padrões. Isto ocorre sempre que um segundo sistema é modelado e projetado, ou quando surgem novas alterações em uma aplicação já em produção.

Um sistema *groupware* suporta aplicações que apoiam equipes e grupos de trabalho auxiliando na comunicação e na colaboração nas empresas (O'BRIEN, 2001). *Groupware* pode ser entendido como a tecnologia baseada em mídia digital que dá suporte às atividades de pessoas organizadas em grupos de trabalhos para realizarem tarefas recorrentes e comuns, mesmo trabalhando em locais e horários distintos. Neste caso, o modelo 3C de colaboração é baseado na concepção de que, para colaborarem, os membros de equipe comunicam-se, coordenam-se e cooperam entre si (PIMENTEL; FUKS, 2011).

O modelo 3C é frequentemente usado para classificar sistemas *groupwares*, pois dá uma ideia geral dos elementos que estão contemplados em qualquer tipo de colaboração. Entretanto, o desenvolvimento de *groupware* não é trivial. Como todo software, há aspectos tecnológicos e sociais envolvidos no desenvolvimento. Segundo Gadelha (2011), há muitos serviços comuns entre diferentes aplicações *groupware*, como fóruns de discussão, salas de chat, repositório de arquivos, dentre outros (GADELHA, 2011). Assim sendo, surge a necessidade de se auxiliar os desenvolvedores de *groupware* na criação de aplicações nos aspectos colaborativos necessários para o trabalho em equipe.

Segundo o autor, como consequência da similaridade tecnológica e funcional entre aplicativos do tipo *groupware*, é possível desenvolver um núcleo comum para essas aplicações, de modo que as diferenças entre um *groupware* e outro sejam notadas em função de aspectos funcionais relacionados com o trabalho em grupo que se pretende realizar. Neste contexto, a modelagem conceitual consiste numa atividade importante no projeto de sistemas, pois é a partir dela que se obtém a estruturação dos conceitos abstraídos de um domínio do mundo real, permitindo sua incorporação em um Sistema de Informação (SI) (VILLELA; OLIVEIRA; BRAGA, 2004).

Como as atividades de modelagem ocorrem normalmente durante as fases iniciais do processo de desenvolvimento de sistemas, erros e omissões que não forem detectados inicialmente podem ter repercussões custosas. No entanto, estudos mostram que a modelagem conceitual deve ser uma descrição adequada da realidade do domínio do problema. Em outras palavras, a mesma deve apresentar informações precisas e claras, que podem ser obtidas a partir de uma análise mais detalhada das propriedades dos objetos de um domínio, denominada análise ontológica.

Tal análise ontológica pode ser utilizada como fonte de conhecimento para modeladores conceituais, possibilitando minimizar esses erros semânticos em modelos conceituais. Foi pensando na melhora semântica dos modelos conceituais que escolhemos apenas trabalhar com UFO-A porque, como veremos a seguir, a linguagem OntoUML (*Ontological Unified Modelling Language*), que iremos adotar neste trabalho, até o momento, se utiliza somente dos construtos definidos em UFO-A.

Segundo Guizzardi (2005a), a OntoUML é um perfil UML2.0 (*Unified Modeling Language*) que permite que os modeladores façam distinções de modelagem minuciosas entre diferentes tipos de classes e relações, de acordo com distinções ontológicas reunidas numa ontologia de fundamentação denominada UFO, i.e., UFO-A, a qual lida com os aspectos mais gerais da fundamentação ontológica, os endurantes[1] (também conhecidos como continuantes). A OntoUML incorpora um conjunto de padrões de modelagem de ontologias (*ontological design patterns*) para solução de alguns problemas clássicos de modelagem (GUIZZARDI et al., 2004).

[1] Endurantes - são entidades que não mudam sua essência ao longo do tempo, preservando sua identidade durante as mudanças que ocorrem durante o período de sua existência: “Você é hoje a mesma pessoa que você foi ontem” (GRENON; SMITH, 2004).

Na modelagem, os padrões são blocos (fragmentos) de algum modelo que encapsula conhecimentos previamente discutidos. Esses fragmentos são considerados um dos meios mais eficazes para nomear, organizar e raciocinar sobre algum tipo de conhecimento (FALBO et al., 2013a). Os Padrões Ontológicos (POs) têm sido apontados como ferramentas interessantes para propiciar a reutilização por se mostrarem uma abordagem promissora, favorecendo o reúso de experiências codificadas e de boas práticas (SOUZA, 2017).

Os POs podem ser organizados em uma Linguagem de Padrões (LP), que representa os mesmos e suas relações em uma rede e define um processo para seleção e utilização deles (dos POs) para resolução sistemática de problemas (QUIRINO, 2016). Nesse sentido, uma Linguagem de Padrões Ontológicos (OPL) fornece, além dos padrões, diretrizes para sua seleção, incluindo a sequência em que podem ser usados, as variações existentes e os caminhos possíveis, entre outros.

Assim, OPLs (Linguagens de Padrões Ontológicos) devem ser representadas visualmente por meio de diagramas de atividades da UML (*Unified Modeling Language*) adaptados que representem os POs propriamente ditos, as relações entre eles e o processo que guia sua aplicação (FALBO et al., 2014). Dentre os diferentes tipos de POs existentes para modelagem conceitual, destacamos os Padrões Ontológicos Relacionados ao Domínio (*Domain-Related Ontology Patterns - DROPs*[2]) do catálogo de DROPs de colaboração que foram extraídos diretamente da Ontologia de Colaboração (CONTO), que iremos adotar como um tipo de padrão ontológico de modelagem de conteúdo que captura um fragmento reutilizável extraído de uma ontologia de domínio de referência, apoiando a construção de modelos conceituais de *groupware* ontologicamente bem fundamentados.

Neste trabalho, faremos uso dos POs de domínio (ou DROPs) que foram agrupados no catálogo de DROPs de colaboração baseado na CONTO, incorporado à ferramenta OLED (*OntoUML Lightweight Editor*), através do módulo OLED Estendido criado por Souza (2017), como meio de edição e reutilização de POs de domínio (ou DROPs). Dessa forma, o modelador fica com uma solução completa, podendo aproveitar suas ferramentas de modelagem, verificação e validação. Assim, espera-se que, com o uso dessa ferramenta, possam ser construídos modelos conceituais mais manuteníveis e flexíveis.

[2] DROPs - Apesar desta dissertação ser escrita em português, será utilizado o acrônimo DROP porque é um acrônimo conhecido para padrões ontológicos relacionado ao domínio em inglês. Logo, o acrônimo em inglês será adotado.

Diante do exposto, surge o método ÔMEGA para mitigar esforços no desenvolvimento de modelagem conceitual de sistemas colaborativos (ou *groupware*), já que a OntoUML tem sido considerada altamente complexa para modeladores iniciantes, pois requer, do modelador, além do conhecimento acerca do domínio, o conhecimento acerca das distinções ontológicas e restrições sintáticas da linguagem (GUIZZARDI; GRAÇAS, 2010). O ÔMEGA é munido de um modelo de processo de uso ALFA (AP-OPL - *ALFA-based Software Process Ontology Pattern Language*), tem por objetivo nortear *o* reúso de DROPs (origem) que serão usados na representação esquemática de modelos conceituais no contexto específico (alvo) de *groupware*.

Inicialmente, manusearemos os artefatos de colaboração[3] como aporte conceitual acerca do conhecimento capturado do domínio de colaboração (alvo). Posteriormente, novos artefatos de colaboração serão utilizados. Salienta-se, ainda, que o método não possui uma fase de captura de requisitos, mas sugere a utilização dos paradigmas da estruturação de requisitos fundamentados nas abordagens de Engenharia de Requisitos Orientadas a Objetivos (em inglês, *Goal Oriented Requirements Engineering* ou GORE), com o uso da Técnica de Cenários e preceitos da Teoria da Atividade.

Esta pesquisa caracteriza-se dentro de uma abordagem quali-quantitativa, fazendo uso de dois estudos de caso com metodologia investigativa, tendo a finalidade de avaliar o emprego dos DROPs na construção de modelos conceituais e suas inter-relações em dois exemplos de uso. Ambos os exemplos, aplicados no domínio de colaboração em ambientes de *groupware* reais: um, de complexidade simples e o outro, com nível de complexidade maior. Assim, espera-se aferir esta pesquisa junto a profissionais de tecnologia da informação (TI) em organizações de desenvolvimento de *software* e/ou departamentos de instituições acadêmicas do nível superior.

1.1 MOTIVAÇÃO

A principal motivação para se adotar mecanismos de reúso na modelagem conceitual de sistemas colaborativos (ou *groupware*) está relacionada ao aumento dos níveis de qualidade e produtividade no desenvolvimento de *software*. O aumento de qualidade é uma consequência do reúso de fragmentos de modelos que foram previamente testados e

[3] Artefatos de colaboração - o catálogo “Catálogo de DROPs de Colaboração”, a OPL que representa o “Modelo de Processo da C-OPL” e as "Ontologias de Colaboração (Três subontologias de CONTO)”.

aprovados. Já o aumento da produtividade é resultado de uma redução no tempo de desenvolvimento, evitando-se a reconstrução de partes do sistema que já existam. É inegável que a reutilização de *software,* quando feita de forma intencional, planejada e controlada, leva a alcançarem-se benefícios relevantes.

Muitos dos problemas encontrados após a implementação de um banco de dados (BD) têm sua origem em uma modelagem conceitual feita sem uma análise mais aprofundada da realidade, porque nem sempre se consegue antever exatamente todos os elementos que estão contidos naquela realidade. Assim, percebe-se que a estruturação dos conceitos abstraídos de um domínio do mundo real é uma parte importante de sistemas de informação, pois projetistas podem reutilizar modelos já validados, reduzindo custos e melhorando a qualidade do sistema.

Diante do exposto, entender-se que o aprimoramento de metodologias que guiem a construção da modelagem conceitual é de suma importância. Então, nesta pesquisa, a motivação está pautada na tentativa de mitigar esforços no desenvolvimento de modelagem de sistemas colaborativos (ou *groupware*), por entendermos que modelos conceituais com qualidade elevada tendem a facilitar a detecção e a correção de erros (WAND; WEBER, 2002) e, portanto, o custo e a complexidade da correção de erros em modelos conceituais são muito menores do que a correção de erros durante a execução desses modelos (MOODY, 2005).

1.2 CARACTERIZAÇÃO DO PROBLEMA

Atualmente, muitas empresas de desenvolvimento de *software* ainda encontram dificuldades no desenvolvimento rápido de novas aplicações, com tecnologias modernas. Diante das necessidades de *softwares* cada vez mais funcionais e sofisticados, os desenvolvedores precisam encontrar soluções para superar os obstáculos e atender a demanda de um público cada vez mais exigente, incluindo as grandes corporações (GUTWIN; GREENBERG, 2000). Neste trabalho, portanto, pode-se descrever de forma muito sucinta o problema de pesquisa focado:

Q_P: Como melhorar a qualidade do desenvolvimento de modelos conceituais em *groupware*?

Nesta dissertação, investiga-se, pois a melhoria na qualidade do desenvolvimento de modelos conceituais de sistemas colaborativos (ou *groupware*), com foco no reúso em nível de modelagem, pois, diante das necessidades de *softwares* cada vez mais funcionais e sofisticados, os projetistas precisam encontrar soluções para superar os obstáculos e atender a demanda de um público cada vez mais exigente. Vale ressaltar que os custos com retificação de bugs aumentam em dez vezes mais para cada etapa que o projeto avança.

1.3 OBJETIVOS DA PESQUISA

O objetivo principal deste trabalho de pesquisa, portanto, pode ser descrito da seguinte maneira:

O_0: propor o método ÔMEGA, que é um método baseado em padrões ontológicos de colaboração para construção de modelos conceituais de *groupware.*

Esse objetivo principal pode ser descrito em etapas menores, como segue:

O_1: Propor um modelo de estruturação de requisitos baseado em GORE;

O_2: Propor um modelo de processo de uso ALFA (AP-OPL) para nortear o reúso de modelos conceituais bem fundamentados ontologicamente.

O_3: Propor dois exemplos de utilização do modelo de processo de uso ALFA (AP-OPL) fundamentados no método ÔMEGA, considerando-se um de complexidade simples e outro com nível de complexidade maior; ambos os exemplos, aplicados no domínio de colaboração em ambientes de *groupware* reais.

O_4: Avaliar os resultados obtidos através do uso do método ÔMEGA e seu respectivo modelo de processo de uso ALFA (AP-OPL).

1.4 ABORDAGEM DE SOLUÇÃO

Neste trabalho, foi realizada a construção do ÔMEGA, que é um método baseado em padrões ontológicos de colaboração para construção de modelos conceituais de *groupware.*

Nesse contexto, a padronização dos aspectos tecnológicos dos objetos que estão inclusos nos modelos nos possibilitou lidar com projetos de colaboração em um alto nível, pois os POs (Padrões Ontológicos) de colaboração possuem diferentes níveis de granularidade[4]. A presente pesquisa considera a uso do catálogo de padrões DROPs extraídos da CONTO como

[4] Granularidade - significa maior detalhamento (menor sumarização) dos dados.

um dos recursos de referência na geração, personalização ou construção controlada de modelos.

Diante do exposto, este estudo visou analisar evidências empíricas baseadas na observação sistemática e controlada, a partir de fatos verificáveis na prática do reúso de modelos conceituais, em que projetistas podem consultar o catálogo DROPs de Colaboração, pois o catálogo contém vários POs de domínio oriundos de modelos que já foram testados e validados. Neste caso, a principal motivação para o reúso dos fragmentos de modelos é a aplicação do catálogo na Engenharia de *Groupware* (EG), reutilizados por meio de CONTO, ampliando o conhecimento sobre o domínio da Colaboração (SOUZA, 2017).

Vale ressaltar que o uso de ontologias por parte dos projetistas faz com que os mesmos conheçam a natureza dos objetos do domínio a ser modelado (QUIRINO 2006). Além disto, a ontologia é um conjunto de conceitos dispostos de forma hierarquizada, que descrevem um domínio específico do conhecimento, podendo ser usado para criar uma base de conhecimento (GUARINO, 1997). Neste contexto, a Ontologia de Colaboração (CONTO) segue o Modelo 3C de Colaboração e foi subdividida em três subontologias, a saber: Cooperação, Comunicação e Coordenação.

Recentemente, vários autores da comunidade de Engenharia de Ontologias (EO) já propuseram POs e mecanismos para aplicá-los. No entanto, um método baseado em padrões ontológicos de colaboração para auxiliar na construção de modelos conceituais de *groupware* na prática são raros. Para preencher essa lacuna, esta dissertação propôs o método ÔMEGA e seu respectivo modelo de processo de uso ALFA (AP-OPL). Neste âmbito, experimentos têm mostrado que o uso de POs (DROPs) na criação de modelos conceituais tem um impacto direto sobre a qualidade da ontologia resultante, tornando a modelagem conceitual mais consistente do que as construídas a partir do zero (PRESUTTI et al., 2009).

Na modelagem conceitual, os padrões ontológicos são blocos (fragmentos) de algum modelo que encapsula conhecimentos previamente discutidos. Esses fragmentos são considerados os mais efetivos para nomear, organizar e raciocinar sobre algum tipo de conhecimento (FALBO et al., 2013b). Por definição, temos que um padrão descreve um problema particular recorrente que emerge de um contexto específico e que se apresenta como uma solução aprovada para este problema (BUSCHMANN; HENNEY; SCHMIDT, 2007).

Entretanto, o método é aplicado a partir de uma descrição de requisitos, cuja metodologia de captura e notação não é abordada no espectro de sua atuação. O método parte do princípio de que já exista um documento de especificação dos requisitos de *groupwar*e construído. Neste caso, a captura de requisitos será normalmente executada por um usuário, o qual deve fornecer o documento inicial de requisitos; assim sendo, o ÔMEGA não possui uma fase de captura, mas sugere o uso de um modelo de estruturação de requisitos orientados a objetivos, a fim de facilitar a modelagem ontológica.

O método ÔMEGA está embasado teoricamente num modelo de estruturação de requisitos extraído das abordagens de Engenharia de Requisitos orientadas a objetivos (em inglês, *Goal Oriented Requirements Engineering* ou GORE), com a utilização de cenários e preceitos da Teoria da Atividade. Por conseguinte, o modelo de estruturação de requisitos tem sua argumentação nas premissas de que clientes e usuários normalmente expressam suas necessidades através de objetivos almejados e que a ação humana deve ser analisada dentro de um contexto para que possa fazer sentido e ser compreendida.

Este trabalho discute em detalhes o reúso e a seleção dos DROPs que serão empregados na modelagem, com o intuito de auxiliar projetistas em suas atribuições. Deste modo, foi criado o processo de uso ALFA (AP-OPL) do método ÔMEGA para a projeção do reúso dos fragmentos de modelos. Neste caso, a OPL criada baseia-se nos detalhes de seleção dos POs (origem) para nortear quais POs usar, apoiando-se, assim, na representação esquemática dos fragmentos utilizados no contexto específico (alvo), considerando-se que a pesquisa visou discutir a problemática enfrentada por projetistas de *software* colaborativos em suas dificuldades de desenvolver modelos conceituais bem fundamentados.

1.5 MÉTODO DE PESQUISA

Este trabalho de pesquisa foi desenvolvido seguindo os seguintes passos:

i) Análise do catálogo de DROPs de colaboração já identificados em OntoUML: analisar o catálogo de padrões ontológicos já definidos para linguagem de padrões ontológicos (OPLs) relacionados ao domínio extraídos da ontologia de colaboração (CONTO).

ii) Revisão sistemática da Literatura: neste passo, ocorreu a aquisição de conhecimento sobre os temas relacionados ao trabalho de pesquisa, a saber: a string de busca foi

construída com base na combinação dos termos encontrados que foram concatenados através dos operadores booleanos “OR” e “AND”, como apresentado na Tabela 1.

Tabela 1: String de busca

STRING DE BUSCA
[“Uso de padrões ontológicos na construção de software colaborativos” **OR** “*Use of ontology patterns in the construction of collaborative software*”] **AND** [“Uso de padrões ontológicos de linguagem” **OR** “*Use of ontology patterns language*”]; **AND** [“Uso de padrões ontológicos de software colaborativos” **OR** “*Use of ontology collaborative software patterns*”]; **AND** [“Padrões de colaboração no desenvolvimento de software” **OR** “*Collaboration patterns in software development*”]; **AND** [“Uso de padrões de Ontológicos” **OR** “*Use of Ontology Patterns*”]; **AND** [“Modelo de padrões de colaboração” **OR** “*Collaboration of model patterns*”]; **AND** [“Aplicação da linguagem de padrões Ontológicos” **OR** “*Ontology Pattern Language application*”].

Fonte: Idealizado pela autora

iii) Investigação do Estado da Prática: neste passo, ocorreu a análise das OPLs existentes e a identificação de oportunidades de melhorias.

iv) Desenvolvimento do método e processo de uso: neste passo, foi elaborado um método para auxiliar na construção de modelos conceituais de *groupware,* usando o catálogo de DROPs de colaboração incorporado à ferramenta OLED, através do módulo OLED Estendido.

v) Prova de Conceito: neste passo, foram realizados dois estudos de caso como metodologia investigativa, a fim de se avaliar se os resultados obtidos através da utilização do método ÔMEGA e respectivo modelo do processo de uso ALFA (AP-OPL) trouxeram contribuições para a construção de modelos conceituais de *groupware* baseados em padrões ontológicos de colaboração.

vi) Avaliação da proposta: neste passo, foi realizado um *survey* com projetistas de *software*, com o intuito de avaliar os resultados obtidos através do uso do método ÔMEGA e respectivo modelo do processo de uso ALFA (AP-OPL), considerando a perspectiva dos profissionais de Sistemas da Informação.

vii) Análise dos dados coletados: neste passo, foi realizada a análise dos resultados obtidos durante a execução dos passos anteriores.

1.6 ESTRUTURA DA DISSERTAÇÃO

Esta dissertação está organizada em oito capítulos. Além desta introdução, há mais sete capítulos e quatro apêndices com o seguinte conteúdo:

- ***Capítulo 2*** **- Revisão sistemática da literatura:** neste capítulo, apresentamos uma revisão sistemática da literatura, utilizada como fundamentação teórica para o melhor entendimento e condução desta pesquisa.
- ***Capítulo 3*** **- Aspectos relevantes sobre os sistemas colaborativos:** neste capítulo, apresentamos alguns aspectos relevantes sobre os sistemas colaborativos.
- ***Capítulo 4*** **- Ontologias e aspectos relevantes sobre o domínio da colaboração**: neste capítulo, apresentamos algumas definições e conceitos sobre ontologias e alguns aspectos relevantes sobre o domínio da colaboração.
- ***Capítulo 5*** **- Procedimentos para construção do método e processo de uso:** neste capítulo, apresentamos os procedimentos utilizados na criação do método ÔMEGA e seu respectivo modelo de processo de uso ALFA (AP-OPL).
- ***Capítulo 6*** **- Dois exemplos de uso:** neste capítulo, descrevemos dois exemplos de uso do método ÔMEGA e seu respectivo modelo de processo de uso ALFA(AP-OPL), que é apoiado pelo módulo OLED Estendido do Editor OLED. Dessa forma, esta seção serve como validação do presente trabalho, explorando e verificando a viabilidade do que aqui foi proposto e desenvolvido. Para isso, buscaram-se dois estudos de caso no domínio de colaboração: um, de complexidade simples (*Brainstorming online)* e outro, com nível de complexidade maior (*CollabCode* - Ferramenta para apoio ao desenvolvimento distribuído e colaborativo de software).
- ***Capítulo 7*** **- Estudo experimental para avaliação do método e do processo de uso:** neste capítulo, apresentamos um estudo experimental para avaliar os resultados obtidos através do uso do método ÔMEGA e seu respectivo modelo do processo de uso ALFA (AP-OPL), considerando-se a perspectiva dos profissionais de Sistemas da Informação.
- ***Capítulo 8*** **- Considerações finais:** neste capítulo, apresentamos as considerações finais do trabalho, contribuições, dificuldades e limitações, perspectivas futuras e aprimoramentos do trabalho.

Apêndice A (Lista de publicações incluídas na revisão sistemática)

Apêndice B (*Checklist* de avaliação dos requisitos de groupware)

Apêndice C (Formulário utilizado no estudo experimental)

Apêndice D (Questionário utilizado nas entrevistas do estudo experimental)

CAPÍTULO 2. REVISÃO SISTEMÁTICA DA LITERATURA

2.1 PROCEDIMENTOS METODOLÓGICOS

Na revisão sistemática de literatura, buscou-se identificar as oportunidades, os desafios e as propostas para a melhoria de modelos conceituais de sistemas colaborativos (ou *groupware*). Além disso, buscou-se verificar a existência de abordagens para a identificação automatizada dessas melhorias.

O objetivo principal foi tentar identificar características similares que definam lacunas de pesquisa (ou oportunidades). Para a realização deste estudo, foi definido um processo composto por quatro etapas principais, baseado no processo de revisão sistemática em SE aplicado por Petersen et al. (2008), a seguir:

2.1.1 ETAPA 1 – DEFINIÇÃO DO ESCOPO DA REVISÃO SISTEMÁTICA

Compreendeu a construção do protocolo da pesquisa, com o planejamento do estudo e a definição do processo de revisão, das questões de pesquisa e dos critérios de busca. Foi definida uma única questão de pesquisa: "Como melhorar a qualidade do desenvolvimento de modelos conceituais em *groupware*?".

2.1.2 ETAPA 2 – BUSCA E SELEÇÃO DE ESTUDOS PRIMÁRIOS

A busca de publicações ocorreu de forma automática, através de mecanismos de busca disponibilizados no acervo das bibliotecas digitais. A seleção ocorreu pela análise do título e pelo resumo das publicações. A lista completa de estudos incluídos no mapeamento pode ser encontrada no Apêndice A desta dissertação. As pesquisas selecionadas foram identificadas pelo código [ETXXX].

Foram selecionadas bibliotecas digitais que possuem mecanismos de busca que permitam o uso de expressões lógicas (ou equivalentes) e que comportem a busca no texto completo (ou em campos específicos das publicações). Assim, foram selecionadas as seguintes bibliotecas digitais:

- MINERVA (em modo *Complete Search*) : *< https://minerva.ufrj.br>.*
- IEEE (em modo *Advanced Search*): *<http://ieeexplore.ieee.org>*;
- Scopus (em modo *Advanced Search*): *<http://www.scopus.com>*;

- ACM Digital Library (em modo *Advanced Search*): *<http://dl.acm.org>*;
- DBLP (em modo *Complete Search*): *<http://dblp.org>.*

A pesquisa nas bibliotecas digitais foi feita buscando, como palavras-chaves "Uso de padrões ontológicos na construção de *software* colaborativo". Nesse contexto, foi estipulado o período de 2010 a 2017, para que fossem retornados apenas publicações relativamente recentes e, assim, mais relevantes para a nossa pesquisa.

A *string* de busca utilizada foi ["Uso de padrões ontológicos na construção de *software* colaborativo" **OR** "*Use of ontology patterns in the construction of collaborative software*"] AND ["Uso de padrões ontológicos de linguagem" **OR** "*Use of ontology patterns language*"]; AND ["Uso de padrões ontológicos de *software* colaborativo" **OR** "*Use of ontology collaborative software patterns*"]; AND ["Padrões de colaboração no desenvolvimento de *software*" **OR** "*Collaboration patterns in software development*"]; AND ["Uso de padrões de Ontológicos" **OR** "*Use of Ontology Patterns*"]; AND ["Modelo de padrões de colaboração" **OR** "*Collaboration of model patterns*"]; AND ["Aplicação da linguagem de padrões ontológicos" **OR** "*Ontology Pattern Language application*"].

2.1.3 ETAPA 3 – EXTRAÇÃO DE DADOS

Nesta etapa, buscou-se extrair informações de cada uma das publicações encontradas, de forma a responder à pergunta de pesquisa. Os retornos das pesquisas nas bibliotecas digitais estão relacionados na Tabela 2:

Tabela 2: Retorno das bibliotecas digitais

BIBLIOTECA DIGITAL	TOTAL DE RESULTADOS
MINERVA	18
IEEE	203
Scopus	19
ACM Digital Library	17
DBLP	3
TOTAL: 260 resultados	

Fonte: Idealizado pela autora.

Para a seleção dos resultados, foram usados os critérios de inclusão e de exclusão, expostos na Tabela 2.1:

Tabela 2.1: Critérios de inclusão e de exclusão

CRITÉRIOS DE INCLUSÃO	CRITÉRIOS DE EXCLUSÃO
Textos relacionados com conceitos básicos de colaboração (ou *groupware).*	Textos escritos em idiomas diferentes de inglês e português.
Textos contendo propostas de solução para conceitos de modelagem, ontologias em geral, ontologias na modelagem, ontologias, padrões (*patterns*) e OPL (ou LPO).	Textos não relacionados com o assunto que se deseja estudar e pesquisar.
Textos relacionados com a resolução de problemas de forma similar (Ontologia na colaboração / *groupware*, reúso em colaboração, padrões (*patterns*) para sistemas de *groupware*, OPL (ou LPO) para sistemas de *groupware*, componentes no desenvolvimento de *groupware*).	-

Fonte: Idealizado pela autora

Os resultados foram tabelados em uma planilha *MS Excel*, para identificação de duplicatas e análise dos critérios de inclusão e exclusão citados na Tabela 2.1, usando-se como argumentos os campos "autores", "título" e "resumo" das publicações.

2.1.4 ETAPA 4 – ANÁLISE E SÍNTESE

A partir dos dados extraídos na fase anterior, deu-se o processo de interpretação dos resultados, criação de tabelas e gráficos para exibição dos dados e descrição das evidências identificadas nos estudos primários selecionados. Após o processo de consolidação dos resultados iniciais da pesquisa, chegou-se a um total de 15 artigos, de acordo com o ano de publicação, conforme exibido na Figura 1:

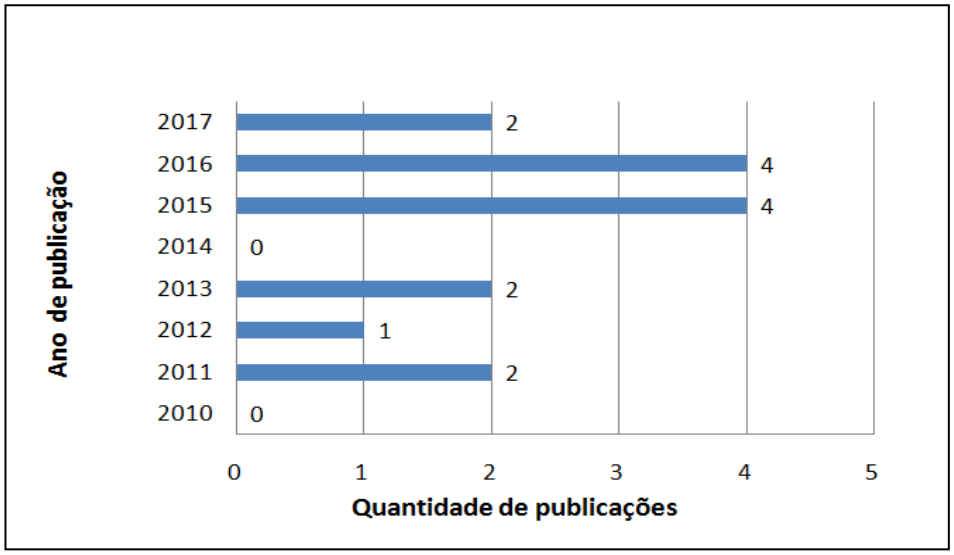


Figura 1: Evolução das pesquisas
Fonte: Idealizado pela autora

As publicações foram distribuídas nas seguintes categorias, propostas pela autora desta dissertação, conforme é descrito na Tabela 2.2 e totalizado na Figura 2:

Tabela 2.2: Categorias das pesquisas

CATEGORIAS DAS PESQUISA	
CATEGORIAS DE PUBLICAÇÕES	**DETALHES**
Conceitos básicos de colaboração (ou *groupware).*	Conceitos básicos de colaboração (ou *groupware*).
Conceitos básicos de solução.	Conceitos de modelagem, ontologias em geral, ontologias na modelagem, ontologias, padrões (*patterns*) e OPL (ou LPO).
Resolução de problemas de forma similar.	Ontologia na colaboração / *groupware*, reúso em colaboração, padrões (*patterns*) para sistemas de *groupware*, OPL (ou LPO) para sistemas de *groupware*, componentes no desenvolvimento de *groupware*.

Fonte: Idealizado pela autora.

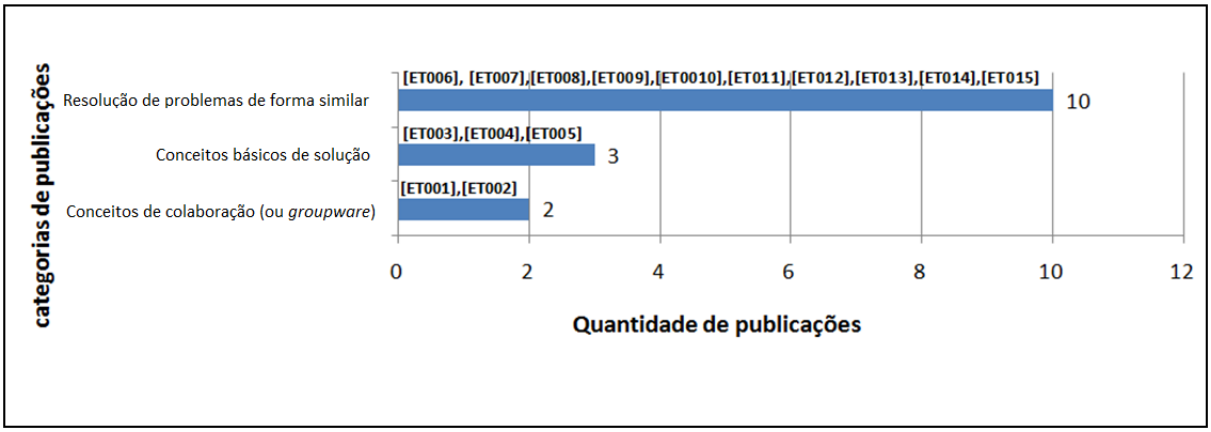


Figura 2: Distribuição das pesquisas por categoria
Fonte: Idealizado pela autora.

2.2 CONSIDERAÇÕES FINAIS DA REVISÃO SISTEMÁTICA

Por meio das publicações descritas a seguir é possível observar que existem trabalhos que relacionam ontologias e modelagem conceitual, seguindo basicamente duas linhas distintas. A primeira linha utiliza modelos ontológicos como ferramentas de validação de elementos de modelagem conceitual, no sentido de fornecer uma definição precisa dos mesmos e, assim, obter modelos conceituais com uma semântica clara e bem definida. A segunda linha utiliza ontologias específicas de domínio como uma ferramenta de apoio à tarefa de modelagem conceitual, no sentido de fornecer conhecimento do mundo real sobre o domínio a ser modelado.

No entanto, faltam propostas que mostrem como adequar a modelagem ontológica às linguagens e aos processos de especificação de *software*. Nesse contexto, observa-se a

relevância do tema "um método baseado em padrões ontológicos de colaboração para construção de modelos conceituais de *groupware*" para as recentes pesquisas acadêmicas. Neste trabalho de pesquisa, as publicações foram estabelecidas em categorias, de acordo com a análise do contéudo, em: publicações relacionadas com conceitos básicos, conceitos de solução e resolução de problemas de forma similar ao tema aqui proposto.

As publicações [ET001], [ET002] foram incluídas na categoria "Conceitos de colaboração (ou *groupware*)". A publicação [ET001] aborda o método colaborativo de observação, a fim de entender coletivamente os ambientes complexos. A publicação [ET002] apresenta um processo de desenvolvimento de *groupware* (*software* para dar suporte ao trabalho em grupo). O processo proposto é resultado de oito anos de experiência com o desenvolvimento dos serviços do ambiente AulaNet, e, mais especificamente, dos cinco anos de pesquisa e desenvolvimento de versões do Mediated Chat.

As publicações [ET003], [ET004], [ET005] foram incluídas na categoria "Conceitos básicos de solução". A publicação [ET003] apresenta uma abordagem sistemática que auxilia na construção de definições consistentes de conceitos que podem ser utilizadas no âmbito computacional, por meio de diretrizes baseadas em ontologia de fundamentação, resgatando sua essência. Já a publicação [ET004] apresenta uma análise arquitetural, ontológica e a proposta de modelo de referência para a recomendação ITU-T G.805[5]. A publicação [ET005] apresenta uma abordagem orientada por modelo para modelagem conceitual calcada nas preocupações ontológicas e informativas.

A publicação [ET006] discute o uso de POs, considerando o contexto de colaboração, levando em conta serem eles aplicavéis a uma ampla gama de áreas, tais como *groupware*, desenvolvimento de *software*, sistemas de gestão de emergência, entre outros domínios. E aborda o uso da ferramenta OLED, através do módulo OLED Estendido, a fim de apoiar o reúso e o gerenciamento de POs específicos de domínio. Tal extensão inclui um catálogo de padrões específicos para domínio da colaboração. É importante esclarecer que faremos uso do catálogo de DROPs de colaboração, incorporado à ferramenta OLED, através do módulo OLED Estendido em nossa pesquisa, visando aproveitar o apoio desse editor preexistente.

[5] ITU-T G.805 - é uma importante recomendação para redes de transportes, pois descreve uma arquitetura funcional genérica independente de tecnologias para este domínio e é usada como base para outras recomentações que descrevem a arquitetura funcional de redes, a gerência, a avaliação de desempenho e a especificação funcional de equipamentos (ITU-T, 2000).

A publicação [ET007] apresenta um sistema de desenvolvimento de ontologia - SEAM (Manutenção Semi-Automatizada de Ontologia) que se utiliza de práticas na informação e de extração de relacionamento de texto para agilizar o processo de geração de estruturas de conhecimento; neste caso, usado especificamente para ontologias de especialidades médicas. Já a publicação [ET008] aborda Collabcode, que é uma ferramenta para apoio ao desenvolvimento distribuído e colaborativo de *software*. Cabe ressaltar que utilizaremos a modelagem da aplicação real do Collabcode no domínio da colaboração como um de nossos estudos de caso.

A publicação [ET009] teve duplo objetivo. Em primeiro lugar, uma revisão da noção de padrões ontológicos na Engenharia de Ontologia (EO) para introduzir a noção de Linguagem de Padrões Ontológicos como forma de organizar Padrões de Ontologia. Em segundo lugar, uma visão geral de uma Linguagem de Padrão de Ontologia do processo de *software*. Já a publicação [ET0010] apresenta o catálogo de padrões para linguagem OntoUML, visando prover, aos usuários mais avançados, o aumento de produtividade e de melhoria na acurácia dos modelos. Além disso, o uso de Linguagens Ontológicas (LPs) tendem a aumentar a complexidade do processo de modelagem. Neste trabalho, foi feito o uso da linguagem OntoUML, derivada da Ontologia de Fundamentação UFO parte A (UFO-A), para construção de modelos conceituais bem fundamentados ontologicamente.

Chamou a atenção a existência da publicação [ET011], escrita por Quirino (2016), que pesquisou "Uma notação visual para representação de linguagens de padrões ontológicos". Podemos fazer uma correlação desse entendimento com o modelo de processo de uso ALFA (AP-OPL - *Alfa-based Software Process Ontology Pattern Language)* usado no método ÔMEGA proposto neste trabalho. Já publicação [ET012] teve como objetivo apresentar a noção de Linguagem de Padrões Ontológicos (OPL) como forma de organizar DROPs. A mesma aborda como uma OPL pode ser usada para construir uma ontologia de domínio. Além disso, cita um exemplo de aplicação do ISP-OPL para a construção de uma ontologia de domínio sobre o processo de definição de requisitos das partes interessadas e apresenta uma visão geral das OPLs existentes.

A publicação [ET014] aborda os esforços feitos para a modelagem e a padronização de processos de *software* ISO/IEC 24744, a utilização de uma Linguagem de Padrões Ontológicos (OPL) como o principal componente dessa infraestrutura ontológica e apresenta

a ISP-OPL (Processo de software baseado em ISO OPL), que é uma OPL que pode ser aplicada como base para a harmonização dos padrões ISO relacionados ao processo de *software*, favorecendo a reutilização. Tendo em vista que os padrões desenvolvidos pelo subcomitê ISO são frequentemente inconsistentes e até contraditórios, isso levou à necessidade de um grupo de estudos do ISO investigar a criação de uma infraestrutura ontológica para estabelecer uma conceituação comum para sustentar todos os padrões SC7.

A publicação [ET015] aborda uma Linguagem de Padrões Ontológicos, denominada S-OPL, fornecendo uma rede de padrões de modelagem de ontologia interligados que cobrem a conceituação básica de serviços. O S-OPL baseia-se em UFO-S, uma ontologia básica baseada em compromissos para serviços. Os padrões S-OPL suportam tipos de modelos de clientes e provedores, bem como as principais fases do ciclo de vida do serviço, a saber: oferta de serviços, negociação ou acordo de serviços e entrega de serviços. O uso do S-OPL é demonstrado em um caso real no domínio do serviço de TIC.

Em síntese, dentro do escopo de aplicação desta revisão sistemática de literatura, nenhuma das publicações pesquisadas abordou, na íntegra o tema proposto neste trabalho de pesquisa. Além disso, faltam propostas na literatura que mostrem como adequar a modelagem ontológica às linguagens e aos processos de especificação de *software* para modeladores de sistemas colaborativos (ou *groupware*). Dessa forma, o aprimoramento de metodologias que guiem a construção da modelagem conceitual é de suma importância, reforçando o valor da contribuição do método ÔMEGA e seu respectivo modelo de processo de uso ALFA (AP-OPL) para a academia e, assim, propiciando o preenchimento de uma lacuna.

CAPÍTULO 3. ASPECTOS RELEVANTES SOBRE OS SISTEMAS COLABORATIVOS

3.1 SISTEMAS COLABORATIVOS OU GROUPWARE

A tradução "Sistemas colaborativos" é adotada no Brasil para designar ambas as denominações: CSCW (Trabalho Cooperativo Auxiliado por Computador, do inglês *Computer Supported Cooperative Work*) e *groupware* (junção das palavras inglesas *group* (grupo) e *software* (*programas de computação*)), conforme Pimentel e Fuks (2011). Segundo os autores, o termo *groupware* costuma ser usado como sinônimo de CSCW, porém alguns autores identificam uma tendência diferenciada no emprego desses termos.

Entretanto, é mais comum associar CSCW com a área de pesquisa do trabalho em grupo e como os computadores podem apoiá-lo; já *groupware* tem sido usado para designar a tecnologia (*hardware* e/ou *software*) gerada pela pesquisa em CSCW (GRUDIN, 1994; ELLIS; GIBBS; REIN, 1991; QUATERMAN, 1990). Para Ellis et al. (1991), um sistema colaborativo (ou *groupware*) foi definido como: "Sistemas baseados em computador que auxiliam grupos de pessoas envolvidas em tarefas comuns (ou objetivos) e que provê interface para um ambiente compartilhado". Assim sendo, *groupware* trata-se de uma série de ferramentas que permitem que as pessoas trabalhem melhor juntas, o que facilita a integração e possibilita mais criatividade e inovação dentro de empresas.

Como exemplos de *groupware,* temos os sistema de comércio eletrônico (inteligência coletiva), repositório de código-fonte (integração, lock, permissão, compartilhamento), as videoconferências, chats e redes sociais, workflow e editores de texto colaborativos (edição cooperativa, revisão, comentários), na medida em que promovem a comunicação entre os membros de um grupo de trabalho e que contribuem, com isso, para que o resultado seja maior que a soma das contribuições individuais de cada membro do grupo.

3.2 CLASSIFICAÇÃO DE GROUPWARE

Segundo Rodden e Blair (1991), podemos classificar as interações das aplicações *groupware* no espaço colaborativo como vulneráveis às variáveis de interação e ao local onde

acontecem. O que resulta numa taxonomia[6] em que as interações podem ser representadas em quatro dimensões de tempo e espaço:

- **Interação síncrona** (ou face a face) - ocorre em tempo real, mas no mesmo local;
- **Interação síncrona remota ou distribuída** - ocorre em tempo real, mas em diferentes lugares;
- **Interação assíncrona** - ocorre em tempos diferentes, mas no mesmo local;
- **Interação assíncrona remota ou distribuída** - ocorre em tempos diferentes e lugares diferentes.

A classificação espaço e tempo é uma das formas mais tradicionais de classificar os sistema colaborativos (ou *groupware*), definida por Ellis et al. (1991). A Figura 3 ilustra a taxonomia das interações na matriz espaço e tempo. Em relação ao eixo tempo, os participantes interagem de maneira síncrona ou assíncrona, enquanto o eixo do espaço indica se eles estão fisicamente próximos ou não. Quando os interlocutores precisam estar simultaneamente conectados ao ambiente para que a colaboração se efetive, o sistema é considerado síncrono. Quando esta simultaneidade não for necessária, o sistema é considerado assíncrono.

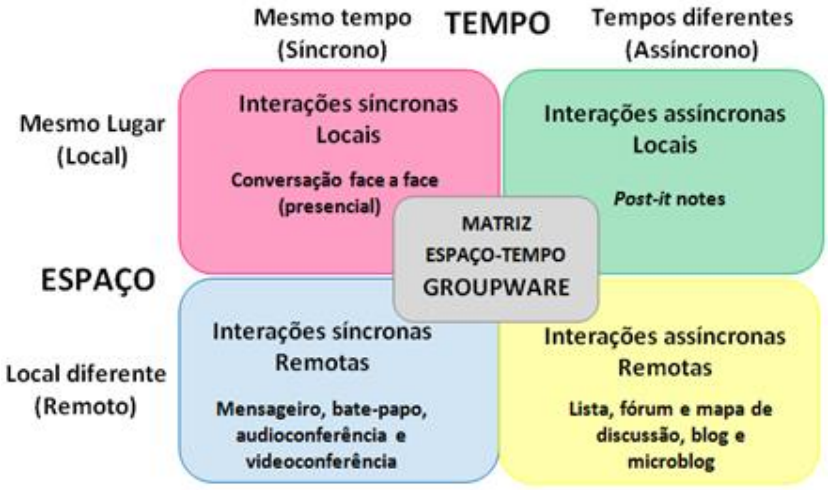


Figura 3: Matriz espaço e tempo de groupware
Fonte: Classificação espaço e tempo, com base em Ellis et al. (1991).

É preciso ocorrer comunicação, coordenação e cooperação, conforme representado no Modelo 3C, para que um trabalho seja caracterizado como colaborativo (PIMENTEL; FUKS, 2011). Para os autores, os sistemas colaborativos são posicionados em um espaço triangular cujos vértices são as dimensões da colaboração, conforme ilustrado na Figura 3.1. Dividindo-

[6] Taxonomia - ciência ou técnica de classificação.

se o espaço triangular em três seções, obtém-se a classificação dos sistemas colaborativos em função do Modelo 3C. No caso, o posicionamento de cada sistema decorre do grau de suporte a cada um dos Cs (BORGHOFF; SCHILCHTER, 2000). Ainda que o objetivo principal de um sistema seja dar suporte a um determinado C, também é preciso dar suporte para os outros dois Cs.

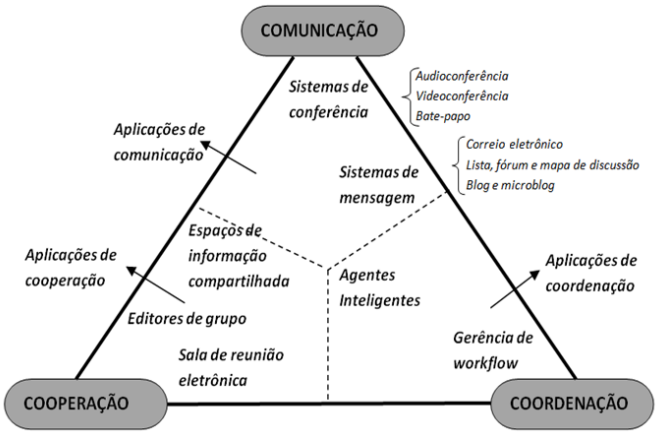


Figura 3.1: Posicionamento dos sistemas colaborativos no espaço em função do Modelo 3C
Fonte: Adaptado de Borghoff e Schilchter (2000).

Por exemplo, em uma videoconferência, mesmo sendo ela um sistema projetado para dar suporte à comunicação interativa, à transmissão da imagem e da voz entre duas ou mais pessoas que estejam em locais diferentes, utilizando câmeras de vídeo, microfones, monitores de vídeo e caixas de som, a sensação que os interlocutores têm é que se encontram no mesmo local através da comunicação audiovisual em tempo real (FUKS; RAPOSO; GEROSA, 2002). Desse modo, ela permite não só a comunicação entre um grupo, mas também a comunicação face a face, e ainda contém elementos para a coordenação (lista de participantes presentes na conversa) e para a cooperação (registro da conversa), conforme ilustrado na Figura 3.2.

Figura 3.2: Elementos para comunicação interativa numa videoconferência
Fonte: Idealizado pela autora.

3.3 ENGENHARIA DE GROUPWARE

Segundo Khoshafian e Buckiewicz (1995), as organizações buscavam novos modelos de gestão e de organização social do trabalho, orientados para o trabalho em equipe, mais participativo e com uma estrutura hierárquica mais reduzida e flexível, o que as tornaria mais competitivas e dinâmicas. Com isso, as ferramentas computacionais e os Sistemas de Informações (SI) passaram a ter uma nova função determinante: dar suporte a grupos de usuários que trabalham de forma colaborativa.

Nesse contexto, as organizações passaram a ter consciência da importância da gestão do seu conhecimento, e esta etapa foi acompanhada pela inserção de novas tecnologias e SI, o que permitiu tratar a informação com qualidade e precisão. Portanto, acompanhando essas necessidades nas organizações, o desenvolvimento e análise dos SI evoluíram para o desenvolvimento de sistemas colaborativos. Para modelar estes SI colaborativos, tem-se a Engenharia de *Groupware*, que estuda o desenvolvimento de ferramentas computacionais e de SI para dar apoio ao trabalho em equipe (CHAFFEY, 1998).

Um *groupware*, geralmente, é composto por um conjunto de ferramentas colaborativas que possibilitam a interação entre múltiplos usuários. Como os processos de trabalho entre os indivíduos são muito específicos e evoluem com o tempo, a tecnologia de *groupware* deve prover flexibilidade suficiente para se adapta às necessidades de cada grupo e à evolução dos processos de trabalho. Assim, o objetivo de um *groupware* é fornecer um ambiente de múltiplos usuários, no qual os participantes possam compartilhar informações, avaliar as contribuições dos outros e, por meio de um processo colaborativo de atividades focadas em diálogo, desenvolver idéias e tomar decisões (THIERAUF, 1989).

3.4 TIPOS DE GROUPWARE

Há uma variedade de produtos de *groupware* disponíveis para apoiar o CSCW no mercado (WILD; WINNIFORD, 1993). Para Schmidt e Bannon (1993), esses programas variam de simples a muito complexos, com alguns direcionados ao usuário médio do computador, enquanto outros são projetados para aplicativos altamente especializados. Com novos programas de *groupware*, recursos e aplicativos aparecendo quase que mensalmente, a terminologia pode ser confusa e inconsistente. No entanto, os produtos de *groupware* compartilham alguns recursos em comum.

A maioria dos *groupwares* oferece recursos de conferência eletrônica (ou de computador), além de recursos básicos de gerenciamento de informações, como banco de dados de referência ou biblioteca de informações, um banco de dados de discussão que contém um registro de todos os comentários e um banco de dados de rastreamento de atividades (GLYN-JONES, 1995). Além disso, os sistemas de *groupwares* aprimorados (ou específicos de aplicativos) podem incluir agendamento de grupo, gerenciamento de projeto, videoconferência, roteamento[7] de fluxo de trabalho e outros recursos (COLEMAN; KHANNA, 1995; THE, 1995).

Apesar da diversidade, é possível classificar o *software* que suporta CSCW em três categorias, que não são nem excludentes nem exaustivas. Em ordem de complexidade, os tipos de *groupware* capazes de suportar CSCW são apresentados a seguir:

- ***Groupware*** "**E-mail e Boletins Eletrônicos** (*E-mail and Electronic Bulletin Boards*)": São métodos de comunicação baseados em computador, que permitem a comunicação assíncrona, não ocorre em "tempo real", entre os membros do grupo em qualquer local (SULLIVAN, 1995). Neste contexto, as mensagens podem ser enviadas e serão armazenadas eletronicamente até serem recuperadas e lidas (FINHOLT; SPROULL, 1990). No entanto, eles não suportam facilmente os altos níveis de interação de grupo necessários para CSCW. Exemplos de produtos: ***Windows Live Mail Desktop, Mozilla Thunderbird*** e **Boletim eletrônico do Instituto Federal**.

- ***Groupware*** "**EMS - Reunião ou Conferências Eletrônicas** (*Electronic Meeting System ou Computer* ou *Electronic Conferences*)": São reuniões eletrônicas nas quais os participantes podem enviar e receber mensagens, editar e modificar as contribuições de outros membros do grupo, desenvolver documentos colaborativos que representem o consenso do grupo e que preservem no banco de dados um registro das atividades e informações do grupo (EDWARDS, 1995; VELAYO, 1994; ZUBOFF, 1988). Independem da geografia, permitem a interação em tempo real (ou mensagens assíncronas) e oferecem suporte total ao CSCW. As conferências (ou reuniões eletrônicas) podem substituir (ou complementar) as reuniões

[7] Roteamento - No contexto das redes de computadores, o encaminhamento ou roteamento de pacotes (em inglês: *routing*) designa o processo de reencaminhamento de pacotes, que se baseia no endereço IP e máscara de rede dos mesmos.

presenciais, as conferências telefônicas e as videoconferências (ELLIS et al., 1991; HUNT; BURFORD, 1994). Exemplos de produtos: ***Collabra Share***, ***IBM WorkGroup*** e ***CollabCode*****.**

- ***Groupware*** "**SADG - Sistemas de Apoio à Decisão em Grupo** (*Group Decision Support Systems*)": Um sistema de apoio à decisão em grupo foi concebido para promover a discussão e a análise de problemas e para melhorar a velocidade e a qualidade do processo de tomada de decisão. As atividades do SADG normalmente ocorrem em uma sala de reunião eletrônica (EMR) com terminais para participantes, uma tela grande visível para todos, um moderador, um banco de dados acessível, com informações relevantes, e um *software* para apoiar a votação, registros e processos de decisão de grupo (BEAUCLAIR, 1990) (ELLIS et al., 1991) (PETROVIC; KRICKL, 1994). Exemplos de produtos: Os produtos SADG comerciais incluem ***Group Systems*** da Ventana, ***Brainstorming*** da Auctus, ***Team Focus*** da **IBM**, entre outros.

3.5 REQUISITOS DE GROUPWARE

Os requisitos das ações e atividades em um CSCW referem-se às funcionalidades gerais e específicas do sistema (SCHNEIDERMAN, 1992). Para o autor, esses requisitos podem ser considerados básicos, não apenas para ambientes de *groupware*, mas para quaisquer interações humano e computador. Os requisitos principais (ou gerais) são os seguintes:

- Identidade individual dos membros: saber quem está logado;
- Posição do usuário: nomes e/ou ícones visíveis na tela;
- Conhecimento: outros integrantes devem saber da presença de um usuário no grupo;
- Visão pessoal ou padrão: visão especificada por cada usuário para um problema;
- Visão convergente: princípio WYSIWIS (*What You See Is What I See*, em português: O que você vê é o que eu vejo);
- Visões particular e pública: partes podem ser vistas por grupos de usuários específicos;
- Controle da palavra: algumas situações necessitam de somente um usuário manipulando um aplicativo (ou emitindo opiniões);
- Controle de acesso: direitos de acesso para cada usuário;
- Atualização e sincronismo (para não haver divergências na edição e ou na entrada e saída de dados);

- Coordenação da informação: coordenação de atividades (ou grupo de atividades), o que aumenta a eficiência.

Geralmente, um *groupware* apresenta requisitos recorrentes (GEROSA, 2006). Para o autor, não é possível especificar completamente um *groupware* a partir de uma lista de requisitos; as listas ajudam a iniciar o processo, instrumentando o desenvolvedor. Nesta seção, são apresentados os requisitos propostos por Tietze (2001), por serem voltados para *groupware* baseado em componentes. Esses requisitos são demonstrados no Capítulo 5 na análise dos requisitos de *groupware* obtidos.

Há diversos trabalhos na literatura (TIETZE, 2001; SCHMIDT E RODDEN, 1996; MANDVIWALLA e OLFMAN, 1994) que catalogam requisitos para instrumentar a especificação, a análise e a avaliação de *groupware*. Como exemplo, temos Tietze (2001), que divide seu catálogo de requisitos em dois grupos: Requisitos do Usuário (RU) e Requisitos do Desenvolvedor (RD). Para Tietze (2001), os requisitos do usuário impactam de modo direto os usuários finais do sistema colaborativo e os requisitos do desenvolvedor impactam os desenvolvedores que utilizam a arquitetura de componentes para criar e adaptar as ferramentas colaborativas.

O autor propõe um total de 20 requisitos de *groupware*, dos quais onze são Requisitos do Usuário (RU) e outros nove são Requisitos do Desenvolvedor (RD). É apresentada, a seguir, nas Tabelas 3 e 3.1, uma breve definição de cada requisito proposto por Tietze (2001), conforme síntese de Oliveira (2011).

Tabela 3: Requisitos de Groupware do Usuário (RGU)

REQUISITOS DO USUÁRIO (RU)	DESCRIÇÃO
RU1 – Acesso aos objetos compartilhados e às ferramentas de colaboração	Consiste em facilitar o acesso do usuário aos itens (arquivos, ferramentas, etc.) colaborativos, garantindo a persistência dos dados e seu compartilhamento.
RU2 – Auxílio na escolha das ferramentas apropriadas	É necessário que o sistema ajude o usuário a escolher a melhor ferramenta para determinada tarefa. Caso haja mais de uma maneira de executá-la, o sistema deve indicar para o usuário a melhor forma de execução.
RU3 – Fornecimento de informações de percepção	O sistema deve emitir alertas referentes à presença de novos usuários conectados à mesma tarefa. Assim sendo, todos os usuários terão a percepção de quem está executando determinada tarefa (e como isto é feito).
RU4 – Colaboração síncrona e assíncrona	Um *groupware* deve oferecer meios de comunicação entre os usuários, de forma síncrona e assíncrona.
RU5 – Acesso ao ambiente independente da	O sistema deve prover acesso remoto aos objetos

estação de trabalho	compartilhados.
RU6 – Fornecimento de espaço privativo e público e transição entre eles	Apesar de ser colaborativo, é ideal que um *groupware* tenha um espaço restrito para cada usuário; assim, cada um terá mais autonomia e privacidade antes de compartilhar algum arquivo.
RU7 – Extensão dinâmica do ambiente	Outra característica desejável dos *groupware*s refere-se à sua capacidade de extensão dinâmica, ou seja, de agregar novos recursos sem a necessidade de reinicialização do sistema.
RU8 – Sincronização entre ferramentas diferentes	Consiste em garantir a sincronia das ações efetuadas sobre um mesmo objeto através de ferramentas distintas.
RU9 – Mobilidade	Esse requisito trata da necessidade de adaptar um groupware para utilização em dispositivos móveis (realizando as limitações necessárias para tal).
RU10 – Agrupamento de ferramentas	Esse agrupamento assegura aos usuários que futuramente, o sistema possa adquirir uma configuração já utilizada anteriormente. Isto é importante para quando ocorrer uma interação similar a outra; com isso, a colaboração poderá ser feita de forma mais rápida.
RU11 – Alta performance	uma das características essenciais para um bom *groupware* é que ele seja minimamente afetado pela latência das operações realizadas. É preciso projetá-lo primando sempre pela eficiência. Se o *software* for demasiadamente lento, o seu uso tende a se tornar inviável.

Fonte: Requisitos de Groupware do Usuário, em Oliveira (2011).

Tabela 3.1: Requisitos de Groupware do Desenvolvedor (RGD)

REQUISITOS DO DESENVOLVEDOR (RD)	DESCRIÇÃO
RD1 – Reúso da experiência e conhecimento anteriores	Este requisito tende a facilitar a integração de novas pessoas à equipe de desenvolvimento, e com isso, aproveitar ao máximo o conhecimento adquirido na implementação de sistemas monousuários.
RD2 – Aproveitamento do modelo de dados	Reutilizando modelos de dados já existentes, o tempo de desenvolvimento é diminuído, pois se aproveitam esforços anteriores.
RD3 – Compartilhamento transparente de dados	Durante a fase de projeto do *groupware*, os projetistas precisam especificar toda a infraestrutura para acesso e compartilhamento dos dados; essa não é uma atribuição do programador.
RD4 – Suporte a dados locais e compartilhados	É necessário que a infraestrutura do software seja desenvolvida de modo a permitir que o desenvolvedor implemente dados locais e compartilhados.
RD5 – Acesso às informações de percepção	O desenvolvedor precisa ter acesso às informações necessárias para oferecer as informações perceptivas aos usuários.
RD6 – Disponibilização de novas ferramentas	As novas ferramentas devem ser disponibilizadas, sem empecilhos arquiteturais, a todos os usuários, a fim de que todos utilizem a mesma versão.

RD7 – Escalabilidade	À medida que novos usuários se conectem, a performance do sistema não pode ser perceptivelmente degradada.
RD8 – Integração com ferramentas externas	É interessante que um *groupware* ofereça a possibilidade de integrar novos recursos externos, sendo ou não recursos adquiridos de terceiros.
RD9 – Suporte a ferramentas localizadas no servidor	Esse requisito trata o aspecto da confiabilidade do sistema. Alguns recursos do sistema devem ser executados no servidor para melhorar o controle de concorrência do acesso a determinados dados. Mas essa diferença na implementação deve ser imperceptível para o usuário.

Fonte: Requisitos de Groupware do Desenvolvedor, em Oliveira (2011).

Para Sommerville (2003, 2008) e Pressman (2006), os requisitos de *software* são definidos como uma funcionalidade (ou condição) que o sistema deve possuir; eles são classificados em vários tipos como: **Funcionais**, **Não Funcionais**, **Domínio**, **Usuário**, **Sistema** e **Interface**.

- **Requisitos funcionais** são as atividades que o sistema deve fornecer, como o sistema deve reagir a determinadas entradas específicas e também verificam o comportamento do sistema em determinadas situações.
- **Requisitos não funcionais** são restrições sobre serviços ou funções oferecidas pelo sistema. Dentre elas, destacam-se as restrições de tempo, sobre o processo de desenvolvimento e de padrões. A descrição das restrições complementa a definição de requisitos (PAULA FILHO, 2000).
- **Requisitos de domínio** são originados do domínio da aplicação do sistema e refletem as características desse domínio, podendo ser requisitos funcionais ou não funcionais.
- **Requisitos de usuário** são declarações, em uma linguagem natural com diagramas, de quais serviços o sistema deve oferecer e as restrições que ele deve realizar. Deve ser compreensível pelos usuários do sistema que não possuam conhecimento técnico detalhado, expressando, apenas, o comportamento externo do sistema. Recomenda-se evitar a utilização de características técnicas de projeto nos requisitos de usuário.
- **Requisitos de sistema** detalham funcionalidades e restrições. Esse documento pode inclusive servir como um contrato entre as partes envolvidas no projeto. Ele é escrito para os profissionais técnicos de nível sênior e para gerentes de projeto.

- **Requisitos de interface** especificam itens externos com os quais o sistema deve interagir com usuários humanos ou não (ENGHOLM JUNIOR, 2016).

3.6 REÚSO EM COLABORAÇÃO

O desenvolvimento de aplicações do tipo *groupware* apresentam diferentes graus e níveis de cooperação. O suporte à cooperação está fortemente ligado às funcionalidades providas por essas aplicações (PICHILIANI, 2006). Neste caso, as aplicações de um sistema *groupware* são compostas por várias características funcionais que nem sempre estão presentes em todos os *groupwares*, porém qualquer sistema que seja considerado *groupware* deve possuir ao menos algumas dessas características funcionais.

Pesquisadores identificaram que grande parte do desenvolvimento de *software* pode ser reutilizada para construcão de outros *softwares* (FERREIRA; HIRAN; NAVES, 2011). Nesse contexto, o reúso de uma aplicação para originar outra deve ter requisitos[8] (ou arquiteturas[9]) similares. Vale destacar que os sistemas de *software* de mesmo domínio apresentam similaridades lexicais nos nomes de classes, métodos e atributos, uma vez que são escolhidos para serem expressivos sobre seus objetivos e as características específicas de um domínio (CYBULSKI; REED, 2000).

Dessa forma, os sistemas do tipo *groupware* no domínio da colaboração apresentam similaridades lexicais nos nomes de classes, métodos e atributos. A análise do domínio pode ser definida como o processo pelo qual a informação usada para o desenvolvimento de *software* é identificada, capturada e organizada para que seja reutilizável na criação de novos sistemas (DIAS, 2015). Além disso, a análise permite identificar requisitos que serviram como base para a implementação de soluções similares, mas que poderiam passar despercebidos durante o processo de levantamento.

Neste âmbito, temos diversos métodos de análise de domínio, sendo estes aplicados em vários níveis conceituais, mas pressupõe-se que qualquer um deles possa ser definido pelos seguintes elementos:

[8] Requisitos - são objetivos ou restrições estabelecidas por clientes e usuários do sistema que definem as diversas propriedades do sistema. Os requisitos de software são, obviamente, aqueles, dentre os requisitos de sistema, que dizem respeito a propriedades do software.

[9] Arquitetura - é a organização ou a estrutura dos componentes significativos do sistema que interagem por meio de interfaces, com elementos constituídos de componentes e interfaces sucessivamente menores.

- Uma ontologia de domínio, assim como a taxionomia dessa ontologia, aparecendo estas no modelo de domínio.
- Um processo a ser seguido para a construção do modelo de domínio.

O modelo do domínio representa a compreensão e a informação adquirida acerca do domínio. O processo de obtenção desse modelo segue os seguintes passos:

- Caracterização do domínio e planejamento do projeto: fase de análise e planejamento.
- Levantamento de dados. As fontes de levantamento de informação são diversas e podem abranger a análise de documentação, consulta a especialistas, etc.
- Análise de dados. O propósito desta fase é construir descrições de componentes reutilizáveis, identificando similaridades e diferenças entre eles.
- Classificação. A informação modelada no passo anterior é refinada, agrupada e hierarquizada.
- Avaliação do modelo de domínio. Esta atividade visa avaliar o modelo resultante obtido, sendo aqui efetuadas as correções necessárias.

3.7 COLABORAÇÃO EM GROUPWARE

A evolução de ambientes computacionais possibilitou a realização de atividades conjuntas por pessoas que não se encontram fisicamente no mesmo local de trabalho, via sistemas que permitem a comunicação entre aplicações distribuídas (MORAES; ZORZO 2000). Para que isto aconteça, segundo os autores, é necessário que as aplicações possuam características que facilitem a interação entre os usuários e forneçam serviços para comunicação e colaboração entre eles.

A interação entre os usuários é uma forma de relacionamento em que há trocas e influência mútua (GEROSA, 2006). Neste âmbito, para caraterizar a colaboração é necessário saber as intenções e objetivos dos participantes, pois, na colaboração, os grupos trabalham juntos em um esforço coodenado (DILLENBOURG e SELF, 1992), com o intuito de atingir um determinado fim, produzindo um resultado. Desta forma, a colaboração é uma maneira de trabalhar em grupo, em que os membros do grupo atuam em conjunto, visando ao sucesso do projeto, sendo que a falha de um dos participantes geralmente implica na falha do grupo como um todo (GROSZ, 1996).

Existem, na literatura, diversos motivos pelos quais a colaboração é importante. Um deles é que os participantes se ajudam, buscando o sucesso das tarefas em conjunto, com objetivos comuns e compartilhados. Outro motivo a destacar é a diversidade de opiniões em um grupo, que possibilita a análise de questões sob diferentes pontos de vista, o que potencialmente resulta numa avaliação melhor (FUKS; GEROSA; LUCENA, 2002). Assim, a colaboração é de grande valia no ambiente de trabalho, tornando possível ao grupo tratar de tarefas complexas e que requeiram habilidades multidisciplinares. No contexto deste trabalho, a colaboração é analisada para nortear o reúso de DROPs de colaboração na construção de modelos conceituais de *groupware.*

3.8 O MODELO 3C DE COLABORAÇÃO

Os sistemas colaborativos (ou *groupware*) que dão suporte ao trabalho em grupo são classificados em três dimensões: Comunicação, Coordenação e Colaboração. Essa classificação deu origem ao modelo 3C de colaboração, que é baseado na concepção de que, para colaborar, os membros de um grupo comunicam-se, coordenam-se e cooperam. Assim, para projetar uma aplicação *groupware* de qualidade, é necessário entender de colaboração, pois a mesma é intrisicamente complexa e norteará o desenvolvimento de *groupware* (GEROSA, 2006).

Na Figura 3.3, é apresentada a esquematização do modelo 3C de colaboração. Nesse modelo, a separação em dimensões foca nos aspectos relevantes para a análise da colaboração; contudo, os C's se inter-relacionam para que a colaboração ocorra.

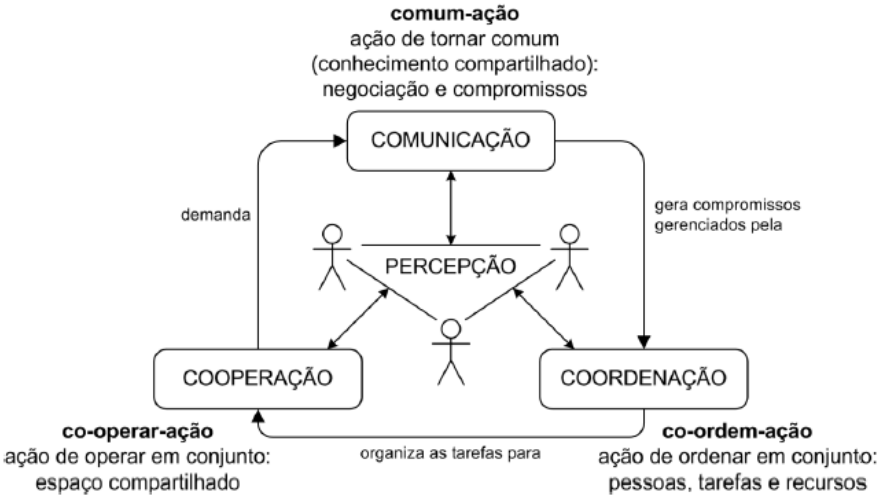


Figura 3.3: Modelo 3C de Colaboração
Fonte: Livro "Sistema Colaborativos" (PIMENTEL; FUKS, 2011).

No trabalho em grupo, a comunicação é voltada para a ação. Enquanto se comunicam, as pessoas negociam e tomam decisões. Enquanto se coordenam, os membros do grupo lidam com os conflitos e organizam as atividades para evitar o desperdício de comunicação e dos esforços de cooperação. Diante disso, a necessidade de renegociar e de tomar decisões sobre situações imprevistas que ocorram durante a cooperação demanda nova comunicação, visto que há necessidade de coordenação para reorganizar tarefas. Por meio de informações de percepção, o indivíduo obtém comentários de suas ações e o ponto de vista das ações de seus colegas (PIMENTEL; FUKS, 2011), o que pode embasar novas decisões.

A colaboração eficaz é um fator determinante para se alcançar os objetivos e, por isso, deve se utilizar todos os recursos disponíveis. As empresas estão cada vez mais dependentes de sistemas colaborativos, devido ao excelente desempenho destes, que, aliado ao seu uso de forma consciente, tem proporcionado bons resultados nos negócios e nos processos empresariais (LAUDON; LAUDON, 2004). Acredita-se que tais sistemas facilitam o uso da informação e da gestão do conhecimento, servindo de suporte à informação e ao trabalho do conhecimento.

3.9 MODELOS PARA SISTEMAS DE INFORMAÇÃO

Os modelos são esquemas abstratos da realidade construídos de forma a selecionar os fenômenos de maior relevância para o problema do objeto de investigação, conforme a visão de modelos representativos. Tal abstração possibilita capturar as características essenciais de um domínio de conhecimento. Exemplos dessa visão são encontrados em Chorley e Haggett (1975) e em Sayão (2001). Nesse ponto de vista, os modelos são definidos como uma estruturação simplificada da realidade, que apresentam características ou relações sob forma generalizada.

Nessa perspectiva, os modelos não incluem todas as observações, mensurações e medições associadas, mas, como tais, são valiosos por ocultarem detalhes secundários e permitirem o aparecimento dos aspectos fundamentais da realidade (CHORLEY; HAGGETT, 1975). Para Sayão (2001), modelo é uma criação cultural, um “mentefato”, destinado a representar uma realidade, a fim de torná-lo descritível e, algumas vezes, observável. Modelos conceituais construídos para o desenvolvimento de SI são exemplos dessa visão.

Os modelos de dados proliferam nas organizações como forma de representar o que deve ser codificado em SI. Ao longo do desenvolvimento de SI, a etapa em que são criados modelos com vistas ao entendimento humano é conhecida como modelagem conceitual. Os modelos conceituais são obtidos a partir de abstrações de aspectos da realidade, na perspectiva de uma pessoa ou de um grupo de pessoas. Neste contexto, as abstrações são formas de especificar as entidades e as relações entre elas, em um domínio do conhecimento de interesse para o sistema em construção.

A modelagem conceitual é uma fase do desenvolvimento de SI, que tem por finalidade descrever a realidade, representando os processos de interesse para desenvolvimento de um sistema. Um problema comum de SI em uso nas instituições é a dificuldade de interoperabilidade entre sistemas desenvolvidos por diferentes equipes, em diferentes períodos e para propósitos diversos. Tal problema tem origem, principalmente, em atividades de modelagem deficientes, conduzidas de forma ad-hoc e sem correspondência com o mundo real (SMITH; WELTY, 2001; GUARINO, 1998).

Em linhas gerais, o objetivo da modelagem conceitual é construir uma representação de alta qualidade de um fenômeno selecionado em algum domínio. Entretanto, os modelos conceituais resultantes devem facilitar o projeto, a implementação, a operação e a manutenção de SI. Como as atividades de modelagem ocorrem, normalmente, durante as fases iniciais do processo de desenvolvimento de sistemas, erros e omissões de modelagem que não forem detectados inicialmente podem ter repercussões custosas. Assim sendo, o aprimoramento das metodologias que orientem a construção de modelos conceituais é de suma importância.

3.9.1 MODELAGEM ONTOLÓGICA NO APOIO À MODELAGEM CONCEITUAL

A modelagem conceitual é uma atividade importante no projeto de sistemas, pois é a partir dela que se obtém estruturação dos conceitos abstraídos de um domínio do mundo real, permitindo sua incorporação em um SI (VILLELA; OLIVEIRA; BRAGA, 1997). Segundo esses autores, para que a modelagem conceitual possa ser uma descrição adequada da realidade do domínio do problema, ela deve apresentar informações precisas e claras, que podem ser obtidas a partir de uma análise mais detalhada das propriedades dos objetos de um domínio.

Tal análise, denominada análise ontológica, pode ser utilizada como fonte de conhecimento para modeladores conceituais, possibilitando minimizar a ocorrência de erros semânticos em seus modelos. Neste sentido, um modelador precisa entender seu contexto de trabalho, para ser capaz de criar uma representação que transmita sua exata percepção[10]. Nem sempre os profissionais responsáveis pela tarefa de modelagem possuem um conhecimento claro sobre os objetos do domínio e os termos que os denotam, o que pode levar a modelos conceituais que apresentem falhas sob o aspecto semântico[11].

É nesse contexto que as ontologias se apresentam com potencial para melhorar a estrutura semântica dos dados na Ciência da Computação e Informação (BRILHANTE, 2005). No entanto, a origem etimológica das ontologias está centrada diretamente no campo da filosofia. “A ontologia trata do ser, daquilo que existe, suas qualidades básicas. É a teoria do ser enquanto tal” (CASTRO, 2008). No fim da década de 1980, observa-se um crescente interesse no uso dessas ontologias de fundamentação no processo de avaliação e de (re) engenharia de linguagens de modelagem conceitual (GUIZZARDI et al., 2011b).

Ainda para Guizzardi et al. (2011b), a modelagem ontológica pode ser definida como a criação de uma ontologia específica de domínio, através da captura das entidades relevantes do domínio e da incorporação das mesmas em um conjunto de categorias que revelam sua natureza, por meio de uma linguagem de especificação de ontologia. Como se pode perceber, existe uma forte semelhança entre esses dois tipos de modelagem, uma vez que ambos objetivam capturar e modelar elementos do mundo real.

Enquanto, na modelagem conceitual busca-se estabelecer as relações entre os conceitos abstraídos de um domínio, na modelagem ontológica o objetivo é identificar os objetos e entender sua natureza, por meio da descrição de suas propriedades. Para Poli (2001), a primeira possui um caráter epistemológico[12], em oposição ao caráter ontológico da segunda. Portanto, pode-se deduzir que as duas modelagens são possivelmente complementares, isto é, a modelagem ontológica pode constituir-se numa base para a modelagem conceitual, no sentido de prover ao projetista, de forma clara e sem

[10] Percepção - Percepção é substantivo feminino, com origem no latim *perceptione,* que descreve o ato, efeito ou capacidade de perceber alguma coisa.

[11] Aspecto semântico - A semântica é o estudo científico do significado de todas as unidades linguísticas, tanto quanto a mudança que ocorre com os significados, no decorrer do tempo.

[12] Epistemológico - significa ciência, conhecimento, é o estudo científico que trata dos problemas relacionados com a crença e o conhecimento, sua natureza e limitações. É uma palavra que vem do grego.

ambiguidades, o conhecimento necessário sobre o domínio a ser modelado (VILLELA et al., 2004).

3.9.2 MODELOS DE DOMÍNIO E ONTOLOGIAS

Os modelos conceituais de domínio resultantes das operações de conceituação em ontologias informacionais podem ser definidos como modelos que representam o conhecimento de um domínio (área de assunto, de missão ou de problema) e que devem servir como uma representação de consenso (ou modelo de referência) de uma conceituação compartilhada por determinada comunidade (CAMPOS, 2017).

Para essa autora, as ontologias informacionais vêm sendo consideradas como sistemas formais para representar conhecimento em determinado domínio, atuando assim como um modelo formal para sistematizar a expressão do conhecimento que manipulam. Nesse contexto, uma ontologia é, formalmente, uma declaração de uma teoria lógica definindo um conjunto de primitivas de representação, com as quais modelamos o domínio do conhecimento; tais primitivas são classes, atributos e relações. As definições explicitam esse entendimento, enquanto os axiomas[13] formais restringem a interpretação.

Diante disto, os modelos conceituais de domínio são utilizados, principalmente, pela comunidade de reúso de *software* (ARANGO, 1994). Tais modelos devem ser independentes da solução física adotada e devem conter apenas elementos referentes ao domínio do problema em questão, ficando para a fase de projeto os elementos da solução, isto é, todos os conceitos que se referem a computadores, como: interfaces, formas de armazenamento (banco de dados), segurança de acesso, comunicação, etc. O uso combinado de modelos conceituais de domínio e ontologias já foi abordado em outros trabalhos; neste, usamos ambas as técnicas, a fim de elaborar modelos conceituais de dominio para representação do mundo a ser modelado.

[13] Axioma(s) - axiomas são verdades inquestionáveis universalmente válidas, muitas vezes utilizadas como princípios na construção de uma teoria ou como base para uma argumentação (ALMEIDA; BAX, 2009).

CAPÍTULO 4. ONTOLOGIAS E ASPECTOS RELEVANTES SOBRE O DOMÍNIO DA COLABORAÇÃO

4.1 ONTOLOGIAS

O estudo de Ontologias na Ciência da Computação é ramo de pesquisa que surgiu no final dos anos 1980, propondo alternativas para representar o conhecimento (GUARINO, 1997). Existem, na literatura, várias definições para o termo ontologia. Uma, interessante para a nossa pesquisa, é apresentada por Fensel et al. (2001): "Uma ontologia é uma especificação formal[14] explicíta[15] de uma conceitualização[16] compartilhada".

Outra definição de ontologias que, de certa forma, complementa, a descrição anterior, é a apresentada por Gómez-Pérez (1999): "Uma ontologia é um conjunto de termos ordenados hierarquicamente para descrever um domínio que pode ser usado como um esqueleto para uma base de conhecimentos". Compreende-se que uma ontologia estuda uma série de formalismos capazes de representar os conceitos, as relações entre os conceitos e a semântica de um domínio do conhecimento.

De acordo com a perspectiva de Malucelli (2002), as ontologias têm como objetivo "deter um conhecimento consensual de forma genérica, podendo ser reusáveis e compartilhadas entre aplicações (*software*) e por grupos de pessoas". Desse modo, convém salientar que, quanto mais precisos formos na representação de um determinado domínio, maiores são as chances de termos sistemas computacionais consistentes e compatíveis com a realidade desse domínio (SCHEIDEGGER, 2016).

Na Figura 4, temos o diagrama apresentado por Guizzardi (2005a), que expressa os conceitos relacionados com a modelagem conceitual: conceituação, abstração, modelo e linguagem de modelagem. Na modelagem conceitual, os conceitos são usados para compor abstrações de uma realidade e são representados por uma linguagem de modelagem. Esta, por sua vez, é usada para compor um modelo que representa uma abstração, completando o ciclo do diagrama apresentado. Geralmente, um modelo conceitual é uma especificação

[14] Especificação formal - legível por computadores.
[15] Especificação explícita - conceitos, propriedades, relações, funções, restrições e axiomas explicitamente definidos.
[16] Conceitualização - coleção de objetos, conceitos e outras entidades que se assume existirem em um domínio.

que corresponde a uma conceitualização de domínio e é escrito em uma linguagem de modelagem conceitual (CARRARETTO, 2012).

Figura 4: Conceitualização, Modelo, Linguagem e Especificação

Fonte:Traduzido pela autora, com base em Guizzardi (2005a).

Existem abordagens que empregam modelos de linguagens de representação "L" mais expressivas de acordo com Guarino (1998). Aderindo a um compromisso ontológico, uma descrição da conceitualização pretendida para uma teoria lógica (GUARINO; GIARETTA, 1995), emprega-se uma linguagem de representação para gerar um conjunto de modelos representativos da realidade. O papel da ontologia é tornar explícitos axiomas que restringem modelos, de forma a igualá-los, tanto quanto possível, aos modelos que contêm o significado pretendido (Figura 4.1).

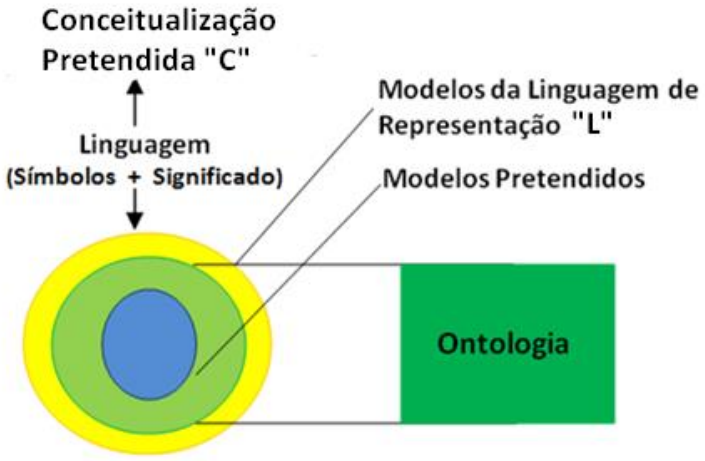


Figura 4.1: Ontologia e linguagem de representação

Fonte: Adaptado de Guarino (1998).

Uma explicação formal deve considerar uma conceitualização “C” representada por uma linguagem de representação “L” que adere a um compromisso ontológico “K=<C, I >”.

Na verdade, “L” se compromete com o domínio “D” através de “K”, do qual “C” é a conceitualização subjacente. A variável “I” representa uma função de interpretação, a qual mapeia elementos de “D” para símbolos do vocabulário “V”. Todos os modelos de “L” que são compatíveis com “K” são modelos pretendidos de “L” de acordo com “K”. Portanto, o papel da ontologia é restringir os modelos de “L” de forma que eles se tornem modelos pretendidos “I_kL”.

Nesse contexto, se uma representação for pouco expressiva, boa parte da conceitualização não será representada, pois uma linguagem de modelagem deve ter os construtos essenciais para representar a especificação do modelo, com suas entidades e relações (GUIZZARDI, 2005a). Para Scheidegger e Campos (2015), quanto mais conhecemos um determinado domínio e quanto mais precisos formos em sua representação, maiores são as chances de termos sistemas computacionais consistentes e compatíveis com a realidade desse domínio.

Ter uma representação precisa de uma conceitualização se torna ainda mais importante e crítica quando queremos compor diferentes modelos que compartilham um mesmo domínio. Atualmente, as empresas convivem com o descasamento entre conceitos definidos em glossários e a representação desses conceitos em modelos conceituais (SCHEIDEGGER, 2016).

4.2 ONTOLOGIAS DE FUNDAMENTAÇÃO

Uma ontologia de fundamentação denominada UFO (*Unified Foundational Ontology*) foi desenvolvida por Guizzardi (2005a), a fim de prover princípios sólidos e robustos para linguagens de modelagem conceitual. As ontologias de fundamentação são sistemas de construtos filosoficamente bem fundamentados, independentes de domínio, e têm sido utilizadas com sucesso para melhorar a qualidade de modelos conceituais e ontologias (PADILHA; BAIÃO; REVOREDO, 2012).

Uma ontologia de fundamentação tem por objetivo identificar categorias gerais de certos aspectos da realidade que não são específicos de um campo científico, descrevendo o conhecimento a despeito da linguagem, de um estado em particular das coisas ou, ainda, do estado de agentes (GUIZZARDI, 2005b).

Borgo e Masolo (2009) definem ontologias de fundamentação a partir de quatro itens:
(i) têm grande alcance;

(ii) podem ser altamente reutilizáveis em cenários de modelagem diferentes;
(iii) são filosófica e conceitualmente bem fundamentadas; e
(iv) são semanticamente transparentes e, portanto, ricamente axiomatizadas.

Segundo Campos e Campos (2012), as ontologias de fundamentação detêm forte fundamentação da Filosofia, permitindo que a estrutura real de um domínio, seu compromisso ontológico, seja representada de forma fiel, clara e consistente. Isto permite que a representação realizada detenha uma semântica baseada no mundo real, restringindo interpretações sobre seus conceitos. Desta forma, as ontologias de fundamentação permitem que a construção de uma teoria sobre o domínio possibilite testar e validar um modelo conceitual.

As autoras ainda identificam que, ao diferenciar os tipos de elementos que compõem um domínio e permitir sua representação, a ontologia de fundamentação explicita conceitos a partir de sua tipologia, estabelecendo sua posição em uma cadeia de elementos. Isto permite, a partir da noção de cada tipo de elemento, que a estruturação do domínio seja construída de forma a evitar proposições errôneas, aferindo qualidade à modelagem conceitual realizada.

Já para Dziekaniak (2010), as ontologias são, talvez, o melhor instrumento para classificação e representação do conhecimento até então pensado e desenvolvido, fornecendo ampla possibilidade de uso inclusive por atores que não dominam a linguagem de máquina. Nesse caso, as ontologias são divididas em classificações de diferentes tipos, de acordo com o nível de generalidade necessária. A seguir, são apresentados quatro tipos de ontologias segundo essa classificação (MAEDCHE, 2002):

- Ontologias de alto nível, que incluem vocabulário relacionado a conceitos globais, não pertencendo a um único tema ou domínio. Esses conceitos são genéricos e podem ser especificados por ontologias de domínio ou tarefa.
- Ontologias de domínio, que representam um determinado domínio, podendo ser reutilizáveis dentro desse domínio. Os conceitos desse domínio e seus relacionamentos com as atividades relacionadas desse domínio são representados.
- Ontologias de tarefa, que representam conjunto de conceitos usados para resolver problemas que podem ser ou não de um mesmo domínio. Dessa forma, inclui nomes de conceitos genéricos, podendo ser reaproveitados em diferentes domínios.

- Ontologias de aplicação, que são ontologias criadas a partir de conceitos de um domínio particular, em certos casos, da aplicação de uma tarefa em particular em um domínio específico.

Para Guarino (1998), as ontologias se relacionam segundo o nível de generalidade, conforme representado na Figura 4.2.

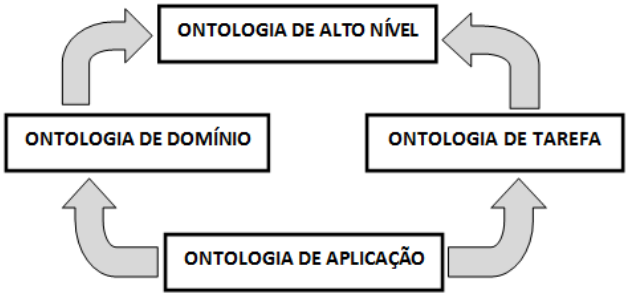


Figura 4.2: Diferentes tipos de ontologias
Fonte: Adaptado de Guarino (1998).

Por meio dessas descrições, pode-se observar que as ontologias de domínio são ontologias reutilizáveis em um domínio genérico, que oferecem vocabulários sobre conceitos, seus relacionamentos, sobre atividades e regras que os governam; por ex., medicina, automóveis, etc.

Segundo Silva, Souza e Almeida (2008), tais ontologias necessitam das ontologias de fundamentação para a concepção de sua estrutura inicial, pois as ontologias de fundamentação são ontologias de alto nível, que representam conceitos globais, não restritos a um domínio de representação. Esse tipo de ontologia pode ser usado para a representação das metapropriedades de uma linguagem para modelagem conceitual (LOPES, 2011).

4.3 UFO: UNIFIED FOUNDATIONAL ONTOLOGY

A Ontologia de Fundamentação Unificada UFO (*Unified Foundational Ontology*) é resultado da combinação da GFO (*General Formal Ontology*), da DOLCE (*Descriptive Ontology for Linguisitics and Cognitive Engineering*) e de princípios da metodologia *OntoClean*. Como síntese de outras ontologias, a UFO (*Unified Foundation Ontology*) oferece maior número de construtos para avaliação de linguagens de modelagem e modelos conceituais. As metapropriedades desses construtos auxiliam o modelador a identificar e classificar os conceitos de um domínio (GUIZZARDI, 2000; 2005a).

A UFO propõe um sistema de categorias formado por **Universais** (*Universals*) e **Individuais** (*Individuals*), baseado no modelo aristotélico, para categorizar os entes que existem. Em modelagem conceitual, a categoria **Universal** refere-se a tipos (classes, conceitos) e a categoria Individual refere-se às instâncias de tipos. Universal é a categoria de padrões de características que podem ser instanciados em diferentes **Individuais**. Exemplo: pessoa, móvel, animal, etc. Os **Universais** (*Universals*) podem, também, ser categorizados quanto à natureza do ser e o tempo. Desse modo, os **Universais** podem ser **Universal Duradouro** (*Endurant Universal*), que define tipos gerais que existem em sua totalidade em cada instante do tempo; e **Universal Evento** (*Perdurant ou Event Universal*), que define tipos gerais que ocorrem parte a parte em cada instante do tempo.

Os Universais subdividem-se em **Universal Substancial** (*Substantial Universal*) e **Momento Universal** (*Moment Universal*). Um *Substantial Universal* é um Universal que é instanciado apenas por ***Substantial Individuals***. Por sua vez, *Moment Universal* é um Universal que é instanciado apenas por *Moment Individuals*. Os *Substantial Universals* subdividem-se em **Sortais Universais** (*Sortal Universal*) e **Mistos Universais** (*Mixin Universal*) segundo (GUIZZARDI, 2005a, p.106;315).

- **Sortal Universal** (*Sortal Universa*l) é um **Universal** que fornece um princípio de identidade e individualização para suas instâncias. Significa afirmar que todas as instâncias de um **Sortal** possuem o mesmo princípio de identidade. A classe "*Sortal Universa*l" especializa-se em quatro subclasses: "*Kind*", "*SubKind*", "*Phase*" e "*Role*". Por exemplo: pessoa, árvore, mesa. Os **Sortais Universais** podem ser **rígidos Sortal** (*RigidSortal*) ou **anti-rígido Sortal** (*AntiRigidSortal*).

- **Misto Universal** (*MixinUniversal*) ou Não-Sortal Universal é um Universal e representa os tipos dispersivos. Os dispersivos são entidades que cobrem conceitos com princípios de identidade diferentes e, portanto, não são sortais. Todos os elementos que são especializados de um sortal carregam o princípio da identidade. O *MixinUniversal* representa propriedades abstratas que são comuns a tipos diferentes. A classe "*MixinUniversal*" especializa-se em três subclasses: "*Category*", "*RoleMixin*" e "*Mixin*". Por exemplo: o conceito de Cliente que tanto pode ser um Cliente Pessoa Física quanto um Cliente Pessoa Jurídica.

Na UFO, apenas Sortais podem fornecer um princípio de identidade e individualização para suas instâncias. Em Guizzardi (2005a, p. 98-99) é proposto um conjunto de postulados e definições relacionados aos Sortais e sobre o conceito de rigidez dos objetos. Dentre eles, destacam-se (Tabela 4):

Tabela 4: Postulado e definições relacionados aos Sortais

POSTULADO	DESCRIÇÃO
Postulado 3.1	Todo Individual[17] em um modelo conceitual do domínio deve ser uma instância de um tipo de modelagem conceitual (CM-Type) representando um Sortal[18].
Postulado 3.2	Um Individual representado em um modelo conceitual do domínio deve instanciar exatamente um CM-Type representando o último Sortal que fornecerá o princípio de identidade para suas instâncias. Este último Sortal é denominado Substancial Sortal.
Postulado 3.3	Um CM-Type Rigid Universal não pode ser uma especialização de um CM-Type Anti-Rigid Universal.
Postulado 3.4	Um CM-Type representando um *Dispersive Universal*[19] não pode especializar um CM-Type representando um Sortal.

Fonte: Baseado em Guizzardi (2005a).

O conceito de rigidez também é importante nas definições das categorias de *Sortals* e *Mixins*. Os *Sortals* podem ser **Rígidos Sortais** (*RigidSortal*) e **Anti-rígidos Sortais** (*AntiRigidSortal*). Já os Mistos Universais (*MixinUniversal*) ou Não-Sortal Universal podem ser **Rígidos Mistos** (*RigidMixin*) e **não-rígidos Mistos** (*NonRigidMixin*). Os **Não-rígidos Mistos** (*NonRigidMixin*), ainda se subdividem, em: **Anti-rígidos Mistos** (*AntiRigidMixin*) e **Semi-rígidos Mistos** (*Mixin*), como mostra a Tabela 4.1:

Tabela 4.1: Definições das categorias de Sortals e MixinUniversal (ou Não-Sortals)

DEFINIÇÃO	DESCRIÇÃO
Definição 3.1	**Tipo rígido**: É todo aquele que se aplica necessariamente a todas as suas instâncias, ou seja, se x instancia um type T em um dado mundo w, então x deve instanciar type T em todo mundo possível w´. Exemplo: pessoa, animal, planta.
Postulado 3.2	**Tipo anti-rígido**: É todo tipo que se aplica contingentemente a todas as suas instâncias, ou seja, se x instancia T em um dado mundo w, então existe um mundo possível w´ em que x não instancia T. Exemplo: o papel de estudante, as fases de uma pessoa: criança, adulto, idoso.
Postulado 3.3	**Tipo semi-rígido**: É todo tipo que tem as características de ser eventualmente aplicável a algum de seus indivíduos e necessariamente aplicável a outros. Ou seja, é um tipo que não é nem rígido nem anti-rígido. Exemplo: coisas que possuem múltiplas finalidades/atribuições, como um objeto "assentável" ou itens asseguráveis.

Fonte: Baseado em Guizzardi (2005a).

[17] Um individual - não pode submeter-se a dois princípios de identidade incompatíveis.
[18] Sortal Type é todo tipo que provê a individualização de suas instâncias e garante o princípio da identidade única (se A=X e B=X então A=B). O princípio da identidade única baseia-se na unidade da essência aristotélica.
[19] Dispersive universal - são conceitos com diferentes princípios de identidade. Os conceitos dispersivos não denotam sortals apesar de serem considerados como "nomes comuns" em linguagem natural e não poderem ter instâncias diretas (GUIZZARDI, 2005). É o caso dos mixins.

A categoria de *Individual* pode ser dividida em **Duradouros** (*Endurants*) e **Eventos** (*Perdurants*). Os **Duradouros** (*Endurants*) são os indivíduos que "são no tempo", não havendo alteração na essência do objeto, podendo haver alterações nos aspectos acidentais do objeto com o passar do tempo. Exemplo: pessoa, carro, animal, planta, etc. Em contraste, os **Eventos** (*Perdurants*) são os indivíduos compostos de partes temporais que acontecem no tempo (GUIZZARDI et al., 2013). Exemplo: o evento que dá início a uma partida de futebol e o evento que inicia uma conversa.

Simetricamente aos *Universals,* os individuos *Endurants* subdividem-se em *Substantial* e *Moment.* Enquanto *Moments* são instâncias de um *Moment Universal, Substantials* são instancias de um **Universal Substancial** (*Substantial Universal),* completando o quadro ontológico aristotélico mencionado em Guizzardi (2005a, p. 273).

As categorias (ou fragmentos) da UFO estão distribuídas em UFO-A, UFO-B e UFO-C:

- **UFO-A** - **Ontologia de duradouros ou objetos** (*Ontology of Endurants*) é a camada que define os tipos cujas instâncias persistem no tempo, não havendo alteração na essência do objeto com o passar do tempo.
- **UFO-B** - **Ontologia de eventos** (*Ontology of Perdurants*) é a camada construída sobre a UFO-A que define indivíduos (individuals) compostos de partes temporais que acontecem no tempo.
- **UFO-C** - **Ontologia de entidades sociais** (*Ontology of Social and Intentional Entities*) é a camada que relaciona conceitos e relações referentes à construção da realidade social.

Escolhemos trabalhar somente com UFO-A porque, como veremos a seguir, a linguagem OntoUML (que iremos adotar neste trabalho), até o momento, se utiliza apenas dos construtos definidos em UFO-A. A UFO-A trata dos objetos (*Endurants*). Os *Endurants* são entidades que perduram através do tempo. Mesmo que o valor de suas propriedades se alterem (por ex.: idade, tamanho, peso), sua identidade é preservada. Eduardo, uma pessoa, pode mudar algumas de suas características, tais como peso e idade, mas continua sendo o mesmo Eduardo ao longo de sua existência. Ele é classificado como elemento da UFO-A.

Na Figura 4.3, temos uma categoria (ou fragmento) da UFO-A a partir dos **Universais Substanciais** (*Substancials Universals)*. Nesse contexto, as categorias (ou fragmentos) foram

apresentados na língua portuguesa, exceto na parte pontilhada do diagrama, que foi mantida em inglês, por ser a língua original dos padrões ontológicos utilizados no OLED.

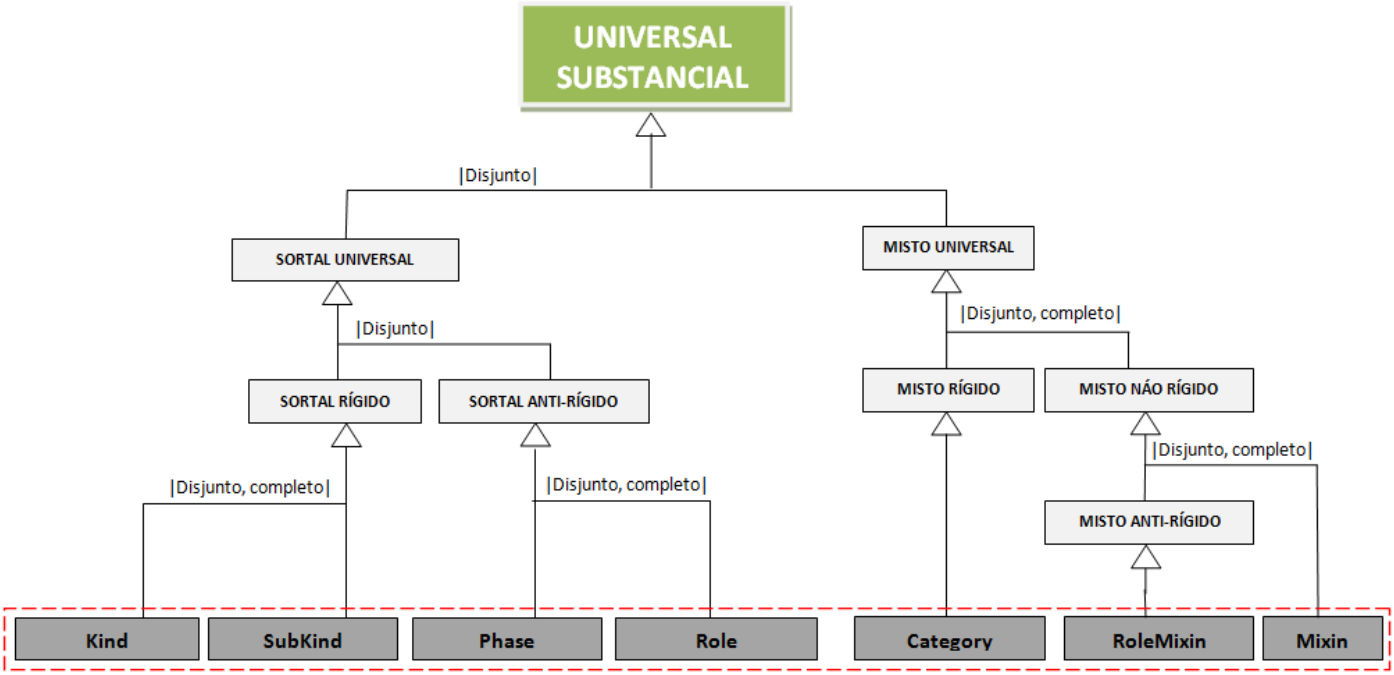


Figura 4.3: Parte da Ontologia de Fundamentação Unificada (UFO-A)

Fonte: Traduzido pela autora, com base em Guizzardi (2005a).

Assim, o presente trabalho surge como mais uma iniciativa de se utilizar as metapropriedades da OntoUML derivadas da UFO, i.e., UFO-A, as quais governam a forma como o processo de modelagem funciona, a fim de propor um método baseado em padrões para auxiliar na construção de modelos conceituais de *groupware* bem fundamentados, usando os padrões ontológicos de colaboração.

4.3.1 PRESSUPOSTO TÉORICO: A LINGUAGEM ONTOUML

A linguagem OntoUML (*Ontological Unified Modelling Language*) foi primeiramente apresentada em Guizzardi (2005a), como uma extensão da versão 2.0 da UML (*Unified Modeling Language*). A OntoUML, de acordo com Guizzardi et al., (2009, p.3), foi construída seguindo um processo no qual: **(i)** o metamodelo da linguagem original (no caso, a UML 2.0) é reparado para garantir um isomorfismo[20] em seu mapeamento para a estrutura definida pela ontologia de referência (no caso, UFO-A); **(ii)** em segundo lugar, a axiomatização da ontologia de fundamentação é transferida para o metamodelo da linguagem, por meio de restrições formais incorporadas a esse metamodelo.

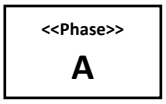
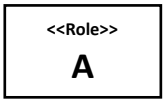
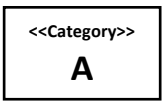
[20] Isomorfismo - [Figurado] Relação de semelhança entre uma coisa e outra.

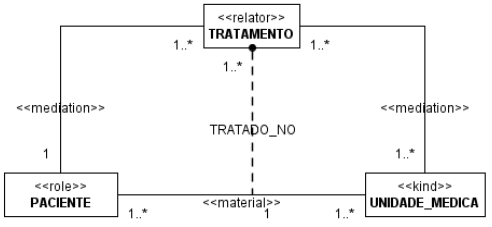
É importante destacar que a OntoUML estende o construto Class de UML para capturar as distinções ontológicas presentes na UFO-A. Com essa extensão, a OntoUML acrescenta várias distinções axiomáticas inter-relacionadas, fornecendo fundamentos de modelagem para todos os conceitos principais de modelagem conceitual, incluindo teorias para tipos e estruturas taxonômicas (incluindo papéis), relações parte-todo, eventos, relações formais e materiais, atributos e valores de atributo e espaços de medição. E também permite a produção de ontologias de qualidade e usá-las como um modelo conceitual de alta expressividade.

A Tabela 4.2 apresenta as categorias ontológicas de diferentes tipos de Classes que implementam o metamodelo ONTOUML. As entidades que serão mais relevantes neste trabalho são as seguintes: *Kind*, *Subkind*, *Category*, *Role*, *RoleMixin*, Mixin, *Quantity*, *Relator* e *Modo*.

Tabela 4.2: Categorias ontológicas de tipos de Classes

ESTEREÓTIPO	CLASSE BASE	DESCRIÇÃO	RESTRIÇÕES
UNIVERSAL SUBSTANCIAL SORTAL			
SORTAIS RÍGIDOS			
<<Kind>> **A**	Classe	Um **tipo** «Kind» representa um Universal do tipo *Substancial Sortal* cujas instâncias são complexos funcionais que agregam indivíduos com o mesmo princípio de identidade. Por exemplo, incluem instâncias de tipos naturais (como Pessoa, Cão, Árvore) e artificiais (Cadeira, Carro, Televisão).	1. Um Kind não pode ter como supertipo um membro de {«Phase», «Role», «RoleMixin»}.
<<SubKind>> **A**	Classe	Um **subtipo** «SubKind» representa um *Universal* do tipo *Substancial Sortal* cujas instâncias são tipos rígidos e relacionalmente independentes que carregam o princípio de identidade de *Universal* do tipo *Substancial*. Por exemplo, temos os Subkinds Homem e Mulher do Kind Pessoa. Neste caso, temos que em ambos os Subkinds estão adicionadas as propriedades de gênero do Kind Pessoa.	1. Um subkind não pode ter como supertipo um membro de {«Phase», «Role», «RoleMixin»}.
<<Quantity>> **A**	Classe	Uma **quantidade** «Quantity» representa um *Substancial Sortal* cujas instâncias são quantidades. Exemplos são aquelas coisas Universais que são tipicamente referidas em linguagem natural por termos gerais de massa (por exemplo, Ouro, Água, Areia, Argila).	
<<Collective>> **A**	Classe	Um **coletivo** «Collective» representa um *Substancial Sortal* cujas instâncias são coletivas, ou seja, são coleções de complexos que possuem uma estrutura uniforme. Exemplos incluem um baralho de cartas, uma floresta, um grupo de pessoas, uma pilha de	1. Um coletivo «Collective» pode ser extensional. Isso significa que todas as suas partes são essenciais e a mudança (ou destruição) de qualquer uma de suas partes termina com a

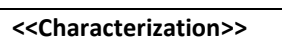
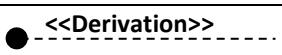
		tijolos. Os coletivos podem se relacionar tipicamente a complexos por meio de uma relação de constituição. Por exemplo, uma pilha de tijolos que constitui uma parede, um grupo de pessoas que constitui um time de futebol. Nesse caso, os coletivos geralmente têm um princípio de identidade extensional, em contraste com os complexos que constituem. Por exemplo, os Beatles estavam em um dado mundo w constituído pelo coletivo {John, Paul, George, Peter} e em outro mundo w' constituído pelo coletivo {John, Paul, George, Ringo}. A substituição de Peter Best por Ringo Star não altera a identidade da banda, mas cria um grupo qualitativamente diferente de pessoas.	existência do coletivo. Usamos o símbolo {extensional} para representar um coletivo extensional (*extension*).
SORTAIS ANTIRÍGIDOS			
	Classe	Um **papel** «Role» representa um Universal do tipo *Substancial Sortal* cujas instâncias são tipos anti-rígidos e relacionalmente dependentes, os quais representam os papéis dos tipos envolvidos em um relacionamento.	1. Um papel «Role» não pode aparecer em um modelo conceitual como um supertipo de um universal rígido. 2. Um papel «Role» é um tipo anti-rígido e precisa carregar um princípio de identidade fornecido por um substancial sortal. 3. Um papel «Role» é externamente dependente de um *Relator*.
	Classe	Uma **fase** «Phase» representa um Universal do tipo *Substancial Sortal* cujas instâncias são tipos anti-rígidos e relacionalmente independentes definidos como parte de uma partição de um *Substancial Sortal*.	1. Uma fase «Phase» não pode aparecer em um modelo conceitual como um supertipo de um universal rígido. 2. As fases {P1...Pn} que formam uma partição de fase de um substancial sortal S são representadas em um diagrama de classes como um conjunto de generalização disjunta e completa. Em outras palavras, um GeneralizationSet com (isCovering = true) e (isDisjoint = true) é usado em um mapeamento de representação como a representação para o conceito ontológico de uma partição de fase.
UNIVERSAL SUBSTANCIAL MIXIN			
	Classe	Uma **categoria** <<Category>> representa um universal do tipo substancial mixin cujas instâncias são tipos rígidos e relacionalmente independentes. No geral, eles agregam propriedades que são comuns a diferentes *Substanciais Sortals*.	1. Uma «categoria» não pode ter um «roleMixin» como um supertipo. Em outras palavras, junto com a condição 1 para todos os mixins, temos que uma "categoria" só pode ser incluída

			por outra “categoria” ou um “mixin”.
<<RoleMixin>> **A**	Classe	Um **papel misto** <<RoleMixin>> representa um universal do tipo substancial mixin cujas instâncias agrupam tipos anti-rígidos e com diferentes princípios de identidade. Assim como *Roles*, os *RoleMixins* também são relacionalmente dependentes, o que significa que eles precisam estar relacionados direta ou indiretamente (por algum de seus supertipos) com algum *Relator*.	1. Um papel misto <<RoleMixin>> agrupa tipos anti-rígidos (e.g, *Role* e *RoleMixin*); eles também precisam estar relacionados a outros tipos através de associações que representam as condições relacionais de instanciação de suas instâncias (dependência relacional), tais como relações materiais e de mediação.
<<Mixin>> **A**	Classe	Um **misto «**Mixin**»** representa propriedades que são essenciais para algumas de suas instâncias e acidentais a outras (semi-rigidez). Um exemplo é o Mixin Sentável, que representa uma propriedade que pode ser considerada essencial para os tipos de cadeira e assento sem encosto, mas acidental para caixa de madeira, caixa de papel.	1. Um «mixin» não pode ter um «roleMixin» como um supertipo.
MOMENTO UNIVERSAL			
<<Mode>> **A**	Classe	Um **Modo** «Mode» é um universal momento intrínseco universal. Cada instância do **Modo Universal** é existencialmente dependente de exatamente uma entidade. Exemplos incluem habilidades, pensamentos, crenças, intenções, sintomas, objetivos privados.	1. Cada «mode» deve ser (direta ou indiretamente) conectado a um fim de associação de pelo menos uma relação de «caracterização».
<<Relator>> **A**	Classe	Um **relacional «**relator**»** é um momento relacional universal. Cada instância do relator universal é existencialmente dependente de pelo menos duas entidades distintas. *Relators* são a instanciação de propriedades relacionais como casamentos, beijos, apertos de mão, compromissos e compras. Um exemplo pode ser observado na Figura 4.4, na qual o Relator “TRATAMENTO” conecta um “PACIENTE” a um “UNIDADE_MEDICA”. Uma relação material é representada pelo estereótipo <<material>> (UML é a classe básica Association).  	1. Todo «relator» deve estar (direta ou indiretamente) ligado a um fim de associação em pelo menos uma relação de «mediação». 2. Seja R um relator universal e seja {C1...C2} um conjunto de universais mediados por R (relacionado a R através de uma relação de mediação). Finalmente, seja $menor_{Ci}$ o valor da restrição de cardinalidade mínima da extremidade de associação conectada a Ci na relação de mediação. Então, nós temos isso: $\left(\sum_{i=1}^{n} menor_{Ci}\right) \geq 2$

Fonte: Traduzido pela autora, com base em Guizzardi (2005a).

Em OntoUML temos uma grande quantidade de estereótipos para as relações, e cada uma possui um propósito específico. Devido a isto, as relações são divididas entre **Relações de Dependência** (*Dependency Relations*) e **Relações de Todo-Parte** (*Meronymic Relations*). Na Tabela 4.3, apresentamos o primeiro caso, em que encontramos as relações de dependência: **Material** (*Material*), **Mediação** (*Mediation*), **Caracterização** (*Characterization*), **Formal** (*Formal*) e **Derivação** (*Derivation*).

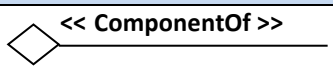
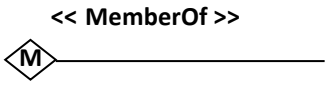
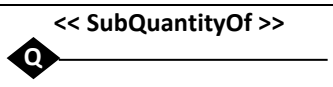
Tabela 4.3: Relações de dependência (Dependency Relationship)

RELAÇÕES DE DEPENDÊNCIA	DESCRIÇÃO
	Uma relação que identifica que um *mode* está acrescentando características que identificam um conceito, em um dado momento, durante certo período.
	Liga um *relator* e cada entidade que faz parte do evento que o *relator* representa.
	Uma relação que identifica que um *mode* está acrescentando características que identificam um conceito, em um dado momento, durante certo período.
	Representa relações formais, seja uma relação de comparação ou uma relação interna.
	Identifica que o relacionamento *material* é derivado de um *relator*, e é identificado por uma linha pontilhada com um círculo preto, ligando o *relator* e o relacionamento *material* derivado.

Fonte: Traduzido pela autora, com base em Guizzardi (2005a).

Na Tabela 4.4, apresentamos o segundo caso, em que encontramos as relações entre o todo e as suas partes (Todo-Parte): **componente-de** (*ComponentOf*), **subquantidade-de** (*SubQuantityOf*), **subcoleção-de** (*SubCollectionOf*) e **membro-de** (*MemberOf*).

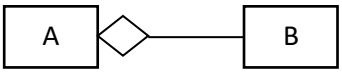
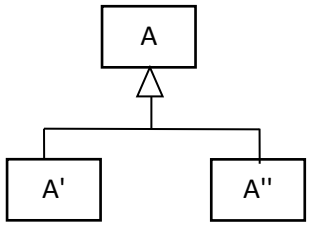
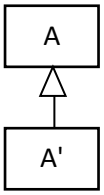
Tabela 4.4: Relações de Todo-Parte (*Meronymic Relations*)

RELAÇÕES DE TODO-PARTE	DESCRIÇÃO
	Representa um relacionamento de parte do todo entre conceitos complexos, como *kind* e *role.*
	Representa um relacionamento parte-todo entre um conceito complexo (como um *kind* ou *role*) ou um *collective* (como uma parte) e outro *collective* (como um todo).
	Representa um relacionamento parte-todo entre *collectives*, limitado à cardinalidade de, no máximo, um no fim da associação.
	Representa um relacionamento parte-todo entre quantidades, sempre não compartilhável e limitado à cardinalidade de, no máximo um no fim da associação

Fonte: Traduzido pela autora, com base em Guizzardi (2005a).

A Tabela 4.5 ilustra os relacionamentos que mostram as interação entre classes e classificadores que foram estendidos para o metamodelo revisado (ONTOUML).

Tabela 4.5: Relacionamentos em Diagramas de Classes (*Relationships in Class Diagrams*)

TIPOS DE RELACIONAMENTOS	DESCRIÇÃO
<<Association >>	**Associação simples** - Representa uma ligação entre duas Classes. Ainda podem expressar a cardinalidade (ou multiplicidade) e a navegação (sentido) da associação.
	Associação recursiva (ou auto-relacionamento) - Acontece quando uma Classe se conecta a ela mesma, e a associação tem alguma semântica no modelo.
	Associação Exclusiva - Quando algumas combinações de associações não são compatíveis no domínio do problema. É uma restrição entre duas ou mais associações.
	Associação de Classe - Uma classe pode ser associada a uma associação. Serve para adicionar informações extras à associação existente.
	**Agregação** - Este é um caso particular de associação. Indica que um elemento é parte ou está contido em outra classe. Representa uma relação do tipo parte/todo.
	Agregação de Composição - É um relacionamento em que um elemento está contido em outro, ou seja, a vida de um depende do outro, e os seus tempos de vida são os mesmos.
 Único pai com vários filhos	**Generalização ou Herança** - A generalização é um relacionamento entre uma classe mais geral e uma mais específica.
 Único pai com um único filho	

Fonte: Traduzido pela autora, com base em Guizzardi (2005a).

4.3.2 ANÁLISE ONTOLÓGICA

Segundo Marcondes (2015), considera-se que a análise ontológica das definições de informação possa trazer aportes interessantes para a compreensão de seus referentes e ajudar a elucidar a natureza última do fenômeno da informação. Dentro desse contexto, as definições de informação são analisadas para determinar a que entidades e relações os

termos ali encontrados se referem, tendo em vista que uma ontologia de fundamentação permite a explicitação dos compromissos ontológicos da ontologia que representa um domínio, agregando fidelidade, consistência e clareza na representação (GUIZZARDI, 2005a).

Na literatura, muitos esforços têm mostrado os benefícios da análise ontológica, como em Guizzardi (2009, 2010, 2011a) que inclui; entre eles:

(i) a definição rigorosa de modelos, em termos de semântica do mundo real;

(ii) a identificação de problemas na definição, interpretação ou uso de conceitos; e

(iii) recomendações de melhorias em modelo formalidade.

Faremos uso da análise ontológica por meio de metapropriedades, proposta por Guarino e Welty (2004). Tal análise concentra-se na natureza ontológica dos argumentos de uma relação de generalização ou especialização, no sentido de verificar se tal relação está ontologicamente correta. Essa verificação é feita por meio das metapropriedades de rigidez, identidade e dependência, as quais impõem diversas restrições na estrutura taxonômica da ontologia.

Para Scheidegger (2016), as metapropriedades podem ser classificadas como "**I**" e "**O**" para identidade, "**R**" para rigidez, "**D**" para dependência e "**U**" para unidade, acrescidos de sinais indicadores de presença (ou ausência) da metapropriedade.

+ (sinal positivo) presença da metapropriedade;

- (sinal negativo) ausência da metapropriedade;

~ (til) indicador "anti". Ex: anti-rígido.

- **Identidade (I, O)** - A análise ontológica por meio da metapropriedade identidade é focada em permitir identificar um indivíduo específico dentre os demais indivíduos. Isso é feito através da definição de quais características tornam um indivíduo único. Um conceito pode fornecer identidade (**+O**) ou apenas carregá-la (**+I**). Por exemplo, para PESSOA, podemos deduzir que a biometria de validação por verificação da íris e da retina é um critério de identidade que torna aquela instância de PESSOA única, diferenciando-a das demais instâncias de PESSOA. Há casos em que não é possível identificar uma única propriedade como critério de identidade e sim o conjunto de propriedades que são necessárias para identificar uma instância do indivíduo. Por exemplo, uma pessoa fornece identidade (+**O**), pois possui um conjunto de características que permitem a distinção com relação a outra

pessoa. Entretanto, um ESTUDANTE não fornece identidade, pois ela é fornecida pela PESSOA à qual se aplica a propriedade de estudante. Logo, Estudante carrega a identidade (+**I**), mas não a fornece.

Propriedades que, na verdade, são atributos não possuem identidade alguma, como a propriedade BONITA, e possuem o rótulo (-**I**). Da mesma forma que a rigidez, a identidade também restringe a taxonomia, impedindo que propriedades com o rótulo (+**I**) sejam declaradas como especializações de propriedades com o rótulo (-**I**); por exemplo, uma PESSOA não pode ser uma subclasse de BONITA. Ou seja, as noções de identidade e rigidez são utilizadas para definir as possibilidades de estruturação hierárquica entre objetos representados, garantindo que pertençam à hierárquia objetos que compartilhem os mesmos princípios de identidade. Assim, apenas HOMEM e MULHER são efetivamente subclasses de PESSOA, porque carregam identidade e são especializações reais do conceito PESSOA.

- **Rigidez (R) -** A análise ontológica por meio da metapropriedade rigidez pode assumir os valores Rígida (+**R**), Não-Rígida (ou semi-rígida) (-**R**) e Anti-Rígida (~**R**). Uma propriedade é classificada como rígida se ela é essencial para todos os indivíduos que a possuem, ou seja, caso algum indivíduo deixe de ter a propriedade, ele também passará a não existir mais. Por exemplo, a propriedade SER HOMEM poderia ser aplicada a um indivíduo do sexo masculino**;** uma vez que ele adquire a propriedade SER HOMEM, ele não pode deixar de tê-la sem deixar de existir; logo, é caracterizada como uma propriedade rígida. Entretanto, a propriedade COMER BANANA pode não ser compartilhada por todos de um grupo; logo, não é essencial e, portanto, não é rígida; no caso, é semi-rígida. Propriedades que são essenciais para alguns indivíduos, mas não para outros, são classificadas como semi-rígidas. Propriedades que não são essenciais para todos os seus indivíduos são classificadas como anti-rígidas. Uma restrição taxonômica imposta por essa propriedade é que propriedades rígidas não podem ser especializações (herança) de propriedades não rígidas. Por exemplo, a propriedade SER HOMEM não pode especializar a propriedade SER HOMEM DE CIRCO, pois esta última é semi-rigida, enquanto a primeira é rígida. O indivíduo que perder a propriedade SER HOMEM deixará de existir, enquanto o indivíduo que perder a propriedade SER HOMEM DE CIRCO permanecerá sendo HOMEM. Dessa forma, o correto é que as

propriedades não rígidas sejam especializações de propriedades rígidas, para evitar insconstências.

- **Dependência (D)** - A análise ontológica por meio da metapropriedade dependência rotula com (**+D**) as propriedades em que seus indivíduos dependam da exisência de indivíduos de outra propriedade para existirem. Dessa forma, pode-se dizer que uma propriedade X pode ser externamente dependente de uma propriedade Y. Caso essa dependênca não exista, a propriedade é rotulada com (**-D**). Por exemplo, a propriedade PAI só pode existir caso a propriedade FILHO exista, ou seja, uma é extremamente dependente da outra. Essa propriedade restringe a taxonomia, impedindo que propriedades com rótulo (**+D**) sejam especializações de propriedades com o rótulo (**-D**).
- **Unicidade (U)** - A análise ontológica por meio da metapropriedade unicidade busca identificar se o objeto pode ser reconhecido a partir de suas partes limites. Isso é feito através da análise da composição do objeto, identificando-se é unitário ou a soma de objetos unitários. O conjunto de condições necessárias para determinar se as propriedades possuem instâncias ou indivíduos unitários é chamado de critério de unicidade. As propriedades cujos indivíduos são unitários e possuem o mesmo critério de unicidade são rotuladas com (**+U**). Por exemplo, indivíduos de SER UMA FAIXA DE AREIA possuem rótulo (**+U**), pois é possível identificar os limites do objeto. Propriedades cujos indivíduos são unitáríos, porém possuem diferentes critérios de unicidade, são rotuladas como (**-U**), ou seja, não carregam unicidade. Por exemplo, a propriedade de SER QUADRÚPEDE, pode possuir diferentes critérios de unicidade entre seus inidivíduos, cada um podendo possuir um critério de unicidade distinto. Propriedades, cujos inidivíduos não são unitários são rolutados com (**~U**), ou seja, carregam anti-unicidade. Por exemplo, SER (UMA QUANTIDADE DE) AREIA possui indivíduos não unitários, pois não é possivel estabelecer limites bem definidos. Essa propriedade também restringe a taxonomia de tal forma que propriedades (**+U**) não podem especializar propriedade (**-U**).

4.3.3 FERRAMENTA OLED PARA MODELAGEM CONCEITUAL ONTOLÓGICA

O OntoUML Lightweight Editor (OLED) é um framework para o desenvolvimento de modelos OntoUML que provê inúmeras funcionalidades para auxiliar os modeladores na construção de modelos completos (representando todos os conceitos pertinentes ao domínio modelado) e corretos (alinhados e respeitando as regras de fundamentação de UFO). O

OLED é uma ferramenta multiplataforma, ou seja, é possível executá-lo nos sistemas operacionais mais comuns (Mac OS, Windows e Linux) e é distribuído como um executável java (.jar).

O OLED é baseado, principalmente, no editor TinyUML que agrega muitas contribuições diretas e indiretas para os membros do Grupo NEMO (Núcleo de Estudos em Modelagem Conceitual e Ontologias). Nele, também existe suporte para importar modelos de ontologias feitos no *Enterprise Architect* (EA) da *Sparx Systems*[21], em que se pode importar modelos concebidos com a EA no OLED e se beneficiar de todas as capacidades do editor (NEMO, 2017).

Na Figura 4.5 temos as fases do processo de desenvolvimento de ontologias do OLED conforme Guerson et al. (2015), em que os modeladores especificam suas ontologias de domínio em OntoUML conforme Guizzardi (2005a), restringindo-os usando a OCL-Linguagem para Especificação de Restrições em Objetos[22] (OMG, 2014). Nesse contexto, a ferramenta contém um conjunto funcionalidades integradas para acelerar a atividade de modelagem, verificação, validação e implementação de modelos OntoUML (representados em caixas cinza). Atualmente, não há suporte para a elicitação de requisitos (representado em caixa branca).

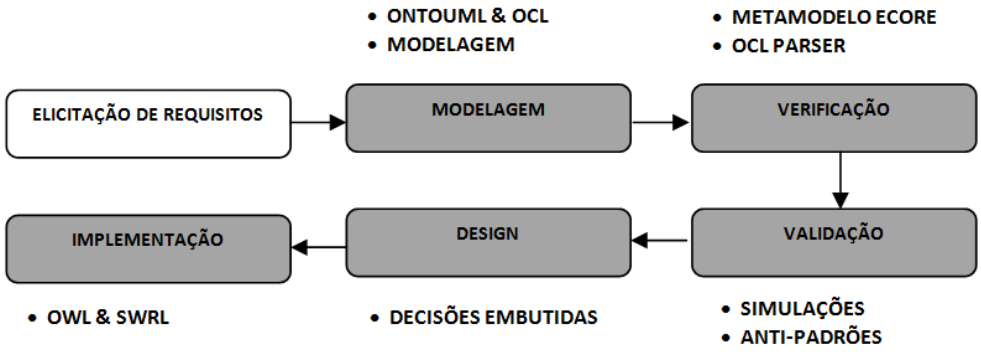


Figura 4.5: Fases do processo de desenvolvimento de ontologias do OLED
Fonte: Traduzido pela autora, com base no Artigo "OntoUML Lightweight Editor" em Guerson et al. (2015).

Ainda na Figura 4.5, a ferramenta fornece um conjunto de padrões de *design* integrados para acelerar a atividade de modelagem através da reutilização e, assim, melhorar a qualidade dos modelos construídos com OLED. Também fornece uma verificação sintática automática juntamente com dois recursos de validação (Simulação visual

[21] Sparx Systems - http://www.sparxsystems.com.au/
[22] OCL - http://www.omg.org/spec/OCL/

(BENEVIDES et al., 2014) e Anti-padrões[23] (SALES e GUIZZARDI, 2014)). Já, para aplicar o conhecimento formalizado no OntoUML em aplicativos da web semântica, o OLED apresenta um número de transformações automáticas predefinidas no *Web Ontology Language* **(OWL**), possivelmente aprimorada com regras de *Semantic Web Rule Language* (**SWRL**) (ALBUQUERQUE, 2014). No entanto, o OLED não oferece transformações em UML, nem em bancos de dados relacionais.

4.3.4 FERRAMENTA OLED ESTENDIDO

Esta dissertação propõe o uso do módulo OLED Estendido, que é uma extensão do *OntoUML lightweight editor* (OLED); optou-se por utilizá-lo como ferramenta para trabalhar os aspectos de utilização dos DROPs de colaboração, com o intuito de aproveitar suas ferramentas de modelagem através da reutilização, verificação, transformação e validação. Dessa forma, os modeladores (ou projetistas) terão não só uma coleção de padrões de domínio (DROPs) de CONTO, mas também um catálogo consolidado (organizado) e um processo de uso criado especificamente para orientar a escolha dos DROPs.

4.4 PADRÕES ONTOLÓGICOS

O termo **padrões ontológicos** (*ontology patterns* - POs) tem sua origem mais remota na área da Arquitetura, mais especificamente nos trabalhos do arquiteto Christopher Alexander e colegas, que, no final da década de 1970, propõem uma solução para resolver problemas recorrentes de projetos arquitetônicos. Christopher Alexander percebeu que alguns problemas eram recorrentes e tinham soluções idênticas que, se fossem descritas e padronizadas, poderiam poupar tempo e esforços (GAMMA et al., 1994).

No âmbito da SE, os padrões de projeto (*design patterns*) são também, como na Arquitetura, soluções para problemas recorrentes em contextos específicos, só que, no âmbito do desenvolvimento de uma aplicação de *software*, o seu objetivo é a melhoria da qualidade do *software* pelo uso de soluções reconhecidas como bem sucedidas, e poupar tempo e esforços. Nesse contexto, a aplicação de **padrões** (*patterns*) específicos para a construção e reúso de ontologias é relativamente recente, tendo sua origem na década de 2000, em trabalhos como os de Reich (2000), no âmbito da Biologia Molecular. Conforme

[23] Anti-pattern (ou anti-padrão) - apresentam uma solução comum que refaz o sistema de modo a maximizar os benefícios e minimizar as consequências (aspectos negativos).

apontam (FALBO et al., 2013b), o uso de padrões ontológicos se justifica tanto pela complexidade existente na construção de ontologias, quanto pela opinião dos que são experientes no assunto, sendo que, sob esse aspecto, o reúso de soluções modulares e adequadas se torna aconselhável.

No escopo da EO, o termo é cunhado como **padrão de projetos ontológicos** (*ontology design patterns*), à semelhança do termo em uso na área de SE. O conceito é definido como "soluções de modelagem para resolver problemas recorrentes no desenvolvimento de ontologias", para (Presutti et al., 2009), tendo sido constatado por experimentos como útil por (Falbo et al., 2013a) e, ainda, conforme (Falbo et al., 2016), apresentando uma solução bem experimentada para problemas recorrentes que surgem em contextos específicos de desenvolvimento.

Entretanto, Falbo et al. (2013a) propõem o uso do termo **padrões ontológicos** (*ontology patterns*), em vez de **padrões de projeto ontológicos** (*ontology design patterns*), de modo a diferenciar o seu uso na SE, em que os aspectos tecnológicos podem estar envolvidos. Embora caiba ressaltar que, na literatura, os dois termos sejam usados para denotar a mesma coisa. Para Gangemi e Presutti (2009), existem diversas categorias de padrões ontológicos. Assim, para os propósitos do presente projeto de pesquisa perpassa pelos **Padrões Ontológicos de conteúdo** (COPs - *Content Ontology Patterns*).

Ainda segundo os autores, os COPs se dividem em dois tipos: **Padrões de Ontológicos Fundacionais** (FOPs - *Foundational Ontology Patterns*) e **Padrões de Ontológicos Relacionados ao Domínio** (DROPs - *Domain-Related Ontology Patterns*) (FALBO et al., 2013b). A Figura 4.6 ilustra os vários tipos de padrões ontológicos (POs) e o foco deste trabalho de pesquisa (representado em caixa cinza).

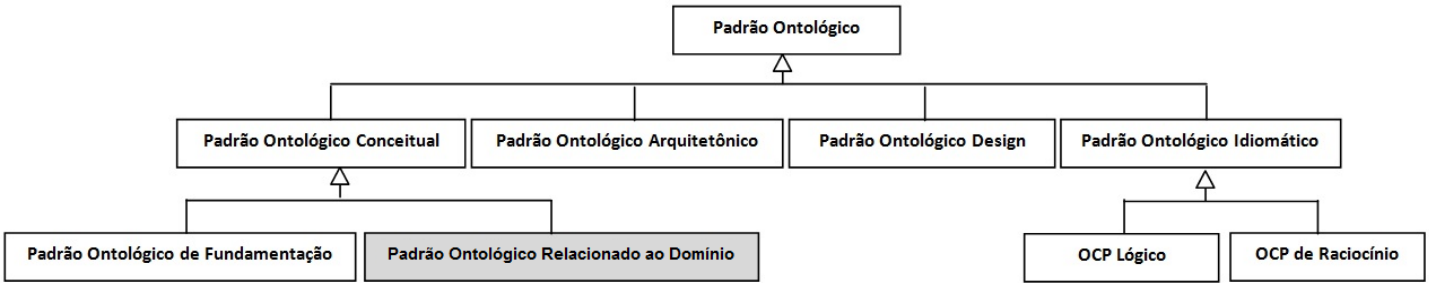


Figura 4.6: Tipos de padrões ontológicos
Fonte: Traduzido pela autora, com base em Falbo et al. (2013b).

Os padrões ontológicos de conteúdo podem ser considerados como pequenas ontologias, com documentação explícita dos princípios adotados para a sua elaboração, e devem refletir boas práticas de modelagem, sendo usados como componentes modulares de modelagem por estarem associados a um documento que descreve o problema, o qual o padrão ontológico visa solucionar (BLOMQVIST; GANGEMI; PRESUTTI, 2009). Existem padrões ontológicos de conteúdo que são independentes do domínio (FOPs), sendo derivados de uma ontologia de fundamentação, sendo mais genéricos, enquanto outros são dependentes do domínio (DROPs) e são mais específicos e voltados para a resolução de problemas desse domínio (FALBO et al., 2013a).

Segundo Ruy et al. (2015a), os padrões são dependentes do domínio que expressam suas noções centrais, sendo fragmentos nucleares para a construção de ontologias desse domínio. Para os autores, alguns casos são compostos por poucos conceitos (pelo menos dois), mas, às vezes, podem ser mais complexos e podem dar origem a dois ou mais padrões de domínio que expressam pontos de vista alternativos (ou mais detalhados), podendo, ainda, haver padrões ontológicos de domínio que são parcialmente sobrepostos.

A Figura 4.7 apresenta a arquitetura dos níveis de generalidade das camadas ontológicas dos padrões ontológicos associados aos FOPs e DROPs.

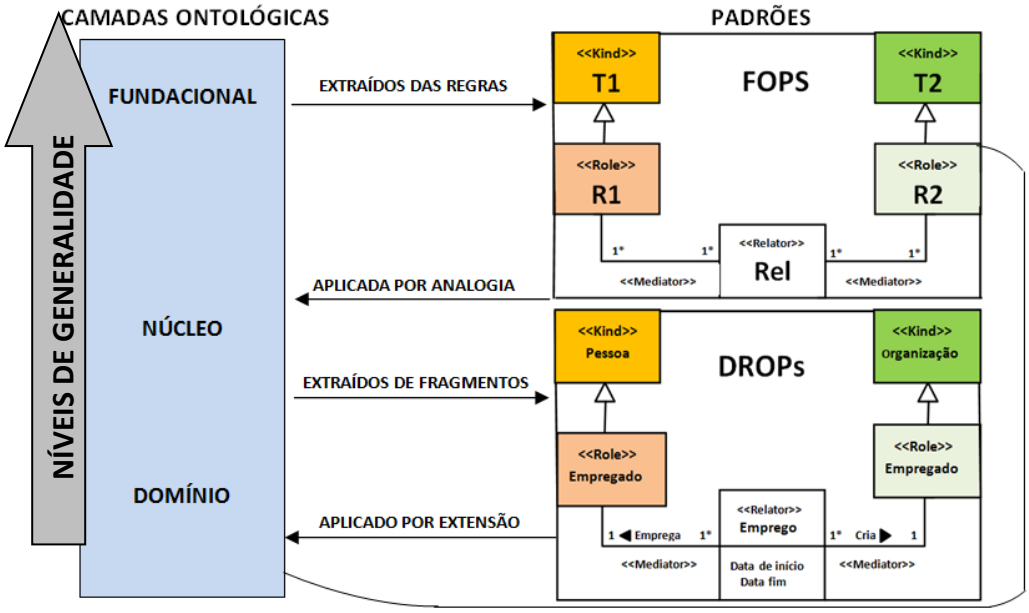


Figura 4.7: Níveis de generalidade e padrões ontológicos
Fonte: Adaptado pela autora, com base em Ruy et al. (2015a).

Ainda para os autores, os FOPs são extraídos das bases e regras de uma ontologia de fundamentação. Logo, FOPs são fragmentos reusáveis extraídos das regras das ontologias de fundamentação ou núcleo, uma vez que as ontologias de fundamentação definem conceitos genéricos e independentes de domínio. Assim, os FOPs possuem a capacidade de poderem ser usados em quaisquer domínios. Por outro lado, os DROPs capturam o conhecimento central do domínio alvo e podem ser extraídos diretamente dos fragmentos de ontologia núcleo ou domínio.

Vale ressaltar que, quando um DROPs é extraído de uma ontologia de núcleo ou de domínio já modelado reutilizando FOPs, temos como resultado um DROP mais rico, com conhecimento estrutural de domínio, caracterizando um aplicativo de cadeia em cadeia no nível de domínio. Para Mattsson e Bosch (1998), a reutilização de DROPs ocorre por extensão, ou seja, seus conceitos e relações se tornam parte da ontologia do domínio e geralmente são estendidos, dando origem a novos conceitos e relações mais específicas.

Ambos DROPs e FOPs podem ser aplicados para desenvolver ontologias de domínio. Se não houver um DROP que satisfaça os requisitos do domínio, pode-se criar um novo fragmento de ontologia (ou aplicar um FOP) e/ou reutilizar sua estrutura. Neste caso, o modelador deve, primeiro, tentar encontrar um DROP aplicável, que capte o conhecimento do núcleo e, possivelmente, uma estrutura fundamental, adequada para o problema de modelagem em questão, haja vista que as ontologias para o domínio da modelagem não se limitam à aplicação direta de padrões. Os fragmentos de ontologias de domínio criados a partir das COPs estão inter-relacionados e precisam ser reunidos.

Dessa forma, o modelador pode procurar DROPs relacionados e também usar FOPs para combinar a estrutura inerente aos diferentes fragmentos. Como citado acima, tanto os DROPs como os FOPs podem ser aplicados na fase de desenvolvimento da modelagem conceitual. Cabe ressaltar que o conjunto formado por eles, juntamente com suas especificações, formam uma modelagem especifica da aplicação *groupware*. Assim, chega-se ao método ÔMEGA e seu respectivo modelo de processo de uso ALFA-OPL(AP-OPL), baseado em padrões ontológicos de colaboração para auxiliar na construção de modelos conceituais de *groupware*.

No contexto deste trabalho, o foco será na construção de modelos conceituais especificos de *groupware* baseados no reúso de padrões ontológicos de colaboração

(DROPs). Vale destacar que, inicialmente, manusearemos, como aporte conceitual os artefatos de colaboração: “**Catálogo de DROPs de Colaboração**”, a OPL que representa o “**Modelo de Processo da C-OPL**” e as "**Ontologias de Colaboração** (três subontologias de CONTO)”. Posteriormente, manusearemos os novos artefatos de colaboração (gerados): “**Catálogo de DROPs de Colaboração Consolidado**", a OPL que representa o “**Modelo de Processo de uso ALFA-OPL (AP-OPL)**” e as "**Ontologias de Colaboração** (três subontologias de CONTO)”.

Como ferramenta, usaremos o Editor OLED, em específico, o módulo OLED Estendido, com capacidade de gerenciamento de catálogo de DROPs de colaboração. Exemplos de DROPs de colaboração podem ser encontrados no Capítulo 4 (seção 4.7.1, 4.7.2 e 4.7.3) desta dissertação.

4.5 LINGUAGEM DE PADRÕES ONTOLÓGICOS

No âmbito de SE, uma Linguagem de Padrões Ontológicos (*Ontology Pattern Language* - OPL) é uma rede de padrões inter-relacionados que define um processo para resolução sistemática de problemas relacionados ao desenvolvimento de *software* (DEUTSCH, 2004). Assim, uma Linguagem de padrões (LP) organiza padrões ontológicos em uma rede, de forma que possam ser usados como blocos de construção que podem ser combinados para tratar os problemas relevantes no desenvolvimento de uma ontologia para o domínio da OPL.

Além da rede de padrões, uma OPL provê um processo que guia a ordem de aplicação dos padrões e sugere quais padrões utilizar, de acordo com os problemas a serem tratados. Esse processo tem como objetivo fornecer suporte global para a utilização dos padrões, de modo a resolver problemas relacionados a um domínio técnico ou de alguma aplicação específica. Essa visão holística deve fornecer orientação explícita sobre os problemas que podem surgir no domínio, informar os possíveis meios de resolvê-los e sugerir um ou mais padrões ontológicos para resolver cada problema específico (FALBO et al., 2013b).

As linguagens de padrões refletem o fato de que os padrões tendem a ser fortemente acoplados, e é difícil, ou até mesmo impossível, utilizá-los de forma isolada (FALBO et al., 2013b). Por esse ângulo, para que uma LP seja eficaz no seu objetivo de guiar o usuário na aplicação dos padrões e trate de maneira adequada a aplicação combinada de vários padrões, é necessário que as relações entre os padrões sejam definidas e explicitadas na LP.

Afinal, uma OPL dá orientações concretas e bem fundamentadas para o desenvolvimento de modelagem em um determinado domínio, abordando, no mínimo, as seguintes questões:

- Quais são os principais problemas a resolver no domínio de interesse?
- Em que ordem esses problemas devem ser tratados?
- Que alternativas existem para resolver um determinado problema?
- Como as dependências entre os problemas devem ser tratadas?
- Como resolver cada problema individualmente de forma mais eficaz na presença dos problemas que o cercam?

O Grupo NEMO, por meio de seu trabalho de pesquisa, defende explicitamente a necessidade de uma abordagem multidisciplinar para o desenvolvimento de Ontologias de Fundamentação para dar suporte à modelagem conceitual (GUIZZARDI et al., 2009). No Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO) têm sido desenvolvidas OPLs para aplicações em domínios diversos. Até o momento, foram desenvolvidas as seguintes OPLs (ou LPOs):

- **SP-OPL** - *Software Process Ontology Pattern Language,* em Falbo et al. (2013a);
- **E- OPL** - *Enterprise Ontology Pattern Languag,* em Falbo et al. (2014);
- **M-OPL** - *Measurement Ontology Pattern Language,* em Barcellos et al. (2014);
- **ML-OPL** - *Ontology Pattern Laguage Modeling Language,* em Quirino et al. (2017);
- **ISP-OPL** - *ISO-based Software Process Ontology Pattern Languag,* em Ruy et al. (2015b);
- **S-OPL** - *Service Ontology Pattern Language*, em Falbo et al. (2016);
- **C-OPL** - *Collaboration Ontology Pattern Language,* em Souza (2017).

As OPLs citadas acima foram representadas visualmente por seus autores por meio de diagramas de atividades da UML adaptados. Segundo FALBO et al. (2013a), o diagrama de atividades da UML é a notação UML padrão para representar restrições de sequenciamento temporais entre tipos de atividade e, portanto, para especificar a possível ordem de execução entre as atividades. Ainda que tenham sido utilizadas adaptações do diagrama de atividades da UML para representar as OPLs desenvolvidas, diferentes adaptações foram feitas para tratar cada OPL (ou LPO).

4.6 CONTO: ONTOLOGIA DE COLABORAÇÃO

Em conformidade com a estrutura sugerida pelo Modelo 3C de Colaboração, a **Ontologia de Colaboração** (*Collaboration Ontology* – CONTO) foi subdivida em três subontologias, a saber: Cooperação, Comunicação e Coordenação. Assim sendo, o Modelo 3C é útil para guiar o desenvolvimento de *groupware* (*software* para dar suporte ao trabalho em grupo), e cada C é profundamente analisado (PIMENTEl, 2006). Tal divisão em CONTO visou facilitar o processo de desenvolvimento, permitindo um melhor entendimento de cada parte da colaboração e dos aspectos específicos de cada um dos Cs.

Segundo SOUZA (2017), as três subontologias de CONTO seguem a estrutura do modelo 3C: cooperação, comunicação e coordenação. Essas três ontologias representam o conhecimento capturado do domínio específico de colaboração com os respectivos conceitos, relacionamentos, propriedades de domínio e um conjunto de axiomas que restringem o domínio colaboração.

A Figura 4.8 ilustra essas ontologias a partir de um diagrama de pacote UML, modelando as três ontologias e as relações entre elas (OLIVEIRA, 2009; SOUZA, 2016).

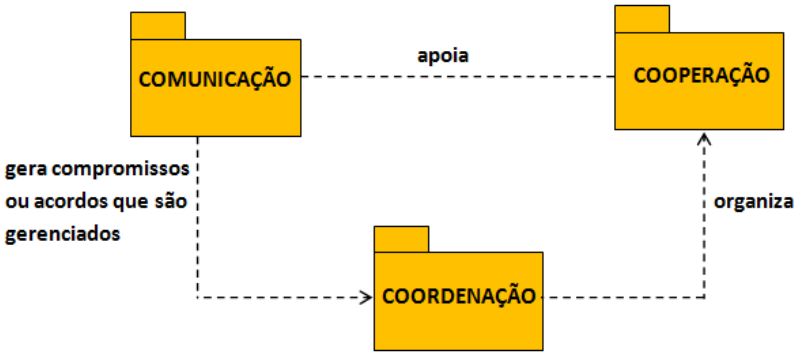


Figura 4.8: Três subontologias CONTO e suas relações
Fonte: Baseada em Oliveira (2009); Souza (2016).

A Ontologia de Comunicação promove o apoio necessário para a realização da cooperação; portanto, ela apoia as interações entre agentes. Além disso, a comunicação gera compromissos ou acordos que são gerenciados pela coordenação durante o processo de comunicação e cooperação. Por fim, a coordenação organiza e harmoniza os processos de cooperação.

Em complemento aos modelos conceituais a seguir apresentados, foram necessárias a definição e formalização de um conjunto de axiomas. Portanto, em sequência, são

apresentadas as três subontologias que formam a Ontologia de Colaboração, representadas pelos modelos conceituais, axiomas e as suas respectivas descrições.

4.6.1 SUBONTOLOGIA DE COOPERAÇÃO

A Subontologia de Cooperação tem por objetivo representar os elementos que são relacionados ao trabalho conjunto entre indivíduos, ou melhor, o trabalho cooperativo entre eles, interagindo para a concretização de uma tarefa. Fazem parte da cooperação o ambiente onde a cooperação acontece e as ações que esses indivíduos realizam, dentre outros. O problema com que esta subontologia lida está relacionado, portanto, ao contexto em que a cooperação acontece e às relações que as entidades têm nesse contexto.

A Figura 4.9 exibe a Subontologia de Cooperação; conceitos em cinza são provenientes das ontologias de fundamentação **UFO** (**A**), (**B**) e (**C**), previamente apresentadas (na seção 4.3), e, destacados em branco, estão conceitos da Ontologia de Cooperação.

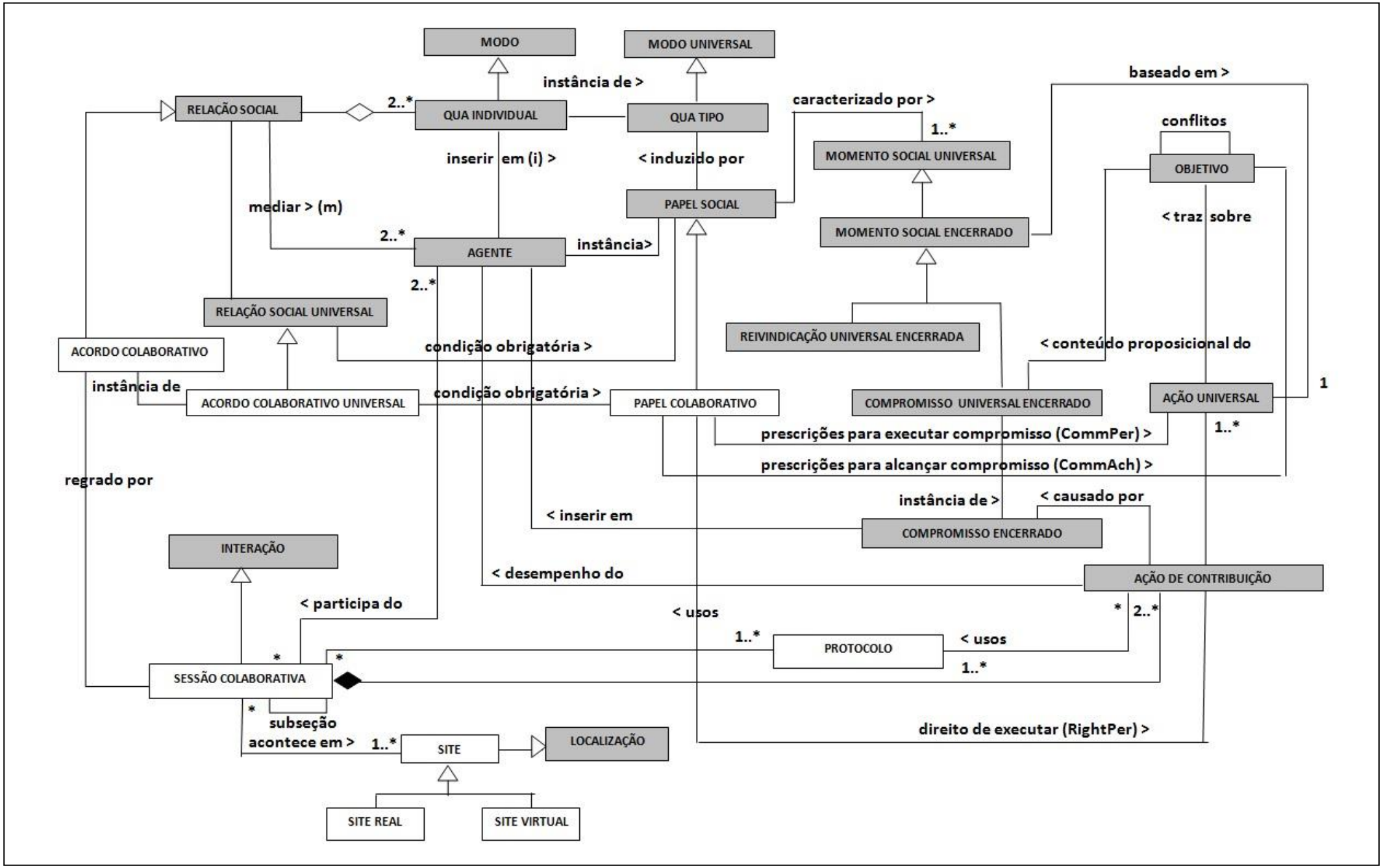


Figura 4.9: Subontologia de Cooperação
Fonte: Traduzido pela autora, com base em Oliveira (2009).

Na Figura 4.9, os **Agentes** podem **participar** (*participates of*) de uma **Sessão Colaborativa** (*Collaborative Session*) (SC). Uma SC deve ter pelo menos dois participantes, por definição. As **Sessões Colaborativas** são compostas por **Participação de Recurso** (*Resource Participation*) ou por **Ações de Contribuição** (*Action Contribution*). Uma **Participação de Recurso** é a participação de um recurso não intencional, isto é, ele participa de forma inanimada (sendo utilizado, criado, terminado ou modificado) (GUIZZARDI et al., 2008c).

Em contrapartida, uma **Ação de Contribuição** é aquela executada por p**erformance** de (*performance of*) um **Agente** (*Agent*) que tem a intenção de contribuir. As Participações de uma SC e ela própria são organizadas por meio da coordenação, que é representada no modelo pelo conceito **Protocolo** (*Protocol*), o qual é detalhado posteriormente na Subontologia de Coordenação. A **Sessão Colaborativa** é uma **Interação** (*Interaction*) entre Agentes **deliberada por** (*ruled by*) **Acordos Colaborativos,** os quais implicam não possuir **Objetivos** (*Goal*) conflitantes, como, por exemplo, em uma partida de basquetebol, em que os jogadores do mesmo time objetivam a vitória, ou em uma reunião de projeto, na qual seus participantes objetivam a resolução de determinados objetivos compartilhados. Nesses exemplos existe um acordo colaborativo que rege essas colaborações.

A natureza desses acordos colaborativos pode derivar de objetivos compartilhados ou das denominadas **trocas sociais** (*social exchange*) (HOMANS, 1958), nas quais um participante adota o objetivo do outro e vice-versa, mediando um acordo. A **Sessão Colaborativa** aqui definida é o mesmo tipo de interação definido por Neumann e Morgenstern (1944), e ocorre quando os objetivos inerentes aos participantes da sessão não conflitam, como consequência, o resultado dessa colaboração é uma soma não zero, ou melhor, os resultados da SC são positivos (ou negativos), não existindo, portanto, uma competição, como os definidos em jogos de soma zero.

Durante uma SC, **Agentes** instanciam **Papéis Colaborativos** *(Collaborative Roles)* que são **caracterizados por** (*characterized by)* um Tipo de **Compromisso Universal Encerrado** (*Closed Commitment Universal*) que tem como conteúdo **proposicional** (*propositional content of*) um ***Objetivo*** (*Goal*). Tal objetivo é satisfeito quando são criadas instâncias de tipos de **Ações Universais** (*Action Universals*), ou seja, essa **ação leva** (*brings about*) a uma situação em que esse objetivo é satisfeito (GUIZZARDI et al., 2008c). Dessa forma, os **Papéis**

Colaborativos, quando instanciados por **Agentes**, os levam a realizar ações que são causadas pelo **Compromisso Encerrado** de instanciar esse papel. Voltando ao exemplo da partida de basquetebol, o papel colaborativo assumido por um jogador de jogar como armador esquerdo é cumprido quando ele desempenha o papel durante a partida com ações de um armador. Lembrando que, quando esse jogador assumiu esse papel, ele concordou em aceitar os objetivos inerentes ao papel, objetivos que são satisfeitos quando ele executa ações específicas desse papel.

A partir das definições acima derivam-se algumas relações. A primeira é denominada **prescreve compromisso para alcançar** (*prescribes commitment to achieve* (CommAch)) e ocorre entre **Papéis Colaborativos** e **Objetivos.** Logo, se **CommAch** (***r, g***) então existe um compromisso universal encerrado **c** (*closed commitment universal* ***c***) a **ser realizado** (*fulfilled*) que tem como conteúdo ***g*,** caracterizando então o papel colaborativo ***r***. Isto reforça que um papel colaborativo ***r*** tem um compromisso de realizar um objetivo que foi acordado através de um compromisso assumido, como definido pelo axioma **A1**. A segunda relação é derivada entre **Papéis Colaborativos** e **Tipos de Ações** (*Action Universal*) e é denominada **prescreve compromisso para executar** (*prescribes commitment to perform* (CommPer)). Conforme definido pelo axioma **A2**, se **CommPer** (***r, a***), então existe um **Compromisso Universal Encerrado c** (*closed commitment universal* ***c***), tal que **c** caracteriza o papel colaborativo ***r*** e ***c*** é baseado em um **ação universal** (*action universal*), ou seja, para satisfazer esse compromisso, o papel tem que executar um tipo de ação específica ***a***. Análogo ao **CommPer**, definido no axioma **A3**, o papel tem o direito de executar uma determinada ação, ou seja, assim como ele faz um **comprometimento** (*Commits*) de executar, em contrapartida ele ganha o **direito de executar** (RigthPer).

A1 - *commAch (r, g) ≡ ∃c compromisso universal encerrado (c)* ∧ caracterizado_por *(r, c)* ∧ *conteúdo_propositivo_do (g, c))*

A2 - *commPer (r, a) ≡ ∃c compromisso Universal Encerrado (c)* ∧ caracterizado_por *(r, c)* ∧ baseado_em *(c, a)*

A3 - *rigthPer (r, a) ≡ ∃c reclamação universal encerrado (c)* ∧ caracterizado_por *(r, c)* ∧ baseado_em *(c, a)*

Como mencionado, uma SC é **deliberada** (*ruled by*) por **Acordos Colaborativos** (*Collaborative Agreements*) que são acordados pelos participantes desta SC. Esses **Acordos Colaborativos** são tipos de **Relações Sociais** (*Relator Social*) e instanciam **Tipos de Acordos Colaborativos** (*Collaborative Agreement Universal*) que estão atrelados aos **Papéis Colaborativos**. Em outras palavras, a relação de um **Acordo Colaborativo Universal** (*Collaborative Agreement Universal*) e um **Papel Colaborativo** é uma **condição de ligação** (*binding condition of*) que o primeiro impõe sobre o segundo (ver axioma **A4**). Dessa forma, através da comunicação, **Acordos Colaborativos** podem ser firmados entre **Agentes** de tal forma que sejam refletidos nos **Papéis Colaborativos** que esses agentes se comprometem a desempenhar em uma SC. Um exemplo disso é quando um jogador acorda com o seu time que irá desempenhar o papel de armador esquerdo.

Conforme definido pelo axioma **A5** e acima descrito, se **regrado por (***RuledBy* (*s,c*)) então existe um ***c***, tal que ***c*** é instância de **acordo de colaboração universal** (*collaborative agreement universal* (**CAU**)) e para toda ação ***a*** que faz parte de ***s***, então existe uma condição de ligação entre o papel colaborativo ***r*** orientada pelo **CAU,** tal que ***r*** tem o compromisso de executar ***a***. Isto exemplifica que as ações realizadas em uma sessão colaborativa são orientadas por acordos colaborativos através dos papéis colaborativos instanciados pelos participantes da sessão. Já no axioma **A6,** tem-se que, dado uma **CAU *ca***, para todo **objetivo** (*goal*) ***g,g'*** existe um papel colaborativo ***r*** tal que ***r*** é orientado pelo ***ca*** e, se ***r*** prescreve um comprometimento de atingir ***g*** *e* ***g'***, então ***g*** não conflita com ***g'***. Em outras palavras, considerando uma **CAU** que orienta um papel colaborativo, tem-se que esse papel não pode prescrever um comprometimento de atingir um objetivo que conflita com um objetivo que ele prescreve.

A4 - p*apel colaborativa (r)* ≡ $\forall$ *u condição_obrigatória_de(u,r)* $\rightarrow$ *acordo colaborativo universal (u)*

A5 - *regrado_por (cs,ca condição_obrigatória_de)* ≡ *sessão colaborativa (cs)* $\wedge$ *acordo colaborativo (ca)* $\wedge$ $\exists U_{ca}$ *instância_de (ca, Uca)* $\wedge$ ($\forall \alpha$ *parte_de (a,cs)* $\rightarrow$ $\exists r$ *condição_obrigatória_de* (U_{ca}, *r)* $\wedge$ (*CommPer(r,a))*

A6 - *acordo colaborativo universal (u)* ≡ $\forall g,g'$ $\exists r,r'$ *condição_obrigatória_de* (u, r) $\wedge$ *comAch* (r, g) $\wedge$ *condição_obrigatória_de*(u, r') $\wedge$ *comAch (r', g')* $\rightarrow$ $\neg$ *conflito (g, g')*

Um **Ambiente** (*Site*) representa onde uma SC acontece (*happens in*), podendo ser esse um **Ambiente Virtual** (*Virtual Site*) ou **Real** (*Real Site*). Como pode ser observado no modelo, um ambiente é um tipo de **Localização** (*Location*), ou seja, um espaço conceitual cujos valores (**qualia**) podem ser usados como valores de propriedades de localização espacial de **entidades concretas** (*endurants* e *perdurants*). Um Ambiente é dito virtual se estiver sendo simulado por uma ferramenta de *software*, perceba que um ambiente virtual também existe no "mundo real". Em compensação, um **Ambiente Real** é aquele que existe no mundo real, existindo em um espaço físico concreto. Exemplos de **Ambientes Virtuais** são: Fóruns, Blogs, Gerenciadores de Documentos, etc. E exemplos de **Ambientes Reais** são: a sala onde acontece a reunião, a quadra do estádio onde acontece a partida de basquetebol, dentre outros. A Tabela 4.6 apresenta um dicionário de dados da Subontologia de Cooperação, com uma entrada para cada conceito.

Tabela 4.6: Dicionário de Termos da Subontologia de Cooperação

TERMOS	DESCRIÇÃO
Sessão Colaborativa	Eventos complexos nos quais participantes interagem através de participações (ou contribuições) com o propósito de colaborar.
Site	Local onde a sessão colaborativa acontece.
Site Virtual	Ambiente que é simulado por uma ferramenta de software, onde uma sessão colaborativa acontece.
Site Real	Ambiente concreto do mundo real onde a sessão colaborativa acontece.
Acordo Colaborativo	Representa um tipo de papel social que delibera uma sessão colaborativa.
Acordo Colaborativo Universal	Tipo de relação social universal que pode ser instanciado por um acordo colaborativo e fornece a condição de ligação de um papel colaborativo.
Papel Colaborativo	Papel social que agentes se comprometem a desempenhar devido aos acordos por eles assumidos.

Fonte: Traduzido pela autora, com base em Oliveira, 2009; Souza (2016).

A conceituação discutida nesta seção pode ser resumida da seguinte forma: agentes participam de uma sessão colaborativa desempenhando papéis colaborativos. Esses papéis são assumidos por agentes que são mediados por um acordo colaborativo. Os Papéis Colaborativos são caracterizados por comprometimentos encerrados, ou seja, compromissos de atingir certos objetivos realizando ações específicas. Ao assumir um papel colaborativo, um agente adota os objetivos prescritos para aquele papel. Uma sessão colaborativa é

materializada por participações (ou contribuições) que são resultados dos compromissos encerrados assumidos por esses papéis. Os Papéis que são ligados pelo mesmo acordo colaborativo não podem prescrever objetivos conflitantes. Esta condição diferencia um acordo colaborativo de um compromisso social genérico.

4.6.2 SUBONTOLOGIA DE COMUNICAÇÃO

A Subontologia de Comunicação tem o objetivo de representar os elementos inerentes ao domínio de comunicação. Logo, são abordados nessa Subontologia elementos que representam a comunicação e as demais entidades envolvidas durante o processo. Na Figura 4.10, são destacados conceitos e relações presentes na Subontologia de Comunicação. Nela é feita uma distinção entre **Contribuições Materiais** (*Material Contribution*) e **Atos Comunicativos** (*Communicative Acts*) e **Percepção** (*Perception*), que são tipos de **Ações de Contribuição**. As primeira são contribuições que modificam um estado de mundo de forma material (física). Essas contribuições não carregam **Mensagens** (*Messages*). Exemplos de contribuições materiais são: em um jogo de basquetebol, o ato de passar o bastão para o seu companheiro, ajudar a mover uma cadeira. Já os **Atos Comunicativos** e as **Percepções** estão associados a uma **Mensagem,** essa mensagem carrega o conteúdo que se deseja transmitir (SEARLE, 1983). Melhor dizendo, a mensagem é o conteúdo proposicional de um ato comunicativo. Essas **Mensagens** podem ser expressas de **forma codificada** (*coded in*) em uma **Linguagem Idiomática**.

Particularmente, os **Atos Comunicativos** são tipos de **atos ilocucionários** (*illocutionary acts*) (SEARLE, 1983). Exemplos desses atos são informar, requisitar, prometer, ordenar, dentre outros. Os **Atos Comunicativos** carregam as informações que são trocadas durante uma SC. Tais informações são trocadas por ações **executadas por** (*performance of*) por **Agentes**. Os **Atos Comunicativos** e as **Percepções** são partes essenciais de uma **Interação Comunicativa** (*Communicative Interaction*).

Diante do exposto, verifica-se que a **Interação Comunicativa** é essencialmente composta por pelo menos um **Ato Comunicativo** (*Communicative Act*) e uma **Percepção** (*Perception*). Em outras palavras, para que uma **Interação Comunicativa** possa existir, é necessário a execução de um **Ato Comunicativo** e uma **Percepção** deste ato (*Perception*) como em (SHANNON, 1949), (SCHRAMM, 1964), (BERLO, 1972), (MELO, 1973), (KOTTER, 1977) e (PENTEADO, 1993). Essa essencialidade é ilustrada pelo seguinte exemplo: quando

um escritor (*Sender*) publica um livro, ele realiza um **Ato Comunicativo** (publicação de informação); posteriormente, esse livro é lido por outro indivíduo (*Receptor* ou *Receiver*); esse ato de leitura é uma **Percepção** (*Perception*) dessa informação, fechando a cadeia do processo de comunicação e, assim, ocasionando uma **Interação Comunicativa**

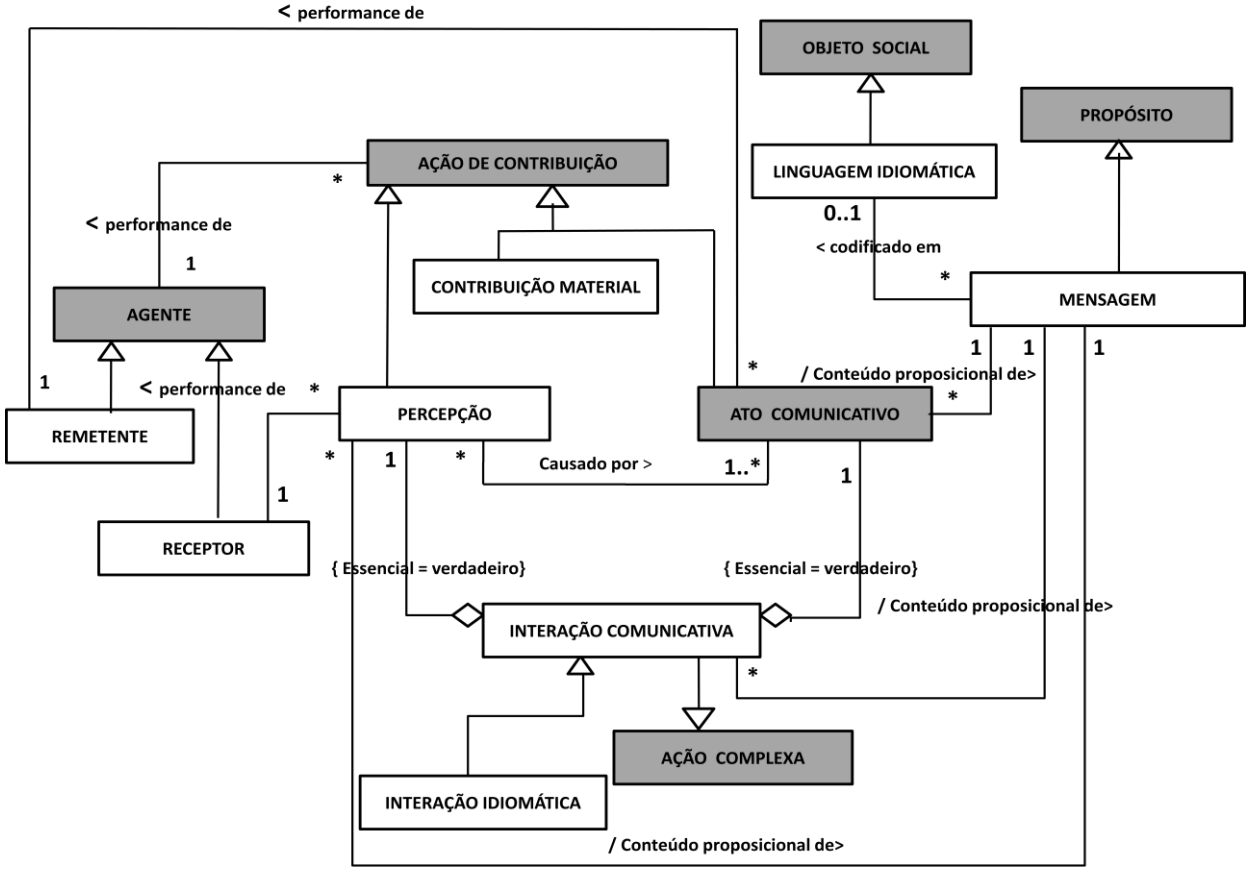


Figura 4.10: Subontologia de Comunicação
Fonte: Traduzido pela autora, com base em Oliveira (2009).

Uma **Interação Comunicativa** tem como conteúdo proposicional uma **Mensagem** que é a mesma atrelada ao Ato Comunicativo essencial para essa interação; portanto, essa relação deriva da relação entre mensagem e o ato. Por outro lado, uma **Percepção,** devido ao fato de ser parte de uma **Interação Comunicativa,** pode conceber um conteúdo proposicional. Assim sendo, uma **Percepção** pode ter como conteúdo proposicional uma **Mensagem**.

As **Interações Comunicativas** são **Ações Complexas** (*Complex Actions*) que podem ainda ter como subtipo uma *Idiomatic Interation***,** na qual representam interações em que há a utilização de uma **Linguagem Idiomática** (*Idiomatic Language*) para o intercâmbio de

conhecimentos tácitos, ideias e informações (BELTRÃO, 1977). Exemplos desse tipo de comunicação são as trocas de informação realizadas pela sociedade humana, por meio de diálogos, jornais, revistas, televisão, livros, etc. As **Interações Comunicativas** podem ser utilizadas de uma forma mais geral, que acontece quando a mensagem não está expressa em uma **Linguagem Idiomática**. Tal comunicação, por exemplo, pode ser utilizada por meio da linguagem corporal (gestos, olhares e postura). A Tabela 4.7 apresenta um dicionário de termos da Ontologia de Comunicação, descrevendo seus principais conceitos:

Tabela 4.7: Dicionário de Termos da Subontologia de Comunicação

TERMOS	DESCRIÇÃO
Interação Comunicativa	Representa a comunicação entre dois ou mais agentes através de troca de informações: envio e percepção.
Contribuição Material	Ação de Contribuição em que não ocorre troca de mensagens.
Mensagem	A mensagem é o conteúdo proposicional trocado de um ato de comunicação e, portanto, de uma interação comunicativa entre agentes.
Ato Comunicativo	Representa a ato de enviar uma mensagem.
Percepção	Representa a ação de perceber uma mensagem.
Linguagem Idiomática	Linguagem que usa um idioma para a sua representação.
Remetente	Individuo capaz de emitir uma mensagem.
Receptor	Individuo capaz de perceber uma mensagem.
Interação Idiomática	Interação Comunicativa que envolve uma mensagem codificada em uma linguagem idiomática.

Fonte: Traduzido pela autora, com base em Oliveira (2009); Souza (2016).

Na Subontologia de Comunicação, para se realizar uma **Interação Comunicativa,** a **Percepção** deve se dar somente após o **Ato Comunicativo**. Portanto, como representado pelo axioma **A7**, para toda interação comunicativa ***c***, ato comunicativo a parte de ***c*** e percepção ***p*** parte de ***c***, então ***a*** tem que ser realizado antes de **p**, ***a*** precede ***p*** no tempo, e a mensagem presente em ambas as ações é a mesma. Já o axioma **A8** representa a necessidade existencial de uma **Interação Comunicativa** em ter duas **Ações de Contribuição** (*Communicative Act e uma Perception*). A idéia é que uma interação comunicativa tenha um ato comunicativo ***c*** executado por um agente emissor ***a*** e uma percepção ***c'*** executada por um agente ***a'***, então uma interação de comunicação é formada pela composição de ***c*** e ***c'***.

A7 - $\forall$ *x,y,z interação comunicativa(x)* $\wedge$ *ato comunicativo (y)* $\wedge$ percepção *(z)* $\wedge$ parte_de *(y, x)* $\wedge$ *parte_de (z, x)* $\rightarrow$ *(precede (y, z)* $\wedge$ $\forall$*m1,m2 contéudo_proposicional_de (m1, y)* $\wedge$ *contéudo_proposicional_de (m2, z)* $\rightarrow$ *m1 = m2)*

A8 - $\forall x,y,z$ *interação comunicativa (x)* $\wedge$ *ato Comunicativo (y)* $\wedge$ percepção *z)* $\wedge$ *parte_de (y, x)* $\wedge$ *parte_de (z, x)* $\rightarrow \exists$ *s,r Remetente(s)* $\wedge$ *performance_de(y, s)* $\wedge$ receptor *(r)* $\wedge$ *performance_de (z, r)* $\wedge$ *(s r)*

Por fim, os **Agentes** participam de sessões colaborativas através de atos intencionais chamados de **contribuições de ação** (*Action Contribution*) que podem ser do tipo material ou comunicativa. Atos comunicativos possuem (além de uma força ilocucionária) um conteúdo proposicional, ou mensagem, que pode ou não ser codificada em uma linguagem idiomática (um objeto social compartilhado por uma comunidade de agentes). Quando uma interação comunicativa do tipo **ação complexa** ocorre, tem-se que a mesma é composta de um ato comunicativo e uma percepção desse ato comunicativo por agentes diferentes. Quando uma interação comunicativa ocorre, o conteúdo proposicional do ato comunicativo efetuado por um dos agentes é o mesmo percebido pelo outro.

4.6.3 SUBONTOLOGIA DE COORDENAÇÃO

A Subontologia de Coordenação representa os elementos do domínio que são utilizados para promover organização e harmonia para os conceitos presentes nas duas subontologias previamente apresentadas. Assim sendo, essa ontologia tem como objetivo organizar e harmonizar a colaboração como um todo, capturando as relações de dependência entre ações, potenciais de colaboração e possíveis conflitos, dado que uma premissa para que a colaboração aconteça entre indivíduos é que os mesmos não assumam objetivos conflitantes no contexto da colaboração.

Segundo Crowston (1994), a Teoria da Coordenação define uma taxonomia de dependências entre ações, restringindo a maneira como as atividades são executadas. Vale enfatizar que não é o objetivo desta seção representar todos os tipos de dependências entre ações, mas definir algumas das mencionadas nessa teoria, como apresentado a seguir. As Figuras 4.11-14 apresentam esses vários tipos de interdependência entre ações. O primeiro tipo, denominado dependência de **Fluxo** (*Flow*) (ver Figura 4.11), é um tipo de dependência entre Ações na qual uma delas produz um Recurso que é requerido por outra Ação. Dessa forma, uma Ação produz um **pós-estado** (*post-state*), isto é, uma Situação no qual esse Recurso passa a existir. Essa situação, em outras palavras, é o **pré-estado** (*pre-state)* de uma outra Ação que requer a utilização desse Recurso. Tem-se dado fluxo (*flow (a,a´,r*)) então o **pós-estado** (*pos-state)* de *a* (no qual *r* passa a existir) é o **pré-estado** (*pre-state* de *a´*). Isto

deixa claro que ***a*** e ***a´*** são necessariamente temporalmente ordenadas. Exemplos de dependência de fluxo são comuns em cadeias de produção, como a produção automotiva, na qual existe uma cadeia de fabricação e uma cadeia de montagem, e as atividades requerem produtos produzidos na linha de fabricação para a utilização na linha de montagem. O axioma **A9** representa a formalização dessa dependência.

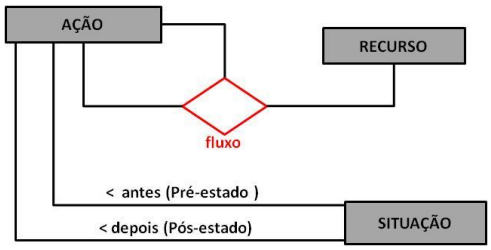


Figura 4.11: Subontologia de Coordenação dependência de Fluxo (*Flow*)
Fonte: Traduzido pela autora, com base em Oliveira (2009).

A9 - *fluxo (a, a', r) ≡ cria (a, r) ∧ requer (a', r)*

A dependência de **Usabilidade** (*Usability*) é um tipo de dependência de **Fluxo** (*Flow*) na qual o Recurso deve atender às características específicas para as necessidades de uma determinada Ação; em outras palavras, o Recurso deve ser apropriado para a utilização satisfazendo os requisitos da Ação. A dependência de **Usabilidade** é destacada na Figura 4.12; nela, o *Recurso* gerado pela Ação precisa atender uma determinada caracterização (*characterization*) definida por um **Momento Universal** (*Moment Universal* **j**), que define o tipo de propriedade necessária para esse Recurso. Portanto, a Ação ***a*** deve gerar uma Situação ***s*** como **pós-estado** (*post-state*), na qual exista esse Recurso ***r***, e ele é apropriado para a Ação ***a'*** (ou seja, ***r*** é caracterizado por uma propriedade específica) que, por sua vez requer *r*, dessa forma ***s*** é um **pré-estado** (*pre-state*) de ***a'*** (ver axioma **A10**). No exemplo da produção automotiva, a dependência de usabilidade garante que as peças encaminhadas para a montagem satisfarão critérios necessários para que o carro possa ser montado.

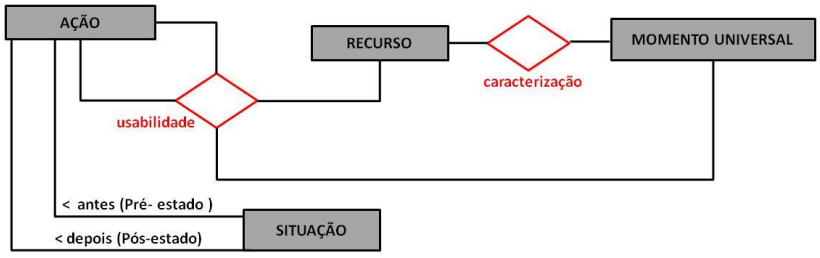


Figura 4.12: Subontologia de Coordenação dependência de Usabilidade (*Usability*)
Fonte: Traduzido pela autora, com base em Oliveira (2009).

A10 - *usabilidade(a, a', r,φ) ≡ fluxo (a, a', r) ∧ caracterização (r, φ, s) ∧ (s = pré-estado (a') ∧ pós-estado(a))*

Levando em consideração os requisitos de localização, a dependência de **Transferência** (*Transfer*) (ver Figura 4.13) define que um Recurso deve ser transferido do local onde foi criado, para o local onde ele será utilizado, ou seja, o Recurso deve estar no local apropriado para que a ação que o requisita o utilize. Essa dependência é uma dependência de **Fluxo** com uma necessidade de atender a um requisito adicional, o de garantir a presença do Recurso em uma **Localização** (*Location*) específica. Dessa forma, dada uma **transferência** (*transfer* (***a***, ***a'***, ***r***, ***l***)), então se tem um fluxo (*flow* (***a***,***a'***,***r***)), e tal recurso ***r*** tem uma localização específica ***l***. Logo, a Ação *a* deve gerar uma *situação s* como **pós-estado** (*post-state*) na qual ***r*** deve estar localizado em ***l***, onde ***l*** é apropriado para a Ação ***a'***, dessa forma ***s*** é um **pré-estado** (*pre-state)* de ***a'***. Novamente, no exemplo da produção automotiva, essa dependência quando atendida garante que os produtos produzidos na linha de produção estejam no lugar correto da linha de montagem para a sua utilização. Também são exemplos desse tipo de dependência as compras realizadas pela internet, nas quais se adquire um determinado produto e esse produto, posteriormente, deve ser entregue no local determinado pelo comprador. O axioma **A11** define formalmente essa dependência.

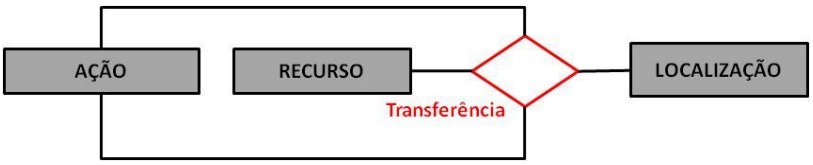


Figura 4.13: Subontologia de Coordenação dependência de Transferência (*Transfer*)
Fonte: Traduzido pela autora, com base em Oliveira (2009).

A11 - *usabilidade(a, a', r,φ) ≡ fluxo (a, a', r) ∧ caracterização (r, φ, s) ∧ (s = pré-estado (a') ∧ pós-estado(a))*

O tipo de dependência de **Precedência** (*Precedence*) é uma consequência da dependência de **Fluxo,** em que uma **Ação requer** (*requires*) um determinado Recurso como **pré-condição** (*precondition*) para a sua execução. Dessa forma essa condição é essencial para habilitar aquela atividade. Conforme o axioma **A12**, onde se tem que para todo ***a***, ***a'*** e ***r*** onde **a** requer um recurso ***r*** e ***a'*** cria ***r,*** então a Ação ***a'*** deve preceder a Ação ***a***, para que a

Ação ***a*** seja realizada sem restrições. Isto exemplifica que uma restrição do tipo **fluxo** (*flow)* implica um tipo de precedência temporal.

A12 - $(\forall\ a, a', r\ requer\ (a, r) \wedge criar\ (a', r) \rightarrow precede(a', a))$

Por fim, a dependência de **Encaixe** (*Fit*), apontada por Crowston e Osborn (2003), é um tipo de **Relação Formal** (*Formal Relation*) entre **Ações** (*Actions)*, na qual duas ou mais Ações são executados simultaneamente para produzir um ou mais Recursos; assim, esses Recursos são gerados de forma coletiva pelos eventos que os criam (ver Figura 4.14). Dessa forma, dado um **Encaixe** *(fit* (*f*)), então existe uma ação complexa ***c***, composta de pelo menos duas Ações ***c',c"*** que agem de forma sincronizada em ***f*** de tal forma a criar um Recurso ***r***. Logo, ***r*** é criado por ***c*** satisfazendo a dependência de ***f***. Um exemplo ilustrativo dessa dependência seria uma atividade de serrar uma árvore com um serrote que somente pode ser manipulado por duas pessoas ao mesmo tempo, gerando um recurso (lenha).

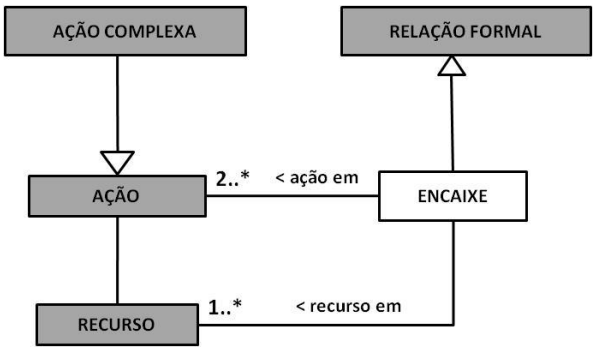


Figura 4.14: Subontologia de Coordenação dependência de Encaixe (*Fit*)
Fonte: Traduzido pela autora, com base em Oliveira (2009).

Como descrito anteriormente e ilustrado na Figura 4.15, um **Recurso** (*Resource***)** pode participar de uma SC, e, para tal, é necessário definir algumas características com a finalidade de se determinar melhor o seu aproveitamento e a detecção de oportunidades de colaboração, visto que, durante um trabalho colaborativo, esse Recurso pode ser demandado em cenários distintos, logo, ocorrendo a necessidade do seu compartilhamento. Um Recurso é considerado **Compartilhável** (*Sharable Resource*) quando é passível de acesso por mais de um agente (ou ação) ao mesmo tempo e, por sua vez, ele é considerado **Exclusivo** (*Exclusive Resource*), quando não é este o caso. Aliás, quando duas ou mais participações de um mesmo Recurso não podem acontecer ao mesmo **tempo** (*overlaps*), ele é dito **Recurso Exclusivo** (*Exclusive Resource*). Um Recurso *r* é dito **Exclusive Resource** se

para toda participação ***p***, ***p'*** de ***r*** então o **Intervalo de Tempo** (*time interval*) ***i*** e ***i'*** respectivamente de ***p*** e ***p'*** não se sobrepõem. A ideia é que não exista mais de uma participação de ***r*** sobrepostas em um intervalo de tempo (ver axioma **A14**). Em contrapartida, como ilustrado pelo axioma **A15**, um Recurso ***r*** é dito **Compartilhável** (*sharable*) se ele não é exclusivo; portanto, a ele é permitido ter participações sobrepostas em um determinado intervalo de tempo.

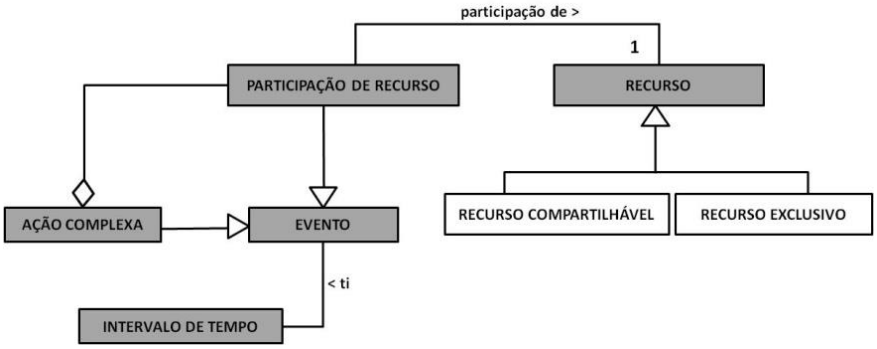


Figura 4.15: Fragmento da Subontologia de Coordenação
Fonte: Traduzido pela autora, com base em Oliveira (2009).

A14 - *recurso exclusivo (r)* $\rightarrow \forall$ *p, p' participação_de (p,r)* $\wedge$ *(p',r) participação_de* $\wedge$ *(p _ p')* $\rightarrow \neg$ *(tempo (ti (p), ti (p'))*

A15 - *recurso compartilhável (r)* $\rightarrow \neg$ *recurso exclusivo (r)*

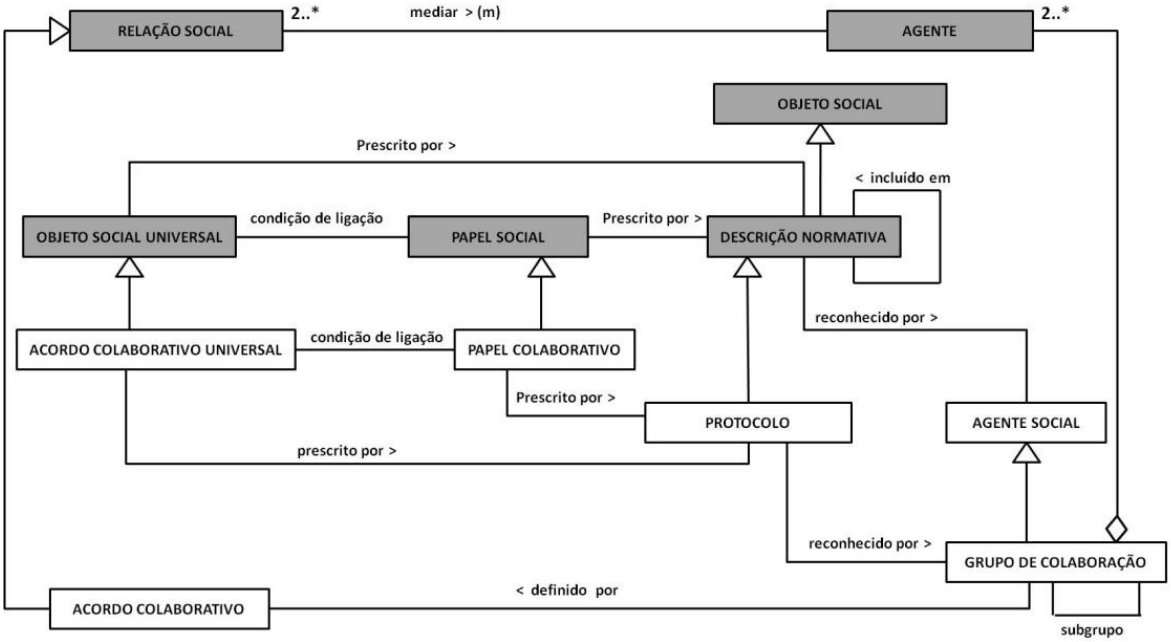


Figura 4.16: Fragmento da Ontologia de Coordenação
Fonte: Traduzido pela autora, com base em Oliveira (2009).

Grupos Colaborativos (*Collaboration Groups*) são formados quando ocorre uma união entre *Agentes,* em especial definida por (*defined by*) **Acordos Colaborativos** (*Collaborative Agreement*) que são mediados entre os membros do grupo (ver axioma **A-16**). Como toda sessão colaborativa (SC) é deliberada por um **Acordo Colaborativo** no qual agentes devem assumir papéis, tem-se que toda SC tem pelo menos um **Grupo Colaborativo**, mesmo que este esteja presente de forma implícita. Esses acordos podem ser vistos como critérios que definem a formação do grupo, como o expresso em Lipnack e Stamps (2000). Além disso, um **Grupo Colaborativo** reconhece (*recognizes*) um **Protocol**o, que é uma descrição das normas e regras que regem o grupo.

Os **Grupos Colaborativos** ainda podem formar *subGrupos* (*subGroup*), que são formados por subconjunto de participantes dentro desse grupo que especializam **Acordos Colaborativos** específicos. Em outras palavras, um *subGrupo* tem os mesmo **Acordos Colaborativos** que o *superGrupo* e, como diferencial, **Acordos Colaborativos** especializados, compartilhados por um subconjunto de Agentes.

Grupos Colaborativos, Agentes, Ações de Contribuições e Sessões Colaborativas são harmonizados por meio de um conjunto de regras denominado **Protocolo**, como ilustrado no modelo da Figura 4.16. O conceito **Protocolo** rege as ações de colaboração durante uma SC. O **Protocolo** (*Normative Description*) tem a descrição da definição dos **Papéis Colaborativos** e seus objetivos adotados através de **Compromissos Fechados**, ou seja, um **Papel Colaborativo** é prescrito por um **Protocolo,** e , tal relação é a manifestação da relação entre *Social Role* e *Normative Description*. É importante salientar que se dois **Papéis Colaborativos** são definidos em um protocolo que descreve um mesmo **Acordo Social**, então:

(*i*) eles não implicam no compromisso de adotar objetivos conflitantes;

(*ii*) as ações, nas quais os compromissos fechados são baseadas, são devidamente ordenadas pelas relações de dependência entre ações descritas, incluindo aquelas descritas nesta seção. Além disso, o **Protocolo** é conhecido e reconhecido pelos **Agentes** e **Grupos**

A16 - *grupo colaborativo (g)* $\equiv \exists$ *c acordo colaborativo (c)* $\wedge$ *(*$\forall$*x parte_de (x ,g)* $\leftrightarrow$ *mediar (c, x))*

A Tabela 4.8 apresenta um dicionário de termos da Subontologia de Coordenação descreve seus principais conceitos:

Tabela 4.8: Dicionário de Termos da Subontologia de Coordenação

TERMOS	DESCRIÇÃO
Protocolo	Conjunto de regras que estabelece coordenação para a harmonia e melhoria da Sessão Colaborativa.
Grupo Colaborativo	Conjunto de agentes definidos por Acordos Colaborativos.
Recurso Exclusivo	Recurso que não pode ser utilizado de forma simultânea.
Recurso Compartilhável	Recurso que pode ser utilizado de forma simultânea.
Fluxo	Dependência entre ações na qual uma delas cria um recurso que é requisitado pela outra.
Usabilidade	Especialização de uma dependência do tipo fluxo (Flow), na qual o recurso produzido deve possuir também características específicas.
Transferência	Especialização de uma dependência do tipo fluxo (Flow), na qual o recurso produzido deve ser disponibilizado em uma localização específica.
Encaixe	Dependência de dois ou mais eventos que necessitam serem executados simultânea para a criação de um recurso

Fonte: Traduzido pela autora, com base em Oliveira (2009); Souza (2016).

Por último, as ações criam Recursos que podem ter tipos de dependências conhecidas como dependências de **Fluxo** (*flow)* que basicamente estão relacionadas a criação, a localização e à usabilidade. Além dessas, ainda é importante mencionar a dependência de **Encaixe** (*fit)*, na qual um Recurso é produzido de forma coletiva através da execução de mais de uma Ação ao mesmo tempo. O reconhecimento desses diversos tipos de dependência pela coordenação é essencial para uma análise das possíveis estratégias de coordenação. Grupos colaborativos são compostos por Agentes e são definidos por Acordos colaborativos. Tais grupos, reconhecem e respeitam regras conhecidas como protocolo, tornando, assim possível que a colaboração entre os mesmos aconteça. Esse protocolo ainda é responsável pelas regras que definem os Papéis colaborativos assumidos pelos Agentes. Levando em consideração os aspectos colaborativos, um Recurso é definido como Exclusivo ou Compartilhável; o primeiro caso acontece quando um Recurso não pode utilizado de forma simultânea, em oposição ao que se dá no segundo caso.

4.7 PADRÕES ONTOLÓGICOS DE COLABORAÇÃO

Os padrões ontológicos de colaboração foram subdivididos em três subontologias que compõem a Ontologia de Colaboração (CONTO) em: Padrões da subontologia de Cooperação, Padrões da subontologia de Coordenação e Padrões da subontologia de

comunicação (SOUZA, 2017). Nesse contexto, a descrição[24] dos DROPs (POs de domínio) que foram extraídos da CONTO inclui os seguintes itens:

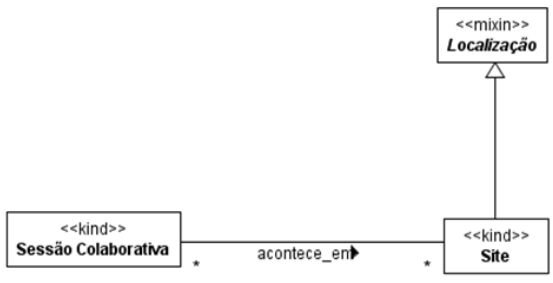
- **NOME/ ACRÔNIMO:** fornece o nome do padrão.
- **PADRÕES ALTERNATIVOS:** lista os padrões alternativos para um padrão.
- **INTENÇÃO:** descreve o propósito do padrão.
- **FUNDAMENTAÇÃO:** descreve a lógica subjacente ao padrão. Uma breve declaração, respondendo a seguinte pergunta: Qual é a lógica do padrão?
- **QUESTÃO DE COMPETÊNCIA:** descreve as questões de competência que o padrão pretende responder.
- **MODELO CONCEITUAL:** representa o diagrama OntoUML, representando os elementos do padrão.

4.7.1 PADRÕES DA SUBONTOLOGIA DE COOPERAÇÃO

4.7.1.1 CSL– LOCALIZAÇÃO DA SESSÃO COLABORATIVA

O padrão CSL tem por objetivo representar a localização onde ocorre a sessão colaborativa (*Collaborative Session*). Um local (*Site*) pode ter nenhuma ou várias sessões colaborativas. A Tabela 4.9 apresenta a descrição detalhada de algumas características do padrão (CSL).

Tabela 4.9: Especificação parcial do padrão(CSL).

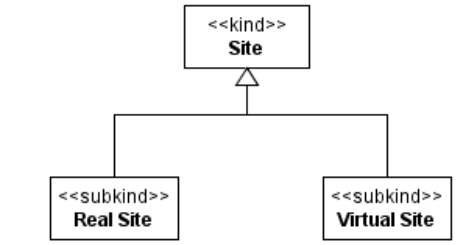
CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Localização da Sessão Colaborativa (CSL)
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar a localização do *site* onde ocorre a sessão colaborativa.
FUNDAMENTAÇÃO	Uma sessão colaborativa acontece em um site específico, que é um local. Esse padrão torna esse site explícito.
QUESTÃO DE COMPETÊNCIA	- Onde está a sessão colaborativa?
MODELO CONCEITUAL	

[24] Descrição - Apesar de as especificações dos padrões no OLED Estendido serem originalmente em inglês, nesta dissertação, nós os especificaremos em português.

4.7.1.2 LT – TIPO DE LOCALIZAÇÃO

O padrão LT tem por objetivo representar os tipos de local (*Site*) onde ocorre a sessão colaborativa (*Collaborative Session*). O local pode ser decomposto em real (*Real Site*) ou virtual (*Virtual Site*). A Tabela 4.10 apresenta a descrição detalhada de algumas características do padrão (LT).

Tabela 4.10: Especificação parcial do padrão (LT)

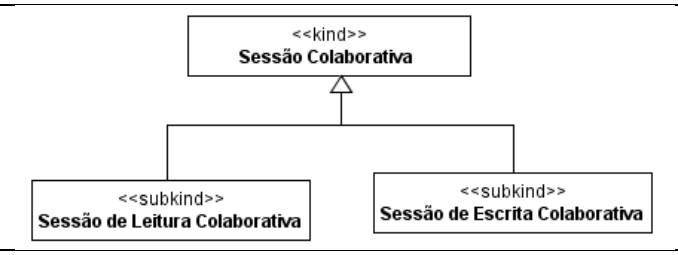
CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Tipo de Localização (LT)
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar o tipo de local em que a sessão colaborativa acontece.
FUNDAMENTAÇÃO	O site em que a sessão de colaboração acontece pode ser um site real ou um site virtual.
QUESTÃO DE COMPETÊNCIA	- O site da sessão colaborativa é real ou virtual?
MODELO CONCEITUAL	

4.7.1.3 CSS - TIPO DE SESSÃO COLABORATIVA

O padrão CSS tem por objetivo representar os tipos de sessão colaborativa (*Collaborative Session*), onde o processo de interação ocorre num nível abstrato. A sessão colaborativa pode ser decomposta em sessão de leitura colaborativa (*Collaborative Reading Session*) ou sessão de escrita colaborativa (*Collaborative Writing*). A Tabela 4.11 apresenta a descrição detalhada de algumas características do padrão (CSS).

Tabela 4.11: Especificação parcial do padrão (CSS)

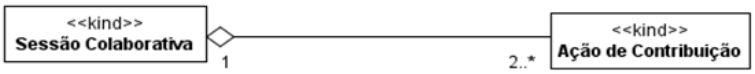
CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Tipo de Sessão Colaborativa (CSS)
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar o tipo de sessão colaborativa, onde o processo de interação acontece.
FUNDAMENTAÇÃO	A sessão em que a interação colaborativa acontece pode ser uma sessão de leitura colaborativa ou uma sessão de escrita colaborativa.
QUESTÃO DE COMPETÊNCIA	- A sessão colaborativa é de leitura ou escrita?

MODELO CONCEITUAL	

4.7.1.4 CSAC – AÇÕES DE CONTRIBUIÇÃO DA SESSÃO COLABORATIVA

O padrão CSAC tem por objetivo representar as ações de contribuição (*Action Contributions*) que compõem uma sessão colaborativa *(Collaborative Session*), isto é, duas ou várias ações de contribuição são partes integrantes de uma sessão colaborativa. A Tabela 4.12 apresenta a descrição detalhada de algumas características do padrão (CSAC).

Tabela 4.12: Especificação parcial do padrão (CSAC)

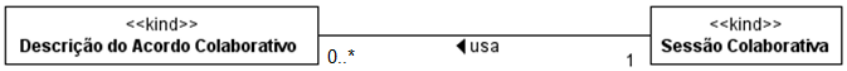
CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Ações de Contribuição da Sessão Colaborativa (CSAC)
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar as ações de contribuição que compõem a sessão colaborativa.
FUNDAMENTAÇÃO	Duas ou várias ações de contribuição que compõem uma sessão colaborativa.
QUESTÃO DE COMPETÊNCIA	- Quais são as ações de contribuições que compõem uma sessão colaborativa?
MODELO CONCEITUAL	

4.7.1.5 CSAD – DESCRIÇÃO DO ACORDO DE SESSÃO COLABORATIVA

O padrão CSAD tem por objetivo representar a descrição dos acordos colaborativos (*Collaborative Agreement Description*) usados em uma sessão colaborativa *(Collaborative Session),* isto é, uma sessão colaborativa (*Collaborative Session*) tem uma descrição de acordos colaborativos (*Collaborative Agreement Description*), que são um tipo de contrato que define os direitos e as obrigações dos papéis que podem participar de uma sessão colaborativa (*Collaborative Session*). A Tabela 4.13 apresenta a descrição detalhada de algumas características do padrão (CSAD).

Tabela 4.13: Especificação parcial do padrão (CSAD)

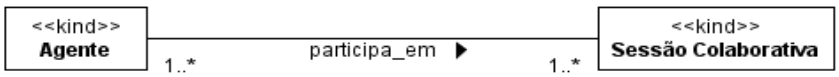
CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Descrição de Acordo da Sessão Colaborativa (CSAD)
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar a descrição dos acordos colaborativos usados em uma sessão colaborativa.

FUNDAMENTAÇÃO	Uma sessão colaborativa tem uma descrição de acordo, que é uma espécie de contrato definindo os direitos e as obrigações dos papéis que podem participar de uma sessão colaborativa.
QUESTÃO DE COMPETÊNCIA	- Qual é a descrição do acordo da sessão colaborativa?
MODELO CONCEITUAL	

4.7.1.6 CSA – AGENTES DE SESSÃO COLABORATIVA

O padrão CSA tem por objetivo representar os agentes participantes da sessão colaborativa (*Collaborative Session*), isto é, uma sessão colaborativa (*Collaborative Session*) só existe se houver dois ou mais agentes (Agents), participando em uma sessão colaborativa (*Collaborative Session*). A Tabela 4.14 apresenta a descrição detalhada de algumas características do padrão (CSA).

Tabela 4.14: Especificação parcial do padrão (CSA)

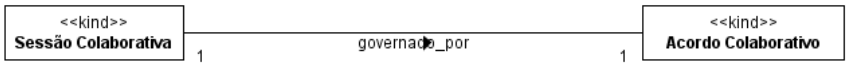
CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Agentes de Sessão Colaborativa (CSA)
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar os agentes que participam da sessão colaborativa.
FUNDAMENTAÇÃO	Dois ou mais agentes participam de uma sessão colaborativa.
QUESTÃO DE COMPETÊNCIA	- Quais são os agentes que participam da sessão colaborativa?
MODELO CONCEITUAL	

4.7.1.7 CA – ACORDO COLABORATIVO

O padrão CA tem por objetivo representar o acordo colaborativo (*Collaborative Agreement*) que rege a sessão colaborativa (*Collaborative Session*), isto é, uma sessão colaborativa é regida por um agente social chamado acordo colaborativo (*Collaborative Agreement*). A Tabela 4.15 apresenta a descrição detalhada de algumas características do padrão (CA).

Tabela 4.15: Especificação parcial do padrão (CA)

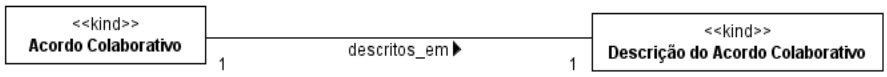
CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Acordo Colaborativo (CA)
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar o acordo colaborativo que rege a colaboração sessão.
FUNDAMENTAÇÃO	Uma sessão colaborativa é regida por um agente social chamado acordo de colaboração.

QUESTÃO DE COMPETÊNCIA	- Qual é o acordo colaborativo que regula a sessão colaborativa?
MODELO CONCEITUAL	

4.7.1.8 CAD – DESCRIÇÃO DO ACORDO COLABORATIVO

O padrão CAD tem por objetivo representar a descrição do acordo colaborativo (*Collaborative Agreement Description*), que é um tipo de contrato que define os direitos e as obrigações dos papéis que podem participar em uma sessão colaborativa (*Collaborative Session*). A Tabela 4.16 apresenta a descrição detalhada de algumas características do padrão (CAD).

Tabela 4.16: Especificação parcial do padrão (CAD)

CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Descrição do Acordo Colaborativo (CAD)
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar a descrição de acordos de colaboração.
FUNDAMENTAÇÃO	Um contrato de colaboração tem uma descrição, que é um tipo de contrato que define os direitos e as obrigações dos papéis que podem participar de uma sessão colaborativa.
QUESTÃO DE COMPETÊNCIA	- Qual é a descrição dos acordos de colaboração?
MODELO CONCEITUAL	

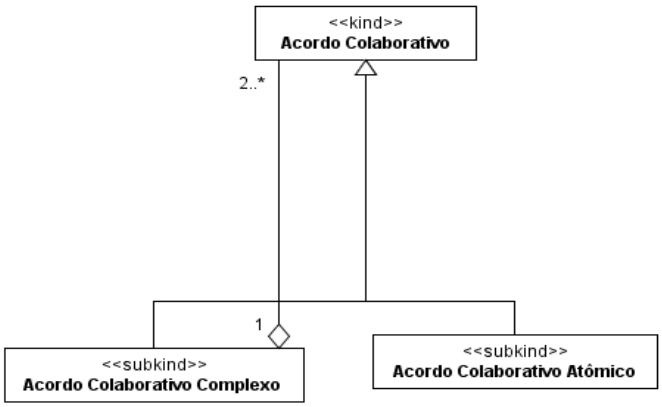
4.7.1.9 CAC – COMPOSIÇÃO DO ACORDO COLABORATIVO

O padrão CAC tem por objetivo representar a decomposição mereológica de um acordo colaborativo (*Collaborative Agreement*). Portanto, um acordo colaborativo (*Collaborative Agreement*) pode ser decomposto em acordo colaborativo complexo (*Complex Collaborative Agreement*) ou acordo colaborativo atômico[25](*Atomic Collaborative Agreement*). A Tabela 4.17 apresenta a descrição detalhada de algumas características do padrão (CAC).

Tabela 4.17: Especificação parcial do padrão (CAC)

CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Composição do Acordo Colaborativo (CAC)
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar a decomposição mereológica de um acordo colaborativo.

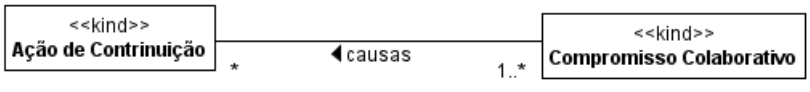
[25] Atômico - Refere-se à ideia de haver um limite para a pequenez das partículas.

FUNDAMENTAÇÃO	O acordo colaborativo pode ser decomposto em acordo colaborativo complexo ou acordo colaborativo atômico. Logo, dois ou mais acordos colaborativos são partes integrantes de um acordo colaborativo complexo.
QUESTÃO DE COMPETÊNCIA	- Este acordo colaborativo pode ser decomposto? - Quais são os componentes deste acordo colaborativo?
MODELO CONCEITUAL	

4.7.1.10 CCAC – COMPROMISSO COLABORATIVO PARA AÇÕES DE CONTRIBUIÇÃO

O padrão CCAC tem por objetivo representar o compromisso colaborativo (*Collaborative Commitment*) que causa ações de contribuição *(Action Contributions*), isto é, uma ação de colaboração (*Action Contribution*) é causada por um compromisso colaborativo (*Collaborative Commitment*). A Tabela 4.18 apresenta a descrição detalhada de algumas características do padrão (CCAC).

Tabela 4.18: Especificação parcial do padrão (CCAC)

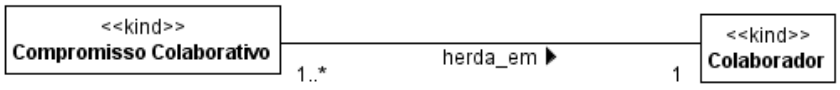
CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Compromisso Colaborativo para Ações de Contribuição (CCAC)
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar o compromisso colaborativo que causa a contribuição da ação.
FUNDAMENTAÇÃO	Uma contribuição de ação é causada por um compromisso colaborativo.
QUESTÃO DE COMPETÊNCIA	- Qual é o compromisso colaborativo que causa ações de contribuição?
MODELO CONCEITUAL	

4.7.1.11 CCIC – COMPROMISSO COLABORATIVO DO COLABORADOR

O padrão CCIC tem por objetivo representar o compromisso colaborativo (*Collaborative Commitment*) que é inerente a um colaborador (*Collaborator*). Assim, um colaborador herda um ou vários compromissos colaborativos, e um compromisso colaborativo é herdado por

somente um colaborador. A Tabela 4.19 apresenta a descrição detalhada de algumas características do padrão (CCIC).

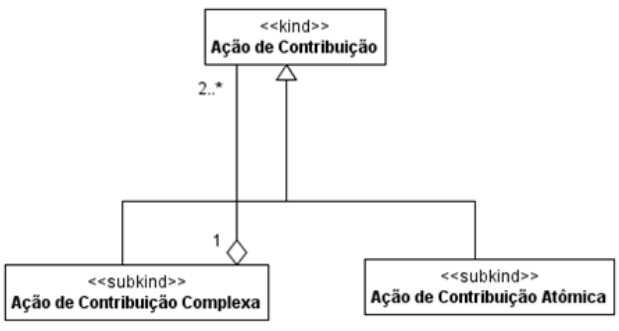
Tabela 4.19: Especificação parcial do padrão (CCIC)

CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Compromisso Colaborativo do Colaborador (CCIC)
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar o compromisso colaborativo que é inerente a um colaborador.
FUNDAMENTAÇÃO	Um compromisso colaborativo é inerente a um colaborador.
QUESTÃO DE COMPETÊNCIA	- Qual compromisso colaborativo é inerente a um colaborador?
MODELO CONCEITUAL	

4.7.1.12 ACC – COMPOSIÇÃO DA AÇÃO DE CONTRIBUIÇÃO

O padrão ACC tem por objetivo representar a decomposição mereológica de uma ação de contribuição (*Action Contribution*). Uma ação de contribuição pode ser decomposta em ação de contribuição complexa (*Complex Action Contribution*) ou ação de contribuição atômica (*Atomic Action Contribution).* A Tabela 4.20 apresenta a descrição detalhada de algumas características do padrão (ACC).

Tabela 4.20: Especificação parcial do padrão (ACC)

CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Composição da Ação de Contribuição (ACC).
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar a decomposição mereológica de uma ação de contribuição.
FUNDAMENTAÇÃO	Uma ação de contribuição pode ser decomposta em ação de contribuição complexa ou ação de contribuição atômica. Assim sendo, duas ou mais ações de contribuição são partes integrantes de uma ação de contribuição complexa.
QUESTÃO DE COMPETÊNCIA	- Quais são os componentes de uma ação de contribuição?
MODELO CONCEITUAL	

4.7.1.13 ACD – DEPENDÊNCIA DA AÇÃO DE CONTRIBUIÇÃO

O padrão ACD tem por objetivo representar a dependência entre ações de contribuição (*Action Contribution*), ou melhor, uma ações de contribuição (*Action Contribution*) depende de outras ações de contribuição (*Action Contribution*). Ou seja, há um relacionamento entre objetos da mesma classe. A Tabela 4.21 apresenta a descrição detalhada de algumas características do padrão (ACD).

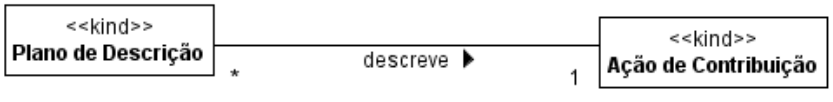
Tabela 4.21: Especificação parcial do padrão (ACD)

CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Dependência da Ação de Contribuição (ACD).
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar uma ação de contribuição que depende de outras ações de contribuição.
FUNDAMENTAÇÃO	Uma ação de contribuição que depende de outras ações de contribuição, ou seja, um relacionamento entre objetos da mesma classe.
QUESTÃO DE COMPETÊNCIA	- Esta ação de contribuição depende de outras ações de contribuição?
MODELO CONCEITUAL	

4.7.1.14 ACPD – DESCRIÇÃO DO PLANO DA AÇÃO DE CONTRIBUIÇÃO

O padrão ACPD tem por objetivo representar o plano de descrição (*Plan Description*) das ações de contribuição (*Action Contribution*). A Tabela 4.22 apresenta a descrição detalhada de algumas características do padrão (ACPD).

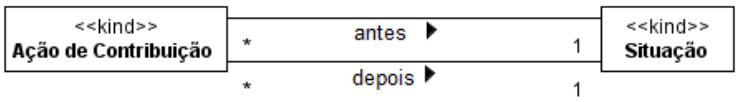
Tabela 4.22: Especificação parcial do padrão(ACPD)

CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Descrição do Plano da Ação de Contribuição (ACPD).
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar a descrição do plano das ações de contribuição.
FUNDAMENTAÇÃO	Uma descrição do plano descreve as ações de contribuição.
QUESTÃO DE COMPETÊNCIA	- Esta ação de contribuição é descrita por uma descrição do plano?
MODELO CONCEITUAL	

4.7.1.15 ACPPS – SITUAÇÃO ANTES E DEPOIS DA AÇÃO DE CONTRIBUIÇÃO

O padrão ACPPS tem por objetivo representar a situação (*Situation*) antes (*pre*) e depois (*post*) da ação de contribuição (*Action Contribution*). A Tabela 4.23 apresenta a descrição detalhada de algumas características do padrão (ACPPS).

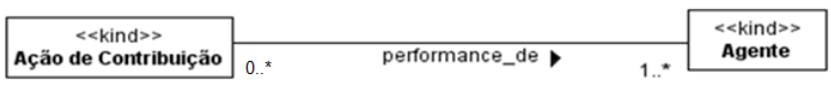
Tabela 4.23: Especificação parcial do padrão (ACPPS)

CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Situação Antes e Depois da Ação de Contribuição (ACPPS).
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar a situação antes e depois das ações de contribuição.
FUNDAMENTAÇÃO	Uma situação antes e depois das ações de contribuição.
QUESTÃO DE COMPETÊNCIA	- Qual é a situação antes da ação de contribuição? - Qual é a situação após a ação de contribuição?
MODELO CONCEITUAL	

4.7.1.16 APAC – AGENTE REALIZA AÇÃO DE CONTRIBUIÇÃO

O padrão APAC tem por objetivo representar os papéis colaborativos instanciados pelos agentes que participam da sessão colaborativa. Os agentes que participam de uma sessão colaborativa assumem papéis colaborativos. A Tabela 4.24 apresenta a descrição detalhada de algumas características do padrão (APAC).

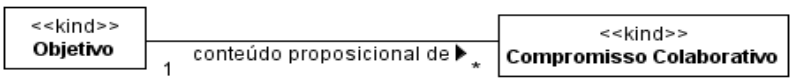
Tabela 4.24: Especificação parcial do padrão (APAC)

CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Agente Realiza Ação de Contribuição (APAC).
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar os papéis colaborativos instanciados pelos agentes que participam da sessão colaborativa.
FUNDAMENTAÇÃO	Os agentes que participam de uma sessão colaborativa assumem papéis colaborativos.
QUESTÃO DE COMPETÊNCIA	- Quais papéis colaborativos são instanciados pelos agentes?
MODELO CONCEITUAL	

4.7.1.17 GPCCC – OBJETIVO COMO CONTEÚDO PROPOSICIONAL DE UM COMPROMISSO COLABORATIVO

O padrão GPCCC representa o objetivo (*Goal*) como um conteúdo proposicional de um acordo colaborativo (*Collaborative Commitment*). A Tabela 4.25 apresenta a descrição detalhada de algumas características do padrão (GPCCC).

Tabela 4.25: Especificação parcial do padrão (GPCCC)

CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Objetivo como Conteúdo Proposicional de um Compromisso Colaborativo ((GPCCC).
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar o objetivo que é o conteúdo proposicional de um compromisso colaborativo.
FUNDAMENTAÇÃO	Um objetivo que é o conteúdo proposicional de um compromisso colaborativo.
QUESTÃO DE COMPETÊNCIA	- Qual é o objetivo que é o conteúdo proposicional de um compromisso colaborativo?
MODELO CONCEITUAL	

4.7.1.18 GC – CONFLITO ENTRE OBJETIVOS

O padrão GC tem por objetivo representar o conflito entre objetivos (*Goal*). A Tabela 4.26 apresenta a descrição detalhada de algumas características do padrão (GC).

Tabela 4.26: Especificação parcial do padrão (GC)

CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Conflito Entre Dois Objetivos (GC).
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar o conflito entre outros objetivos.
FUNDAMENTAÇÃO	Um objetivo entra em conflito com outro objetivo.
QUESTÃO DE COMPETÊNCIA	- Esse objetivo entra em conflito com outro objetivo?
MODELO CONCEITUAL	

4.7.1.19 GD – DEPENDÊNCIA ENTRE DOIS OBJETIVOS

O padrão GC tem por objetivo representar a dependência entre objetivos (*Goal*). A Tabela 4.27 apresenta a descrição detalhada de algumas características do padrão (GD).

Tabela 4.27: Especificação parcial do padrão (GD)

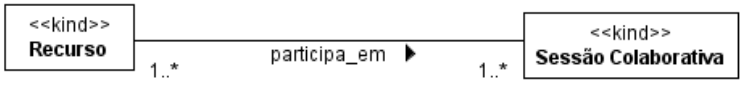
CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Dependência Entre Dois Objetivos (GD).
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar a dependência entre objetivos.
FUNDAMENTAÇÃO	Um objetivo depende de outros objetivos.
QUESTÃO DE COMPETÊNCIA	- Esse objetivo depende de outro objetivo?
MODELO CONCEITUAL	

4.7.2 PADRÕES DA SUBONTOLOGIA DE COORDENAÇÃO

4.7.2.1 CSRS – RECURSOS SIMPLES DE SESSÃO COLABORATIVA

O padrão CSRS tem por objetivo representar a participação de recursos (*Resources*) em sessão colaborativa (*Collaborative Session*), sem a mediação de um agente de participação de recursos (*Resource Participation*). Portanto, em uma sessão colaborativa participam um ou mais recursos, e um recurso participa em uma ou mais sessões colaborativas. A Tabela 4.28 apresenta a descrição detalhada de algumas características do padrão (CSRS).

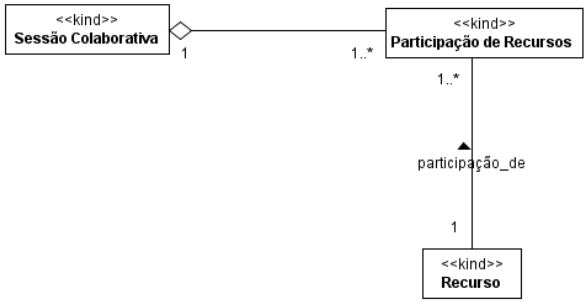
Tabela 4.28: Especificação parcial do padrão (CSRS)

CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Recursos Simples da Sessão Colaborativa (CSRS)
PADRÕES ALTERNATIVOS	Recurso de sessão colaborativa (CSR)
INTENÇÃO	Representar os recursos que participam da sessão colaborativa.
FUNDAMENTAÇÃO	Em uma sessão colaborativa participam um ou mais recursos, e um recurso participa em uma ou mais sessões colaborativas.
QUESTÃO DE COMPETÊNCIA	- Quais são os recursos simples que participam da sessão colaborativa?
MODELO CONCEITUAL	

4.7.2.2 CSR – COLLABORATIVE SESSION RESOURCE

O padrão CSR tem por objetivo representar a participação de recursos (*Resource)* em sessão colaborativa (*Collaborative Session*), com a mediação de um agente de participação de recursos (*Resource Participation*). Uma sessão colaborativa (*Collaborative Session*) tem uma ou mais participações de agentes de recursos (*Resource Participation*). A Tabela 4.29 apresenta a descrição detalhada de algumas características do padrão (CSR).

Tabela 4.29: Especificação parcial do padrão (CSR)

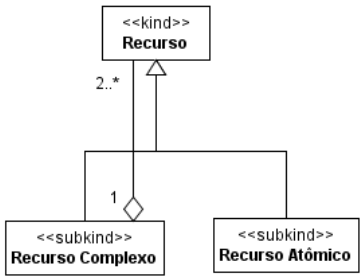
CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Recursos da Sessão Colaborativa (CSR)
PADRÕES ALTERNATIVOS	Recursos simples da sessão colaborativa (CSRS)
INTENÇÃO	Representar a participação de recursos em sessão colaborativa, com a mediação de um agente de participação de recursos.
FUNDAMENTAÇÃO	Uma sessão colaborativa possui um ou mais agentes de participação de recursos.
QUESTÃO DE COMPETÊNCIA	- Quais são os recursos que participam da sessão colaborativa?
MODELO CONCEITUAL	

4.7.2.3 RC – COMPOSIÇÃO DE RECURSO

O padrão RC tem por objetivo representar a decomposição mereológica dos recursos (*Resource*). Um recurso pode ser decomposto em recurso complexo (*Complex Resource*) ou recurso atômico (*Atomic Resource).* Os recursos complexos podem ser ainda mais decompostos, enquanto os recursos atômicos não podem ser decompostos. A Tabela 4.30 apresenta a descrição detalhada de algumas características do padrão (RC).

Tabela 4.30: Especificação parcial do padrão (RC)

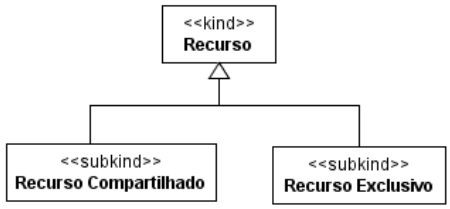
CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Composição de Recurso (RC)
PADRÕES ALTERNATIVOS	--
INTENÇÃO	Representar a decomposição mereológica de recursos.
FUNDAMENTAÇÃO	Um recurso pode ser composto de outros recursos, ou recursos complexos (que podem ser ainda mais decompostos) ou recursos simples (que não podem ser decompostos).

QUESTÃO DE COMPETÊNCIA	- Esse recurso pode ser decomposto? - Quais são os componentes desse recurso?
MODELO CONCEITUAL	

4.7.2.4 RN – NATUREZA DE RECURSO

O padrão RN tem por objetivo especificar os tipos de recursos (*Resource*) em termos de compartilhamento. Assim sendo, um recurso pode ser compartilhável (*Sharable Resource*) ou exclusivo (*Exclusive Resource*). A Tabela 4.31 apresenta a descrição detalhada de algumas características do padrão (RN).

Tabela 4.31: Especificação parcial do padrão (RN)

CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Natureza de Recurso (RN)
PADRÕES ALTERNATIVOS	--
INTENÇÃO	Representar os tipos de recursos em termos de compartilhamento.
FUNDAMENTAÇÃO	Um recurso pode ser compartilhável ou exclusivo.
QUESTÃO DE COMPETÊNCIA	- Esse recurso pode ser compartilhado?
MODELO CONCEITUAL	

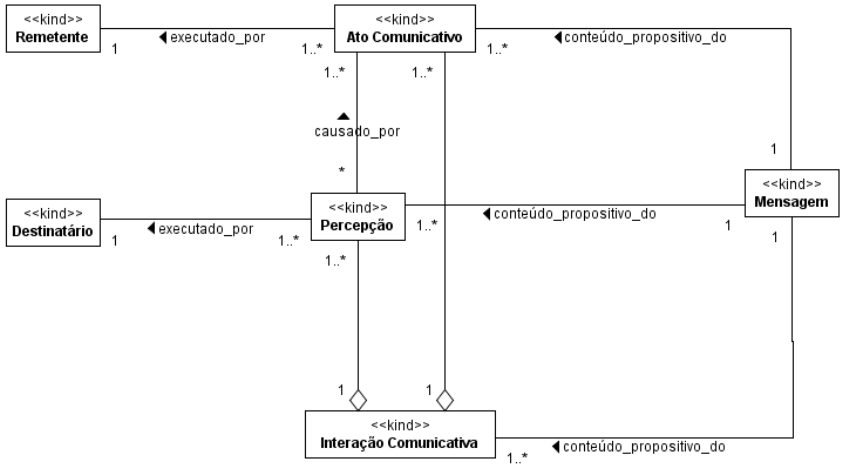
4.7.3 PADRÕES DA SUBONTOLOGIA DE COMUNICAÇÃO

4.7.3.1 CIP – PADRÃO DE INTERAÇÃO DE COMUNICAÇÃO

O padrão CIP tem por objetivo representar o ato comunicativo (*Communicative Act) exe*cutado por um remetente (*Sender*) e uma percepção realizada por um destinatário (*Receiver*), em que a mensagem (*Message*) é o conteúdo proposicional do ato comunicativo (*Communicative Act*), da interação comunicativa (*Communicative Interaction*) e da

percepção (*Perception*). A Tabela 4.32 apresenta a descrição detalhada de algumas características do padrão (CIP).

Tabela 4.32: Especificação parcial do padrão (CIP)

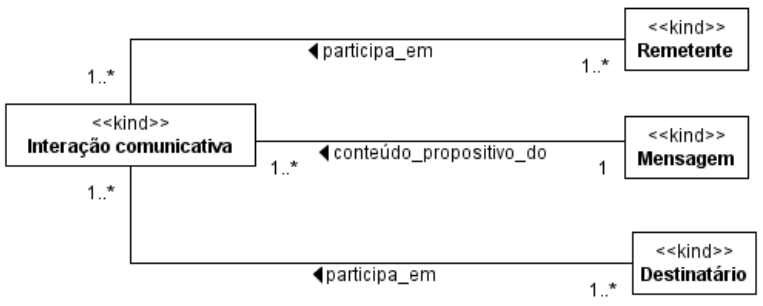
CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Padrão de Interação de Comunicação (CIP)
PADRÕES ALTERNATIVOS	Padrão de interação de comunicação simples (CIPS)
INTENÇÃO	Representar detalhadamente um padrão de interação conectando um remetente e um destinatário.
FUNDAMENTAÇÃO	Uma interação de comunicação consiste em um ato comunicativo realizado pelo emissor e uma percepção realizada pelo receptor. A mensagem é o conteúdo proposicional do ato comunicativo e da percepção
QUESTÃO DE COMPETÊNCIA	- Quem é o remetente na interação comunicativa? - Quem é o destinatário na interação comunicativa? - Qual é a mensagem da interação comunicativa?
MODELO CONCEITUAL	

4.7.3.2 CIPS – PADRÃO DE INTERAÇÃO DE COMUNICAÇÃO SIMPLES

O padrão CIPS tem por objetivo representar uma interação comunicativa entre um remetente (*Sender*) e um destinatário (*Receiver*), em que a mensagem é o conteúdo proposicional da interação comunicativa (*Communicative Interaction*). A Tabela 4.33 apresenta a descrição detalhada de algumas características do padrão (CIPS).

Tabela 4.33: Especificação parcial do padrão (CIPS)

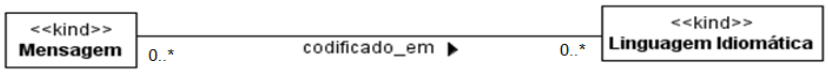
CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Padrão de interação de comunicação simples (CIPS)
PADRÕES ALTERNATIVOS	Padrão de interação de comunicação (CIP)
INTENÇÃO	Representar uma interação comunicativa entre um remetente e um destinatário.
FUNDAMENTAÇÃO	Uma interação de comunicação consiste em um remetente enviar uma mensagem para um destinatário.
QUESTÃO DE COMPETÊNCIA	- Quem é o remetente de cada ato comunicativo?

	- Quem é o destinatário de cada ato comunicativo? - Qual é a mensagem de cada ato comunicativo?
MODELO CONCEITUAL	

4.7.3.3 IL – PADRÃO DE LINGUAGEM IDIOMÁTICA

O padrão IL tem por objetivo representar a linguagem idiomática (*Idiomatic Language*) utilizada para codificar a mensagem (*Message*), ou melhor, a mensagem pode ser codificada em uma linguagem idiomática. A Tabela 4.34 apresenta a descrição detalhada de algumas características do padrão (IL).

Tabela 4.34: Especificação parcial do padrão (IL)

CARACTERÍSTICA	DESCRIÇÃO
NOME / ACRÔNIMO	Padrão de linguagem idiomática (IL)
PADRÕES ALTERNATIVOS	- -
INTENÇÃO	Representar a linguagem idiomática utilizada para codificar a mensagem.
FUNDAMENTAÇÃO	Uma mensagem pode ser codificada em uma linguagem idiomática.
QUESTÃO DE COMPETÊNCIA	- Qual é a linguagem idiomática que codifica a mensagem?
MODELO CONCEITUAL	

4.8 AVALIAÇÃO DOS MODELOS VIA OLED

Os modelos criados no OLED podem ser verificados sintaticamente, validados via simulação e checados via anti-padrões. A validação de uma modelagem ontológica como um todo é ainda uma questão de pesquisa em aberto. Entretanto, esta pesquisa adota a verificação sintática como uma maneira de testar a viabilidade de uso dos DROPs de colaboração na geração de modelos conceituais ontológicos adequados, haja vista que, na literatura, foram encontradas poucas referências sobre a verificação sintática de modelos conceituais ONTOUML no OLED.

A princípio, tal verificação se utiliza do metamodelo da OntoUML, sobre o qual a ferramenta foi implementada, para comparar o modelo submetido às restrições ontológicas do metamodelo. Neste âmbito, uma das premissas da linguagem OntoUML, é manter a consistência com as regras de UML. Ou seja, cada modelo ONTOUML sintaticamente válido também será um modelo UML válido. Para que isso aconteça, os possíveis tipos instanciáveis de OntoUML são representados em UML com estereótipos de uma classe. Assim sendo, uma das restrições que OntoUML impõe para UML é a redução para apenas uma das possibilidades de estereótipo para uma dada classe. Afinal, temos que inúmeros benefícios surgem do fato de OntoUML manter a estrutura sintática de UML inalterada.

Podemos destacar o fato de um modelo OntoUML poder ser criado usando qualquer ferramenta de diagramação que suporte a sintaxe de UML 2.0. Propõe-se que cada PO de domínio (DROPs) de colaboração contido no modelo específico de *groupware* será testado sintaticamente para dar uma ideia se são flexíveis o bastante para se adequar ao contexto de modelagem no domínio em questão. O capítulo 6 desta dissertação ilustra, por meio de estudos de caso, como os POs podem ser utilizados e, como consequência, verifica sintaticamente alguns padrões a partir desses mesmos estudos.

4.9 OPL DE COLABORAÇÃO

A C-OPL (*Collaboration Ontology Pattern Language*), concebida por Souza (2017), foi idealizada para guiar os engenheiros de ontologias na utilização dos DROPs de Colaboração presentes no catálogo criado a partir da CONTO. O autor cita que a C-OPL é uma versão preliminar da linguagem de padrões ontológicos (OPL) para o domínio de colaboração, pois considera apenas uma quantidade limitada de padrões. Isso se deve ao fato de a C-OPL ser um trabalho em andamento e, assim, não finalizada.

A Figura 4.17 apresenta o diagrama de atividade UML que fornece uma visão geral do modelo de processo da C-OPL. Esta versão da C-OPL (1.0) fez uso da linguagem de especificação de OPLs proposta por Quirino et al. (2017), e foi construída com a utilização da ferramenta Astah[26], uma vez que o OLED não suporta a criação de OPLs nele. Em uma visão geral, um modelo de processo descreve em que ordem utilizar os padrões para resolver problemas de modelagem do domínio a que se presta (SOUZA, 2017).

[26] http://astah.net/

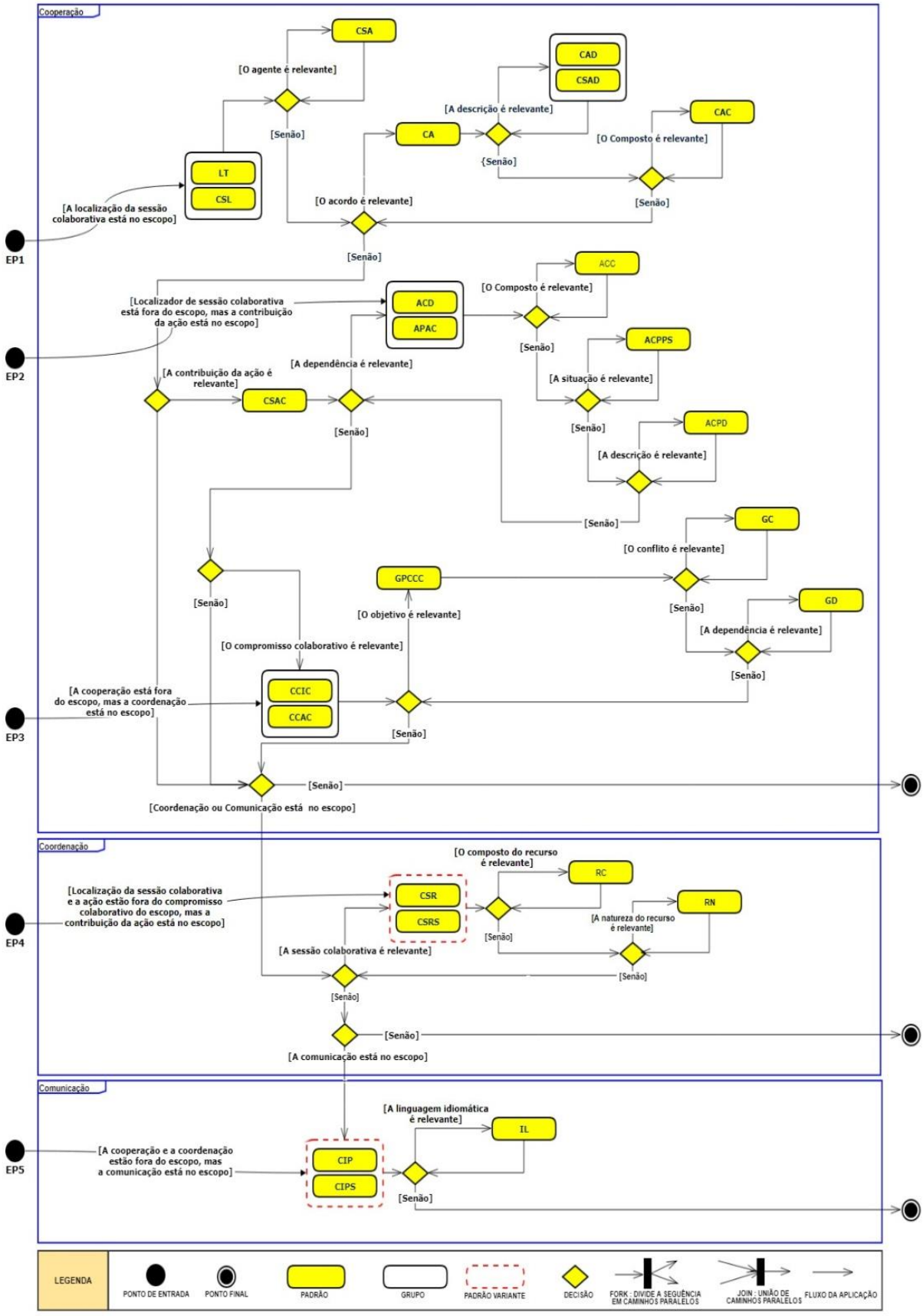


Figura 4.17: Modelo do processo da C-OPL

Fonte: Traduzido pela autora, com base em Souza (2017).

Segundo Souza (2017), a C-OPL ainda está incompleta em sua definição e em estado de pesquisa e alteração. Diante disto, podem ser geradas novas versões como aprimoramentos e especificação dos DROPs. Tendo em vista que faltam propostas que mostrem como adequar a modelagem ontológica às linguagens e aos processos de especificação de *software* (Villela; Oliveira; Braga, 2004), o presente trabalho procura suprir essa lacuna, contribuindo com um método e respectivo modelo de processo e uso para auxiliar na construção de modelos conceituais de *groupware*, baseado em Padrões Ontológicos de Colaboração (CONTO), utilizando o módulo OLED Estendido do Editor OLED e os paradigmas da estruturação de requisitos orientados a objetivos, a fim de facilitar o desenvolvimento de modelagem conceitual.

4.10 CATÁLOGO DROPs DE COLABORAÇÃO

O catálogo DROPs de colaboração foi concebido através do uso da metodologia SABIO (*Systematic Approach for Building Ontologies*), que propõe uma abordagem sistemática para a construção de ontologias de domínio (FALBO et al., 2011b). A Figura 4.18 apresenta uma visão geral do SABIO 2.0.

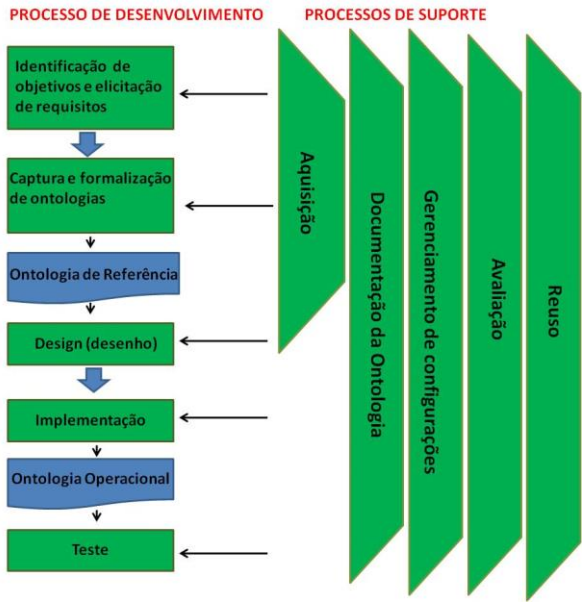


Figura 4.18: Processo da metodologia SABÍO
Fonte: Traduzido pela autora, com base em Falbo et al. (2011b).

Segundo Falbo et at. (2011b), a metodologia SABIO pode ser usada apenas parcialmente para o desenvolvimento de ontologias de referência de domínio ou integralmente visando

desenvolvimento de ontologias operacionais de domínio (FALBO et al., 2011b). Como essa figura mostra, o processo de desenvolvimento da SABIO compreende cinco fases principais:

- **Identificação de objetivos e elicitação de requisitos**: tem por objetivo identificar claramente o propósito da ontologia e seu uso pretendido, isto é, a competência a que a ontologia deverá responder.
- **Captura e formalização de ontologias**: na segunda fase do método, conceitos, relações, restrições e axiomas considerados importantes para a conceituação da ontologia devem ser identificados e organizados. Nessa fase também é caracterizada uma representação (através de uma linguagem formal de representação) da conceituação capturada pela ontologia. A SABiO ainda sugere que modelos que utilizem linguagem gráfica sejam usados para facilitar a comunicação entre os especialistas de domínio e o engenheiro de ontologias. Ao final dessa fase, a ontologia de referência de domínio é obtida.
- **Design**: nesta fase, a especificação conceitual da ontologia de referência deve ser transformada em uma especificação de design para uma ontologia operacional a ser implementada. Essa especificação deve levar em consideração questões arquiteturais e tecnológicas da implementação
- **Implementação**: assim como na SE tradicional, a fase de implementação de ontologias trata da codificação da ontologia para uma linguagem operacional escolhida durante a fase de *design* que seja capaz de ser interpretada por uma máquina.
- **Teste**: a fase de testes é a ultima fase proposta pela metodologia e consiste em verificar e validar, de forma dinâmica e objetiva, o comportamento da ontologia operacional através de conjunto de casos de teste e comparar os resultados com o comportamento esperado, baseando-se nas questões de competência.

O método SABiO sugere que os testes sejam feitos, inicialmente partindo das subontologias que compõem a ontologia de domínio, subindo à medida que as sub-ontologias vão sendo integradas, até que a ontologia final esteja devidamente integrada e possa ser testada funcionando como um todo. Já os processos de suporte são executados em paralelo ao processo de desenvolvimento. Embora a Figura 4.18 sugira um fluxo de trabalho um tanto sequencial, a SABIO não prescreve nenhum modelo específico de ciclo de vida. Assim, modelos de ciclo de vida, como Incremental e Espiral, podem ser adotados. No

entando, o autor recomenda o desenvolvimento incremental e iterativo (FALBO et al., 2011b).

4.11 CARREGANDO O CATÁLOGO DROPs DE COLABORAÇÃO NO OLED

Para fazer uso do catálogo DROPs de colaboração, deve-se importá-lo no Editor OLED. Assim, importar tem o mesmo sentido de carregar, o que significa disponibilizar os POs de domínio (DROPs) para utilização no OLED. A importação do catálogo é feita através da funcionalidade "importando de" que possibilita a execução dessa tarefa de carregamento do catálogo. Na Figura 4.19, podem ser vistos os elementos participantes da opção de importação do catálogo DROPs de colaboração:

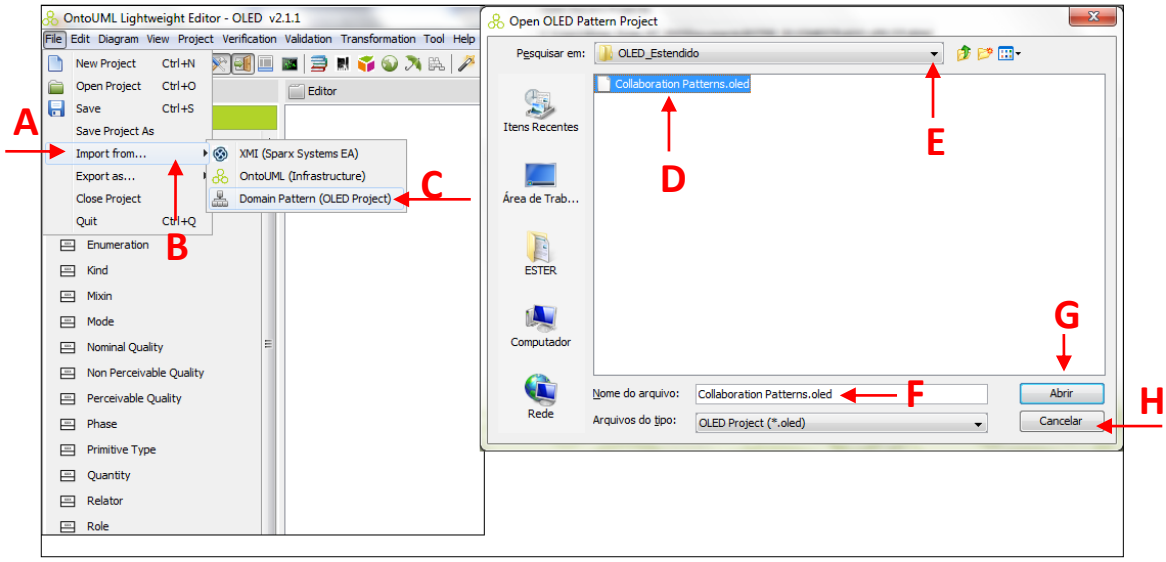


Figura 4.19: Importação catálogo DROPs de colaboração no OLED

Fonte: Ferramenta "OLED Estendido".

O **Menu principal** contém a opção de importação do catálogo DROPs de colaboração que é acessado em *File -> Import from -> Domain Pattern (OLED)*. Uma vez selecionado o item do *menu* para a importação do catálogo, será aberta uma janela para a seleção do projeto do catálogo, conforme Figura 4.19. Nela, são apresentados os elementos que compõem a tela com alguns pontos em destaque que executam as seguintes funções:

(A),(B): O *menu* principal contém opção de importação do catálogo DROPs de colaboração;

(C): Campo de entrada para a informação do nome do arquivo do projeto do catálogo que é preenchido automaticamente com a seleção do arquivo (*A*);

(D): Seleção do projeto do catálogo DROPs de colaboração;

(E): Diretório onde se encontra o arquivo com o projeto do catálogo de DROPs;

(F): Nome do arquivo que contém o projeto (*Collaboration Patterns.oled*);

(G): Botão **Abrir** *(Open)* a importação do catálogo selecionado em (A);

(H): Botão **Cancelar** (*Cancel)* a importação do arquivo que contém o projeto do catálogo.

4.12 QUESTÕES DE COMPETÊNCIA DOS DROPs DE COLABORAÇÃO NO OLED

A elicitação das questões de competência (QC) servem para determinar quais POs podem ser utilizados, pois são elas que nos respondem à pergunta: "**qual é o problema recorrente que o padrão resolve**?". Assim sendo, os padrões têm por objetivo lidar com problemas recorrentes estudados e mapeados (BUSCHMANN; HENNEY; SCHMIDT, 2007). As QCs informadas pelos padrões da CONTO poderão ser visualizadas pelo modelador quando ele posicionar o ponteiro do mouse sobre o PO do catálogo, quando este tiver sido importado no OLED.

Na Figura 4.20, é apresentada a aba **questões de competência** (*Competency Questions*), selecionada com os elementos do padrão CSL que apresentam as configurações do PO selecionado que tem as seguintes funções no OLED:

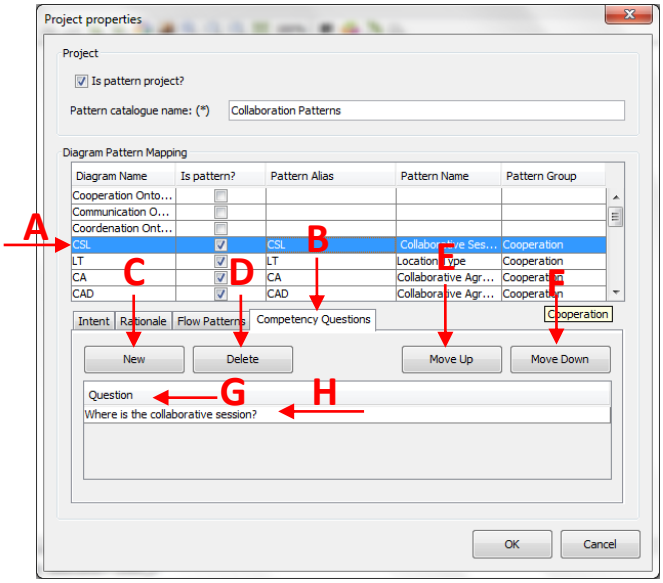


Figura 4.20: Configuração das questões de competência
Fonte: Ferramenta "OLED Estendido".

(A): Registro da tabela de mapeamento em que o padrão selecionado é CSL, como pode ser visto pelo destaque do registro;

(B): Aba **questões de competência** (*Competency Questions*) selecionada, apresentando os elementos de configuração das QCs;

(C): Botão **Novo** (*New*), que é responsável por incluir um novo registro em branco na listagem QCs (*G*), o qual se torna editável, com um dublo clique do *mouse*, permitindo, assim, a inclusão do conteúdo;

(D): Botão **Delete** (*D*), responsável por excluir um registro da listagem de QC (G); basta selecionar o registro na listagem das QCs (*G*), antes de clicar sobre o botão **Delete** (D), para que a QC seja excluída da listagem;

(E): Botão *Move Up*, que é responsável por mover um registro da listagem de QC (G) para cima, bastando selecionar a QC desejada na listagem antes do clique;

(F): Botão *Move Down,* que é responsável por mover um registro da listagem de QC (G) para baixo, bastando selecionar a QC desejada na listagem antes do clique;

(G): Lista *Question,* que contém as QCs cadastradas para o padrão selecionado;

(H): Registro na lista das QCs (G). O registro se torna editável com um duplo clique do *mouse*.

4.13 PADRÕES ALTERNATIVOS OU VARIANTES NO OLED

É importante destacar que um padrão do catálogo DROPs de colaboração pode dar origem a dois (ou mais) padrões alternativos ou variantes. Por exemplo, um PO é alternativo a outro, se ambos lidam com o mesmo problema de diferentes maneiras, mas se comportam de diferentes formas em nível de complexidade. Já um PO variante é construído com base em outro PO existente, uma vez que os POs estão fortemente inter-relacionados para funcionar como um todo.

Na Figura 4.21, pode-se ver a aba **padrões de fluxo** (*Flow patterns*), selecionada com os elementos participantes da configuração relacionados ao PO criado. Nela, estão destacados alguns elementos, que têm as seguintes funções:

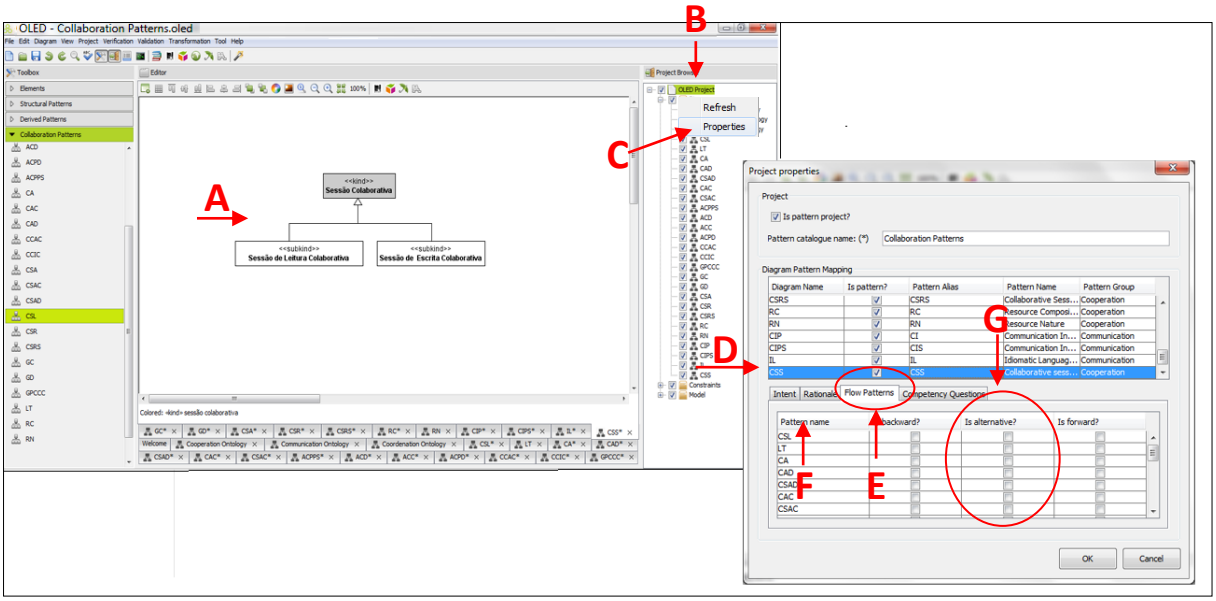


Figura 4.21: Criação do Padrão variante CSS

Fonte: Ferramenta "OLED Estendido".

(A)(B)(C): A partir da Classe (Sessão Colaborativa) do diagrama PO CSL, pode-se generalizá-la em duas subclasses (Sessão de Escrita Colaborativa e Sessão de Leitura Colaborativa) em (A). Feito isso, pode-se clicar na raiz do **Projeto OLED** (OLED *Project*) em (B). Em seguida, a ferramenta irá disponibilizar a opção **Propriedades** (*Properties*) em (C). Após esta ação, se abrirá a janela de **Propriedades do projeto** (*Project Properties*). Nela, devem ser feitas as configurações do novo padrão variante PO CSS.

(D): Registro da tabela de mapeamento do diagrama do PO CSS, neste caso, está selecionado como pode ser visto pelo destaque do registro;

(E)(F): Aba **Padrões de fluxo** (*Flow Patterns*) selecionada apresenta os elementos de configuração relacionados ao PO CSS que é um PO variante (criado a partir de outro PO existente) em (E). A coluna **é Alternativa** (*Is alternative?*) da aba **Padrões de fluxo** contém os POs existentes no catálogo para que o modelador possa marcar os POs alternativos, caso existam em (F).

(G): A coluna **Nome do padrão** (*Pattern name*), que é responsável por apresentar os POs presentes no catálogo, em que cada PO é um registro na tabela de configuração do relacionamento.

CAPÍTULO 5. PROCEDIMENTOS PARA CONSTRUÇÃO DO MÉTODO E PROCESSO DE USO

O método ÔMEGA proposto para auxiliar na construção de modelos conceituais específicos de *groupware* consiste nas seguintes etapas, conforme exibido na Figura 5.

5.1 ETAPAS DO MÉTODO ÔMEGA

- **Etapa 1**- "**Obter artefatos de colaboração**": tem como objetivo obter os artefatos colaborativos aplicáveis ao método ÔMEGA.
- **Etapa 2**- "**Obter requisitos de *groupware***": tem como objetivo obter os requisitos de *groupware* aplicáveis ao método ÔMEGA.
- **Etapa 3**- "**Analisar requisitos de *groupware* obtidos**": tem como objetivo detectar a ocorrência de não-conformidades[27] relacionadas aos requisitos de groupware obtidos.
- **Etapa 4**- "**Solicitar ajustes na especificação de requisitos**": tem como objetivo recomendar ações corretivas por parte do usuário (projetista de sistemas).
- **Etapa 5**- "**Obter ajustes solicitados na especificação de requisitos**": tem como objetivo lograr as correções por parte do usuário (projetista de sistemas).
- **Etapa 6**- "**Localizar informações necessárias para compreensão do domínio**": tem como objetivo localizar as informações necessárias para compreender o domínio no contexto dos requisitos, sem se utilizar do modelo de estruturação de requisitos orientados a objetivos
- **Etapa 7**- "**Estruturar requisitos de *groupware* obtidos**": tem como objetivo organizar e capturar as informações necessárias para compreender o domínio no contexto dos requisitos, utilizando se do modelo de estruturação de requisitos orientados a objetivos para ajudar a definir as principais características da solução.
- **Etapa 8**- "**Adequar requisitos de groupware aos padrões**": tem como objetivo realizar a adequação dos requisitos de *groupware* aos Padrões Ontológicos de Colaboração.
- **Etapa 9**- "**Construir modelo conceitual específico no OLED Estendido**": tem como objetivo ilustrar o processo de construção de modelos conceituais específicos de *groupware*

[27] Não-conformidade - o não atendimento de um requisito pré-estabelecido, inconsistências, omissões, erros ou falhas.

baseados em Padrões Ontológicos de Colaboração no módulo OLED Estendido da ferramenta OLED.

- **Etapa 10**- "**Gerar novos artefatos de uso**": tem como objetivo apresentar "Catálogo de DROPs de Colaboração Consolidado" e seu respectivo "Modelo de Processo de Uso ALFA (AP-OPL)" gerado.

A Figura 5 ilustra de maneira diagramática o primeiro nível do processo de definição das etapas de trabalho do método ÔMEGA.

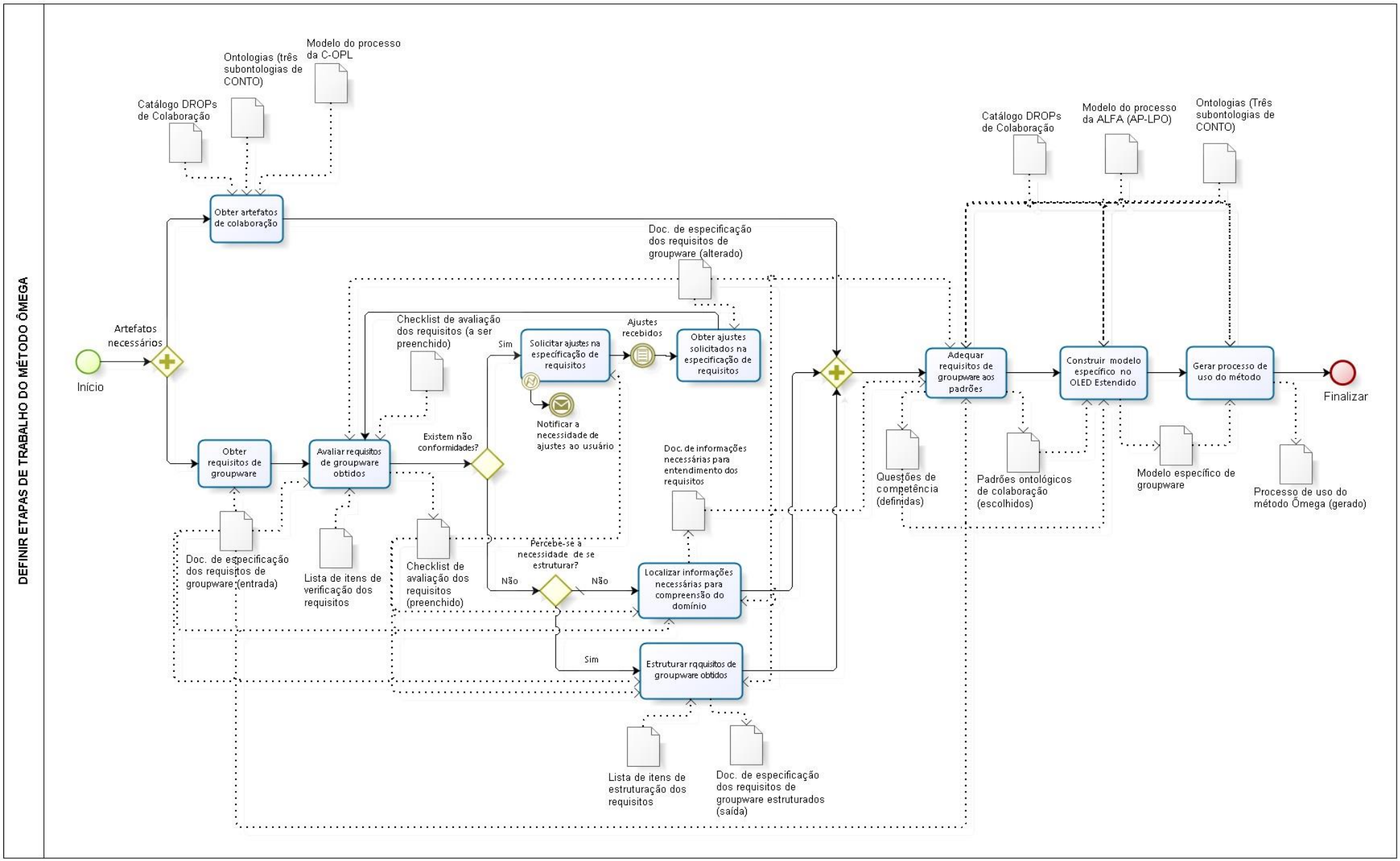


Figura 5: Processo de definição das etapas de trabalho do método ÔMEGA.

Fonte: Processo por meio da modelagem com BPMN idealizado pela autora.

5.2 OBTER ARTEFATOS DE COLABORAÇÃO

Nesta etapa do método ÔMEGA**,** devem ser obtidos os artefatos[28] de colaboração: o "**Catálogo de DROPs[29] de Colaboração**"(seção 4.10), a OPL que representa o "**Modelo de Processo da C-OPL**" (seção 4.9), e das "**Ontologias de Colaboração** (três subontologias de CONTO)"(seção 4.6). Neste caso, tais artefatos servem como aporte conceitual acerca do conhecimento capturado do domínio específico de colaboração. A Figura 5.1 ilustra uma representação abstrata da atividade de obtenção dos artefatos de colaboração.

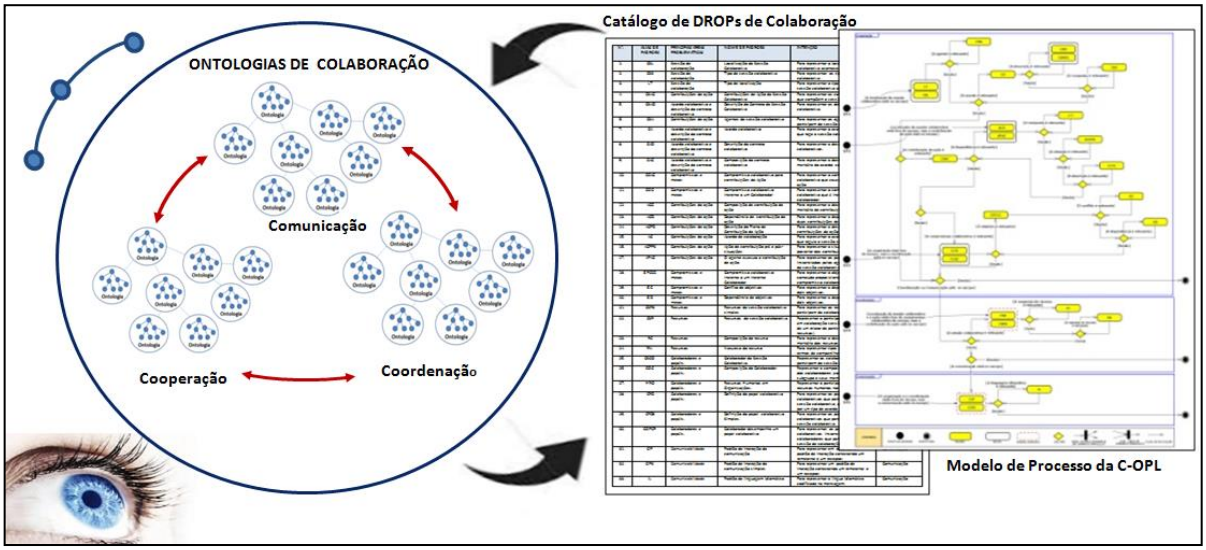


Figura 5.1 - Esboço da atividade de obtenção dos artefatos de colaboração.

Fonte: Idealizado pela autora.

O catálogo tem por objetivo final fornecer um conjunto de POs de domínio (DROPs) de Colaboração criado a partir da CONTO, a fim de apoiar modeladores na criação de novas modelagens. Já a C-OPL tem como objetivo guiar os modeladores na utilização dos DROPs de Colaboração presentes no catálogo, e as Ontologias de Colaboração (três subontologias de CONTO) buscam representar o conhecimento capturado do domínio de colaboração com os respectivos conceitos, relacionamentos, propriedades de domínio e um conjunto de axiomas que restringem o domínio.

[28] Artefato(s) - Termo que representa um conjunto de informações, um documento ou formulário, etc.

[29] DROPs - Apesar de esta dissertação ser escrita em português, será adotado o acrônimo DROP porque é um acrônimo conhecido para Padrões Ontológicos Relacionados ao Domínio em inglês.

5.3 OBTER REQUISITOS DE GROUPWARE

Nesta etapa do método ÔMEGA devem ser obtidos os requisitos do sistema *groupware* (ou uma família de aplicações colaborativas) a serem modelados. Nesta atividade, temos como artefatos de entrada o "**Doc. de especificação dos requisitos de groupware (entrada)**". Tal especificação de requisitos deve descrever aquilo que clientes (ou futuros usuários) necessitam e que será resolvido pela construção do *groupware*. Normalmente, tais requisitos são levantados na fase de análise dos requisitos, em que a atenção está centrada no *software*.

No entanto, o método é aplicado a partir de uma descrição de requisitos, cuja metodologia de captura e notação não é abordada no espectro de sua atuação. O método ÔMEGA parte do princípio de que já exista um documento de especificação dos requisitos de *groupwar*e construído. Vale lembrar que a captura de requisitos será normalmente executada por um usuário, o qual deve fornecer o documento inicial de requisitos; portanto, o ÔMEGA não possui uma fase de captura de requisitos, mas sugere a utilização de um modelo de estruturação de requisitos orientados a objetivos, com o intuito de facilitar a modelagem ontológica.

Salienta-se, então, que o método ÔMEGA está alicerçado em um modelo de estruturação de requisitos fundamentado nas abordagens de Engenharia de Requisitos orientadas a objetivos (em inglês, *Goal Oriented Requirements Engineering* ou GORE), com a utilização de cenários e preceitos da Teoria da Atividade. Por conseguinte, o modelo de estruturação de requisitos tem sua argumentação nas premissas de que clientes e usuários normalmente expressam suas necessidades através de objetivos almejados e que a ação humana deve ser analisada dentro de um contexto, para que possa fazer sentido e ser compreendida.

As abordagens da GORE têm como meta utilizar o conceito de objetivos para dar suporte às fases iniciais da engenharia de requisitos, incluindo as necessidades sociais e suas dependências. Por isso, possuem diversos frameworks e notações como: NFR Framework, I* *Framework*, KAOS, *Problem Frames*, UML, BMPN e Tropos. Eles representam as bases usadas no contexto da engenharia de requisitos para dar suporte aos analistas de requisitos. Além disso, é importante saber trabalhar com, pelo menos, um dos frameworks GORE, nas fases da engenharia de requisitos (concepção ou levantamento, análise e negociação,

especificação e/ou documentação e validação), como também elaborar os diagramas usando uma das notações.

5.4 ANALISAR REQUISITOS DE GROUPWARE

Nesta etapa do método ÔMEGA devem ser avaliados os requisitos de *groupware* obtidos, com o intuito de determinar se o estado dos requisitos é obscuro, incompleto, ambíguo ou contraditório. Em outras palavras, a avaliação tem por objetivo detectar a ocorrência de não-conformidades na especificação de requisitos. Nesta atividade, temos como artefatos de entrada o "**Doc. de especificação dos requisitos de groupware (entrada)**" e/ou o "**Doc. de especificação dos requisitos de groupware (alterado)**", a "**Lista de itens de verificação dos requisitos**" e o "**Checklist de avaliação dos requisitos (a ser preenchido**)". E como artefato de saída o "**Checklist de avaliação dos requisitos (preenchido)**".

Pode-se utilizar o *Checklist* para averiguar se existem não-conformidades na especificação de requisitos. Se for desejável, será possível imprimi-lo e ir marcando cada item, à medida que *os* requisitos forem sendo verificados, no decorrer de sua avaliação. Cabe ressaltar que os itens contidos na "**Lista de itens de verificação dos requisitos**" correspondem aos parâmetros que serão avaliados no "**Checklist de avaliação dos requisitos (A ser preenchido)**". A Figura 5.2, em sequência retrata, a visão geral dos itens que servem de base para a construção da "**Lista**" contidos no *Checklist* (vide apêndice B).

Figura 5.2: Diagrama da "Lista de itens de verificação dos requisitos".

Fonte: Idealizado pela autora.

→ A seguir, vamos explicar cada um dos itens da "**Lista de itens de verificação dos requisitos**" e como são compostos:

1. [**Verifique a existência de requisitos gerais**]: este item está relacionado com a existência dos requisitos gerais do cliente. Por exemplo, temos:

a) Os requisitos de identidade individual dos membros (papéis e responsabilidades de cada um);

b) Os requisitos de posição do usuário (nomes e/ou ícones visíveis na tela);

c) Os requisitos de visão pessoal ou padrão, particular, pública e convergente (uma visão especificada por cada usuário para um problema);

d) Os requisitos de conhecimento de outros integrantes do grupo (uma visão especificada, acerca das atividades dos integrantes que participam do grupo)

e) Os requisitos de controle da palavra ou capacidade necessária (algumas situações necessitam de somente um usuário manipulando um aplicativo ou emitindo opiniões);

f) Controle de acesso (direitos de acesso para cada usuário);

g) Atualização e sincronismo (para não haver divergências na edição e/ou na entrada e saída de dados);

h) Coordenação da informação (coordenação de atividades (ou grupo de atividades), aumentando a eficiência).

2. [**Verifique os tipos de requisitos existentes**]: este item está relacionado com a identificação dos tipos de requisitos existentes na aplicação *groupware* (funcionais e não funcionais):

- Os **requisitos funcionais** estabelecem funcionalidades, características ou ações que o sistema deve fornecer. Normalmente, podem ser visualizados com a ajuda de diagramas de caso de uso. Tais requisitos especificam ações que o sistema deve executar independente de exigências físicas ou tecnológicas e são associados ao modelo conceitual, isto é, o conjunto de necessidades do cliente que devem ser satisfeitas para se resolver um problema.
- Já os **requisitos não funcionais** também são características esperadas de determinado sistema, porém, não estão diretamente ligados a serviços específicos que o sistema oferece a seus usuários, podendo estar relacionados às propriedades emergentes do sistema. Normalmente, especificam (ou restringem) as características de um sistema como um todo, sendo mais críticos que os requisitos funcionais individuais.

Eles surgem das necessidades dos interessados, devido a restrições de orçamento, políticas organizacionais, necessidade de interoperabilidade com outros sistemas de *software* ou *hardware*, ou a partir de fatores externos, como regulamentos de segurança ou até legislações de privacidade. Eles descrevem atributos do ambiente do sistema, tais como: Extensibilidade; Usabilidade; Confiabilidade; Desempenho; Escalabilidade; Reusabilidade; Capacidade de manutenção; Reutilização de código; Performance; Eficiência no desenvolvimento; Confiabilidade nos dados apresentados.

A separação dos requisitos não funcionais em categorias hierarquizadas os dividem em três grandes grupos:

- **Requisitos de produto:** Especificam ou restringem o comportamento do sistema. Exemplos: a rapidez com que o sistema deve executar, quanta memória ele requer, taxa aceitável de falhas, proteção e confiabilidade;
- **Requisitos organizacionais:** São derivados das políticas e dos procedimentos da organização do cliente e do desenvolvedor. Exemplos: definição de como o sistema será usado, qual linguagem de programação será utilizada, o ambiente de desenvolvimento e o ambiente operacional do sistema;
- **Requisitos externos:** Derivam de processos externos ao sistema. Exemplos: especificação de sistema por parte do governo, requisitos legais, dentre outros.

3. [**Verifique a existência de requisitos de cliente**]: este item está relacionado com a existência de requisitos estabelecidos pelo cliente, pela empresa e pelas partes interessadas, que devem ser envolvidas nesse processo para entender o que o produto deve fazer (compreender o contexto do sistema).

4. [**Verifique a existência de requisitos de usuário**]: este item está relacionado com a existência de requisitos estabelecidos para os usuários, acesso aos objetos compartilhados e às ferramentas de colaboração. Por exemplo, temos:

a) Auxílio na escolha das ferramentas apropriadas;
b) Fornecimento de informações de percepção;
c) Colaboração síncrona e assíncrona;
d) Acesso ao ambiente independente da estação de trabalho;
e) Fornecimento de espaço privativo e público e transição entre eles;
f) Extensão dinâmica do ambiente;
g) Sincronização entre ferramentas diferentes; Mobilidade;
h) Agrupamento de ferramentas;
i) Alta performance.

5. [**Verifique a existência de requisitos de desenvolvedor**]: este item está relacionado com a existência de requisitos estabelecidos para desenvolvedores (experiência e conhecimento anteriores). Por exemplo, temos:

a) Aproveitamento do modelo de dados;

b) Compartilhamento transparente de dados;

c) Suporte a dados locais e compartilhados;

d) Acesso às informações de percepção;

e) Disponibilização de novas ferramentas;

f) Integração com ferramentas externas;

g) Suporte a ferramentas localizadas no servidor;

h) Escalabilidade.

6. [Averigúe as restrições do projeto]: este item está relacionado com a existência de restrições que são condições (ou situações) que limitam o planejamento, desenvolvimento e não podem ser eliminadas (ou alteradas no decorrer do projeto). Por exemplo, temos:

a) Dentre as limitações que nos cercam, algumas são auto-impostas e outras o são pela organização e pela sociedade.

b) Algumas limitações são impostas pela equipe do projeto, pois podem afetar o desempenho. Tais limitações devem ser respeitadas, devidamente analisadas e tratadas.

7. [**Verifique a existência de requisitos conflitantes**]: este item está relacionado com a existência de requisitos conflitantes[30] ou irreais (de desenvolvimento impossível). Mesmo assim, a revisão para validação de requisitos pode ser conduzida, desde que possibilite a descoberta de erros nos requisitos. Neste caso, é frequentemente útil examinar cada requisito em face de um conjunto de questões do tipo *checklist*.

8. [Solicite tratamento das não-conformidades]: este item está relacionado com o pedido de tratamento das não-conformidades identificadas na especificação de requisitos para que sejam mantidas, alteradas ou adaptadas pelo usuário (projetista de sistemas).

5.5 SOLICITAR AJUSTES NA ESPECIFICAÇÃO DE REQUSITOS

Nesta etapa do método ÔMEGA, devem-se recomendar ações corretivas por parte do usuário na especificação de requisitos, pois foram identificadas não conformidades nos resultados da avaliação dos requisitos. Nesta atividade, temos como artefato de entrada o "**Checklist de avaliação dos requisitos (preenchido**)" que contém os resultados da avaliação e um fluxo de exceção para disparo de um email de notificação para o usuário.

[30] Conflitante - Que não é compatível com; que vai de encontro a; em que há conflito; contrário ou incompatível.

5.6 OBTER AJUSTES SOLICITADOS NA ESPECIFICAÇÃO DE REQUISITOS

Nesta etapa do método ÔMEGA, devem ser obtidos os ajustes solicitados no documento de especificação de requisitos. Cabe destacar que o usuário deve encaminhar a correção devida para uma nova avaliação quanto a sua conformidade. Nesta atividade, temos como artefato de entrada a "**Doc. de especificação dos requisitos de groupware (alterado)**" e o "**Checklist de avaliação dos requisitos (preenchido)**" que contém os resultados da avaliação.

5.7 LOCALIZAR INFORMAÇÕES NECESSÁRIAS PARA COMPREENSÃO DO DOMÍNIO

Nesta etapa do método ÔMEGA devem ser localizadas as informações necessárias para compreender o contexto dos requisitos do domínio, pois não existem não conformidades identificadas através dos resultados da avaliação. Apesar disso, percebe-se que não existe a necessidade de se estruturar os requisitos, uma vez que se identifica um levantamento de requisitos adequado, por meio da boa definição do projeto e do perfeito diagnóstico obtido. Neste caso, as informações necessárias para compreender o contexto dos requisitos do domínio foram obtidas através do documento de especificação dos requisitos.

Nesta atividade, temos como artefatos de entrada o "**Doc. de especificação dos requisitos de groupware (entrada)**" e/ou o "**Doc. de especificação dos requisitos de groupware (alterado)**" e o "**Checklist de avaliação dos requisitos (preenchido)**", que contém os resultados da avaliação. E como artefato de saída, o "**Doc. de informações necessárias para entendimento dos requisitos**", que deve conter informações necessárias para compreensão do contexto do domínio dos requisitos de software.

Na Figura 5.3 ilustramos os elementos que devem estar contidos na "**Doc. de informações necessárias para entendimento dos requisitos**".

Figura 5.3: Lista de elementos contidos Doc. de informações necessárias.

Fonte: Idealizado pela autora.

→ Em seguida, vamos explicar cada um dos elementos e como são compostos:

1. [Descrição dos requisitos]: aqui, é preciso descrever os requisitos do contexto de *groupware* com base nas necessidades, a fim de se evitarem desperdícios, eliminando-se as não conformidades e os principais problemas, gargalos ou inventários desnecessários. Tal

descrição dos requisitos é facilitada já que os requisitos dos *Stakeholders* são frequentemente revelados pelo processo de reflexão em que as perguntas **"**Como?", **"**Por quê?", **"**De que outra forma?" serão respondidas, produzindo requisitos com menor risco de falta ou excesso de especificação.

2. [Descrição das intenções do usuário]: é preciso descrever as intenções do usuário através da analise dos atores, pois identificamos que o usuário é quem tem objetivos a serem alcançados porque é o agente que tem os objetivos a serem supridos. Neste caso, tais requisitos devem ser classificados por tipos (U = Usuário; O = Objetivo; T = Tarefa e M=Meta).

3. [Descrição das interações dos agentes]: é necessário descrever as interações entre agentes, através da análise dos agentes do domínio, pois há pelo menos um que representa o *software* (sistema). Cabe esclarecer que pode haver vários agentes de *software*s representando diferentes componentes na visão dos clientes e usuários. Neste caso, tais requisitos devem ser classificados por tipos (U = Usuário; O = Objetivo; T = Tarefa e M=Meta.).

4. [Identificação de conflitos]: caso existam conflitos, é importante conhecê-los, saber qual é sua amplitude e como trabalhar com eles. Neste contexto, os conflitos estão focados em problemas entre pares de requisitos não funcionais, o que significa que os dois objetivos não podem ser satisfeitos simultaneamente. A modelagem orientada a objetivos pode identificar interesses divergentes entre as partes interessadas e identificar conflitos, já que o cumprimento de um objetivo pode interferir no cumprimento de outros.

5. [Fragmentação dos objetivos]: caso necessário, deve-se fragmentar os objetivos para alcançar as metas, pois a fragmentação permite que objetivos mais complexos sejam divididos em objetivos mais simples, facilitando a análise e a realização dos mesmos. Por exemplo, se o objetivo é passar em um concurso, as metas podem ser: **realistas**, mas **acomodadas**: estudar uma vez por semana; **realistas**, mas **desafiadoras**: estudar por, pelo menos, uma hora todos os dias; **não realistas** ou superestimadas: estudar durante oito horas todos os dias.

6. [Elabore as questões de competência]: com os requisitos identificados, é preciso elaborar as primeiras perguntas de competência (ou questões de competências) informais. Feito isto, deve-se transformar as questões de competências informais identificadas em formais, por

meio de reconhecimento de linguagens, no sentido amplo da palavra, suas classificações, estruturas, propriedades, características e inter-relacionamentos. Por exemplo, a questão de competência informal "O local onde é ou ocorre o projeto?" é transformada em formal "Onde ocorre a sessão colaborativa?".

7. [Responda as questões de competência]: a partir de um conjunto de perguntas de competência (ou questões de competência), deve-se responder às mesmas, com base no contexto dos requisitos das partes interessadas, relevantes em conformidade com os requisitos aplicáveis e compromissos estabelecidos. Então, deve-se fazer uso das respostas às questões de competência para orientar na escolha de quais padrões ontológicos da CONTO serão utilizados para a modelagem a ser desenvolvida.

5.8 ESTRUTURAR OS REQUISITOS DE GROUPWARE OBTIDOS

Nesta etapa do método ÔMEGA, devem ser organizadas e capturadas as informações necessárias para compreender o domínio no contexto dos requisitos, pois não existem não conformidades identificadas através dos resultados da avaliação. Apesar disso, percebe-se que existe a necessidade de se estruturarem os requisitos, pois identifica-se um levantamento de requisitos inadequado; o resultado é um diagnóstico pobre com conclusões comprometidas, não identificação das causas dos problemas, custos elevados, prazos vencidos ou comprometedores e omissão de processos fundamentais.

Neste contexto, deve-se obter as informações necessárias fazendo uso do modelo de estruturação que foi concebido segundo os conceitos das abordagens orientadas a objetivos com a utilização de cenários e preceitos da Teoria da Atividade. Nesta atividade, temos como artefatos de entrada o **“Doc. de especificação dos requisitos de groupware (entrada)”** e/ou o **“Doc. de especificação dos requisitos de groupware (alterado)”**, o “**Checklist de avaliação dos requisitos (preenchido)”**, que contém os resultados da avaliação e a ”**Lista de itens de estruturação dos requisitos**”, que contém os itens de estruturação**.** E, como artefato de saída, o **“Doc. de especificação dos requisitos de groupware estruturados (saída)**”, que deve descrever o contexto do qual foram extraídas as informações relevantes sobre os requisitos de *software*.

Caso seja necessário estruturar os requisitos, deve-se aplicar o modelo de estruturação orientado a objetivos concebido abaixo. Nesse contexto, a estrutura do modelo de estruturação conta com uma lista de itens para nortear equipes na identificação de

informações relevantes sobre o contexto dos requisitos do domínio de *software*. Na Figura 5.4 ilustramos os itens que estão contidos na **"Lista de itens de estruturação dos requisitos**".

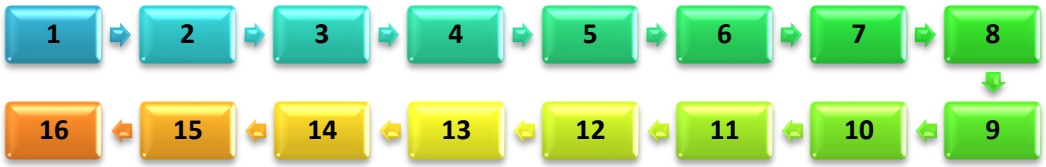


Figura 5.4: Lista de itens de estruturação dos requisitos.

Fonte: Idealizado pela autora.

→ A seguir, vamos explicar cada um dos itens da **"Lista de itens de estruturação dos requisitos**" e como são compostos:

1. [Descubra os objetivos]: este item está relacionado com a identificação dos objetivos que correspondem àquilo que se pretende alcançar (que responde ao "por quê", ao "como", ao "o quê" e ao "de que outra forma"), que proporciona descrições de alto nível, funcionais e não funcionais:

- os objetivos são capturados ao se entrevistar as partes interessadas ou com base na experiência de um engenheiro de requisitos;
- deve-se refinar e abstrair objetivos, pois o refinamento de objetivos é o processo no qual os objetivos são partidos em subobjetivos. O objetivo principal pode ser refinado em subjetivos até modelar os requisitos, que são os objetivos mais básicos, na hierarquia do diagrama de objetivos, sempre associados aos agentes;
- deve-se utilizar de objetivos específicos para capturar as intencionalidades do sistema em diferentes níveis de abstração, a fim de desencadear, elaborar, estruturar, especificar, analisar, negociar, documentar e modificar requisitos para serem efetivamente alcançados;
- pode-se encontrar novos objetivos e cenários com base na observação dos obstáculos de objetivos já descobertos. Adicionalmente, os agentes precisam atingir determinados objetivos para que os objetivos de mais alto nível sejam satisfeitos.

Nota: Objetivo e meta são diferentes entre si. Objetivo é a descrição daquilo que se pretende alcançar. A meta é a definição em termos quantitativos e com um prazo determinado. Por exemplo, o objetivo de uma determinada pessoa é emagracer 40 quilos em um ano. E a meta é emagrecer 33,33% quilos por mês. A relação entre objetivos e metas

é que o objetivo pode ser definido como o lugar onde se deseja chegar e a meta o caminho a ser percorrido para chegar no lugar desejado.

2. [Refine os objetivos]: este item está relacionado com os refinamentos (ou conectores) que relacionam os objetivos com os subobjetivos E (AND) e OU (OR). Tais conectores, representam a decomposição dos objetivos e definem as soluções alternativas para o cumprimento do superobjetivo;

- deve-se avaliar a estrutura do refinamento AND (E)/OR (OU) dos objetivos. Os refinamentos E (AND) são a única condição suficiente para satisfazer ao objetivo e a todos os subobjetivos. E os refinamentos OU (OR), para satisfazer a um dos subobjetivos, e são condição suficiente para satisfazer ao objetivo;
- o refinamento dos objetivos deve parar quando as descrições dos objetivos não mais se referem às propriedades do domínio.

3. [Operacionalize objetivos e responsabilidades]: este item está relacionado à preparação dos objetivos (que responde ao "Quando?") e das atribuições de responsabilidade (que responde ao "Quem?"); traduzem os objetivos, requisitos definidos em objetos (que responde ao "O quê?") que vão compor o sistema, bem como suas operações e comportamentos;

- entende-se por requisito tudo aquilo que deve ser feito no sistema e que compõe o escopo do sistema. Neste ponto, é especificado o domínio da solução e seus limites como o ambiente (domínio do problema), através da identificação dos agentes e de atribuição de suas responsabilidades. E também do estabelecimento das pré e pós condições das operações de identificação dos objetos envolvidos com atendimento aos requisitos.

4. [Descubra as metas]: este item está relacionado com os objetivos, pois as metas são tarefas específicas para alcançá-los, sendo que elas são temporais, ou seja, estipulam prazos;

- deve-se refinar e abstrair metas, pois o refinamento de metas é o processo, no qual as metas são partidas em submetas. Deve-se subdividir as metas em submetas para que as metas fiquem menos comprometidas, haja vista que as submetas são as responsáveis por auxiliar no alcance das metas.

5. [Refine as metas]: este item está relacionado com os refinamentos (ou conectores) que relacionam os metas com os submetas E (AND) e OU (OR). Tais conectores representam a

decomposição das metas e definem as soluções alternativas para o cumprimento da super-meta;

- deve-se avaliar a estrutura do refinamento AND (E)/OR (OU) das metas. Os refinamentos E(AND) são a única condição suficiente para satisfazer à meta e a todas as submetas. E os refinamentos OU (OR), para satisfazer uma das submetas, e são condição suficiente para satisfazer à meta;
- o refinamento das metas deve parar quando as descrições das metas não mais se referirem às propriedades do domínio.

6. [Operacionalize metas e responsabilidades]: este item aborda o estabelecimento de metas e a operacionalização do seu alcance pelos colaboradores participantes, a fim de que se obtenham resultados que satisfaçam às necessidades de seus clientes, assim como às suas próprias. Uma organização é constituída por pessoas que querem algo maior, que não podem conseguir sozinhas. A união dessas pessoas que visam os mesmos ideais, forma a organização. Além de objetivos individuais, passam a ter objetivos conjuntos; para isto, as pessoas aceitam responsabilidades e têm metas determinadas, que são etapas que devem ser seguidas para se alcançar o objetivo maior definido.

7. [Descubra os requisitos]: este item está relacionado com os requisitos que são um tipo especial de objetivos que têm certas restrições na utilização dos valores monitorizados e controlados. Por exemplo, a fase de requisitos é normalmente dividida em levantamento de requisitos (recolha, compreensão, revisão e articulação das necessidades dos *stakeholders*), análise (modelação, verificação de consistência e completude), especificação de requisitos (documentação e ou modelação dos requisitos) e validação de requisitos (garantir que os requisitos especificados estão corretos, de um ponto de vista interno e externo).

8. [Refine os requisitos]: este item está relacionado com os refinamentos (ou conectores) que relacionam os requisitos com os subrequisitos E(AND) e OU(OR). Tais conectores representam a decomposição dos requisitos e definem as soluções alternativas para o cumprimento do super-requisito;

- o refinamento dos requisitos deve parar quando as descrições dos requisitos não mais se referirem às propriedades do domínio.

9. [Operacionalize os requisitos e responsabilidades]: este item aborda o estabelecimento de atribuições de responsabilidades através da operacionalização (estática e a dinâmica) dos

requisitos. A operacionalização estática expressa a necessidade da utilização de dados para que um requisito não funcional seja atendido ou, mesmo, para atender soluções de projeto. A operacionalização dinâmica representa as ações que devem ser implementadas para que os objetivos sejam atingidos.

10. [Identifique os cenários]: este item aborda a técnica de cenários que descreve como componentes de sistemas e seus usuários interagem para fornecer uma funcionalidade nivelada do sistema, de maneira a colaborar com a definição dos objetivos ou metas. Os cenários exemplificam as descrições comportamentais de um sistema e seu ambiente em situações restritas, permitindo que necessidades escondidas sejam descobertas. Eles deixam implícitos propriedades, requisitos e objetivos;

- deve ser definido como uma sequência temporal de interações de eventos entre diferentes agentes em um contexto restrito para obtenção de um propósito explícito. O fundamento desta avaliação é que os cenários devem ser utilizados como um mecanismo de obtenção de exemplos e ilustrações capazes de auxiliar a operacionalizar os objetivos.

11. [Identifique os agentes]: este item está relacionado aos agentes, que são definidos como humanos, programas ou componentes automatizados, que têm, como responsabilidade alcançar as expectativas e os requisitos.

12. [Identifique as atribuições e responsabilidades dos agentes]: este item está relacionado com as atribuições e responsabilidades descritas para cada agente, os requisitos e expectativas pelos quais ele é responsável ou que tenham sido atribuídos a ele. Além de se conectarem aos requisitos, os agentes/atores também se conectam às expectativas. Uma expectativa é uma meta alcançada por um agente que faz parte do ambiente do sistema;

- analisando os agentes, identificamos que o (agente) usuário é quem tem objetivos a serem alcançados porque é o agente que tem objetivos a serem supridos. Os requisitos sobre os agentes que interagem com o sistema de *software* a ser desenvolvido são conhecidos como expectativas no modelo de meta. Eles são introduzidos para mostrar como o sistema de *software* e seu ambiente devem cooperar para atingir as metas do SI.

13. [Identifique as atividades]: este item está relacionado com a Teoria da Atividade. A atividade é o contexto mínimo no qual a ação humana deve ser analisada para que possa fazer sentido e ser compreendida. A atividade é, então, a unidade básica de análise de situações. Esse conceito de atividade implica a existência de um agente que atua sobre

alguma coisa, de maneira que a atividade media a interação entre sujeitos (agentes) e objetos (coisas);

- uma atividade é decomposta em ações, e cada ação é decomposta em operações. Uma atividade é realizada através de ações cooperativas (ou individuais), podendo se estabelecerem cadeias (ou redes de ações) relacionadas umas com as outras por buscarem atingir a mesma meta. Uma a atividade é orientada por um motivo, as ações são orientadas a metas e as operações, orientadas a condições;
- os preceitos da Teoria da Atividade podem contribuir para elicitação de requisitos baseados em cenários. A análise das relações sistêmicas existentes no contexto de uma atividade contribui para uma elicitação de requisitos mais cuidadosa, pois leva quem está realizando a elicitação de requisitos a considerar elementos importantes para um bom entendimento do problema, tais como: **sujeito**, **ferramenta de mediação**, **objeto**, **regras**, **comunidade** e **divisão de trabalho**.

14. [Identifique os conflitos]: este item está relacionado com a identificação dos conflitos que estão focados em problemas entre pares de requisitos não funcionais, o que significa que os dois objetivos não podem ser satisfeitos simultaneamente. É particularmente importante identificar conflitos, o mais cedo possível, no ciclo de vida do software (melhor ainda se isso ocorrer durante o levantamento dos requisitos).

15. [Identifique obstáculos]: este item está relacionado com a identificação de obstáculos de diversas maneiras; uma delas é considerar falhas nos componentes ou no comportamento de pessoas que podem levar a caminhos inesperados. Os obstáculos são as situações em que uma meta é violada. Nesse caso, diz-se que o obstáculo "obstrui" a meta, requisito ou a expectativa. O princípio da análise de obstáculos é ter uma visão pessimista do modelo elaborado, considerando como o sistema pode se desviar do modelo;

- uma maneira de corrigir um obstáculo é definir novos requisitos que impeçam que o obstáculo ocorra. O tratamento de obstáculos permite que os analistas identifiquem e localizem as circunstâncias excepcionais durante a engenharia de requisitos, a fim de produzir, por exemplo, requisitos robustos, ou até mesmo novos requisitos, para evitar ou reduzir o impacto dos obstáculos, resultando num *software* de confiança.
- Uma segunda maneira de corrigir um obstáculo é restaurar o objetivo obstruído, uma vez que o obstáculo ocorra. Os obstáculos impedem que os objetivos de um sistema sejam

alcançados. Lidar com obstáculos é muito importante para a construção de sistemas seguros e robustos.

16. [**Priorize os requisitos**]: este item está relacionado com as prioridades, que podem ter nomes distintos, de acordo com a literatura, mas geralmente, quando falamos de priorização de requisitos são utilizados os termos essencial, importante e desejável;

- o requisito essencial é fundamental para o sistema, sem o qual o sistema não pode ser dado como "completo", ou "apto para produção". Se os requisitos essenciais não são implementados, impedem uma implantação ou a conclusão do sistema. São compulsórios, não sendo possível aplicar soluções de contorno ou paliativos para eles;
- o requisito importante deve ser parte do escopo, mas não bloqueia o sistema a entrar em produção**.** É como se o sistema ficasse com uma "pendência" de escopo - criando débito técnico - que será atendido em momento oportuno. Sem um requisito importante, o sistema poderá rodar, funcionar, ser utilizado. Pode ser simplesmente postergado para pós-implantação ou ser atendido temporariamente por soluções de contorno ou por paliativos.
- o requisito desejável não é indispensável para o sistema estar completo, para entrar em produção. Também não é algo que, mesmo postergado, deverá ser feito obrigatoriamente. Sem um requisito desejável, o sistema deve funcionar de maneira satisfatória, atendendo completamente seu objetivo. Por ser algo que não precisa ser feito para que o sistema esteja completo, é a menor das prioridades e deve ser postergado para, se possível, ser viabilizado no futuro.

17. [**Elabore as questões de competência**]: com os requisitos identificados, deve-se elaborar as primeiras perguntas de competência (ou questões de competências) informais. Feito isto, é necessário transformar as questões de competências informais identificadas em formais, através de reconhecimento de linguagens, no sentido amplo da palavra, suas classificações, estruturas, propriedades, características e inter-relacionamentos.

18. [**Responda as questões de competência**]: a partir de um conjunto de perguntas de competência (ou questões de competência), deve-se responder às mesmas, com base no contexto da modelagem de objetivos em conformidade com os requisitos aplicáveis e compromissos estabelecidos. Então, deve-se fezer uso das respostas às questões de competência para orientar na escolha de quais padrões ontológicos da CONTO serão utilizados para a modelagem a ser desenvolvida.

Destaca-se que um modelo de estruturação orientado a objetivos é considerado completo se todos os objetivos do sistema estão decompostos para o maior detalhamento de suas tarefas e relações. Ademais, os devidos objetivos devem estar delegados aos seus respectivos atores, com suas respectivas tarefas e relações. A composição do modelo baseou-se na definição dos porquês do *software* e na identificação dos requisitos que representam a estrutura de objetivos, metas, requisitos, agentes, cenários, atividades e suas associações, que foram aglutinados na forma de itens na "**Lista de itens de estruturação dos requisitos**", pois as razões de um *software* para a organização para a qual ele é requerido é que de terminam as diretrizes do desenvolvimento (PÁDUA; CAZARINI, 2004).

O entendimento das razões para o desenvolvimento do *software* é importante, pois são elas que determinam as diretrizes de desenvolvimento, auxiliam na definição do sistema, facilitam a cooperação com outros sistemas, bem como sua evolução, dado que uma definição confiável dos requisitos de um *software* depende diretamente da completa e correta compreensão das necessidades do sistema e, consequentemente, de sua representação, de forma adequada ao processo de desenvolvimento. Para tal, o modelo de estruturação definido neste trabalho tem por objetivo prover a construção gradual do entendimento compartilhado entre os envolvidos sobre o domínio do problema e da solução, na concepção e no desenvolvimento do *groupware*.

Convém salientar que abordagens orientadas a objetivos operacionalizadas através de cenários, conjugadas a princípios da Teoria da Atividade oferecem um suporte adequado à estruturação de requisitos, provendo usabilidade ao modelo estruturado. Os cenários são utilizados para descrever as situações de uso do sistema pelos seus usuários e os relacionamentos entre o sistema em desenvolvimento e outros sistemas externos, auxiliando no entendimento e na descoberta de novos requisitos. Já os preceitos da Teoria da Atividade contribuem para análise das relações sistêmicas existentes no contexto de uma atividade, pois levam quem está realizando a elicitação a considerar elementos importantes para um bom entendimento do problema, tais como: sujeito (usuário(s) ou grupo(s)), ferramenta de mediação, objeto, regras, comunidade e divisão de trabalho.

Na Figura 5.5, exemplificamos os elementos mais importantes para um bom entendimento do problema. Tal entendimento é a condição necessária para se evitar o desperdício de recursos e não deve ser negligenciado.

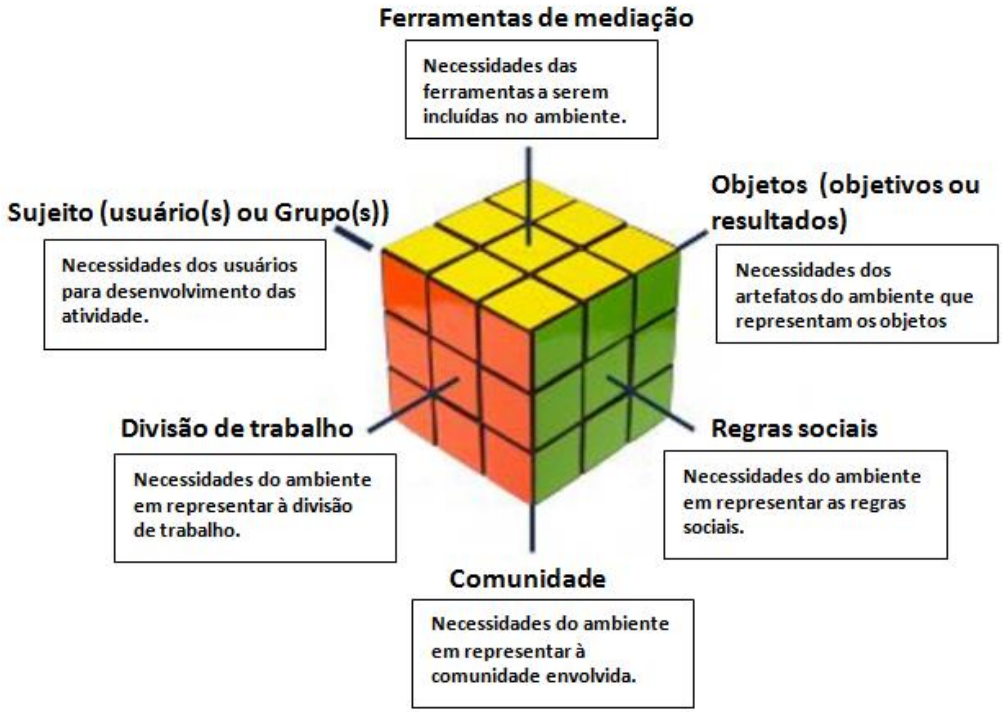


Figura 5.5: Elementos para um bom entendimento do problema.

Fonte: Idealizado pela autora.

É incontestável a importância da definição de requisitos, pois uma definição precisa do que produzir é a condição determinante para a satisfação do(s) cliente(s) e usuário(s). No entanto, uma das maiores dificuldades da engenharia de requisitos (ER) reside na complexidade da própria atividade. Por isso, almejamos uma compressão correta e/ou completa para atender às necessidades, a fim de apoiar as restrições que as organizações têm.

5.9 EXECUTAR O SUBPROCESSO ADEQUAR OS REQUISITOS DE GROUPWARE AOS PADRÕES

Nesta etapa do método ÔMEGA, devem ser adequados os requisitos de *groupware* obtidos frente aos padrões ontológicos de colaboração, com o propósito de moldar os requisitos de *groupware* obtidos para a concepção de modelos conceituais em ONTOUML. Neste caso, utilizaremos como aporte conceitual o "**Catálogo de DROPs de Colaboração**", a OPL que representa o "**Modelo de Processo da C-OPL**" e as "**Ontologias de Colaboração** (três subontologias de CONTO)".

Nesta atividade, temos como artefatos de entrada o "**Doc. de especificação dos requisitos de groupware (entrada)"** e/ou o **"Doc. de especificação dos requisitos de**

groupware (alterado)", o **"Doc. de informações necessárias para entendimento dos requisitos**" e/ou o "**Doc. de especificação dos requisitos de groupware estruturados (saída)**". E, como artefato de saída, o "**Questões de competência (definidas)"** e o **"Padrões ontológicos de colaboração (escolhidos)**".

Para se adaptarem os requisitos de *groupware*, devem ser respondidas as perguntas de competência (ou questões de competência), com base no contexto dos requisitos das partes interessadas relevantes, em conformidade com os requisitos aplicáveis e os compromissos estabelecidos. Então, deve-se fazer uso das respostas às questões de competência para se orientar na escolha de quais padrões ontológicos usar na modelagem conceitual ontológica. Os POs de domínio (DROPs) são muito inter-relacionados, para funcionar como um todo, de modo que a obtenção deles, ou de parte deles, é feita de maneira integrada (um PO por vez).

5.10 CONSTRUIR MODELO ESPECÍFICO NO OLED

Nesta etapa do método ÔMEGA, deve ser ilustrado o processo de construção de modelos conceituais ONTOUML específicos de *groupware,* baseados em padrões ontológicos de colaboração usando o módulo OLED Estendido implementado no OLED para apoiar modeladores ontológicos iniciantes. Uma vez que os requisitos do sistema tenham sido identificados, o projetista de sistemas estará preparado para esboçar a modelagem ontológica do *groupware* a ser modelado.

Nesta atividade, temos como artefatos de entrada o "**Catálogo de DROPs de Colaboração**", a OPL que representa o "**Modelo de Processo da C-OPL**" e as "**Ontologias de Colaboração** (três subontologias de CONTO)", o "**Doc. de especificação dos requisitos de groupware (entrada)**" e/ou o "**Doc. de especificação dos requisitos de groupware (alterado)**", o "**Doc. de informações necessárias para entendimento dos requisitos**" e/ou o "**Doc. de especificação dos requisitos de groupware estruturados (saída)**". E como artefato de saída o "**Modelo específico de groupware**".

No método ÔMEGA, podemos construir modelos conceituais ontológicos específicos de *groupware* em **duas visões distintas**: ☞ **visão a partir do zero**; e **visão de manutenção**. A "**visão a partir do zero**" é usada quando não se tem nada modelado. Já a "**visão de manutenção**" é usada quando a modelagem está bastante adiantada ou se tem

um conjunto de padrões selecionados. Ambas as visões adotam a verificação sintática do OLED como como uma maneira de testar a viabilidade dos POs de domínio (DROPs) de colaboração na geração de fragmentos de modelagem ontologicamente adequados. Quando o conjunto de padrões estiver pronto (ou,ao menos, bastante adiantado), os padrões criados devem ser validados.

Os modelos conceituais ontológicos no ÔMEGA foram subdivididos em **dois procedimentos distintos**: ☞ **procedimentos introdutórios do processo de uso**; e **procedimentos avançados do processo de uso.** O primeiro procedimento equivale a um giro por vez, no cubo mágico (um passo ou ação por vez) para a construção de modelos específicos de *groupware,* por parte dos novatos no OLED. Já o segundo equivale a dois ou mais giros em sequência, por vez, no cubo mágico (passos ou ações repetidas por vezes) para construção de modelos específicos de *groupware,* por parte dos modeladores já iniciados no OLED.

A Figura 5.6 apresenta os oito passos que correspondem aos **procedimentos introdutórios** do processo de uso do método ÔMEGA para a construção de modelos específicos de *groupware* no OLED para novatos.

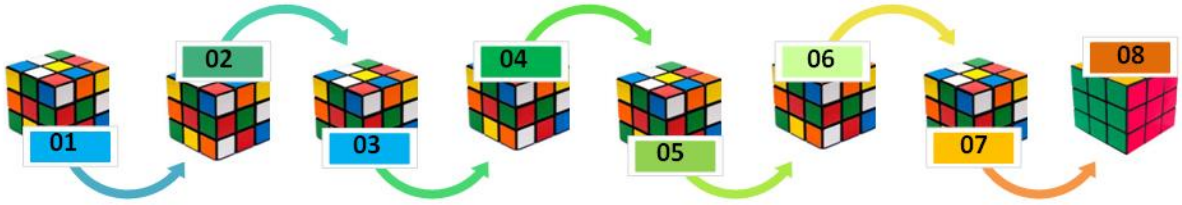


Figura 5.6: Oito passos para construção de modelo conceitual ONTOUML.

Fonte: Idealizado pela autora.

☞ **Procedimentos introdutórios do processo de uso:** equivale a um giro, por vez, no cubo mágico (um passo ou ação por vez).

1. [Carregue o catálogo de DROPs no OLED]: com o catálogo DROPs carregado no OLED estendido, o modelador tem acesso aos POs de domínio (DROPs) do catálogo para construção de modelos conceituais de *groupware*. Neste caso, carregar e importar têm o mesmo sentido, que é o de disponibilizar os DROPs do catálogo para uso no OLED. Em outras palavras, o catálogo deve ser importado no OLED para a construção de ontologias operacionais de domínio específico de colaboração (na seção 4.11).

2. [Compreenda a realidade do domínio]: para que a modelagem conceitual ontológica possa ser uma descrição adequada da realidade do domínio, ela deve apresentar informações precisas e claras, que podem ser obtidas a partir de uma análise mais detalhada das propriedades dos objetos de um domínio e dos termos que o denotam, possibilitando mitigar falhas semânticas e sintáticas, eliminando ambiguidades e contribuindo com um entendimento uniforme da área do problema em questão.

3. [**Delimite o escopo da modelagem**]: o escopo da modelagem se refere àquilo que se pretende atingir, ou seja, é o que delimita o que será feito no esboço do projeto, uma vez que um modelo conceitual baseado em ontologias deve estar apto a instanciar as informações do mundo real, de modo que possa responder perguntas ou recuperar informações rapidamente. Os modelos conceituais gerados não necessitam cobrir todo universo de discurso, mas a parte requisitada para a sua representação (desejável).

4. [**Organize o conhecimento do domínio**]: o conhecimento do domínio descreve a informação estática principal e os objetos de conhecimento do domínio da aplicação. Neste âmbito, é preciso identificar as necessidades e as expectativas (ou seja, os requisitos) das partes interessadas. Feito isso, deve-se organizar o conhecimento do domínio para que a modelagem conceitual seja fiel aos dados levantados e para que os modelos tenham qualidade, e mais precisamente representem a realidade.

5. [**Mapeie seus conceitos fundamentais**]: tenha sempre em mente que um *groupware* normalmente oferece ao participante um ambiente com um conjunto de ferramentas colaborativas usadas em diferentes momentos da colaboração (voltados para o trabalho em grupo) e que proporcionam uma interface única para o ambiente compartilhado. Neste caso, os conceitos fundamentais (ou principais características) funcionam como uma base conceitual comum de compreensão mútua para instrumentar a modelagem conceitual ontológica do *groupware* a ser modelado.

6. [**identifique seus DROPs de Colaboração**]: por meio dos conceitos fundamentais (ou principais características), deve-se identificar as questões de competência (ou perguntas de competência), com o intuido de detectar quais POs de domínio (DROPs) serão utilizados no desenvolvimento do novo modelo, considerando-se que os POs são inter-relacionados para funcionar como um todo, de modo que a obtenção deles, ou de parte deles, é feita de maneira integrada.

7. [Encadeie seu conhecimento]: no aspecto da modelagem é interessante destacar o encadeamento do conhecimento entre as OPLs, POs e ontologias, uma sendo construída com o suporte oferecido pela outra, começando da última para a primeira. Em outras palavras, a construção de um modelo é um processo incremental, isto é, um modelo de um sistema não é construído em um único passo, mas em muitos passos pequenos, muitas pequenas transformações do modelo inicial até o modelo completo.

8. [**Delineie seu modelo conceitual ontológico**]: na construção do modelo, deve-se fazer uso dos artefatos de colaboração, pois os mesmos servirão de guia na construção do modelo conceitual específico de *groupware*. No entanto, o modelador deve ter a percepção de que um modelo é feito em partes, às quais os POs de domínio (DROPs) são adicionados, de forma incremental ao modelo (um por vez). Com o apoio do OLED, através de seu módulo OLED Estendido, interligar os POs do catálogo DROPs de colaboração na construção de modelos conceituais ontológicos se torna mais prático, ágil e sistemático.

A Figura 5.7 apresenta, os cinco passos que correspondem aos **procedimentos avançados** do processo de uso do metodo ÔMEGA para construção de modelos específicos de *groupware* no OLED para modeladores já iniciados.

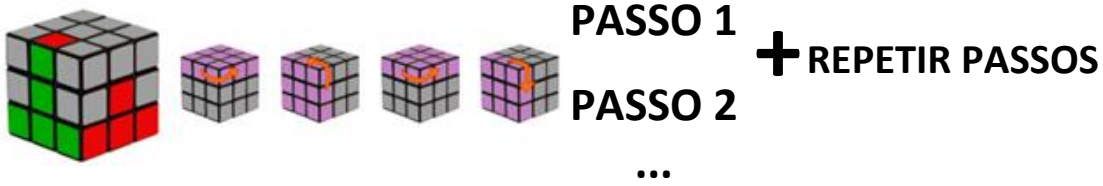


Figura 5.7: Os passos avançados para construção de modelo conceitual ONTOUML.

Fonte: Idealizado pela autora.

☞ **Procedimentos avançados do processo de uso:** equivale a dois ou mais giros em sequência, por vez, no cubo mágico (passos ou ações repetidas por vezes).

1. [**Verifique o contexto da nova modelagem**]: o utilizador deve escolher seu novo ponto de entrada (de **EP1** a **EP5** ou **EP1, EP2 e EP5**) na parte avançada da OPL ALFA (AP-OPL), em função do escopo do seu contexto de modelagem (ou requisitos sistêmicos) do *groupware* que está modelando. Os novos pontos de entrada na parte avançada são:

- O ponto de entrada **EP1**, que deve ser escolhido quando os requisitos para a nova modelagem incluírem a definição de onde acontecerá a colaboração.
- O ponto de entrada **EP2**, que deve ser escolhido quando o objetivo for especificar as ações de contribuição sem importar onde ocorrerão **(EP1** é quem se importa com isso).

- O ponto de entrada **EP3**, que é a opção para quando o interesse estiver nos compromissos de colaboração, sem se importar com as ações de contribuição (**EP2**) e onde elas acontecem (**EP1**).
- O ponto de entrada **EP4**, que deve ser escolhido quando os requisitos incluírem a coordenação no processo de colaboração versando sobre tipos de recursos e modelando a participação desses recursos na seção colaborativa.
- O ponto de entrada **EP5**, que quando os requisitos incluírem a definição da comunicação durante a colaboração sem interesse em como a cooperação se procederá e como a coordenação será feita.
- **EP1**, **EP2** e **EP3**, que são pontos de entrada quando o objetivo para a nova modelagem incluir cooperação; **EP4**, se incluírem a coordenação e **EP5**, quando incluírem comunicação sem cooperação.

2. [**Identifique as características funcionais no contexto**]: a tecnologia de *groupware* permite um acesso amplamente expandido de pessoas, documentos, discussões, parâmetros de mensuração e até ao passado. Enquanto a equipe estiver conectada (ligada), a mesma poderá recorrer ao conhecimento (*know-how*) e à participação de informações de qualquer pessoa dentro ou fora da organização; e qualquer pessoa pode recorrer ao trabalho da equipe, enquanto está sendo realizado o trabalho e quando estiver salvo. É possível, na verdade, com bastante facilidade, que todos os envolvidos leiam documentos, participem de discussões, acompanhem o progresso e o padrão de mensuração, e fiquem a par de qualquer trabalho captado e realizado pela equipe no meio oferecido pela tecnologia.

Portanto, ao fazermos o estudo de como a tecnologia de *groupware* funciona, descobrimos que a mesma possui algumas características funcionais similares. Neste contexto, para que um conjunto de sistemas seja considerado um *groupware* de fato, os mesmos devem possuir, ao menos, algumas das características funcionais similares às abaixo descritas:

→ Note-se abaixo que **negritamos** e **sublinhamos** algumas palavras-chave para exemplificar algumas dessas características funcionais similares:

a) Dado que um ***groupware online tem o*** **acesso universal aos dados** e a **criação de grupos de trabalho** com clientes, fornecedores, parceiros temporários ou permanentes e outros, é

uma tecnologia que permite que um **grupo de pessoas possa trabalhar no mesmo aplicativo** ou sistema, **não importa onde elas estejam**.

- Observe-se que a localização da sessão colaborativa está dentro do escopo da modelagem, pois sistemas colaborativos para ambientes de trabalho devem ser abordados, visando ao acesso universal e indiscriminado. Neste caso, deve-se iniciar a modelagem pelo ponto de entrada **EP1** na utilização do processo da ALFA (AP-OPL), pois os requisitos incluem a definição de onde acontecerá a colaboração. Após esta ação, deve-se fazer uso dos POs da área problemática "**Sessão colaborativa**", um de cada vez, mas de forma integrada. Feito isto, continue a percorrer o caminho d*o* processo da ALFA (AP-OPL). E assim sendo, ir verificando se os POs contidos no caminho correspondem aos requisitos que serão esboçados na nova modelagem.

b) Dado que um *groupware* é uma tecnologia que permite que um grupo de pessoas (usuários) possam trabalhar **no mesmo aplicativo** ou sistema **não importa onde eles estejam**, os mesmos acessam dados idênticos de qualquer dispositivo com conexão à Internet;

- Observe-se que os termos estão relacionados ao compartilhamento de recursos, pois um *groupware* oferece recursos que permitem às equipes trabalharem juntas e participarem de projetos em comum. Diante disso, pode-se utilizar os POs da área problemática "**Recursos**" de acordo com os requisitos do contexto, ou pode-se continuar a percorrer o caminho na ALFA (AP-OPL).

c) Dado que um *groupware* é projetado **para armazenar** e **gerenciar os dados**, oferecendo fácil acesso;

- Observe-se que os termos estão relacionados ao armazenamento, a acessar e a compartilhar dados na Internet. Assim, pode-se utilizar os POs da área problemática "**Ações de Contribuições**", "**Recursos**" e "**Descrição do Acordo Colaborativo**". E então, ir verificando a existência de indícios dos POs no contexto de requisitos. Por exemplo: se há indícios de **recursos** (*Resource*) em **sessão colaborativa (***Collaborative Session*) com a mediação de um agente de **participação de recursos** (*Resource Participation*), deve-se usar, o PO (CSR) a partir da sessão colaborativa.

d) Dado que um *groupware* **registra modificações** feitas **e quem foi o responsável pela modificação**;

- Observe-se que os termos estão relacionados ao armazenamento, a acessar e a compartilhar dados na Internet. Então, pode-se fazer uso dos POs da área problemática "**Ações de Contribuições**" e de "**Compromissos e Objetivos**" de acordo com os requisitos do seu contexto, ou pode-se optar por continuar a percorrer o caminho na ALFA (AP-OPL). Por exemplo: se há indícios de **compromisso colaborativo** (*Collaborative Commitment*) que é inerente a um colaborador (*Collaborator),* deve-se usar, os o PO (CSR) a partir do PO (ACPPS).

e) Dado que um *groupware* melhora a **comunicação entre os usuários** com meios e **comunicação audiovisual e/ou escrita**;

- Observe-se que os termos estão relacionados a comunicação a distância, pois um *groupware* oferece recursos que permitem a várias equipes trabalharem juntas mesmo quando separadas geograficamente. Neste caso, pode-se fazer uso dos POs da área problemática "**Comunicação**" de acordo com os requisitos do contexto, ou pode-se optar por continuar a percorrer o caminho na ALFA (AP-OPL). Por exemplo: se há indícios de interações de comunicação, deve-se verificar qual dos POs se enquadram no contexto de requisitos, por exemplo, os PO (CIP) ou PO (CIPS) e PO (IL).

f) Dado que um *groupware* **combina várias técnicas computacionais** para s**elecionar itens personalizados com base nos interesses dos usuários** e conforme o contexto no qual estão inseridos.

- Observe-se que os termos estão relacionados aos interesses dos usuários que estão relacionadas a três aspectos principais: atividades, artefatos e papéis. Neste caso, pode-se fazer uso dos POs da área problemática "**Papéis e Colaboradores**". Por exemplo: se há indícios de **acordo colaborativo** (*Collaborative Agreement Description)*, que é um tipo de contrato que define os direitos e as obrigações dos papéis que podem participar em uma sessão colaborativa (*Collaborative Session*), deve-se usar, por exemplo, os PO (COPCR) ou PO (APAC) a partir da sessão colaborativa.

g) Dado que numa aplicação *groupware* os ***3Cs interagem*** continuamente entre si;

Observe-se que os termos estão relacionados a interação entre as subontologias de CONTO (comunicação, coordenação e cooperação). A partir dos padrões que se utilizam desse princípio, é possível combinar o comportamento de diversas classes de formas diferentes, obtendo uma grande quantidade de possibilidades de comportamento final.

3. [**Esboçe seus POs de domínio no OLED Estendido**]: primeiro, deve-se esboçar o PO de domínio (DROPs) de origem. Observe-se que o PO (origem) escolhido possui Classes OntoUML. Essas Classes do PO (origem) podem ser interligadas com outras Classes de outros POs (um de cada vez). Neste caso, deve-se ir acrecentando os demais POs de acordo com o contexto. Ou se preferir, optar por percorrer o caminho dos DROPs da linguagem do modelo de processo de uso da ALFA (AP-OPL)(simples (vide Figura 5.8) ou detalhado (vide Figura 5.11)). Cabe destacar que um PO pode gerar dois ou mais padrões (alternativos ou variantes), tendo como origem um PO existente do catálogo (vide seção 4.13), haja vista que os POs de domínio (DROPs) são inter-relacionados e interconectados entre si.

4. [**Verifique os elementos candidatos a Classe identificados**]: um *groupware* possui diferentes elementos necessários à criação de aplicações colaborativas que podem ser candidatos a Classe. As classes são utilizadas para definir o comportamento de objetos do mundo real, segundo a perspectiva (ou abstração) que interessa para a solução de um problema específico. Em outras palavras, uma classe é uma abstração de um conjunto de objetos com propriedades, comportamentos, relacionamentos e semântica comuns. Uma classe pode ser vista como um esqueleto/modelo para criar objetos.

Dicas gerais:

- Utilizar substantivos para nomear as classes.
- Classes devem encerrar uma só abstração do mundo real. Por exemplo, uma classe estudante contendo também o histórico do estudante não é uma boa solução. Melhor é dividi-la em duas classes: estudante e histórico.
- Nas fases iniciais de desenvolvimento, pode-se suprimir os atributos e os métodos deixando somente os compartimentos. Em síntese, as classes especificam a estrutura e o comportamento (operações) dos objetos, que são instâncias das classes.
- Os diagramas de classes ilustram atributos e operações de uma classe e as restrições sobre como os objetos podem ser conectados; descrevem também os tipos de objetos no sistema e os relacionamentos entre esses objetos, que podem ser associações e abstrações.
- Podemos dizer que os diagramas de classes são os principais diagramas estruturais da UML, pois ilustram as classes, as interfaces e os relacionamentos entre elas.

A UML reconhece três tipos mais importantes de relações: dependência, associação e generalização(ou herança). Geralmente, as classes relacionam entre-si através de elementos conhecidos como associações, que podem ser de quatro tipos: especialização ou generalização; composição, agregação e dependência.

→ Observe-se, abaixo que **negritamos** algumas palavras-chave para exemplificar um conjunto de objetos que representam elementos abstratos que podem ser candidatos a Classe**:**

- **Coisas concretas**;
- **Descritores ou especificações**;
- **Lugares**;
- **Transações**;
- **Itens de transação**;
- **Papéis ou funções**;
- **Unidades organizacionais**;
- **Conceitos do negócio**;
- **Eventos ou ocorrências**;
- **Registro fiscal/contratual**;
- **Coleções de outras coisas**;
- **Serviços**;
- **Documentos**.

5. [**Estereotipe os elementos candidatos a Classe identificados**]: os elementos candidatos a Classe identificados através de contexto podem ser acrescidos aos POs de domínio (DROPs) de origem ou aos POs (alternativos ou variante) gerados, mas os mesmos devem ser estereotipados de acordo com suas metapropriedades ontológicas (mais detalhes na seção 4.3.1). Essas metapropriedades permitem aos modeladores construir modelos que tenham uma maior chance de alcançar os critérios de qualidade mencionados, quando comparados com as demais linguagens que não aplicam tais distinções, de acordo com (GUIZZARDI et al., 2004). Um estereótipo é mecanismo de extensibilidade dentro da UML que nos permite definir categorias de Classes que implementam o metamodelo revisado da linguagem ONTOUML. Na ONTOUML, os esteriótipos são representados entre <<>> (por exemplo, <<

estereótipo>>). A OntoUML foi construída para possibilitar a construção de modelos, com base em decisões que envolvem aspectos ontológicos e não computacionais.

5.11 GERAR NOVOS ARTEFATOS DE USO

Nesta etapa do método ÔMEGA, devem ser gerados os novos artefatos de colaboração, o "**Catálogo de DROPs de Colaboração Consolidado**" e seu respectivo "**Modelo de Processo de Uso ALFA (AP-OPL)**". Nesta atividade, temos como artefatos de entrada o "**Catálogo de DROPs de Colaboração**", a OPL que representa o "**Modelo de Processo da C-OPL**" e as "**Ontologias de Colaboração** (três subontologias de CONTO)". E, como artefatos de saída, o "**Catálogo de DROPs de Colaboração Consolidado**" e o "**Modelo de Processo de Uso ALFA (AP-OPL)**".

A Figura 5.8 apresenta uma visão geral da OPL ALFA (AP-OPL) que corresponde ao "**Modelo de Processo de Uso ALFA(AP-OPL)**" do método ÔMEGA, a qual se consistiu na reengenharia da representação visual da C-OPL (vide seção 4.9), com base no uso dos artefatos de colaboração, através da linguagem de especificação de OPLs proposta por Quirino (2016), por intermédio da ferramenta Office (PowerPoint)[31], uma vez que o OLED ainda não suporta a criação, nele de OPLs.

[31] Ferramenta Office (PowerPoint) - https://www.microsoft.com/pt-br/

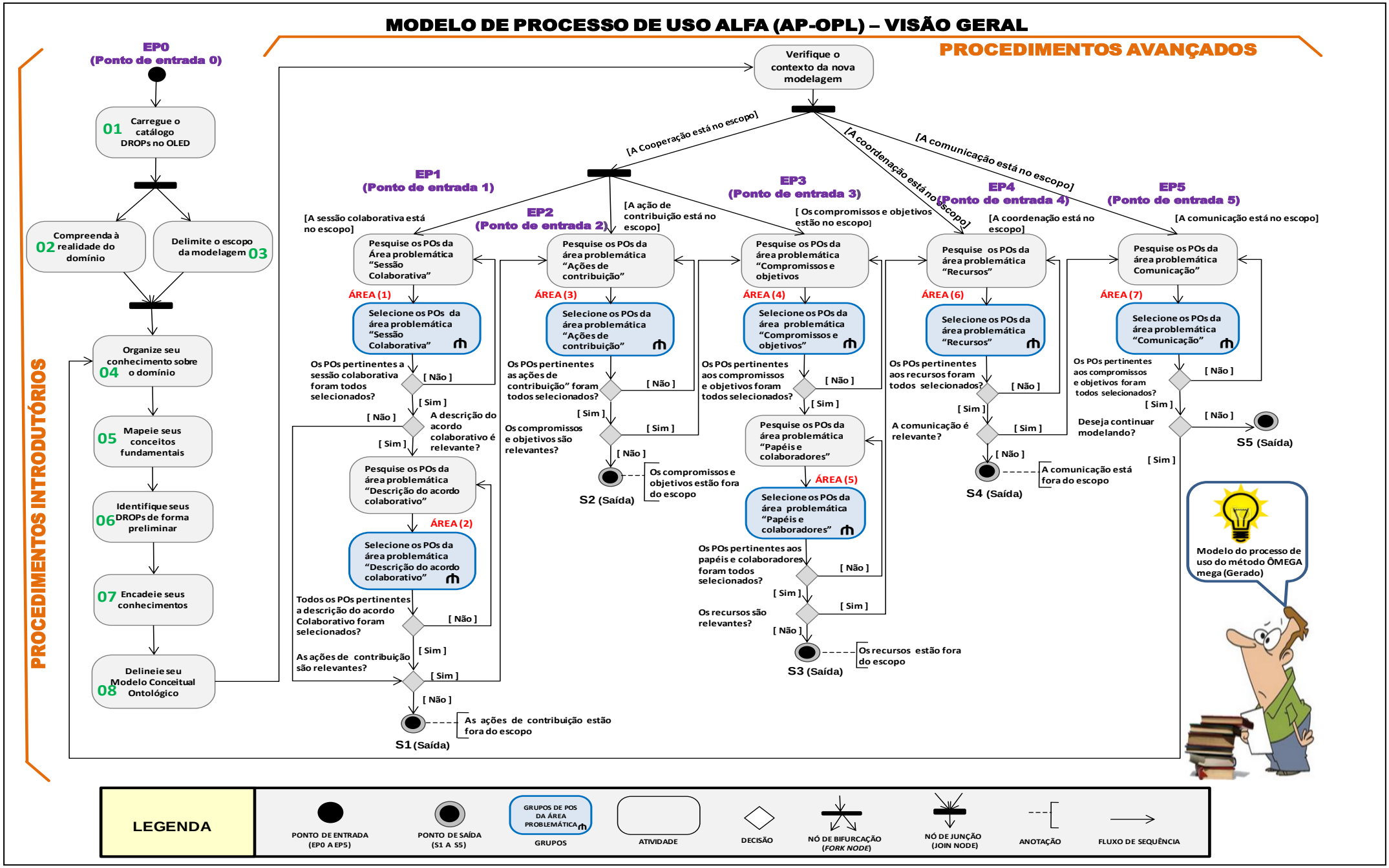


Figura 5.8: Modelo de processo de uso ALFA(AP-OPL) do método ÔMEGA.

Fonte: Idealizado pela autora, com base em Souza (2017).

A OPL aqui proposta se chama ALFA (AP-OPL) e aborda os seguintes aspectos ou áreas problemáticas do catálogo consolidado: **Sessão de Colaboração**; **Descrição do Acordo Colaborativo**; **Ações de Contribuição**; **Compromissos e Objetivos**; **Papéis e Colaboradores**; **Recursos e Comunicação**. De acordo com essa linguagem, temos:

- um conjunto de POs de domínio (DROPs) é representado por retângulos arredondados rotulados;
- os nós iniciais (círculos sólidos) representam pontos de entrada na OPL, ou seja, DROPs da linguagem que podem ser usados sem resolver outros problemas primeiro;
- os nós finais (círculo sólido com auréola externa) representam pontos de saída na OPL, ou seja, DROPs da linguagem que podem ser usados para finalizar caminhos percorridos na linguagem;
- os fluxos de controle (linhas com setas) representam as sequências admissíveis nas quais DROPs podem ser usados;
- os nós de fusão (*merge*) (símbolo no formato de diamante) representam os pontos onde há uma fusão de caminhos na OPL;
- os nós de Junção/Bifurcação (barra sólida) representam a conjunção de caminhos (junção) ou caminhos paralelos possíveis (bifurcação);
- os nós de decisão (*decision nodes*), representados por um losango, são usados para representar caminhos alternativos. Assim, se o utilizador decide seguir o caminho de entrada do nó de decisão, ele deve escolher um e apenas um dos caminhos de saída do nó de decisão;
- uma extensão da notação UML original (linhas pontilhadas com setas) é usada para representar padrões variantes/alternativos, isto é, padrões que podem ser usados para resolver o mesmo problema de diferentes maneiras;

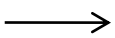
A OPL ALFA (AP-OPL) encontra-se subdividida em dois procedimentos distintos: **Procedimentos introdutórios do processo de uso** e **Procedimentos avançados do processo de uso.** O primeiro procedimento equivale a um giro por vez no cubo mágico (um passo ou ação por vez) para a construção de modelos específicos de *groupware* por parte dos novatos no OLED. Já o segundo equivale a dois ou mais giros em sequência, por vez, no cubo mágico (passos ou ações repetidas por vezes) para a construção de modelos específicos de *groupware* por parte dos modeladores já iniciados no OLED.

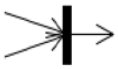
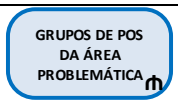
O ponto de partida para utilização de uma OPL se dá pelos pontos de entrada no modelo do processo da linguagem. A OPL ALFA tem seis pontos de entrada. O modelador (ou projetista) deve escolher um deles, dependendo do escopo da modelagem em desenvolvimento. O modelo do processo da ALFA (AP-OPL) prevê **seis pontos de entrada** para a seleção do DROPs:

- O **EP0** deve ser escolhido quando o modelador for novato na construção de modelos conceituais ontológicos de colaboração no OLED.
- Já **EP1** deve ser escolhido quando os requisitos para a nova modelagem incluírem a definição de onde acontecerá a colaboração.
- O **EP2** deve ser escolhido quando o objetivo for especificar as ações de contribuição sem se importar onde elas ocorrerão (**EP1** é quem se importa com isso).
- O **EP3** é a opção para quando o interesse estiver nos compromissos de colaboração sem se importar com as ações de contribuição (**EP2**) e onde elas acontecem (**EP1**).
- O **EP4** deve ser escolhido quando os requisitos incluírem a coordenação no processo de colaboração versando sobre tipos de recursos e modelando a participação desses recursos na seção colaborativa.
- Por último, o **EP5** é a opção para quando os requisitos incluírem a definição da comunicação durante a colaboração, sem interesse em como a cooperação se processará e como a coordenação será feita.
- Os **EP1**, **EP2** e **EP3** são pontos de entrada quando o objetivo para a nova modelagem incluir cooperação; **EP4**, se incluírem a coordenação, e **EP5**, quando incluírem comunicação sem cooperação.

Na Tabela 5 retratamos os elementos visuais utilizados na legenda do modelo de processo de uso ALFA (AP-OPL) visão geral:

Tabela 5: Elementos visuais utilizados na legenda da OPL ALFA (AP-OPL) visão geral

ELEMENTO	REPRESENTAÇÃO	SEMÂNTICA
Ponto de Entrada		Indica os pontos de entrada na OPL, isto é, indica os padrões na OPL que podem ser utilizados sem que seja necessário utilizar outros padrões antes.
Ponto de saída		Indica os pontos de saída na OPL.
Fluxo de sequência		Representa um fluxo sequêncial. Utilizado para indicar uma sequência entre as atividades.
Decisão		Representa uma decisão que pode desviar o

		fluxo ilustrado no diagrama. Utilizado quando lidamos com condições.
Nó de Bifurcação (*Fork Node*)		Nó de Bifurcação tem como finalidade dividir o fluxo em mais de uma direção.
Nó de Junção (*Join Node*)		Nó de junção tem a finalidade de fazer a união de várias direções do fluxo em uma única direção.
Grupo de Padrões		Agrupamentos de POs (padrões ontológicos) da um determinada área problemática.
Atividade		Atividade a ser executada.
Anotação		Anotações.

Fonte: Idealizado pela autora com base em (QUIRINO, 2016).

A Figura 5.9 fornece um modelo de comportamento geral do modelo de processo de uso ALFA (AP-OPL). Nele, mostramos o estudo detalhado dos caminhos a serem percorridos pela OPL, os grupos de Cooperação, Coordenação e Comunicação (CONTO) e as respectivas possibilidades de saída do processo.

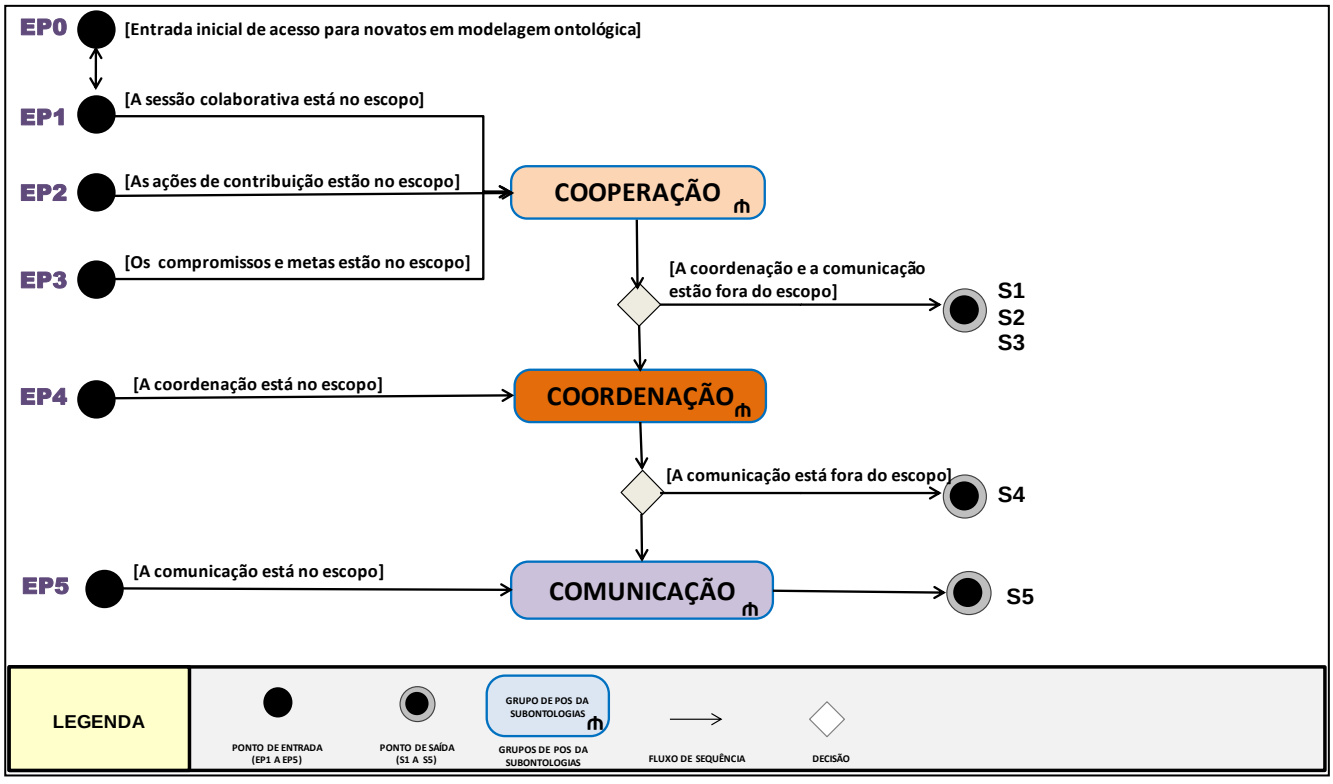


Figura 5.9: Caminhos de entrada do modelo de processo de uso do método ÔMEGA

Fonte: Idealizado pela autora

A Figura 5.10 retrata as áreas (1-7) presentes na ALFA (AP-OPL). Tais áreas são chamadas de áreas problemáticas dos POs de Domínio (DROPs) de colaboração. Na Figura, temos os DROPs que foram agrupados em função de suas áreas problemáticas de acordo com a subontologia a que pertencem (Cooperação, Coordenação e Comunicação), por meio de suas intenções e questões de competência.

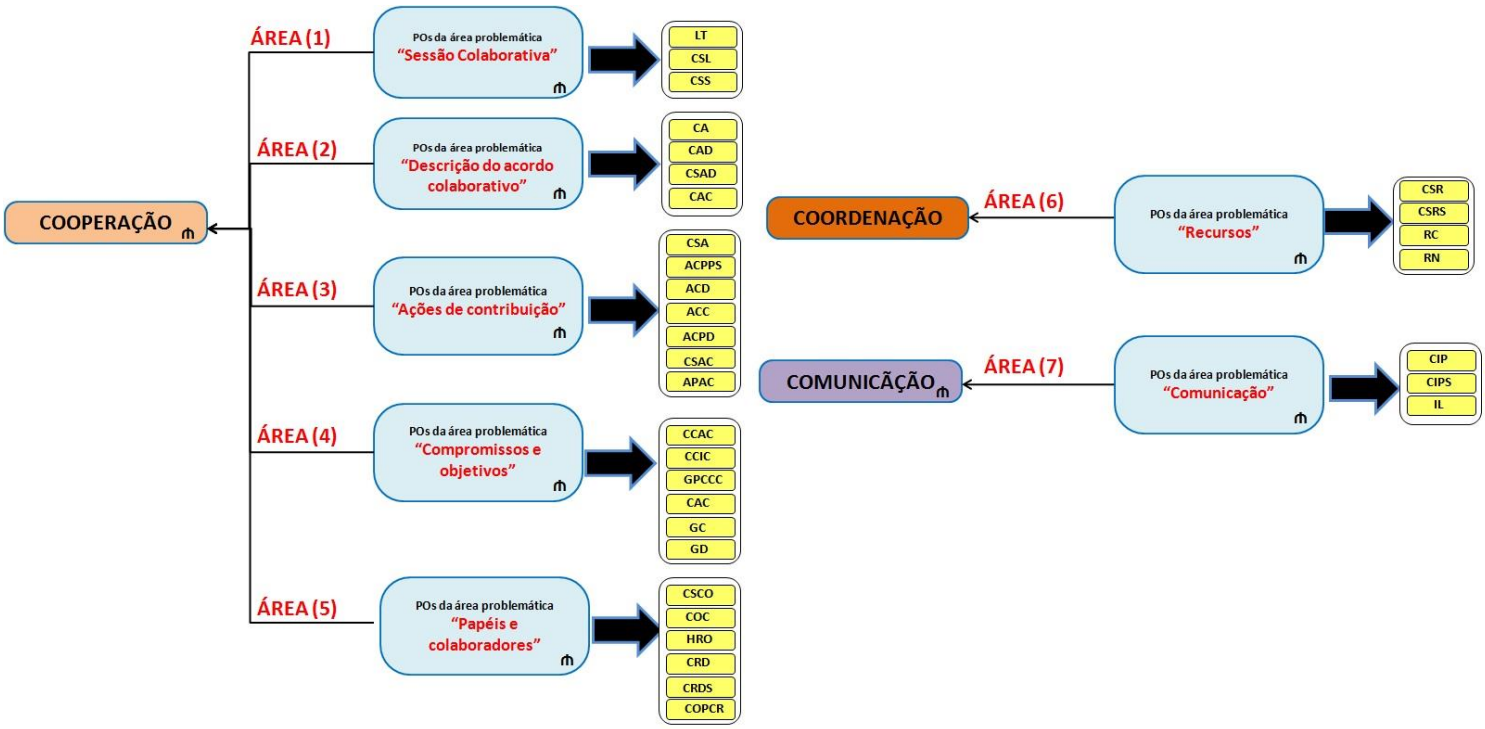


Figura 5.10: Os POs das áreas problemáticas de CONTO.
Fonte: Idealizado pela autora

A Figura 5.11 fornece uma visão detalhada que expande a visão geral do modelo de processo de uso ALFA (AP-OPL) para seleção dos POs de Domínio (DROPs) do Catálogo Consolidado. Na Figura, temos o detalhamento dos DROPs que foram agrupados em função de suas áreas problemáticas, por meio de suas intenções e questões de competência na OPL do método ÔMEGA.

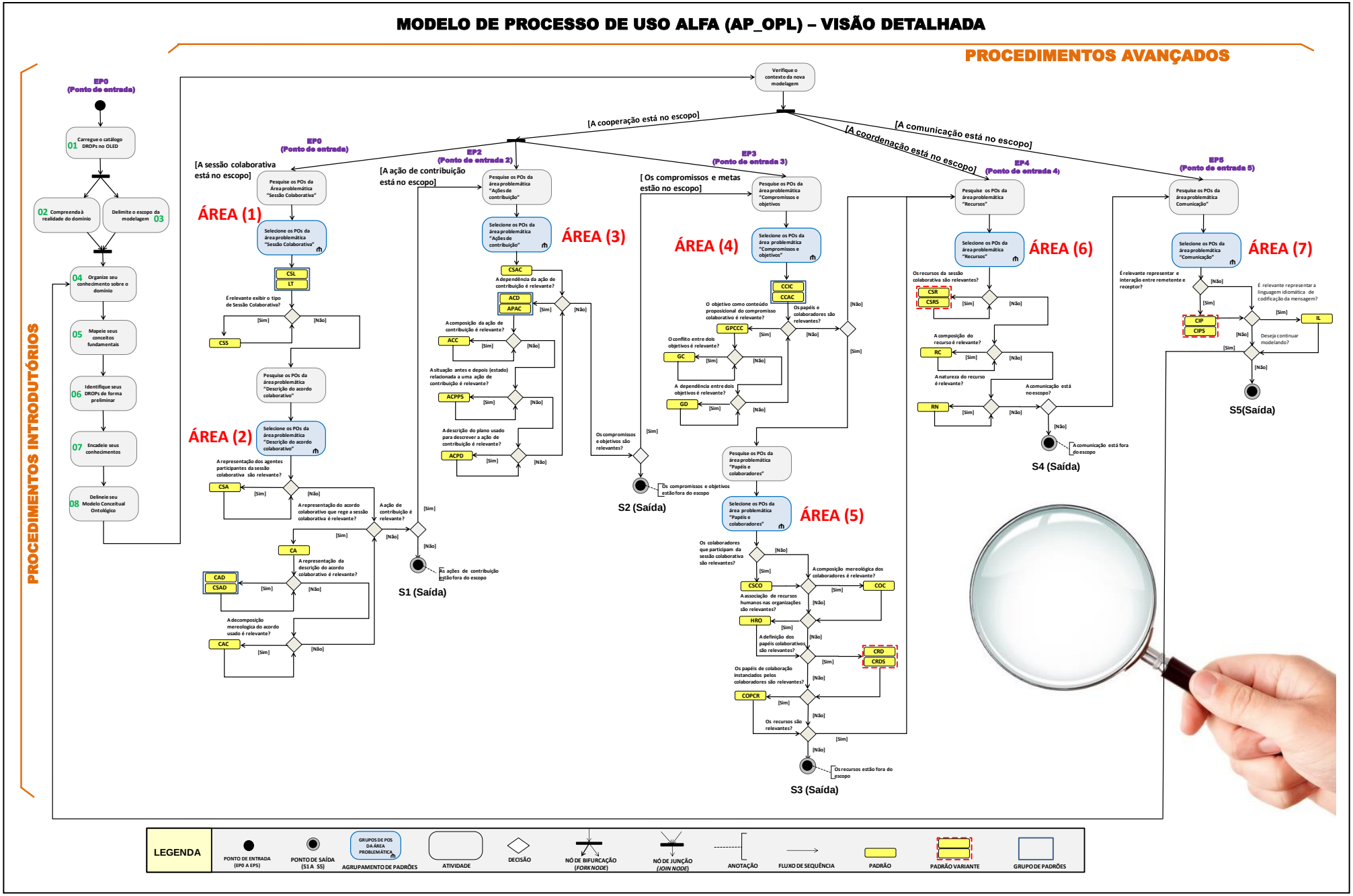


Figura 5.11: Visão detalhada da OPL ALFA (AP-OPL).

Fonte: Idealizado pela autora

A Tabela 5.1 apresenta o "**Catálogo de DROPs de Colaboração Consolidado**", que agrupa as informações dos POs de Domínio (DROPs) de forma estruturada e organizada em: N°., identificador dos POs, áreas problemáticas, nome dos padrões, intenção, questões de competência e subontologia. Aqui, nós nos abstivemos de apresentar todos os axiomas e modelos por questão de espaço (vide seção 4.7).

Tabela 5.1: Catálogo de DROPs de Colaboração Consolidado

N°.:	IDENTIFICADOR DOS POs	ÁREAS PROBLEMÁTICAS	NOME DOS PADRÕES	INTENÇÃO	QUESTÕES DE COMPETÊNCIA
1.	CSL	Sessão de colaboração.	Localização da sessão colaborativa.	Representar a localização do *site* onde ocorre a sessão colaborativa.	***QCs1*** - Onde acontece a sessão colaborativa?
2.	LT		Tipo de localização.	Representar o tipo de local em que a sessão colaborativa acontece.	***QCs2*** - O site da sessão colaborativa é real ou virtual?
3.	CSS		Tipo de sessão colaborativa.	Representar o tipo de sessão colaborativa onde o processo de interação acontece.	***QCs3*** - A sessão colaborativa é de leitura ou escrita?
4.	CSA	Descrição do acordo colaborativo.	Agentes de sessão colaborativa.	Representar os agentes que participam da sessão colaborativa.	***QCs4*** - Quais são os agentes que participam da sessão colaborativa?
5.	CA		Acordo colaborativo.	Representar o acordo colaborativo que rege a colaboração sessão.	***QCs5*** - Qual é o acordo colaborativo que rege a sessão colaborativa?
6.	CAD		Descrição do acordo colaborativo.	Representar a descrição de acordos de colaboração.	***QCs6*** - Qual é a descrição do acordo colaborativo que descreve o tipo de contrato de colaboração?
7.	CSAD		Descrição do acordo da sessão colaborativa.	Para representar a descrição do acordo colaborativo usado na sessão colaborativa.	***QCs7*** - Qual é a descrição do acordo da sessão colaborativa?
8.	CAC		Composição do acordo colaborativo.	Representar a decomposição mereológica de um acordo colaborativo.	***QCs8(A)*** - Este acordo colaborativo pode ser decomposto? ***QCs8(B)*** - Quais são os componentes deste acordo colaborativo?
9.	CSAC	Ação de contribuição.	Ações de contribuição da sessão colaborativa.	Representar as ações de contribuição que compõem a sessão colaborativa.	***QCs9*** - Quais são as ações de contribuições que compõem uma sessão colaborativa?
10.	ACPPS		Situaçâo antes e depois da ação contribuição	Representar a situação antes e depois das ações de contribuição.	***QCs10(A)*** - Quais são as pré-situações da contribuição de ação? ***QCs10(B)*** - Quais são as pós-situações da contribuição de ação?
11.	ACD		Dependência da ação de contribuição.	Representar uma ação de contribuição que depende de outras ações de contribuição.	***QCs11*** - Esta ação de contribuição depende de outras ações de contribuição?
12.	APAC		Agente executa a ação de contribuição.	Representar os papéis colaborativos instanciados pelos agentes que participam	***QCs12*** - Quais são os papéis colaborativos instanciados pelos agentes que participam

				da sessão colaborativa.	da sessão colaborativa
13.	ACC		Composição da ação de contribuição.	Representar a decomposição mereológica de um acordo colaborativo.	***QCs13(A)*** **-** Esta ação de contribuição pode ser decomposta? ***QCs13(B)*** **-** Quais são os componentes dessa ação de contribuição ?
14.	ACPD		Descrição do plano da ação de contribuição.	Representar a descrição do plano das ações de contribuição.	***QCs14(A)*** **-** Qual é a descrição do plano que descreve a ação de contribuição?
15.	CCIC	Compromissos e objetivos.	Compromissos colaborativos inerentes a um colaborador.	Representar o compromisso colaborativo que é inerente a um colaborador.	***QCs15*** **-** Qual são compromissos colaborativos que são inerentes a um colaborador?
16.	CCAC		Compromisso colaborativo para ação de contribuição.	Representar o compromisso colaborativo que causa a contribuição da ação.	***QCs16*** **-** Qual é o compromisso colaborativo que causa a ação de contribuição?
17.	GPCCC		Objetivo como conteúdo proposicional do compromisso colaborativo.	Representar o objetivo que é o conteúdo proposicional de um compromisso colaborativo.	***QCs17*** **-** Qual é o objetivo que é o conteúdo proposicional de um compromisso colaborativo?
18.	GC		Conflito entre objetivos.	Representar o conflito entre outros objetivos.	***QCs18-*** Esse objetivo está em conflito com outro objetivo?
19.	GD		Dependência entre dois objetivos.	Representar a dependência entre objetivos.	***QCs19*** **-** Esse objetivo depende de outra objetivo?
20.	CSCO	Papéis e colaboradores.	Colaborador da sessão colaborativa.	Representar os colaboradores que participam da sessão colaborativa.	***QCs20*** **-** Quais os colaboradores que participam da sessão colaborativa?
21.	COC		Composição de colaboradores.	Representar a composição mereológica de colaboradores (como um grupo, subgrupo e seus membros).	***QCs21-*** Quais são as composições mereológica de colaboradores (como grupo, subgrupo e seus membros)
22.	HRO		Recursos humanos em organizações.	Representar a associação de recursos humanos em organizações.	***QCs22*** **-** Quais são as associações de recursos humanos em organizações?
23.	CRD		Definição de papéis colaborativos.	Representar os papéis colaborativos que participam de uma sessão colaborativa, o qual são definidos por um tipo acordo de colaboração.	***QCs23 (A)*** **-** Quais são os papéis colaborativos que participam de uma sessão colaborativa? ***QCs23 (B)*** **-** Qual é o acordo colaborativo que define os papéis colaborativos que participam de uma sessão colaborativa?
24.	CRDS		Definição de papel colaborativo simples.	Representar os papéis colaborativos que participam de uma sessão colaborativa.	***QCs24*** **-** Quais são os papéis colaborativos que participam de uma sessão colaborativa?
25.	COPCR		Colaborador desempenha papel colaborativo.	Representar os papéis de colaboração que instanciados pelos colaboradores participam da sessão colaborativa	***QCs25*** **-** Quais são os papéis de colaboração que instanciados pelos colaboradores participam da sessão colaborativa?
26.	CSR	Recursos	Recursos da sessão	Representar os recursos que	***QCs26*** **-** Quais são os recursos

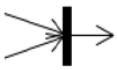
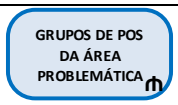
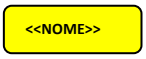
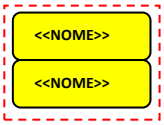
			colaborativa	participam da sessão colaborativa.	que participam da sessão colaborativa?
27.	CSRS		Recursos simples de sessão colaborativa.	Representar a participação de recursos em sessão colaborativa, com a mediação de um agente de participação de recursos.	***QCs27*** **-** Quais são os recursos simples que participam da sessão colaborativa?
28.	RC		Composição de recurso.	Representar a decomposição mereológica de recursos.	***QCs28(A)*** **-** Esse recurso pode ser decomposto? ***QCs28(B)*** **-** Quais são os componentes deste recurso?
29.	RN		Natureza de recurso.	Representar os tipos de recursos em termos de compartilhamento.	***QCs29*** **-** Esse recurso pode ser compartilhável ou exclusivo?
30.	CIP	Comunicação	Padrão de interação de comunicação.	Representar detalhadamente um padrão de interação conectando um remetente e um destinatário.	***QCs30 (A)*** - Quem é o remetente na interação comunicativa? ***QCs30(B)*** - Quem é o receptor na interação comunicativa? ***QCs30(C)*** **-** Qual é a mensagem da interação comunicativa?
31.	CIPS		Padrão de interação de comunicação simples.	Representar uma interação comunicativa entre um remetente e um destinatário.	***QCs31(A)*** - Quem é o remetente na interação comunicativa? ***QCs31(B)*** - Quem é o receptor na interação comunicativa? ***QCs31(C)*** - Qual é a mensagem da interação comunicativa?
32.	IL		Padrão idiomático de linguagem.	Representar a linguagem idiomática utilizada para codificar a mensagem.	***QCs32-*** Qual é a linguagem idiomática que codifica a mensagem?

Fonte: Idealizado pela autora, com base em Souza (2017).

Alguns POs de domínio (DROPs) extraídos da CONTO foram reunidos no catálogo DROPs de colaboração consolidado, com o intuito de estruturar e/ou organizar os DROPs agrupados em função de suas áreas problemáticas, por meio de suas intenções e questões de competência. Vale destacar que a padronização que os POs de domínio (DROPs) de CONTO oferecem torna a modelagem conceitual ontológica muito semelhante a montar formas ou objetos usando o LEGO® (SOUZA, 2017).

Na Tabela 5.2 retratamos os elementos visuais utilizados na legenda do modelo de processo de uso da ALFA (AP-OPL) visão detalhada:

Tabela 5.2: Elementos visuais utilizados na legenda da OPL ALFA (AP-OPL) visão detalhada

ELEMENTO	REPRESENTAÇÃO	SEMÂNTICA
Ponto de Entrada		Indica os pontos de entrada na OPL, isto é, indica os padrões na OPL que podem ser utilizados sem que seja necessário utilizar outros padrões antes.
Ponto de saída		Indica os pontos de saída na OPL.
Fluxo de sequência		Representa um fluxo sequêncial. Utilizado para indicar uma sequência entre as atividades.
Decisão		Representa uma decisão que pode desviar o fluxo ilustrado no diagrama. Utilizado quando lidamos com condições.
Nó de Bifurcação (*Fork Node*)		Nó de Bifurcação tem como finalidade dividir o fluxo em mais de uma direção.
Nó de Junção (*Join Node*)		Nó de junção tem finalidade de fazer a união de várias direções do fluxo em uma única direção.
Grupo de Padrões		Agrupamentos de POs (padrões ontológicos) de uma determinada área problemática.
Atividade		Atividade a ser executada.
Anotação		Anotações.
Nome padrão		Representa os padrões ontológicos de domínio.
Grupo de Padrões Variantes		Agrupa os padrões variantes (padrões que resolvem o mesmo problema, mas de maneiras diferentes e mutuamente exclusivas). Desse conjunto de padrões, apenas um deles pode ser selecionado.

Fonte: Idealizado pela autora, com base em Quirino (2016).

CAPÍTULO 6. DOIS EXEMPLOS DE USO

Para fins de ilustração da aplicação do método ÔMEGA e respectivo modelo de processo de uso ALFA (AP-OPL), buscou-se verificar a viabilidade do que aqui foi proposto e desenvolvido. A verificação foi executada através de dois estudos de caso no domínio de colaboração: um, de complexidade simples (*Brainstorming online*) e o outro, com nível de complexidade maior (*CollabCode* - Ferramenta para apoio ao desenvolvimento distribuído e colaborativo de software).

6.1 PRIMEIRO EXEMPLO DE USO (*BRAINSTORMING ONLINE* - COMPLEXIDADE SIMPLES)

Como **primeiro exemplo de uso** do método ÔMEGA e respectivo modelo de processo de uso ALFA (AP-OPL), iremos construir o modelo conceitual ontológico em ONTOUML específico do *groupware* "*Brainstorming online*" baseado na ferramenta de Lucidchart[32]. Cabe destacar que começaremos a modelar o *groupware* através da "**Visão a partir do zero**" e o verificaremos ao final. O **contexto** do *groupware* em questão é descrito abaixo:

_____

(CONTEXTO): O *brainstorming* (em português "tempestade cerebral") ou tempestade de ideia é uma técnica criativa em grupo que tem várias aplicações, mas é frequentemente usada em:

- **Desenvolvimento de novos produtos** - obter ideias para novos produtos e efetuar melhoramentos nos produtos existentes.
- **Publicidade** - desenvolver ideias para campanhas de publicidade.
- **Resolução de problemas** - consequências, soluções alternativas, análise de impacto, avaliação.
- **Gestão de processos** - encontrar formas de melhorar os processos comerciais e de produção.
- **Gestão de projetos** - identificar objetivos dos clientes, riscos, entregas, pacotes de trabalho, recursos, tarefas e responsabilidades.
- **Formação de equipes** - geração de partilha e discussão de ideias enquanto se estimulam os participantes a raciocinar e a criar: criatividade em equipe. Entretanto, muitas pessoas se

[32] https://www.lucidchart.com/

adaptaram ao *brainstorming* e usam-no como uma técnica criativa individual. De fato, muitos indivíduos podem ser mais criativos sozinhos do que fazendo parte de um tradicional grupo de *brainstorming*. Aliás, a liberdade de estar sempre disponível para um *brainstorming* individual é incrivelmente fácil de atingir.

É sempre possível realizar uma sessão de *brainstorming* eficaz mesmo quando não estão reunidas condições para se fazer tal sessão fisicamente, ou quando não se tem com quem trocar ideias. Pode-se sempre recorrer à ajuda de pessoas de todo o mundo pela internet, para encontrar ideias diferentes e inovadoras, ou, então, usar o meio virtual para fazer a sessão de *brainstorming online* com as pessoas que deseja, mas que se encontram separadas geograficamente.

Vale esclarecer que, antes de começar a fazer o *brainstorming* em grupo, é necessário definir um facilitador, que nada mais é do que uma pessoa neutra, ou seja, que não está ali para dar ideias, mas que desempenha o papel de observar o processo, cuidar do fluxo de sugestões e anotar todas as ideias, evitando que elas sejam esquecidas. Além disso, o indivíduo que atua como facilitador também é incumbido de orientar a equipe para que não perca o seu foco, ajudando-a a voltar ao tema principal sempre que for necessário. Desse modo, se as pessoas começarem a fugir da proposta inicial, tem-se que refazer a pergunta original da dinâmica.

Um grupo típico de *brainstorming* deverá conter de seis a doze pessoas; o grupo de avaliação deverá conter, exatamente, três, cinco ou sete pessoas. Usando um número ímpar, eliminam-se as possibilidades de empate quando é efetuada a votação das possíveis soluções. Esta situação também ajuda quando é procurado um consenso ao invés da votação. A composição dos participantes desse grupo pode variar. Poderá consistir em pessoas que faziam parte do grupo de geração de ideias, ou na combinação de pessoas deste grupo com pessoas externas, ou de um grupo completamente novo de indivíduos.

Utilizar as mesmas pessoas poderá ter a vantagem de assegurar a familiaridade com o problema, enquanto o uso de um grupo de pessoas externas ao grupo original pode ter o benefício da maior objetividade. A tarefa do grupo de avaliação é a de avaliar todas as ideias e selecionar as melhores para uma possível implementação ou estudo adicional. Depois de o líder do grupo receber a lista de ideias, em seguida, deve-se certificar e verificar se elas estão descritas claramente e estão concisas.

As ideias devem ser organizadas segundo categorias lógicas (usualmente, estas categorias vão de cinco a dez) e apresentadas ao grupo de avaliação para revisão. É possível tornar este processo mais fácil e prático utilizando-se um *checklist* organizado segundo determinados critérios como a simplicidade das ideias ou ideias menos custosas em termos de tempo e capital e outros termos similares. O grupo de avaliação deve verificar as melhores ideias de forma a sujeitá-las a testes práticos.

A maioria dos grupos de *brainstorming online* são constituídos por três elementos:

- Facilitador (Líder);
- Avaliadores;
- Participantes.

Há três partes principais no *brainstorming*:

- Encontrar os fatos;
- Geração da ideia;
- Encontrar a solução.

Na busca dos fatos na resolução de um problema existem duas sub partes:

- Definição do problema,
- Preparação.

Inicialmente, define-se o problema. Poderá ser necessário subdividir o problema em várias partes. A técnica de *brainstorming* funciona para problemas que têm muitas soluções possíveis, tal como a geração de ideias para o seu desenho. Depois, é necessário colher toda a informação que possa relacionar-se com o problema.

- Geração de ideias por *brainstorming*.
- Busca da solução.
- Avaliar e selecionar as melhores ideias.

As quatro principais regras do *brainstorming* são:

- **Críticas são rejeitadas**: Esta é provavelmente a regra mais importante. A não ser que a avaliação seja evitada, o princípio do julgamento não pode operar. A falha do grupo ao cumprir esta regra é a razão mais crítica para que a sessão de *brainstorming* não dê resultados. Esta regra é aquela que primariamente diferencia um *brainstorming* clássico dos métodos de conferência tradicionais.

- **Criatividade é bem-vinda**: Esta regra é utilizada para encorajar os participantes a sugerir qualquer ideia que lhes venha à mente, sem preconceitos e sem medo de que isso o vá avaliar imediatamente. As ideias mais desejáveis são aquelas que inicialmente parecem ser sem domínio e muito longe do que poderá ser uma solução. É necessário deixar as inibições para trás enquanto se geram ideias. Quando se segue esta regra, cria-se automaticamente um clima de *brainstorming* apropriado. Isso aumenta também o número de ideias geradas.
- **Quantidade é necessária**: Quanto mais ideias forem geradas, mais hipóteses há de encontrar uma boa ideia. Quantidade gera qualidade.
- **Combinação e aperfeiçoamento são necessários**: O objetivo desta regra é encorajar a geração de ideias adicionais para a construção e a reconstrução sobre as ideias dos outros.

De posse dos novos artefatos de colaboração (**Catálogo de DROPs de Colaboração Consolidado**, **Modelo de processo de uso ALFA(AP-OPL**) e das **Ontologias de Colaboração** (três subontologias de CONTO)) é preciso analisar o contexto do *groupware "Brainstorming online".* Neste caso, é aconselhavél selecionar o ponto de entrada **EP0** para seleção do DROPs, ou seja, executar as atividades de **01** a **08** da ALFA (AP-OPL). Essas atividades correspondem aos procedimentos introdutórios para novatos em desenvolvimento de modelagem conceitual ontológica (mais detalhes na seção 5.10).

Procedimentos introdutórios do processo de uso para o Brainstorming online:

1. [**Carregue o catálogo de DROPs no OLED**]: deve-se importar o "**Catálogo DROPs de Colaboração**"no OLED. Esse módulo OLED Estendido corresponde a um projeto (*Collaboration Patterns.oled*) que contém o catálogo (vide seção 4.11). Feito isso, deve-se escolher qual visão da OPL se deseja utilizar (visão geral (vide Figura 5.8) ou visão detalhada (vide Figura 5.11)).

2. [**Compreenda a realidade do domínio**]: é preciso analisar as informações contidas no contexto[33] do *groupware,* pois as mesmas ajudaram na definição da abrangência (que estabelecem limites) do sistema. É válido ressaltar que os limites do sistema são estabelecidos para definir o que está dentro e o que está fora do

[33] Contexto - é o conjunto de elementos físicos ou situacionais que ajudam o receptor da mensagem a compreendê-la.

sistema. Por exemplo, temos o contexto do *brainstorming* que delimita sua realidade (seu "**de até**").

3. [**Delimite o escopo da modelagem**]: a abrangência do escopo da modelagem depende

dos requisitos que demandam a visão sistêmica, bem como a sua aplicação no contexto informacional. Em outras palavras, essa visão sistêmica é a visão do todo por meio da análise das partes que o formam. Neste caso, os modelos conceituais gerados não necessitam cobrir todo universo de discurso, mas a parte requisitada para a sua representação (desejável). Por exemplo, identificamos como parâmetros do escopo da modelagem do "*Brainstorming online*": Necessidades de ferramentas a serem incluídas no ambiente; Necessidades dos usuários para desenvolvimento das atividades; Necessidade dos artefatos do ambiente que representam os objetos; Necessidade do ambiente em representar a divisão de trabalho; Necessidade do ambiente em representar as regras sociais; Necessidade do ambiente em representar a comunidade envolvida ("**de até**").

4. [**Organize o conhecimento do domínio**]: deve-se identificar as necessidades e

expectativas (ou seja, os requisitos) das partes interessadas do *groupware* de *Brainstorming online.*

A Tabela 6 retrata os elementos para a melhor compreensão do problema no contexto real, suas necessidades e expectativas identificadas. A partir desse entendimento, tomamos conhecimento do domínio, com o intuito de evitar o desperdício ou a negligência quanto aos recursos.

Tabela 6: Organize seu conhecimento do domínio

ELEMENTOS PARA ENTENDIMENTO DO PROBLEMA	NECESSIDADES E EXPECTATIVAS IDENTIFICADAS
Ferramentas de mediação - Necessidades das ferramentas a serem incluídas no ambiente.	Sessão colaborativa é bidirecional.
	Local do site.
	Site real (Sala de reunião) ou vitual (ambiente baseado na Web).
Sujeito (usuários ou grupos) - Necessidades dos usuários para desenvolvimento das atividades.	Grupos de *brainstormin*g (Facilitador (líder), Avaliadores e Participantes).
Objetos (Objetivos ou resultados) - Necessidade dos artefatos do ambiente que representem os objetos.	Post-it (situação antes (Editável) e depois (não editável)).
	Recurso compartilhável (não editável) e exclusivo (Editável).
Divisão de trabalho - Necessidade do ambiente em representar a divisão de trabalho.	Facilitador - Lider da sessão de *brainstormin*g (conduz, define o problema e prepara a sessão); Avaliador - avaliar ideias e gerar solução; Participantes - dar ideias.
Regra sociais - Necessidade do ambiente em representar as regras sociais.	Críticas são rejeitadas.
	Criatividade é bem-vinda.

	Quanto mais ideias forem geradas, mais hipóteses há de encontrar uma boa ideia Combinação e aperfeiçoamento das ideias são necessários.
Comunidade - Necessidade do ambiente em representar à comunidade envolvida.	Três partes da sessão de *Braisntorming:* Encontrar os fatos. Fatos na resolução de um problema: Definição do problema e Preparação. Geração da ideia; Encontrar a solução.

Fonte: Idealizado pela autora.

5. [**Mapeie seus conceitos fundamentais**]: a partir das necessidades e expectativas identificadas, deve-se reconhecer seu conjunto de conceitos fundamentais (ou principais características) que indicam as particularidades do *groupware* analisado. A

Tabela 6.1 retrata os conceitos fundamentais do *groupware* "*Brainstorming online".* Nela, foram descritas as necessidades e expectativas identificadas e, por conseguinte, os conceitos fundamentais.

Tabela 6.1: Conceitos fundamentais

NECESSIDADES E EXPECTATIVAS IDENTIFICADAS	CONCEITOS FUNDAMENTAIS
Sessão colaborativa é bidirecional.	*Site* onde acontece a sessão colaborativa.
Local do site.	Localizacão do site.
Site real (Sala de reunião) ou vitual (ambiente baseado na Web).	O tipo de site real (Sala de reunião) ou vitual (ambiente baseado na Web).
	Sessão colaborativa de leitura ou escrita.
Grupos de *brainstormin*g (Facilitador (líder), Avaliadores e Participantes)	Grupos (Facilitador, Avaliador e Participantes).
Post-it (situação antes (Editável) e depois (não editável).	Post_it.
	Recurso (compartilhável (Leitura) ou exclusivo(escrita)).
Recurso compartilhável (não editável) e exclusivo (Editável).	Situação depois (não editável) e antes (Editável).
Facilitador - Lider da sessão de *brainstormin*g (conduz, define o problema e prepara a sessão); Avaliador - avaliar ideias e gerar solução; Participantes - dar ideias;	Perfil.
	Permissões.
	Histórico.
Críticas são rejeitadas.	-
Criatividade é bem-vinda.	Escrever ação de contribuição.
Quanto mais ideias forem geradas, mais hipóteses há de encontrar uma boa ideia Combinação e aperfeiçoamento das ideias são necessários.	Escrever compromisso.
Três partes da sessão de *Braisntorming:* Encontrar os fatos. Fatos na resolução de um problema: Definição do problema e Preparação. Geração da ideia; Encontrar a solução.	Escrever_ideias, Escrever_avaliação, Escrever_solução e Escrever_Fatos.
	Gerar_fatos (Gerar_problema e Gerar_preparação), Avaliar_ideias, Buscar_solução, Gerar_ideias.

Fonte: Idealizado pela autora.

6. [**identifique seus DROPs de Colaboração**]: por meio de seus conceitos fundamentais (ou principais caracteristicas), deve-se identificar suas questões de competência (ou perguntas de competência), a fim de escolher de forma preliminar quais POs de domínio (DROPs) serão utilizados no desenvolvimento do novo modelo.

Cabe detacar que os POs são inter-relacionados entre si para funcionar como um todo, de modo que a obtenção deles, ou de parte deles, é feita de maneira integrada. A Tabela 6.2 retrata as áreas problemáticas e, por conseguinte, suas respectivas questões de competência. Com isso, conseguimos identificar de forma preliminar os POs que serão utilizados na nova modelagem.

Tabela 6.2: Escolha preliminar dos POs que serão usados

CONCEITOS FUNDAMENTAIS	ÁREA PROBLEMÁTICA	QUESTÕES DE COMPETÊNCIA	PADRÕES (POs)
Site onde acontece a sessão colaborativa.	Sessão colaborativa.	***QCs1*** - Onde acontece a sessão colaborativa?	**CSL**
O tipo de site real (Sala de reunião) ou vitual (ambiente baseado na Web).		***QCs2*** - O site da sessão colaborativa é real ou virtual?	**LT**
Localizacão do site.			
Sessão colaborativa de leitura ou escrita.		***QCs3*** - A sessão colaborativa é de leitura ou escrita?	**CSS**
Grupos (Facilidtador, Avaliador e Participantes).	-	-	-
Post_it.	-	-	-
Recurso (compartilhável (Leitura) ou exclusivo (escrita)).	Recursos.	***QCs29*** - Esse recurso pode ser compartilhável ou exclusivo?	**RN**
		QCs26 - Quais são os recursos que participam da sessão colaborativa?	**CRS**
Situação depois (não editável) e antes (Editável).	Ação de contribuição.	***QCs10(A)*** - Quais são as pré-situações da contribuição de ação? ***QCs10(B)*** - Quais são as pós-situações da contribuição de ação?	**ACPPS**
Escrever ação de contribuição.		**QCs9** - Quais são as ações de contribuições que compõem uma sessão colaborativa?	**CSAC**
Perfil.	-	-	-
Permissões.	-	-	-
Histórico.	-	-	-
Escrever compromisso.	Compromissos e objetivos.	***QCs15*** - Qual são compromissos colaborativos que são inerentes a um colaborador?	**CCIC**

		QCs16 **-** Qual é o compromisso colaborativo que causa a ação de contribuição?	**CCAC**
		QCs17 **-** Qual é o objetivo que é o conteúdo proposicional de um compromisso colaborativo?	**GPCCC**
Escrever_ideias, Escrever_avaliação, Escrever_solução e Escrever_Fatos.	-	-	-
Gerar_fatos (Gerar_problema e Gerar_preparação), Avaliar_ideias, Buscar_solução, Gerar_ideias.	-	-	-

Fonte: Idealizado pela autora.

7. [**Encadeie seu conhecimento**]: é interessante destacar o encadeamento do conhecimento

entre a OPL **ALFA (AP-OPL**), POs e ontologias. Uma sendo construída com o suporte oferecido pela outra, começando da última para a primeira. É interessante que isso cria uma sequência de acontecimentos encadeados, tendo em vista que a modelagem de um modelo de um sistema não é construída em um único passo, mas em muitos passos pequenos, através de pequenas transformações do modelo inicial até o modelo completo.

8. [**Delineie seu modelo conceitual ontológico**]: para delinear um modelo, deve-se fazer uso

dos novos artefatos de colaboração, pois os mesmos irão nortear a construção desse modelo conceitual específico de *groupware*. Deve-se lembrar que um modelo é feito em partes, às quais os POs de domínio (DROPs) são adicionados de forma incremental ao modelo (um por vez). Concluídos, os procedimentos introdutórios do processo de uso da OPL ALFA (AP-OPL) do método ÔMEGA, deve-se dar continuidade à modelagem, executando os procedimentos avançados.

Procedimentos avançados do processo de uso para o Brainstorming online:

1. [**Verifique o contexto da nova modelagem**]: caso tenham sido executados, os procedimentos introdutórios do modelo de processo de uso da OPL ALFA (AP-OPL), deve-se ir para o procedimento (4). Caso contrário, deve-se escolher qual visão da OPL se deseja utilizar (visão geral (vide Figura 5.8) ou visão detalhada (vide Figura 5.11)). Feito isso, deve-se averiguar o seu contexto da modelagem, a fim de verificar qual será o seu novo ponto de partida na ALFA (AP-OPL). Deve-se observar que o *groupware* analisado contém em seus requisitos a definição de onde acontecerá a colaboração. Diante disto, deve-se dar partida na parte avançada da OPL pelo ponto de entrada **EP1**. Com a OPL ALFA, podemos percorrer o caminho dos DROPs da linguagem, selecionando aqueles POs de domíno (DROPs) que sejam de interesses, ou seja, escolhendo aqueles que se enquadram no contexto da nova modelagem.

2. [**Identifique as características funcionais do contexto**]: deve-se identificar as características funcionais[34] do contexto, pois as mesmas indicam as funções essenciais do *groupware* de *Brainstorming online*.

A Tabela 6.3 retrata as características funcionais do *groupware* analisado, as suas áreas problemáticas e, por conseguinte suas respectivas questões de competência. Com isso, conseguimos identificar de forma preliminar os POs que serão utilizados na nova modelagem.

Tabela 6.3: Características funcionais do contexto

CONCEITOS FUNDACIONAIS	ÁREA PROBLEMÁTICA	QUESTÕES DE COMPETÊNCIA	PADRÕES (POs)
Localização da sessão colaborativa.	Sessão colaborativa	***QCs1*** - Onde acontece a sessão colaborativa?	**CSL**
O tipo de site em que a sessão colaborativa acontece.		***QCs2*** - O site da sessão colaborativa é real ou virtual?	**LT**
Localizacão do site.			
Tipo de sessão colaborativa (leitura ou escrita).		***QCs3*** - A sessão colaborativa é de leitura ou escrita?	**CSS**
Grupos (Facilitador, Avaliador e Participantes).	-	-	-
Um tipo de bloco de notas composto por pequenas (Post_it).	-	-	-
Recurso (compartilhável (Leitura) ou exclusivo (escrita)).	Recursos	***QCs29*** - Esse recurso pode ser compartilhável ou exclusivo?	**RN**

[34] Funcionais - relativo a funcionalismo.

		QCs26 **-** Quais são os recursos que participam da sessão colaborativa?	**CRS**
Situação depois (não editável) e antes (Editável).	Ação de contribuição.	***QCs10(A)*** - Quais são as pré-situações da contribuição de ação? ***QCs10(B)*** **-** Quais são as pós-situações da contribuição de ação?	**ACPPS**
Escrever ação de contribuição.	-	**QCs9** - Quais são as ações de contribuições que compõem uma sessão colaborativa?	**CSAC**
Perfil dos usuários.		-	-
Permissões de usuário.		-	-
Histórico da ações.		-	-
Escrever compromisso colaborativos.	Compromissos e objetivos.	***QCs15*** **-** Qual são compromissos colaborativos que são inerentes a um colaborador?	**CCIC**
	-	***QCs16*** **-** Qual é o compromisso colaborativo que causa a ação de contribuição?	**CCAC**
	-	***QCs17*** **-** Qual é o objetivo que é o conteúdo proposicional de um compromisso colaborativo?	**GPCCC**
Escrever_ideias, Escrever_avaliação, Escrever_solução e Escrever_Fatos.	-	-	-
Gerar_fatos (Gerar_problema e Gerar_preparação), Avaliar_ideias, Buscar_solução, Gerar_ideias.	-	-	-

Fonte: Idealizado pela autora.

3. [**Esboçe seus POs de domínio no OLED Estendido**]: primeiro, deve-se esboçar o seu PO (origem) o PO (**CSL**). Note-se que o PO (**CSL**) origem possui as seguintes Classes OntoUML(mais detalhes na seção 4.7): **Sessão Colaborativa** (*Collaborative Session*), **Ambiente** (*Site*) e **Localização** (*Location*). Agora, vamos combinar (ou integrar) o PO (**LT**) ao PO (**CSL**) origem. Cabe destacar que a adição dos demais POs identificados deve ser feita um de cada vez. Feito isso, devem ser analisados os seus elementos candidatos a Classe identificados. Eles são: Post_It, Post_It_De_Leitura, Post_It_De_Escrita, Post_It _Editável, Post_It_Não_Editável, Escrever_Objetivo_Do_ Post-It, Gerar_Fatos, Avaliar_Ideias, Buscar_Solução, Gerar_Ideias, Definição_Problema, Preparação,

Escrever_Compromisso, Escrever_Contribuição, Escrever_Ideias, Escrever Avaliação, Escrever_Solução, Escrever_Fatos, Grupo, Usuário, Facilitador, Avaliador, Participante, Permissões_De_Usuário, Perfil, Histórico,Ambiente_Baseado_Na_Web.

4. [**Verifique os elementos candidatos a Classe identificados**]: um *groupware* possui

diferentes elementos necessários à criação de aplicações colaborativas que podem ser candidatos a Classe. As classes são utilizadas para definir o comportamento de objetos do mundo real, segundo a perspectiva ou abstração que interessa para a solução de um problema específico. Neste contexto, os elementos candidados a Classe podem ser classificados em: **enquadrados** e **não enquadrados.**

No primeiro caso, deve-se averiguar se o elemento candidado a Classe escolhido se enquadra com outro PO descrito no catálogo. Em outras palavras, deve-se verificar se o elemento canditado a Classe pode ser enquadrado em uma das intenções ou questões de competência de outro PO existente no catálogo consolidado. Em caso positivo, deve-se fazer uso do PO identificado, ou se necessário, criar um PO variante. Já no segundo caso, deve-se estereotipar o elemento candidado a Classe, a fim de adicioná-lo a outro PO (já identificado na modelagem).

Por exemplo, uma sessão colaborativa pode ter um ou mais tipos diferentes de ambientes compartilhados. O PO (**CSS**) foi decomposto a partir da classe **Sessão Colaborativa** (*Collaborative Session*) do PO (**CSL**) em **Sessão de Leitura Colaborativa** (*Collaborative Reading Session*) e **Sessão de Escrita Colaborativa** (*Collaborative Writing Session*). As sessões de **Leitura e Escrita** acontecem em um determinado **Ambiente Virtual** (ou Local Virtual)**;** portanto, esse conceito foi especializado para **Um Ambiente Baseado na Web** (*Web-based Enviromment*). O PO (**CSS)** responde a questão de competência: "Que tipos de sessões colaborativas existem?", que condiz com a área problemática "Sessão Colaborativa".

5. [**Estereotipe os elementos candidatos a Classe identificados**]: os elementos candidatos a

Classe que não foram enquadrados com outros POs descritos no Catálogo Consolidado devem ser esteriotipados e acrescidos aos POs existentes. É importante lembrar que os elementos candidatos a Classe,

para serem acrescidos, ou melhor, associados a um modelo conceitual ontológico específico de *groupware,* devem ser estereotipados de acordo com as metapropriedades ontológicas (mais detalhes na seção 4.3.1). A Tabela 6.4 exibe os elementos candidatos a Classe identificados e respectivos estereótipos em ONTOUML.

Tabela 6.4: Estereótipos das Classes identificados

ELEMENTOS CANDIDATOS A CLASSE	ESTERIÓTIPO DA CLASSE ONTOUML
POST_IT.	<<mixin>>
POST_IT_DE_LEITURA, POST_IT_DE_ESCRITA.	<<role>><<role>>
POST_IT _EDITÁVEL, POST-IT_NÃO_EDITÁVEL.	<<rolemixin>><<rolemixin>>
ESCREVER_OBJETIVO_DO_ POST-IT.	<<subkind>>
ESCREVER_COMPROMISSO.	<<subkind>>
ESCREVER_CONTRIBUIÇÃO.	<<subkind>>
DEFINIÇÃO_PROBLEMA , PREPARAÇÃO.	<<subkind>><<subkind>>
GERAR_FATOS, AVALIAR_IDEIAS, BUSCAR_SOLUÇÃO, GERAR_IDEIAS.	<<phase>><<phase>><<phase>> <<phase>>
ESCREVER_IDEIAS, ESCREVER AVALIAÇÃO, ESCREVER_SOLUÇAO, ESCREVER_FATOS.	<<phase>><<phase>><<phase>> <<phase>>
GRUPO.	<<kind>>
USUÁRIO.	<<subkind>>
FACILITADOR, AVALIADOR, PARTICIPANTE.	<<role>><<role>><<role>>
PERMISSÕES_DE_USUÁRIO.	<<kind>>
PERFIL.	<<kind>>
HISTÓRICO.	<<kind>>
AMBIENTE_BASEADO_NA_WEB.	<<subkind>>

Nota. Os elementos candidatos a Classe foram escritos em letras maiúsculas e sem acentuação. Já os estereótipos das Classes ONTOUML foram mantidos em inglês por ser a língua dos padrões da ferramenta OLED

Fonte: Idealizado pela autora.

A Figura 6 retrata o modelo específico do *groupware* de "*Brainstorming online*". Esse modelo foi concebido segundo o método ÔMEGA e respectivo modelo de processo de uso ALFA (AP-OPL). Nele, identificamos os POs de domínio (DROPs) utilizados, os elementos candidatos a Classe estereotipados e associados aos POs existentes (já identificados na modelagem).

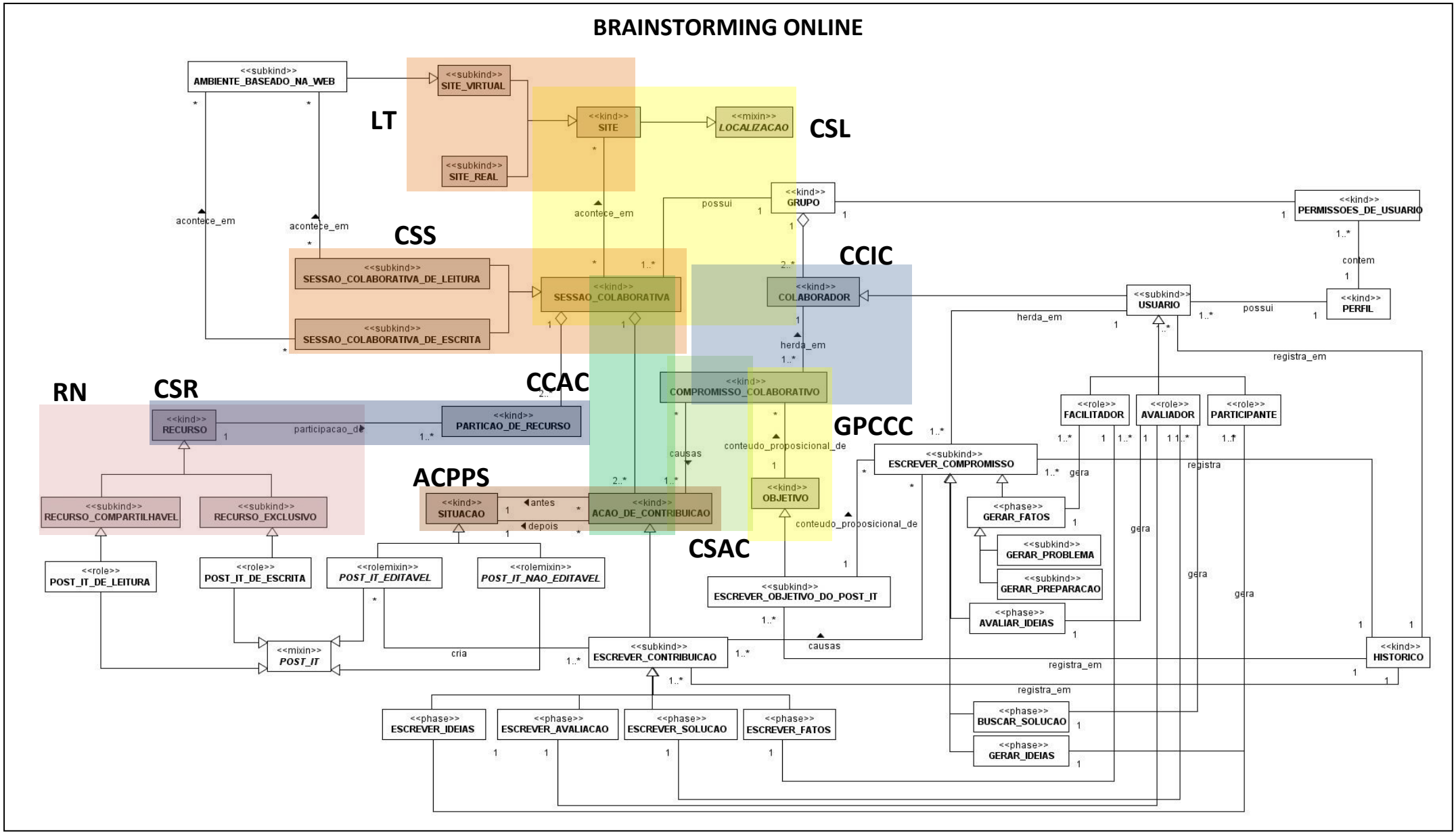


Figura 6: Modelo conceitual específico do Brainstorming Online

Fonte: Idealizado pela autora.

Verificação sintática do modelo no OLED: "Modelo conceitual específico do groupware Brainstorming online"

A Figura 6.1 apresenta a verificação sintática do modelo conceitual específico do *groupware* de "*Brainstorming online*" criado no OLED.

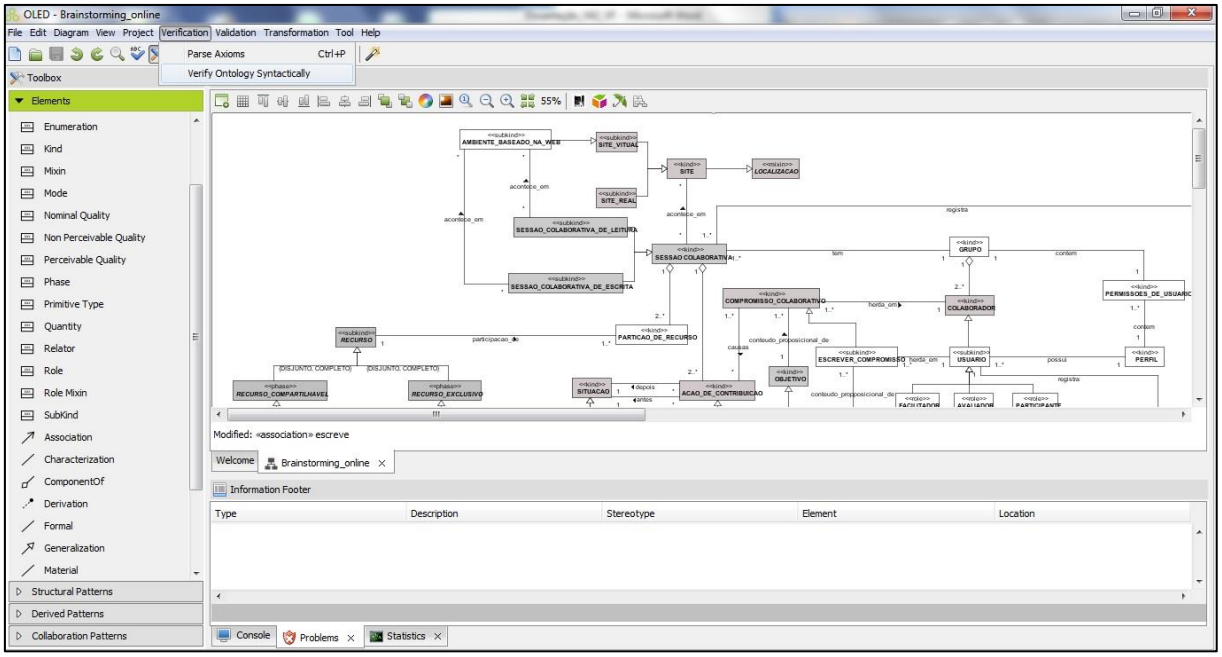


Figura 6.1 : Verificação sintática do groupware de Brainstorming online
Fonte: Idealizado pela autora.

O procedimento de verificação sintática consiste no mapeamento do modelo de "*Brainstorming online*" criado, submetido ao teste de verificação sintática disponível no OLED. Cabe destacar que os resultados errôneos retornados pela ferramenta devem ser ajustados. As etapas de teste e correção são repetidas até que o modelo esteja livre de erros. Ao final, obtém-se um modelo ontologicamente adequado e semanticamente refinado.

É importante ressaltar que a execução de verificações sintáticas são baseadas em critérios ontológicos para comparar o modelo submetido "*Brainstorming online*" às restrições ontológicas do metamodelo da OntoUML. Neste âmbito, é assegurado que o modelo conceitual verificado será sintaticamente correto.

6.2 SEGUNDO EXEMPLO DE USO (*COLLABCODE* - MAIOR COMPLEXIDADE)

Como **segundo exemplo de uso** do método ÔMEGA e respectivo modelo de processo de uso OPL ALFA (AP-OPL), iremos construir o modelo conceitual ontológico em ONTOUML específico do *groupware* "*CollabCode* - Ferramenta para apoio ao desenvolvimento distribuído e colaborativo de software", concebido por Silva (2013). Cabe destacar que começaremos a modelar o *groupware* através da "**Visão de manutenção**" e o verificaremos ao final. O **contexto** do *groupware* em questão segue descrito abaixo:

(CONTEXTO): O *CollabCode* foi projetado visando criar um ambiente de desenvolvimento online capaz de suprir as necessidades de equipes de desenvolvimento de *software* geograficamente distribuídas ou separadas. Tecnologias utilizadas no desenvolvimento do *CollabCode*: Hardware: computador com acesso à Internet; Sistema Operacional: Linux; Linguagens de programação: PHP e JavaScript; Banco e dados: PostgreSQL; IDE: NetBeans; Servidores web: Apache e ShareJS; Versionamento do código-fonte: com o Subversion.

Devido à necessidade de instalar aplicações específicas para que o trabalho de desenvolvimento ocorra e à limitada mobilidade dos membros da equipe, para solucionar o problema de mobilidade, de modo a centralizar todas as ferramentas necessárias para o desenvolvimento de sistemas, mais precisamente o desenvolvimento web, foi projetado o *CollabCode*. O *CollabCode* é *"Groupware Meeting system Web"* que possui um conjunto de ferramentas para comunicação, gerenciamento de atividades e colaboração capaz de suprir as necessidades de ferramentas instaladas nas estações de trabalho, sendo necessário apenas um navegador de Internet para sua utilização.

O *CollabCode* foi fundamentado em quatro requisitos, que precisam ser constantemente verificados e equalizados para que não se tornem armadilhas capazes de colocar em risco projetos ou até mesmo gerar atritos entre as equipes. Neste contexto e, em função de encontrar facilmente ferramentas que atendam isoladamente cada um desses requisitos, a tendência é que projetistas precisem ter em suas estações de trabalho uma série de aplicações que permitam a realização de suas atividades. Neste caso, é comum a utilização de sistemas de troca de mensagens, sistemas de compartilhamento de tela, softwares gerenciadores de atividades, sistema de compartilhamento de arquivos, IDEs, dentre outros, por parte das equipes de desenvolvimento distribuído.

Os principais objetivos do *CollabCode* são: primeiro, fornecer o ferramental necessário para que um projetista ou desenvolvedor consiga codificar dentro do navegador sem a necessidade de utilização de qualquer outra ferramenta. Para isso, a solução dispõe de uma IDE de desenvolvimento com a possibilidade de trabalhar com versionamento de arquivos e utilizar interface com banco de dados. O segundo objetivo é criar uma área de gerenciamento que possibilite a criação de projetos *web,* que utilizam como base a linguagem de programação PHP, e o gerenciamento das atividades inerentes ao projeto.

→ **O Processo de Colaboração no *CollabCode*** - Quando uma atividade deixa de ser motivada pelo desejo do sujeito ou surge outro desejo, esta pode tornar-se uma ação. As ações ou os objetivos conscientes vão determinar o conjunto de passos ou operações. De modo geral, uma atividade é desencadeada pela motivação para a realização de um desejo do sujeito. Por fim, as ações e objetivos conscientes vão determinar o conjunto de passos ou operações (CRUZ, 2001), o que revela "a noção de que o ser humano orienta-se por objetivo/metas planejadas que lhe permitem satisfazer suas necessidades" (LAGNI, 2004).

A Figura 6.2 apresenta os níveis de interações existente no *CollabCode* que são baseados na teoria da atividade.

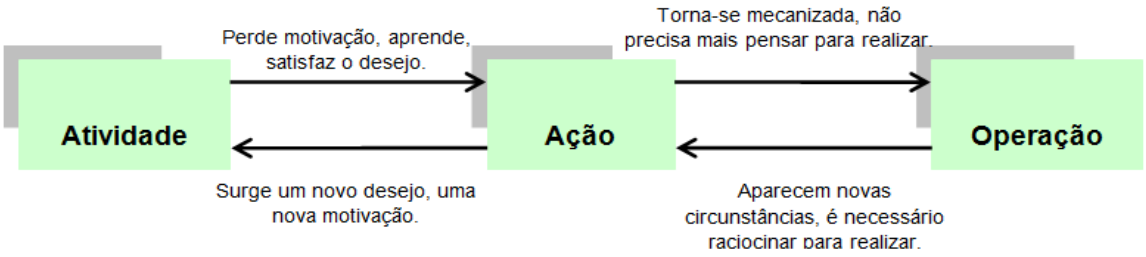


Figura 6.2 : Níveis da teoria da atividade usado no CollabCode.

Fonte: Trabalho de conclusão de curso da UNIVATES (SILVA, 2013).

A Tabela 6.5 ilustra uma visão geral de todas as três dimensões da colaboração (3C) disponíveis no *CollabCode*.

Tabela 6.5: Dimensões da colaboração disponíveis no *CollabCode*

TRÊS DIMENSÕES DA	C'S	DEFINIÇÃO	FERRAMENTAS
	COMUNICAÇÃO	As ferramentas de comunicação assíncronas ajudam a identificar as interações dos usuários com as atividades e fornecem base para a documentação do sistema que está sendo desenvolvido	Bate papo; Videoconferência; Comentários em atividades;

	COORDENAÇÃO	Estas ferramentas devem permitir, ao gerente de projetos, aferir métricas, acompanhar a execução de cada atividade, realocar, sempre que necessário, o recurso humano disponível, dentre outros.	Gerenciamento de projetos; Gerenciamento de atividades;
	COOPERAÇÃO	Estas ferramentas permitem que as interações muitas vezes existentes em equipes presenciais sejam possíveis em equipes distribuídas geograficamente.	Edição colaborativa; Compartilhamento de tela; Quadro branco.

Fonte: Trabalho de conclusão de curso da UNIVATES (Silva, 2013).

Na Figura 6.3 ilustramos todas as ferramentas disponíveis no *CollabCode.* O *CollabCode* foi desenvolvido utilizando diversas bibliotecas e componentes de interface. Dentre elas temos: o framework JQuery UI, a biblioteca Flexigrid, o editor CodeMirror e GanttView. Além disso, diversos *plugins* do JQuery foram utilizados, tais como: JQuery UI Layout e JQuery TE.

Figura 6.3 : Ferramentas disponíveis no CollabCode.

Fonte: Trabalho de conclusão de curso da UNIVATES (SILVA, 2013).

A Figura 6.4 apresenta algumas das interfaces do *CollabCode* construída com os componentes citados acima.

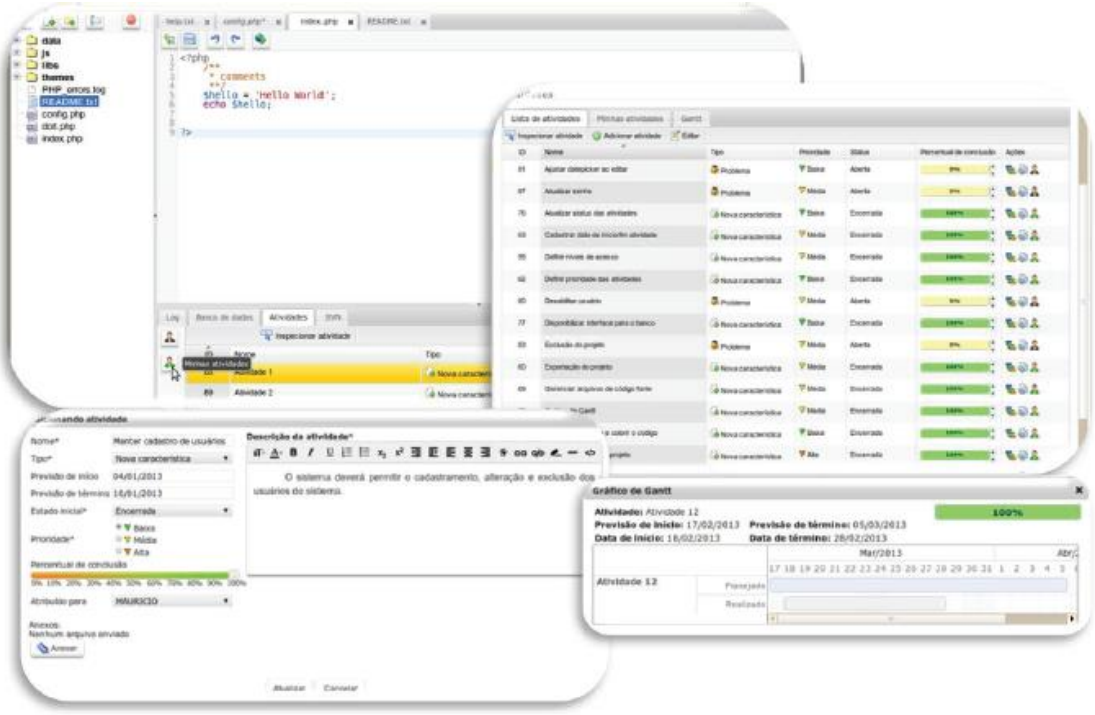


Figura 6.4 : Interfaces do CollabCode.

Fonte: Trabalho de conclusão de curso da UNIVATES (SILVA, 2013).

O controle de acesso do *CollabCode* é baseado em papéis. É possível cadastrar novos papéis e atribuir permissões a esses novos perfis de usuários. Além disso, é possível copiar as permissões de acesso de outros perfis. Na Tabela 6.6, são apresentados os papéis, permissões e as responsabilidades do *CollabCode.*

Tabela 6.6: Papéis, permissões e responsabilidade do CollabCode

PAPÉIS	PERMISSÕES	RESPONSABILIDADES
Gerente ou analista	Definir prioridades das atividades; Cadastrar data de início e fim em cada atividade; Manter atividades do projeto; Manter projetos; Exportar projetos; Manter cadastro de usuários; Gerenciar arquivos de código-fonte; Interagir por bate-papo; Controlar versionamento do código-fonte; Gerenciar reuniões / videoconferências; Atualizar status das atividades; Utilizar interface de banco de dados; Gerenciar bases de dados;	Gerente é o responsável por criar os projetos e alimentá-los com as atividades. Estas atividades podem ser requisitos (características) ou chamados para correção de problemas. Em geral, as atividades podem ser qualquer tarefa que implique em alteração ou criação de código.

	Codificar / editar cooperativamente; Manter documentos e atividades; Acompanhar o projeto através de gráficos .	
Desenvolvedor	Codificar; Editar cooperativamente ; Gerenciar arquivos de código- fonte; Manter documentos de atividades; Participar de reuniões / Videoconferências; Interagir por bate -papo; Atualizar status das atividades; Escolher atividades para trabalhar; Utilizar interface de banco de dados; Controlar versionamento de código fonte.	Abrange, todas as atividades que o desenvolvedor pode desempenhar no sistema. A principal função de um desenvolvedor é a codificação, mas, para que isso seja possível, é necessário que ele escolha uma atividade na qual deve trabalhar. Estas atividades são mantidas pelo gerente, que pode ou não atribuir as tarefas a um determinado desenvolvedor.
Administrador	Todas.	O administrador pode fazer todas as atividades do sistema, além de poder ajustar alguns parâmetros de configuração.

Fonte: Idealizado pela autora, com base no trabalho de conclusão de curso da UNIVATES (SILVA, 2013).

→ **Ferramentas de comunicação -** As ferramentas de comunicação existentes no *CollabCode* permitem que a comunicação entre os membros da equipe transcorra de diversas formas. As ferramentas de comunicação síncronas servem de apoio para que todos os envolvidos possam trocar informações de forma online e no momento em que for necessário. Já as ferramentas de comunicação assíncronas ajudam a identificar as interações dos usuários com as atividades e fornecem subsídios para a documentação do sistema, o qual está sendo desenvolvido. As ferramentas de comunicação disponíveis no *CollabCode* são:

- **Bate-papo:** A solução possui um sistema de bate-papo em que todos os membros do projeto podem interagir de forma síncrona. Além disso, o sistema cria salas que apenas quem está em edição colaborativa possa utilizar;
- **Videoconferência:** O *CollabCode* dispõe de uma ferramenta que permite a conversação por áudio e vídeo. Esta ferramenta possui também opções de compartilhamento de tela e quadro branco;

- **Comentários em atividades:** Quando uma atividade é criada no *CollabCode* os membros da equipe podem trocar informações através de uma área de comentários relacionada a ela. É possível, ainda, adicionar arquivos e trechos de código-fonte.

→ **Ferramentas de Coordenação -** O gerenciamento e a coordenação em projetos DDS são tarefas que precisam funcionar corretamente para o bom andamento dos projetos. Em um *software* que se propõe a gerenciar projetos DDS, são necessárias ferramentas que possibilitem a coordenação de atividades, as quais devem permitir ao gerente de projetos, aferir métricas, acompanhar a execução de cada atividade, realocar, sempre que necessário, o recurso humano disponível, entre outras possibilidades. Como forma de proporcionar esses recursos aos gerentes de projetos, o *CollabCode* dispõe das seguintes ferramentas:

- **Gerenciamento de projetos:** A ferramenta dispõe de uma interface para a criação e o gerenciamento de projetos *web*. Ao adicionar um projeto, o usuário pode optar por importar um projeto através de repositório SVN ou criar um novo projeto. É possível, também, criar bases de dados e acompanhar estatísticas sobre o andamento do projeto;
- **Gerenciamento de atividades:** O *CollabCode* dispõe de uma interface em que é possível criar e gerenciar atividades, além de permitir seu acompanhamento através de gráficos de *Gantt*. Esta ferramenta possibilita, também, o acompanhamento da execução das atividades através de um percentual de conclusão e dos comentários adicionados pelos membros da equipe.

→ **Ferramentas de colaboração -** O *CollabCode* implementa várias ferramentas que possibilitam a colaboração entre os membros da equipe. Essas ferramentas permitem que as interações muitas vezes existentes em equipes presenciais sejam possíveis em equipes distribuídas geograficamente. As ferramentas disponíveis no *CollabCode* para a colaboração são:

- **Edição colaborativa:** A edição de código pode ser compartilhada entre os desenvolvedores. Caso dois ou mais desenvolvedores abram o mesmo arquivo, uma sessão de edição colaborativa é iniciada, permitindo a eles a edição cooperativa em tempo real do código-fonte;
- **Compartilhamento de tela:** Esta ferramenta permite que um membro da equipe possa compartilhar área de trabalho com os demais membros da equipe. A ferramenta permite,

também, que o computador que está sendo compartilhado possa ser controlado remotamente por outro membro da equipe;

- **Quadro branco:** O quadro branco é um recurso disponível na ferramenta de videoconferência. Com ele, é possível que os membros da equipe interajam em tempo real na construção de artefatos de software que facilitem a compreensão de processos e de atividades a serem desenvolvidas.

A Figura 6.5 retrata o modelo conceitual (também chamado Modelo ER, ou simplesmente MER) para análise da modelagem do *groupware CollabCode*.

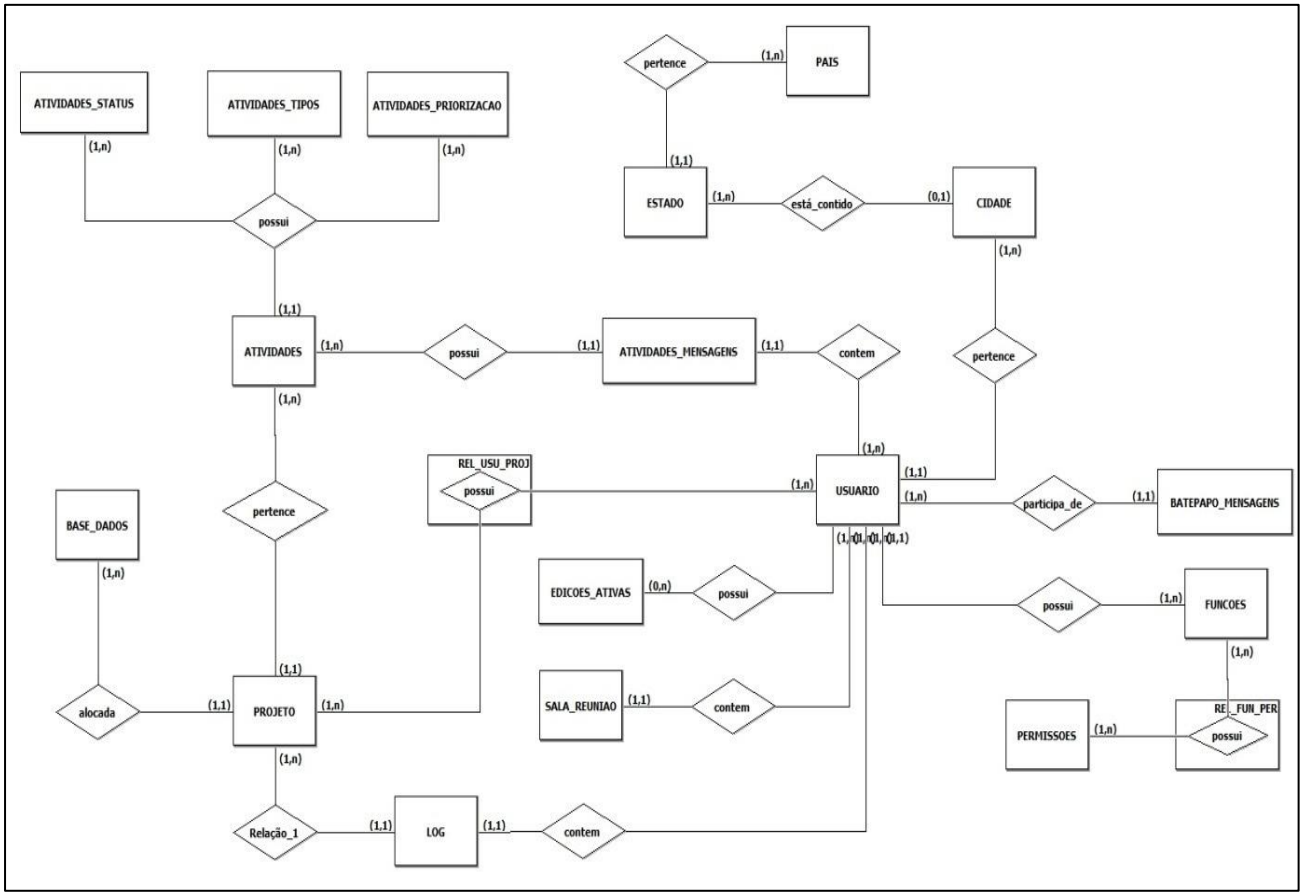


Figura 6.5 : Modelo de Entidade Relacionamento (MER) do *groupware CollabCode.*
Fonte: Idealizado pela autora, com base no trabalho de conclusão de curso da UNIVATES (SILVA, 2013).

Como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos). Por via de regra, este modelo representa de forma abstrata a estrutura que possuirá o banco de dados da aplicação. Evidentemente, o banco de dados poderá conter várias outras entidades, como

chaves e tabelas intermediárias, que podem só fazer sentido no contexto de bases de dados relacionais.

Segundo SILVA, foi criado um modelo físico que tornasse possível a implementação dos requisitos considerados indispensáveis, além de garantir que a estrutura da base possibilitasse a implementação dos requisitos desejáveis sem a necessidade de grandes modificações (SILVA, 2013). A Figura 6.6 apresenta o modelo físico (ou de implementaçâo) da base de dados do *groupware CollabCode* para fins ilustrativos.

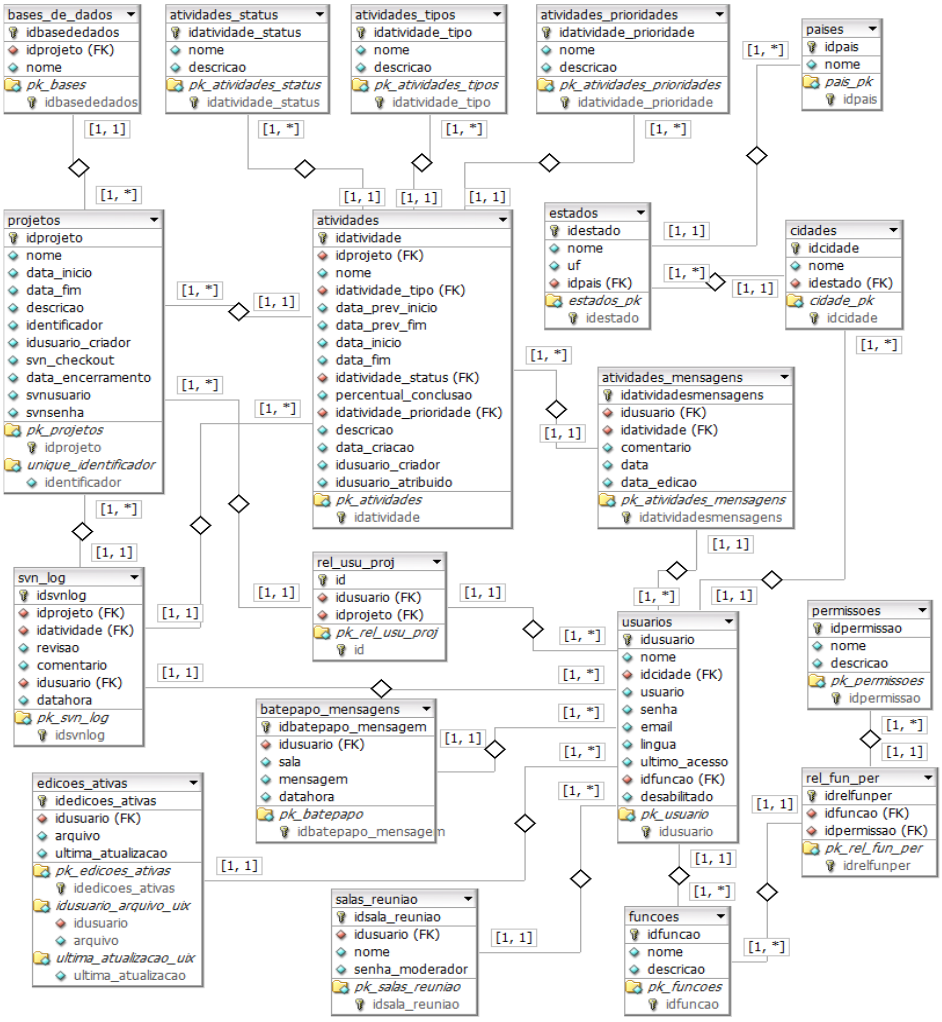


Figura 6.6 : Modelo físico da base de dados do *groupware CollabCode.*

Fonte: Trabalho de conclusão de curso da UNIVATES (SILVA, 2013).

De posse dos novos artefatos de colaboração (**Catálogo de DROPs de Colaboração Consolidado**, **Modelo de processo de uso ALFA (AP-OPL**) e das **Ontologias de Colaboração** (três subontologias de CONTO) é preciso analisar o contexto do*groupware "CollabCode".* Feito isso, deve-se executar as atividades de **01** a **05 do processo de uso ALFA (AP-OPL**). Essas atividades correspondem aos procedimentos avançados para modeladores já iniciados em modelagem ontológica (mais detalhes na seção 5.11).

☞ **Procedimentos avançados do processo de uso para o CollabCode:**

1. [**Verifique o contexto da nova modelagem**]: deve-se **e**scolher qual visão da OPL se deseja utilizar (visão geral (vide Figura 5.8) ou visão detalhada (vide Figura 5.11)). Feito isso, deve-se averiguar o seu contexto da modelagem, com o intuito de verificar qual será o seu novo ponto de partida na OPL ALFA (AP-OPL).

Observe-se que o *groupware* analisado contém em seus requisitos a definição de onde acontecerá a colaboração. Assim sendo, deve-se dar partida na parte avançada da OPL pelo novo ponto de entrada **EP1**; feito isso, deve-se escolher os POs de domínio (DROPs) que se enquadram no contexto da nova modelagem ou, se preferir, percorrer o caminho dos DROPs da linguagem, selecionando aqueles POs de domíno (DROPs) que sejam de interesse, ou seja, escolhendo aqueles que se enquadram no contexto do *CollabCode*.

2. [**Identifique as características funcionais do contexto**]: **deve-se** analisar as características fundamentais às características funcionais[35] do contexto, pois as mesmas indicam as funções essenciais do *groupware* de *Brainstorming online*.

A Tabela 6.7, a seguir, retrata as características funcionais do *groupware* analisado, as suas áreas problemáticas e, por conseguinte, suas respectivas questões de competência. Com isso, conseguimos identificar de forma preliminar os POs que serão utilizados na nova modelagem.

Tabela 6.7: Características funcionais do contexto

CONCEITOS FUNDACIONAIS	ÁREA PROBLEMÁTICA	QUESTÕES DE COMPETÊNCIA	PADRÕES (POs)
Localização da sessão colaborativa.	Sessão colaborativa	***QCs1*** - Onde acontece a sessão colaborativa?	**CSL**

[35] Funcionais - relativo a funcionalismo.

Tipo de localização.		***QCs2*** - O site da sessão colaborativa é real ou virtual?	**LT**
Tipo de sessão colaborativa.		***QCs3*** - A sessão colaborativa é de leitura ou escrita?	**CSS**
Ações de contribuição da sessão colaborativa.	Ação de contribuição	**QCs9** - Quais são as ações de contribuições que compõem uma sessão colaborativa?	**CSAC**
Ação de contribuição pré e pós-situações.		***QCs10(A)*** - Quais são as pré-situações da contribuição de ação? ***QCs10(B)*** **-** Quais são as pós-situações da contribuição de ação?	**ACPPS**
Compromisso colaborativo para ação de contribuição.	Compromissos e objetivos.	***QCs16*** **-** Qual é o compromisso colaborativo que causa a ação de contribuição?	**CCAC**
Compromissos colaborativos inerentes a um colaborador.		***QCs15*** **-** Qual são compromissos colaborativos que são inerentes a um colaborador?	**CCIC**
Objetivo como conteúdo proposicional do compromisso colaborativo.		***QCs17*** **-** Qual é o objetivo que é o conteúdo proposicional de um compromisso colaborativo?	**GPCCC**
Recursos da sessão colaborativa.	Recursos	***QCs26*** **-** Quais são os recursos que participam da sessão colaborativa?	**CRS**
Natureza de recurso.		***QCs29*** **-** Esse recurso pode ser compartilhável ou exclusivo?	**RN**
Interação de comunicação simples.	Comunicação	***QCs30 (A)*** - Quem é o remetente na interação comunicativa? ***QCs30(B)*** - Quem é o receptor na interação comunicativa? ***QCs30(C)*** **-** Qual é a mensagem da interação comunicativa?	**CIPS**
Projeto_leitura, Projeto_escrita.	-	-	-
Versão_pré, Versão_pós.	-	-	-
Projeto.	-	-	-
Ambiente_baseado_na_web.	-	-	-
Sala_reunião.	-	-	-
Escrever_objetivo_do_projeto.	-	-	-
Configuração_do_projeto, Documentação_de_projeto.	-	-	-
Escrever_compromisso.	-	-	-
Atividade_de_projeto.	-	-	-
Gerar_priorização, Gerar_planejamento, Gerar_execução .	-	-	-
Usuário.	-	-	-

Perfil .	-	-	-
Permissões_usuário.	-	-	-
Funções.	-	-	-
Histórico.	-	-	-
Desenvolvedor, Analista, administrador, Gerente.	-	-	-

Fonte: Idealizado pela autora.

3. [**Esboçe seus POs de domínio no OLED Estendido**]: primeiro, deve-se esboçar o seu PO

(origem) o PO(**CSL**). Note-se que o PO(**CSL**) origem possui as seguintes Classes OntoUML (mais detalhes na seção 4.7): **Sessão Colaborativa** (*Collaborative Session*), **Ambiente** (*Site*) e **Localização** (*Location*). Agora, vamos combinar (ou integrar) o PO (**LT**) ao PO (**CSL**) origem. Eles são (POs de domínio): sessão_colaborativa, localização, sessão_de_leitura, sessão_de_escrita, recurso, partição de recurso, ação_de_contribuição, compromisso_colaborativo, objeto, recepção, destinatário, emissor, ato_comunicativo, interação_comunicativa, projeto_leitura, projeto_escrita, versão_pré, versão_pós, ambiente_baseado_na_web, escrever_objetivo_do_projeto, configuração_do_projeto, documentação_de_projeto, escrever_compromisso, funcões, desenvolvedor, analista, administrador e gerente.

4. [**Verifique os elementos candidatos a Classe identificados**]: um *groupware* possui

diferentes elementos necessários à criação de aplicações colaborativas que podem ser candidatos a Classe. As classes são utilizadas para definir o comportamento de objetos do mundo real, segundo a perspectiva ou a abstração que interessa para a solução de um problema específico. Nesse contexto, os elementos candidados a Classe podem ser classificados em **enquadrados** e **não enquadrados.**

No primeiro caso, verifica-se se o elemento candidado a Classe escolhido se enquadra com outro PO descrito no catálogo. Em outras palavras, se o elemento canditado a Classe pode ser enquadrado com uma das intenções ou questões de competência de outro PO existente no catálogo consolidado. Em caso positivo, deve-se fazer uso do PO identificado,

ou se necessário, criar um PO variante. Já no segundo caso, deve-se estereotipar o elemento candidado a Classe, a fim de adicioná-lo a outro PO (já identificado na modelagem).

5. [**Estereotipe os elementos candidatos a Classe identificados**]: os elementos candidatos a Classe que não foram enquadrados com outros POs descritos no Catálogo Consolidado devem ser estereotipados e acrescidos aos POs existentes. É importante lembrar que os elementos candidatos a Classe, para serem acrescidos, ou melhor, associados a um modelo conceitual ontológico específico de *groupware,* devem ser estereotipados de acordo com as metapropriedades ontológicas (mais detalhes na seção 4.3.1). A Tabela 6.8 exibe os elementos candidatos a Classe identificados e os respectivos estereótipos em ONTOUML.

Tabela 6.8: Estereótipos das Classes identificados

ELEMENTOS CANDIDATOS A CLASSE	ESTERIÓTIPO DA CLASSE ONTOUML
PROJETO_LEITURA, PROJETO_ESCRITA	<<role>><<role>>
VERSÃO_PRE, VERSÃO_POS	<<subkind>><<subkind>>
PROJETO	<<mixin>>
AMBIENTE_BASEADO_NA_WEB	<<subkind>>
SALA_REUNIÃO	<<subkind>>
ESCREVER_OBJETIVO_DO_PROJETO	<<subkind>>
CONFIGURAÇÃO_DO_PROJETO, DOCUMENTACAO_DE_PROJETO	<<rolemixin>><<rolemixin>>
ESCREVER_COMPROMISSO	<<subkind>>
ATIVIDADE_DE_PROJETO	<<subkind>>
GERAR_PRIORIZAÇÃO, GERAR_PLANEJAMENTO, GERAR_EXECUÇÃO	<<phase>><<phase>><<phase>>
USUÁRIO	<<subkind>>
PERFIL	<<kind>>
PERMISSÕES_USUÁRIO	<<kind>>
FUNÇÕES	<<kind>>
BASE_DE_DADOS	<<kind>>
ESTADOS	<<kind>>
CIDADES	<<kind>>
PAÍS	<<kind>>
HISTÓRICO	<<kind>>
DESENVOLVEDOR,ANALISTA, ADMINISTRADOR, GERENTE	<<role>><<role>><<role>><<role>>

Nota. Os elementos candidatos a Classe foram escritos em letras maiúsculas e sem acentuação. Já os esteriótipos das Classes ONTOUML foram mantidos em inglês por ser essa a língua dos padrões da ferramenta OLED

Fonte: Idealizado pela autora.

A Tabela 6.9 exibe o mapeamento entre as classes do modelo conceitual ontológico, a ontologia e o modelo de entidade relacionamento do CollabCode.

Tabela 6.9: Mapeamento entre Classe e diagrama MER

CLASSE (FÍGURA 6.7)	ENTIDADE (FÍGURA 6.5)
SITE, SITE_REAL/ SALA_REUNIAO, VIRTUAL / AMBIENTE _BASEADO_NA_WEB	SALAS_REUNIÃO
SESSÃO COLABORATIVA	-
LOCALIZAÇÃO	-
SESSAO_DE_LEITURA, SESSÃO_DE_ESCRITA	-
RECURSO	-
PARTICAO DE RECURSO	-
SITUAÇÃO	EDICOES_ATIVAS
AÇÃO_DE_CONTRIBUIÇÃO	-
COMPROMISSO_COLABORATIVO	-
OBJETO	-
COLABORADOR / USUÁRIO	USUÁRIOS
MENSAGEM	BATE_PAPO_MENSAGENS
RECEPÇÃO	-
DESTINATÁRIO, EMISSOR	-
ATO_COMUNICATIVO	-
INTERAÇÃO_COMUNICATIVA	-
PROJETO_LEITURA, PROJETO_ESCRITA	-
VERSAO_PRE, VERSÃO_POS	-
PROJETO / RECURSO	PROJETOS
AMBIENTE_BASEADO_NA_WEB	-
ESCREVER_OBJETIVO_DO_PROJETO	-
CONFIGURAÇÃO_DO_PROJETO, DOCUMENTAÇÃO_DE_PROJETO	
ESCREVER_COMPROMISSO	REL_USU_PROJ
ATIVIDADE_DE_PROJETO	ATIVIDADES
GERAR_PRIORIZAÇÃO, GERAR_PLANEJAMENTO, GERAR_EXECUÇÃO	ATIVIDADES_STATUS, ATIVIDADES_TIPOS, ATIVIDADES_PRIORIDADES, ATIVIDADE_DE_MENSAGENS

PERFIL	REL_FUN_PER
PERMISSOES_USUÁRIO	PERMISSÕES
FUNÇÕES	FUNÇÕES
HISTÓRICO	SVN_LOG
DESENVOLVEDOR,ANALISTA, ADMINISTRADOR, GERENTE	-
BASE_DE_DADOS	BASE_DE_DADOS
ESTADOS	ESTADOS
CIDADES	CIDADES
PAÍS	PAÍS

Nota. As Classes e Entidades foram escritas em letras maiúsculas e sem acentuação.

Fonte: Idealizado pela autora.

A Tabela 6.10 exibe o comparativo entre o diagrama MER obtido sem e com a utilização de ontologias no *CollabCode*.

Tabela 6.10: Comparação entre entidades

ENTIDADE SEM ONTOLOGIA (FÍGURA 6.5)	ENTIDADE COM ONTOLOGIA (FÍGURA x)
SALAS_REUNIÃO	SALAS_REUNIÃO
EDICOES_ATIVAS	SITUAÇÃO
USUÁRIOS	COLABORADOR /USUARIOS
BATE_PAPO_MENSAGENS	MENSAGEM
PROJETOS	PROJETO
ATIVIDADES	ATIVIDADE_PROJETO
ATIVIDADES_STATUS, ATIVIDADES_TIPOS, ATIVIDADES_PRIORIDADES, ATIVIDADE_DE_MENSAGENS	GERAR_PRIORIZAÇAO, GERAR_PLANEJAMENTO, GERAR_EXECUÇÃO
FUNÇÕES	FUNÇÕES
PERMISSÕES	PERMISSÕES_USUARIO
SVN_LOG	HISTÓRICO
BASE_DE_DADOS	BASE_DE_DADOS
ESTADOS	ESTADOS
CIDADES	CIDADES
PAIS	PAIS
REL_USE_PROJ	ESCREVER_COMPROMISSO
REL_FUN_PER	PERFIL

Nota. As Entidades sem e com ontologia foram escritas em letras maiúsculas e sem acentuação.

Fonte: Idealizado pela autora.

A Figura 6.7 retrata o modelo específico do *groupware* de "*Brainstorming online*". Esse modelo foi concebido segundo o método ÔMEGA e seu respectivo modelo de processo de uso OPL ALFA (AP-OPL). Nele, identificamos os POs de domínio (DROPs) utilizados, os elementos candidatos a Classe estereotipados e associados aos POs existentes (já identificados na modelagem).

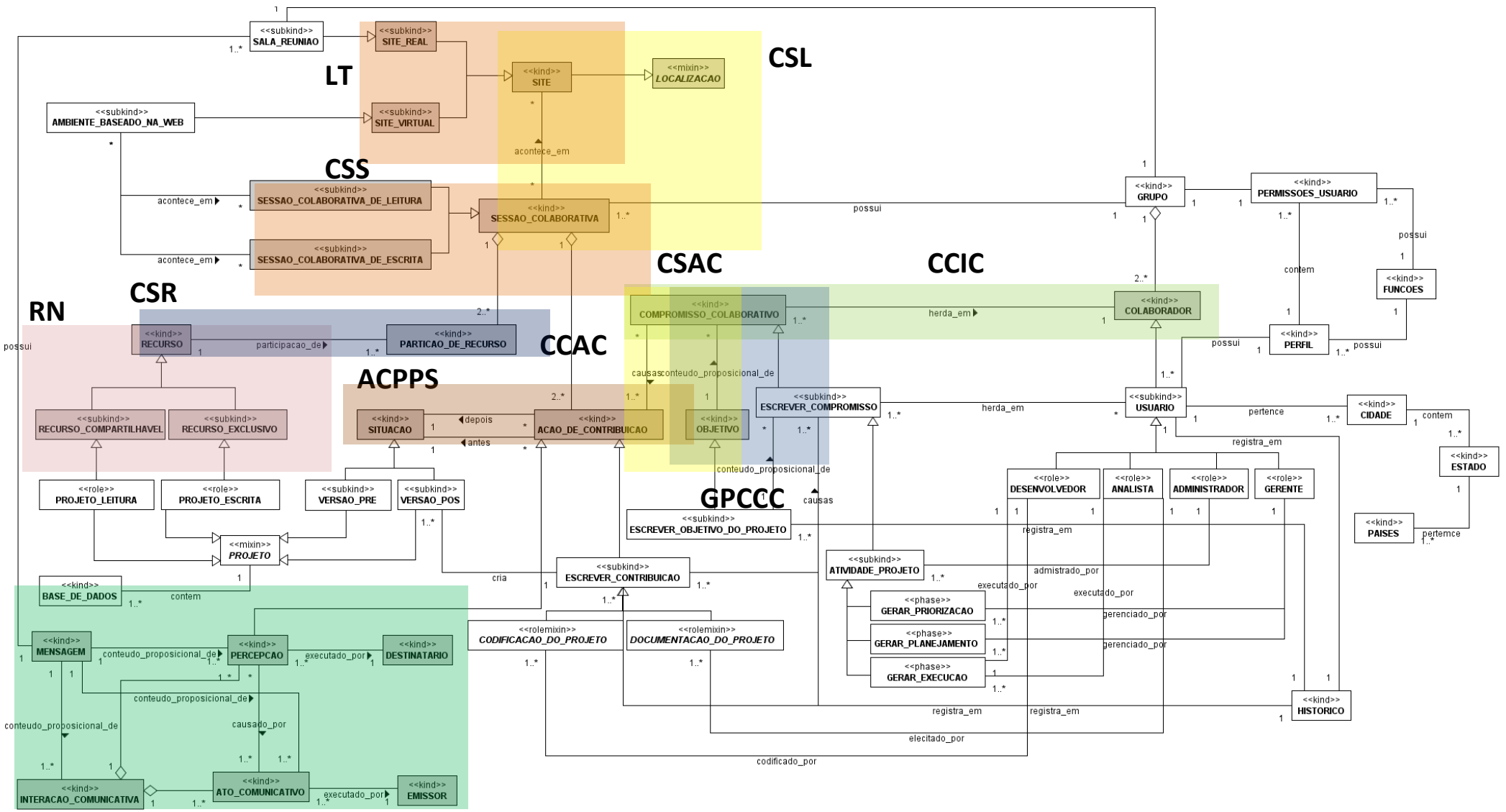


Figura 6.7: Modelo conceitual específico do CollabCode

Fonte: Idealizada pela autora.

Verificação sintática do modelo no OLED: "Modelo conceitual específico do groupware COLLABCODE"

A Figura 6.8 retrata a verificação sintática do modelo conceitual específico do *groupware* de "*CollabCode*" criado no OLED.

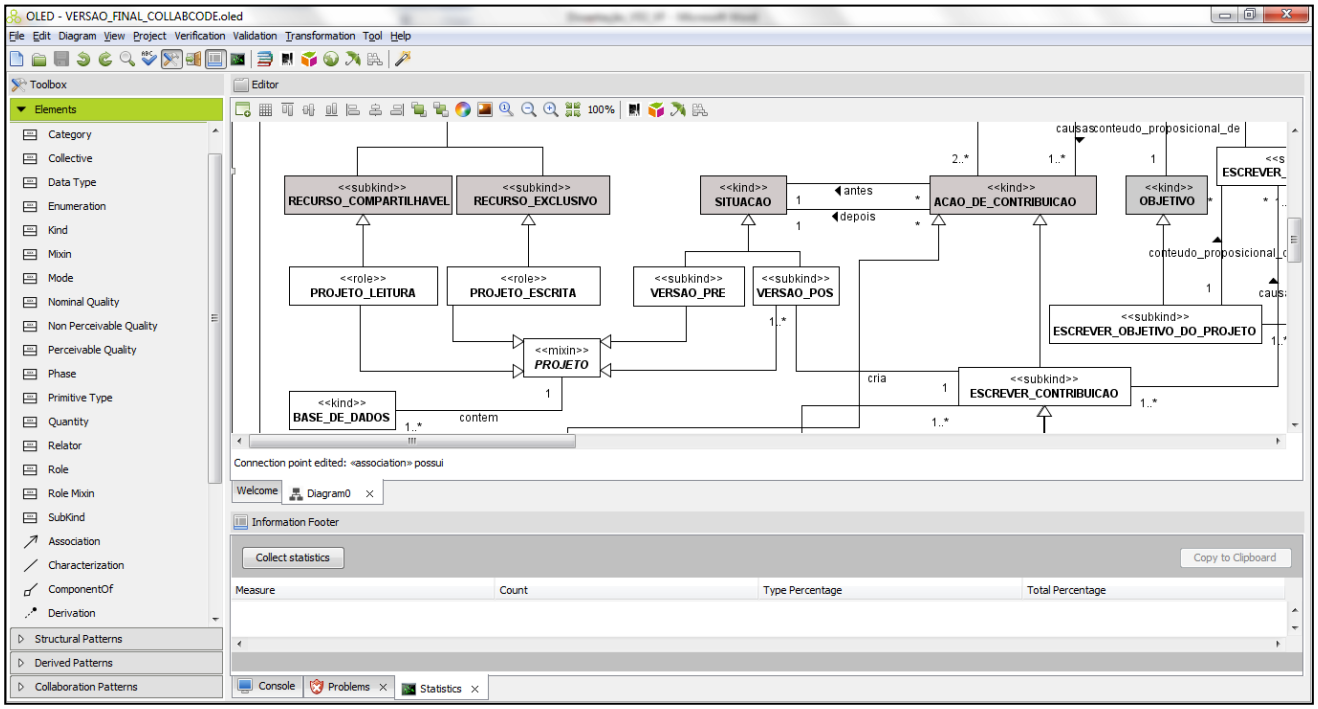


Figura 6.8 : Verificação sintática do groupware CollabCode

Fonte: Idealizado pela autora.

O procedimento de verificação sintática consiste no mapeamento do modelo de "*CollabCode*" criado, submetido ao teste de verificação sintática disponível no OLED. Cabe salientar que, os resultados errôneos retornados pela ferramenta devem ser ajustados. As etapas de teste e correção são repetidas até que o modelo esteja livre de erros. Ao final, obtém-se um modelo ontologicamente adequado e semanticamente refinado.

É importante ressaltar que a execução de verificações sintáticas são baseadas em critérios ontológicos para comparar o modelo submetido "*CollabCode*" às restrições ontológicas do metamodelo da OntoUML. Nesse âmbito, é assegurado que o modelo conceitual verificado será sintaticamente correto.

CAPÍTULO 7. ESTUDO EXPERIMENTAL PARA AVALIAÇÃO DO MÉTODO E MODELO DE PROCESSO DE USO

7.1 PLANEJAMENTO DA AVALIAÇÃO

No âmbito da SE, os estudos experimentais têm sido usados para encontrar indícios e aprimorar a utilização de técnicas no contexto de projetos de software (TRAVASSOS; GUROV, 2002). No âmbito deste trabalho, foram utilizados dois estudos de caso em que as abordagens qualitativas e quantitativas (ou mesmo, quali-quantitativas) foram empregadas para a obtenção de uma análise mais profunda do assunto da pesquisa. Desse modo, a pesquisa teve uma parte quantitativa que reuniu dados que puderam ser codificados de forma numérica. E uma parte qualitativa, para especular quais as causas dos resultados.

O **procedimento** de condução do estudo teve três etapas. Na **primeira etapa**, a autora desta dissertação explicou o contexto do estudo aos participantes; a documentação necessária foi disponibilizada para uso deles, compreendendo uma listagem com a descrição resumida de cada padrão ontológico presente na OPL, os diagramas de processo e descrições, e a descrição detalhada de cada padrão (contendo nome, intenção, questões de competência, modelo conceitual e axiomatização) para testar a abordagem metodológica do ÔMEGA.

Na **segunda etapa**, os participantes do estudo preencheram um questionário online (*survey*[36]). Durante essa etapa, não houve intervenção por parte da pesquisadora. A **terceira etapa** consistiu na aplicação de um questionário para registrar a percepção dos participantes sobre o uso do ÔMEGA e seu respectivo modelo de processo de uso ALFA (AP-OPL). Após receber as respostas do questionário, foram realizadas entrevistas, a fim de se obter informações mais detalhadas sobre a experiência de uso dos participantes.

A **instrumentação utilizada** na condução do estudo constou do seguinte: procedimento de realização do *survey*, procedimento de realização das entrevistas e procedimento para analisar os resultados.

→ O **procedimento de realização do *survey*** consistiu na aplicação de um questionário online contendo questões sobre:

[36] survey - Um *survey* é um estudo experimental realizado em retrospecto quando, por exemplo, uma nova tecnologia ou método é utilizado e deseja-se obter informações que permitam avaliá-los e evoluí-los [Pfle00].

- O perfil profissional do participante;
- Experiência em modelagem conceitual;
- Experiência em desenvolvimento de ontologias;
- Experiência em linguagem de padrões ontológicos (OPL);
- Avaliação do método ÔMEGA e modelo de processo de uso ALFA (AP-OPL);
- O questionário foi dotado de questões objetivas em que o participante pôde justificar sua resposta, expressando sua opinião.

O questionário online (*survey*) realizado pode ser visto na íntegra no Apêndice **C** desta dissertação.

→ **O procedimento de realização das entrevistas** consistiu na aplicação de um questionário, por escrito, aos participantes do experimento, com a finalidade de obter informações adicionais, percepção e esclarecimentos sobre comentários registrados nos questionários respondidos. As perguntas realizadas nas entrevistas são apresentadas no Apêndice (**D**) desta dissertação.

→ **No procedimento para analisar os resultados** foram considerados quatro indicadores:

1- Produtividade do desenvolvimento de modelagem conceitual ontológica;

2- Qualidade da modelagem resultante;

3- Grau de dificuldade para entendimento do ÔMEGA e dos símbolos da ALFA (AP-OPL);

4- Ausência de problemas de representação da notação ALFA (AP-OPL).

O **objetivo da pesquisa** executada foi avaliar a contribuição que o método ÔMEGA e seu respectivo modelo de processo de uso ALFA (AP-OPL) trouxeram para a construção de modelos conceituais de *groupware* baseados em padrões ontológicos de colaboração (avaliação dos resultados). No contexto desse trabalho, os participantes tiveram acesso a:

- Um documento contendo alguns tópicos dos capítulos (4, 5 e 6) para contextualizar e guiar os participantes da avaliação;
- Uma versão da ferramenta para testar a abordagem metodológica do ÔMEGA;
- Um questionário online (*survey*) para encontrar indícios que permitissem avaliar os resultados obtidos através do uso do ÔMEGA e seu respectivo modelo de processo de uso ALFA (AP-OPL). Esse formulário foi criado com auxílio do **google drive** (*forms*)[37] (vide apêndice **C**).

[37] https://www.google.com/forms/about/.

- Um documento contendo a entrevista, a fim de se obter informações mais detalhadas sobre a experiência de uso do método ÔMEGA por parte dos participantes (vide apêndice **D**).

7.2 RESULTADOS OBTIDOS

Nesta seção, são apresentados os resultados obtidos a partir do questionário online respondido pelos participantes da avaliação de "**Um método baseado em padrões ontológicos de colaboração para construção de modelos conceituais de *groupware***". Os participantes do estudo foram 16 profissionais de tecnologia da informação (TI). O primeiro tópico do questionário *survey* foi idealizado para identificar o perfil profissional do participante do estudo. Cabe destacar que as primeiras perguntas do *survey* são relativas aos dados pessoais dos participantes. Essas informações foram mantidas em anonimato na publicação dos resultados da pesquisa.

A primeira pergunta a ser considerada é a "Situação atual de formação acadêmica do participante do estudo". A Figura 7 retrata o status atual de formação acadêmica dos participantes. Nela, obtivemos que 87,9% dos participantes tinham formação em TI (completa) e 12,5% tinham formação em TI (incompleta).

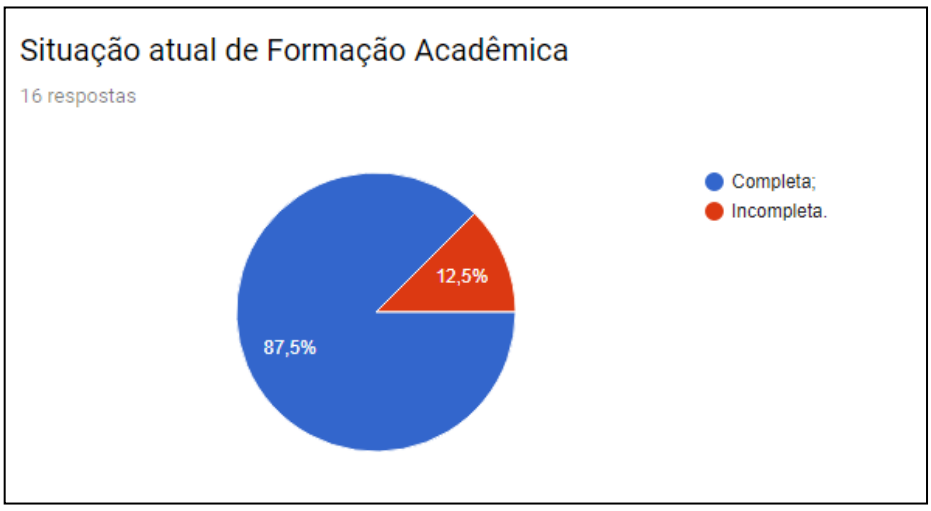


Figura 7 : Situação atual de formação acadêmica
Fonte: Idealizado pela autora

A segunda pergunta a ser considerada é o "Grau de formação acadêmica do participante do estudo". A Figura 7.1 retrata o grau de formação acadêmica dos participantes. Nela, obtivemos 50% de especialistas; 31,9% de mestres; 6,3% de doutores; 6,3% de phd e 6,3% de superior (graduação).

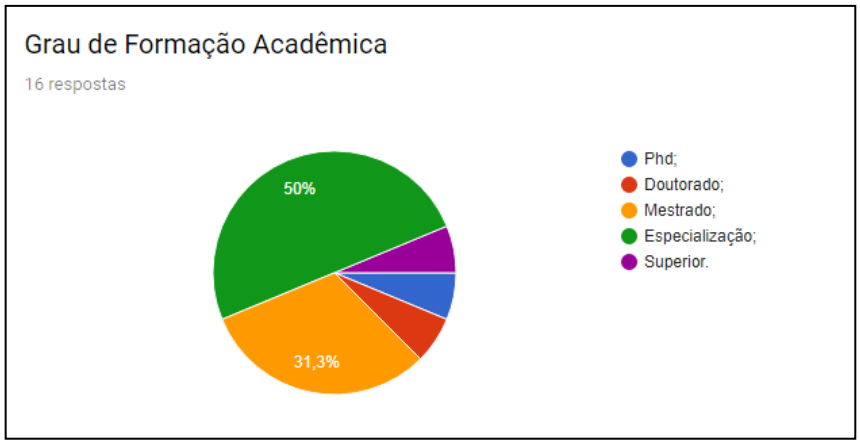


Figura 7.1: Grau de formação acadêmica
Fonte: Idealizado pela autora

O segundo tópico do questionário *survey* foi idealizado para identificar o nível de experiência em modelagem conceitual do participante do estudo. A primeira pergunta do segundo tópico é "Qual seu grau de experiência em modelagem conceitual?". A Figura 7.2 retrata o grau de experiência em modelagem conceitual. Nela, obtivemos que 43,8% têm alto grau (mais de três anos de experiência); 43,8% têm médio grau (entre um e três anos de experiência); 12,3% têm baixo grau (menos de 1 ano de experiência);

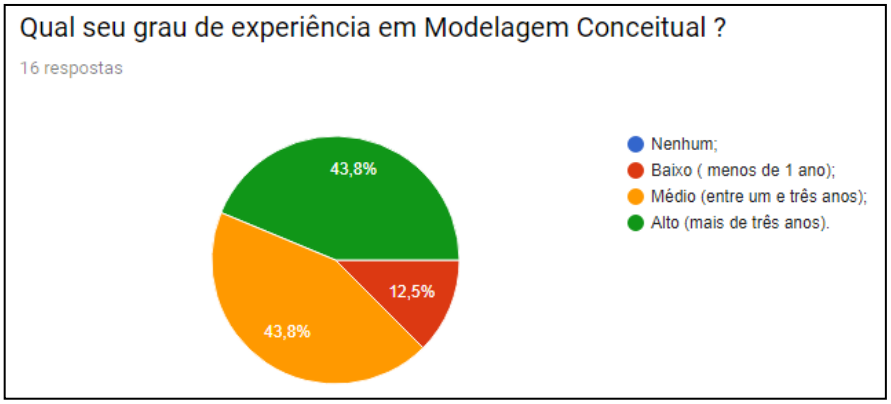


Figura 7.2: Grau de experiência em modelagem conceitual
Fonte: Idealizado pela autora

A segunda pergunta do segundo tópico é: "Caso possua experiência em Modelagem Conceitual, de que forma a mesma foi adquirida?". A Figura 7.3 retrata de que forma a experiência foi adquirida por parte dos participantes do estudo. Nela, obtivemos que 43,8% obtiveram a experiência em projetos; 31,3% em cursos e 25% em disciplinas.

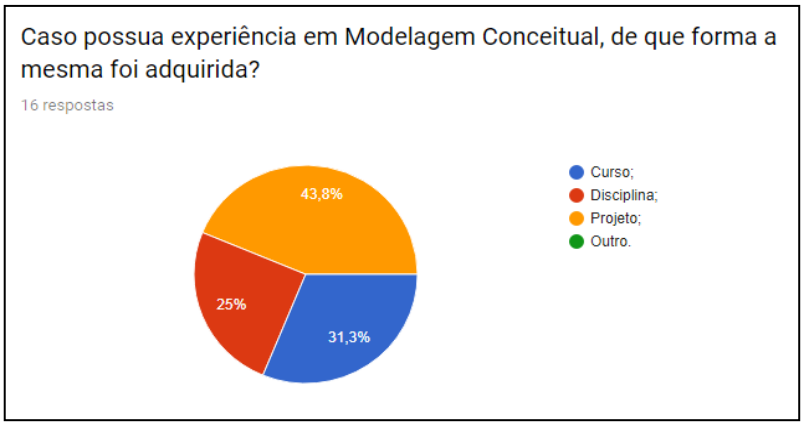


Figura 7.3: De que forma a experiência em modelagem conceitual foi adquirida
Fonte: Idealizado pela autora

A terceira questão do segundo tópico é: “Especifique abaixo o “Curso”, a “Disciplina” e/ou o “Projeto” que você indicou na questão anterior. Cite também a duração do(s) mesmo(s)?”. Dos 16 participantes do estudo, 7 participantes obtiveram experiência através de projetos, 5 participantes obtiveram experiência através de cursos e 4 participantes obtiveram experiência através de disciplinas. Em relação à duração, a mesma varia de 40 horas a 5 anos.

O terceiro tópico do questionário *survey* foi idealizado para identificar o nível de experiência em desenvolvimento de ontologias do participante do estudo. A primeira pergunta do terceiro tópico é “Qual seu grau de experiência em desenvolvimento de Ontologias?”. A Figura 7.4 retrata de que forma a experiência foi adquirida. Nela, obtivemos que 31,5% têm baixo grau (menos de 1 anos); 31,3% não tem conhecimento algum; 25% têm médio grau (entre um a três anos) e 6,3% têm alto grau (mais de três anos).

Figura 7.4: Grau de experiência em desenvolvimento de ontologias
Fonte: Idealizado pela autora

A segunda pergunta do terceiro tópico é “Caso você possua experiência anterior no desenvolvimento de ontologias, de que forma a mesma foi adquirida?”. A Figura 7.5 retrata de que forma a experiência foi adquirida por parte dos participantes do estudo. Nela, obtivemos que 54,5% a adquiriram em disciplina e 45,5% em projetos.

Figura 7.5: Experiência anterior em desenvolvimento de ontologias
Fonte: Idealizado pela autora

A terceira questão do terceiro tópico é: “Especifique abaixo o “Curso”, a “Disciplina” e/ou o “Projeto” que você indicou na questão anterior. Cite também a duração do(s) mesmo(s)”. Dos 16 participantes do estudo, 6 participantes obtiveram experiência através de disciplinas, 5 participantes obtiveram experiência através de projetos e 5 participantes não comentaram. Em relação à duração, a mesma varia de 4 meses a 3 anos.

O quarto tópico do questionário *survey* foi idealizado para identificar o nível de experiência em linguagem de padrões ontológicos (OPL). A primeira pergunta do quarto tópico é “Qual seu grau de experiência em linguagem de padrões ontológicos (OPL)?”. A Figura 7.6 retrata de que forma a experiência foi adquirida por parte dos participantes do estudo. Nela, obtivemos que 50% têm nenhum conhecimento; 31,3% têm baixo grau (menos de 1 ano); 12,5% têm médio grau (entre um a três anos) e 6,3% têm alto grau (mais de três anos).

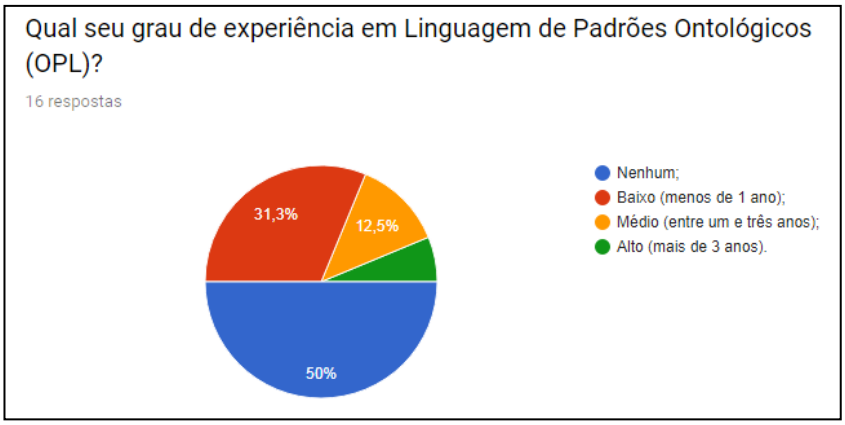


Figura 7.6: Grau de experiência em linguagem de padrões ontológicos (OPL)
Fonte: Idealizado pela autora

A segunda pergunta do quarto tópico é “Caso você possua experiência anterior em OPL, de que forma a mesma foi adquirida?”. A Figura 7.7 retrata de que forma a experiência foi adquirida por parte dos participantes do estudo. Nela, obtivemos que 75% a obtiveram em curso e 25% em projeto.

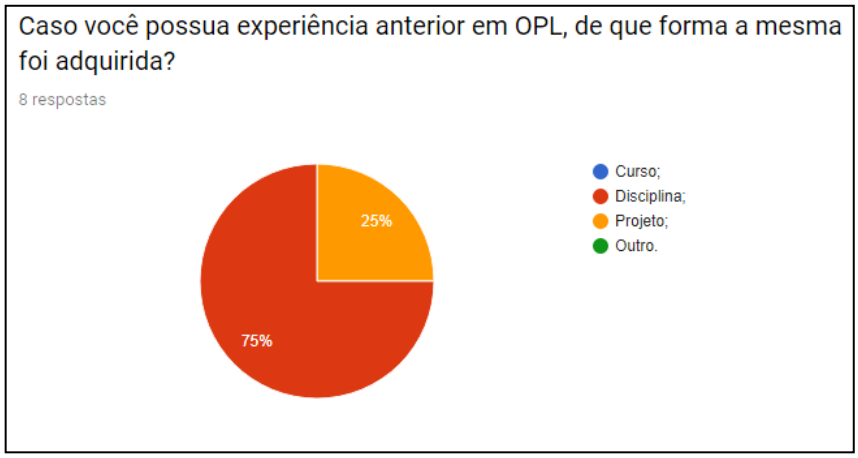


Figura 7.7: De que forma foi adquirida a experiência em linguagem de padrões ontológicos (OPL)
Fonte: Idealizado pela autora

A terceira questão do quarto tópico é: “Especifique abaixo o “Curso”, a “Disciplina” e/ou o “Projeto” que você indicou na questão anterior. Cite também a duração do(s) mesmo(s)". Dos 16 participantes do estudo, 6 participantes obtiveram experiência através de disciplinas, 2 participantes obtiveram experiência através de projetos e 8 participantes não comentaram. Em relação à duração, a mesma varia de 4 meses a 3 anos.

O quinto tópico do questionário *survey* foi idealizado para avaliar o método ÔMEGA e seu respectivo modelo de processo de uso ALFA (AP-OPL). A primeira pergunta do quinto

tópico é “Quão difícil/fácil você considera entender o mecanismo de funcionamento do método ÔMEGA?”. A Figura 7.8 retrata o grau de dificuldade para entendimento do mecanismo de funcionamento do ÔMEGA e do modelo de processo de uso ALFA (AP-OPL). Nela, obtivemos que 37,5% consideram fácil; 31,3% foram imparciais (neutros); 18,8% consideram muito fácil e 12.5% consideram difícil.

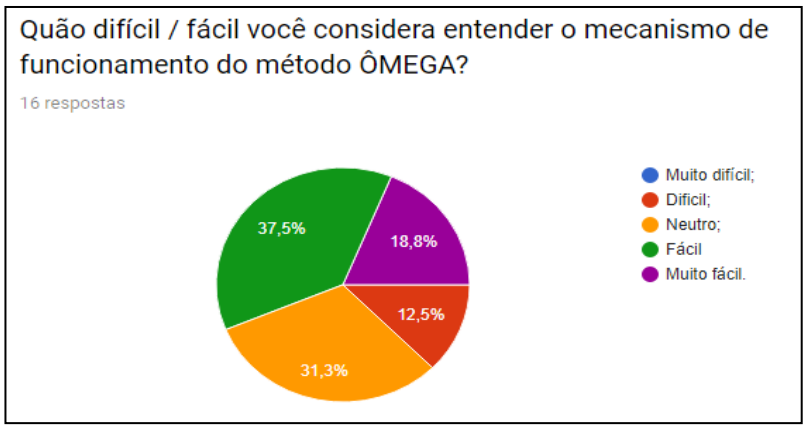


Figura 7.8: Quão difícil/fácil você considera entender o mecanismo de funcionamento do ÔMEGA
Fonte: Idealizado pela autora

A segunda pergunta do quinto tópico é “Por favor, justifique sua resposta”. Dos 16 participantes do estudo, 6 participantes consideram que o método recupera os padrões já validados; portanto, fica fácil associar o problema do mundo real em estudo ao padrão que melhor se adequa à sua representação; 3 participantes consideram que o método é muito fácil e intuitivo; 5 participantes foram imparciais (neutros); 2 participantes não têm conhecimento sobre o método.

A terceira pergunta do quinto tópico é “Quão difícil você considera entender o mecanismo de funcionamento da OPL que corresponde ao "Modelo de Processo de Uso ALFA (AP-OPL)?”. A Figura 7.9 retrata o resultado desta questão. Nela, obtivemos que 43,8% consideram fácil o entendimento; 25% consideram muito fácil; 18,8% foram imparciais (neutros) e 12,5% consideram difícil.

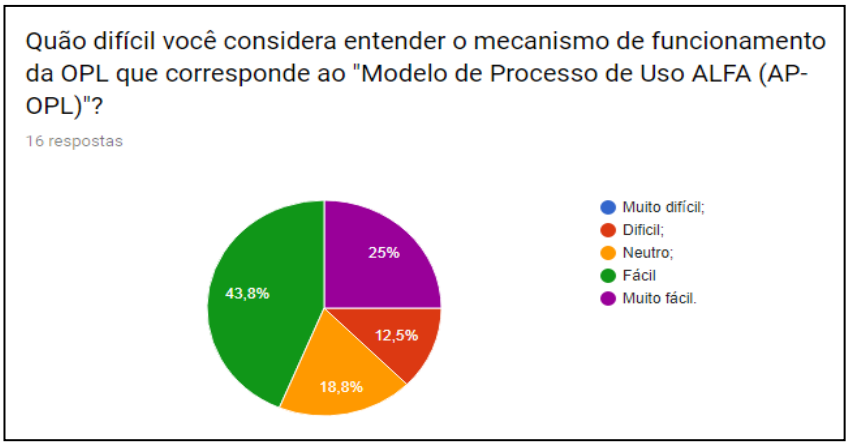


Figura 7.9: Quão difícil você considera entender o mecanismo de funcionamento da OPL
Fonte: Idealizado pela autora

A quarta pergunta do quinto tópico é: “Por favor, justifique sua resposta”. Dos 16 participantes do estudo, 7 participantes consideram interessante a forma como foram abordadas as análises dos pontos de entrada no modelo de processos, permitindo um encadeamento de todas as fases da modelagem, e a forma como foram “linkados” os procedimentos introdutórios e os avançados. Já 4 participantes consideram muito fácil o mecanismo de funcionamento da OPL, pois o entenderam na primeira leitura; 3 participantes foram imparciais (neutros) e 2 participantes consideram difícil.

A quinta pergunta do quinto tópico é “Quão difícil/fácil você considera entender os elementos da ALFA (AP-OPL)?”. A Figura 7.10 retrata o grau de entendimentos dos elementos da ALFA (AP-OPL). Nela, obtivemos que 31,3% consideram fácil o entendimento; 31,3% foram imparciais (neutros); 25% consideram muito fácil e 12,5% consideram difícil.

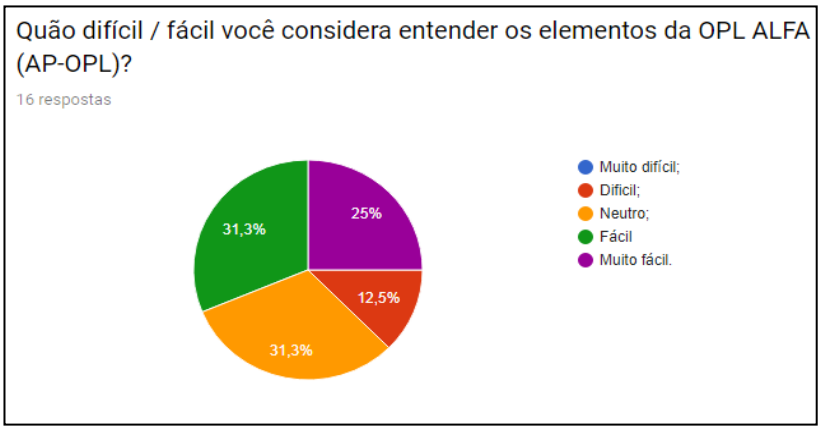


Figura 7.10: Quão difícil/fácil você considera entender os elementos da OPL
Fonte: Idealizado pela autora

A sexta questão do quinto tópico é: “Por favor, justifique sua resposta”. Dos 16 participantes do estudo, 5 participantes consideram que os elementos estão bem detalhados; 5 participantes foram imparciais (neutros); 5 participantes consideram muito fácil para pessoas com um mínimo conhecimento em colaboração. Neste caso, somente 1 participante considera confuso, por não ter conhecimento algum de *groupware* /colaboração.

A oitava pergunta do quinto tópico é “Dentre os elementos listados anteriormente, qual deles você teve maior dificuldade para entender? Justifique sua resposta”. Dos 16 participantes do estudo, somente 1 participante cita que precisava de mais tempo de uso para opInar com confiança, pois dúvidas ou detalhes adicionais podem ser representados ao longo do processo de uso.

A nona pergunta do quinto tópico é “Você considera que essa versão ALFA (AP-OPL) tornou o processo mais claro?”. A Figura 7.11 retrata que a versão da ALFA (AP-OPL) tornou o processo de modelagem ontológica mais claro. Nela, obtivemos que 50% consideram o processo mais claro; 31,3% consideram bem mais claro e 18,8% consideram que a nova versão mantém a mesma clareza da versão anterior.

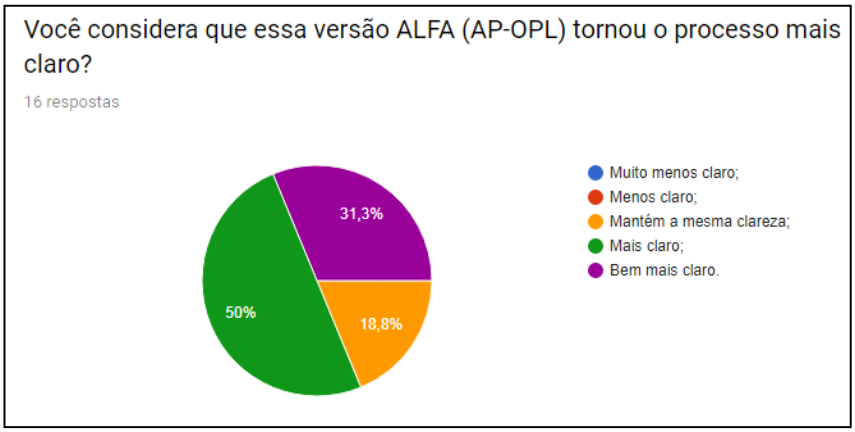


Figura 7.11: Essa versão ALFA (OP_OPL) tornou o processo mais claro
Fonte: Idealizado pela autora

A décima questão do quinto tópico é “Por favor, justifique sua resposta”. Dos 16 participantes do estudo,11 consideram que a versão da ALFA (AP-OPL) destaca os elementos importantes que ajudam a organizar e representar os principais requerimentos de um *groupware*. Também citaram que, visualmente, o processo ficou bem mais fácil de ser lido e entendido, o que é essencial para a sua aplicação.

A décima primeira questão do quinto tópico é “Em relação à facilidade de entendimento do processo da OPL, qual representação você julga melhor?”. A Figura 7.12 retrata que 100% dos participantes do estudo julgam que a nova versão contempla diferentes níveis de abstração, permitindo trabalhar com uma visão geral do processo, com mais detalhes, que podem ser trabalhados no todo.

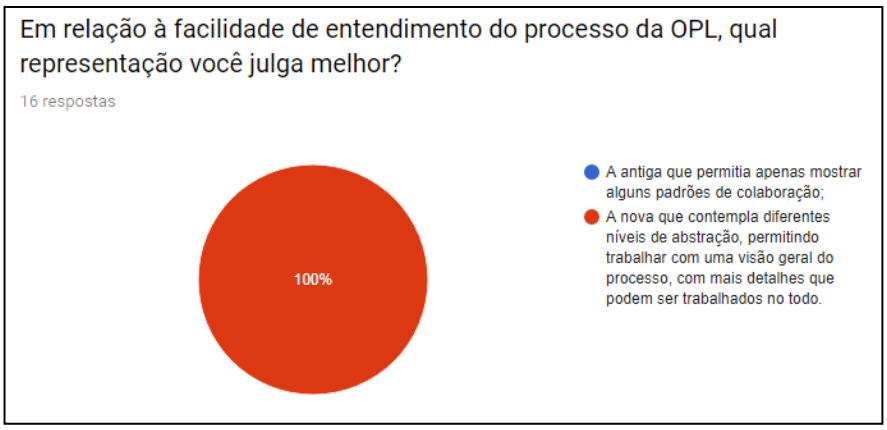


Figura 7.12: Facilidade de entendimento do processo da OPL
Fonte: Idealizado pela autora

A décima segunda questão do quinto tópico é “Por favor, justifique sua resposta”. Segundo os 16 participantes do estudo, trabalhar com uma visão geral do processo permite uma modelagem mais completa, pois a mesma facilita o entendimento da solução a ser adotada. Também citaram que os diferentes níveis de abstração do método permitem um entendimento mais apurado do escopo do problema a ser resolvido.

A décima terceira pergunta do quinto tópico é "Em sua avaliação, no geral, o método ÔMEGA contribui com o processo de Modelagem Conceitual Ontológica (Por exemplo, vide modelo conceitual do Brainstorming online - abaixo descrito)?". A Figura 7.13 retrata que o método ÔMEGA contribuiu com o processo de Modelagem Conceitual Ontológica. Nela, obtivemos que 62,5% concordam que o método é aplicável e eficaz. E 37,6% dos participantes do estudo acham que seria necessário mais tempo de uso para perceber mais nitidamente uma possível contribuição.

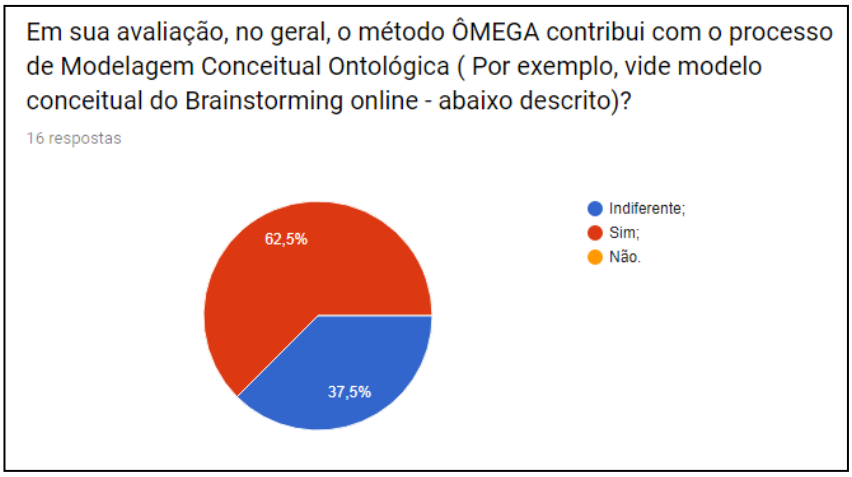


Figura 7.13: Avaliação geral do método ÔMEGA

Fonte: Idealizado pela autora

7.3 PERCEPÇÃO DOS PARTICIPANTES

Nesta seção, são apresentados os principais resultados obtidos, após a aplicação do questionário por escrito, em entrevista com os participantes do estudo. Segundo a opinião geral, o método ÔMEGA auxilia bastante, na medida em que permite a modelagem em etapas, facilitando o entendimento do escopo e a cobertura do problema a ser resolvido em *groupware*. Também foi citado que o ÔMEGA e sua OPL ALFA podem ajudar a identificar/modelar os elementos-chaves da colaboração. Foi sugerida a inclusão de mais exemplos práticos em uso comercial.

Em relação ao desenvolvimento incremental e à abrangência do método OMEGA, alguns participantes comentaram que o mesmo se torna adequado ao desenvolvimento tradicional e ágil de *software* utilizado por suas equipes de trabalho. Segundo os mesmos, o ÔMEGA é recomendado na produção de documentação que realmente será útil, tendo em vista que a linguagem ONTOUML procura fornecer meios para auxiliar no levantamento dos requisitos que irão constituir um sistema**,** além do reúso de fragmentos de modelagem já testados e validados por especialistas de domínio (aumentarem a produtividade).

Com relação à notação visual da ALFA (AP-OPL), a maioria dos participantes destacou que o modelo do processo foi um guia no desenvolvimento de modelagem conceitual, tornando esse processo claro e intuitivo. Porém, um participante sugeriu a utilização de um glossário de termos detalhado para facilitar o entendimento dos elementos do modelo de processos para aqueles que estão tendo o primeiro contato com o método.

7.4 DISCUSSÕES

Conforme estabelecido no planejamento do estudo, seu objetivo foi avaliar a contribuição que o método ÔMEGA e seu respectivo modelo de processo de uso ALFA (AP-OPL) trouxeram para construção de modelos conceituais de *groupware* baseados em padrões ontológicos de colaboração (avaliação dos resultados). Para isso, foram considerados cinco indicadores: **1-** Produtividade do desenvolvimento de modelagem conceitual ontológica; **2-** Qualidade da modelagem resultante; **3-** Grau de dificuldade para entendimento do OMEGA e dos símbolos da OPL ALFA (AP-OPL) e **4-** Ausência de problemas de representação da notação OPL ALFA(AP-OPL).

A maioria dos participantes afirmou haver mais benefícios em criar modelos conceituais ontologicamente bem fundamentados e semanticamente ricos, do que criar um modelo sem esse recurso. Esses participantes destacaram a diminuição no tempo de desenvolvimento da modelagem e a melhoria da qualidade dos modelos resultantes. Segundo as discussões detalhadas sobre as percepções obtidas a partir dos resultados dos questionários e entrevistas e, também, a partir da avaliação, há indicação de melhoria para os indicadores de produtividade do desenvolvimento de modelagem conceitual ontológica e da qualidade da modelagem resultante, indicando que o uso do ÔMEGA e sua ALFA (AP-OPL) impactam positivamente esses aspectos do desenvolvimento de modelagem conceitual ontológica.

Com relação à representação visual adotada para representar o ÔMEGA e seu respectivo modelo de processo de uso OPL ALFA (AP-OPL), a maioria dos participantes considerou fácil de entender o mecanismo de funcionamento, o que significa um baixo grau de dificuldade para entendimento do mecanismo de funcionamento do ÔMEGA e sua ALFA (indicador (3)). Somente um participante relatou ter tido dificuldades no entendimento de alguns símbolos, o que significa a ausência de problemas de representação da notação OPL ALFA (AP-OPL) (indicador (4)).

Em síntese, conjectura-se que novos usuários sem o conhecimento prévio em modelagem conceitual, UML e/ou ontologias tenham uma certa dificuldade de compreender quando e qual padrão aplicar durante o processo de modelagem. Assim, considerando os resultados obtidos, conclui-se que, no contexto geral da avaliação realizada neste experimento, o uso do ÔMEGA e seu respectivo modelo de processo de uso ALFA (AP-OPL)

contribuem para o aumento da produtividade do desenvolvimento de modelagens conceituais e para a qualidade dos modelos resultantes (indicador (2)).

7.5 CONSIDERAÇÕES FINAIS SOBRE A AVALIAÇÃO

Neste capítulo, foi apresentado um estudo experimental realizado com o objetivo de avaliar o uso do método baseado em padrões ontológicos de colaboração para construção de modelos conceituais de *groupware.* Os resultados do estudo corroboram as afirmações encontradas na literatura de que o uso de padrões ontológicos melhora a produtividade do processo e a qualidade da modelagem conceitual ontológica resultante de corporações

Verificou-se que o método ÔMEGA é aplicável e eficiente para equipes de trabalho. O método ÔMEGA e sua OPL ALFA(AP-OPL) trazem grandes possibilidades de uso para equipes pequenas, médias e de complexidade maior. A partir desses resultados, podemos concluir que o método é viável, podendo ser usado tanto por analistas de sistemas quanto por analistas de dados, de forma desacoplada da implementação dos Sistemas de Informação. De acordo com os participantes, o método é aderente ao ciclo proposto pelas metodologias tradicionais e ágeis, o que só aumenta a sua aplicabilidade.

CAPÍTULO 8. CONSIDERAÇÕES FINAIS

8.1 CONSIDERAÇÕES FINAIS DO TRABALHO

No âmbito da modelagem conceitual baseada em ontologias, modelar sistemas a partir do zero, com ou sem especialista de domínio, é uma tarefa árdua. Com o apoio de ferramentas e abordagens metodológicas para construção de modelos conceituais ontologicamente bem fundamentados, o trabalho se torna mais suave. Para tal, optou-se por utilizar o módulo OLED Estendido do Editor OLED como ferramenta computacional, a fim de aproveitar a abordagem de catálogo de DROPs de colaboração implementado no OLED Estendido e sua gama de ferramentas de modelagem, verificação, transformação e validação para criação e utilização de POs de domínio da OPL ALFA (AP-OPL).

Acreditamos que o método ÔMEGA e respectivo Modelo de Processo de Uso ALFA (AP-OPL) apresentados aqui sirvam de guia para auxiliar usuários (misto de novatos e avançados) na construções de modelos ontológicos de *groupware* usando OntoUML. Entretanto, mesmo tendo os novos artefatos de colaboração como aporte conceitual, estima-se que novos usuários sem o conhecimento prévio em modelagem conceitual, UML e/ou ontologias tenham uma certa dificuldade de compreender quando e qual padrão aplicar durante o processo de modelagem.

Diante disso, analisamos o processo de construção de modelos ontológicos buscando definir procedimentos de construção para serem usados por usuários novatos da linguagem, similar ao realizado em Guizzardi et al. (2011b), pois verificamos que, a partir da aplicação de um padrão de colaboração OntoUML, uma série de outros padrões se faz necessária para completar o modelo, gerando, assim, uma cadeia de chamadas para novos padrões. Ademais, realizamos uma análise das regras de fundamentação de UFO-A para identificar a existência de novos padrões. Por fim, apresentamos nosso catálogo de padrões DROPs de colaboração consolidado de forma estruturada.

É importante salientar que a ferramenta OLED Estendido sistematiza a construção de modelagens conceituais baseadas em ontologias operacionais relacionadas ao domínio da colaboração (*groupware)* que podem ser compartilhadas e utilizadas na construção de outras modelagens, considerando-se que, com a utilização do catálogo de DROPs ganharam

a padronização que os DROPs oferecem, tornando a construção de modelos ontológicos muito semelhante a montar formas ou objetos utilizando LEGO®. Assim, entendemos que este trabalho passa a fazer parte de um arcabouço de trabalhos que, unidos contribuem para a realização de um suporte metodológico e computacional na construção de modelos conceituais em OntoUML.

8.2 CONTRIBUIÇÕES

Dessa forma, concluímos nosso trabalho apresentando as principais contribuições desta dissertação:

- Definição de um modelo de estruturação de requisitos orientados a objetivos. E por conseguinte, a criação de um *checklist* de avaliação dos requisitos de *groupware*;
- A criação de novos artefatos de colaboração, o “Catálogo de DROPs de Colaboração Consolidado”, a OPL que representa o “Modelo de Processo da ALFA (AP-OPL)” e a tradução das "Ontologias de Colaboração (três subontologias de CONTO)” com base na evolução dos artefatos de colaboração de origem o “Catálogo de DROPs de Colaboração”, a OPL que representa o “Modelo de Processo da C-OPL” e as "Ontologias de Colaboração (três subontologias de CONTO)”, para o português;
- A criação do método ÔMEGA baseado em padrões ontológicos de colaboração para construção de modelos conceituais de *groupware*;
- A proposta e a apresentação de uma versão da OPL para o domínio de colaboração a ALFA (AP-OPL) com seu modelo do processo para aplicação dos DROPs, o qual consistiu na reengenharia da representação visual da C-OPL.
- Avaliação da contribuição que o método ÔMEGA e sua respectiva OPL para o domínio de colaboração ALFA (AP-OPL) trouxeram para construção de modelos conceituais de *groupware* baseados em padrões ontológicos de colaboração (avaliação dos resultados).

8.3 DIFICULDADES E LIMITAÇÕES

As principais dificuldades e limitações encontradas durante a realização deste trabalho foram:

- O método pressupõe o conhecimento prévio de levantamento de requisitos, portanto, o método ÔMEGA não possui uma fase de captura de requisitos, mas sugere a utilização de

um modelo de estruturação de requisitos orientados a objetivos, com o intuito de facilitar a modelagem ontológica.

- O método também conjectura que novos usuários sem o conhecimento prévio em modelagem conceitual, UML e/ou ontologias tenham uma certa dificuldade de compreender quando e qual padrão aplicar durante o processo de modelagem.
- A falta de documentação da ferramenta OLED escolhida para entendimento das funcionalidades ali existentes, bem como a melhor forma de utilizá-las para alcançar os objetivos. Os modelos conceituais ontológicos resultantes não foram validados via simulação e checados via antipadrões.
- A falta de implementação de outras categorias de OntoUML não pertencentes à UFO-A para a modelagem de ontologias.
- A deficiência na ferramenta OLED no que tange ao gerenciamento de memória, causando vários estouros de memória, travamento e gerando retrabalhos.

8.4 PERSPECTIVAS FUTURAS E APRIMORAMENTO DO TRABALHO

Considerando a pesquisa aqui apresentada, há algumas perspectivas de trabalhos futuros e de aprimoramentos. No âmbito da pesquisa, pode-se destacar:

- Criar catálogos de DROPs em outros domínios para utilização da modelagem conceitual baseada em ontologias, validação e implementação na ferramenta. Por exemplo, o domínio da neurociência;
- Criar outro ambiente de desenvolvimento, avaliação e implementação de ontologias de domínio usando a linguagem de modelagem ontologicamente bem fundamentada com base em Ontologias de Fundamentação Unificada UFO (A, B e C);
- Criar mecanismos para geração de OPLs de forma interativa e incremental;
- Criar repositórios DROPs que possam ser utilizados e construídos de forma colaborativa;
- Realizar novos estudos, a fim de avaliar o desempenho da proposta em outros domínios, com uma maior quantidade de participantes e maior variedade de perfis.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBUQUERQUE, A. **Ontological Foudations for Conceptual Modeling Datatype**. MSc Thesis, Federal University of Espírito Santo, 2013.

ALMEIDA, M; BAX, M. **Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção**. Revista Ciência da Informação, 32(3), 2003. Disponível em: <http://www.scielo.br/readcube/epdf.php?doi=10.1590/S0100-19652003000300002&pid=S0100-19652003000300002&pdf>. Acesso em: 25 mai. 2017.

ALMEIDA, WESLLEY AUGUSTO DE. **Um estudo sobre reutilização de software utilizando linha de produto de software e desenvolvimento dirigido a modelos**. 2017. 51 p. Trabalho de Conclusão de Curso (Bacharel em Sistemas de Informação)- Universidade Estadual do Norte do Paraná, Bandeirantes - PR., 2017. 1. Disponível em:<http://200.201.11.152/bitstream/handle/123456789/227/TCC.pdf?sequence=1&isAllowed=y>.Acesso em: 05 jun. 2018.

ARANGO, G. **Software Reusability - Domain analysis methods**. Ellis HorWood, pp. 17-49. 1994.

A. MAEDCHE: **Ontology for the Semantic Web, The Kluwer International Series In Enginneering and Computer Science**, Volume 665, Kluwer Academic, 2002.

BARCELLOS, M. P.; FALBO, R. DE A.; FRAUCHES, V. G. **Towards on a Measurement Ontology Pattern Language**. First Joint Workshop Onto.Com/ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering. 2014. Rio de Janeiro - RJ, Brazil.

BEAUCLAIR, R. **Group decision support systems and their effect on small group work**. In G. Phillips (Ed.), Teaching how to work in groups (pp. 151 - 172). Norwood, NJ: Ablex Publishing Corporation. 1990.

BERLO, D. K. **O processo da comunicação, introdução a teoria e prática**., Rio, Fundo Cultura, (4ed.), 1972.

BELTRÃO, L. **Teoria geral da comunicação**. Ed. Thesaurus , Brasília, 1977..

BENEVIDES, A.B., GUIZZARDI, G., BRAGA, B.F.B., ALMEIDA, J.P.A. et al. **Validating modal aspects of OntoUML conceptual models using automatically generated visual world structures**. J. Univers. Comput. Sci., vol. 16, pp. 2904–2933, 2011.

BORGHOFF, U.M.; SCHLICHTER, J.H. 2000. **Computer-Supported Cooperative Work: Introduction to Distributed Applications**. Springer, USA.

BORGO, S. MASOLO, C. **Foundational choices in DOLCE**. In: STAAB, S.; Studer, R. (Ed.). Handbook on Ontologies. 2. ed. Springer Verlag, 2009. p. 361-382. Disponível em: ttps://pdfs.semanticscholar.org/695d/8db76d8e044de4c79b399330454d7e73beec.pdf>. Acesso em: 25 mai. 2017.

BUSCHMANN, F.; HENNEY, K.; SCHMIDT, D. C. **Pattern Oriented Software Architecture Volume 5: On Patterns and Pattern Languages**. v. 5, 2007.

BLOMQVIST, EVA; GANGEMI, ALDO; PRESUTTI, VALENTINA. **Experiments on pattern-based ontology *design***. In Proceedings Of The Fifth International Conference On Knowledge Capture (K-CAP '09). ACM, New York, NY, USA, p. 41-48, 2009.

BRILHANTE, VIRGÍNIA. **Ontologias**. Raízes Filosóficas. Disponível em:<www.dcc.ufam.edu.br/~ontologias/slides/ raizesFilosoficas.pdf>. Acesso em: 20 jul. de 2018.

CAMPOS, M. L. A. **Modelização de domínios de conhecimento: uma investigação de princípios fundamentais**. Ciência da Informação, Brasília. v. 33, n. 1, p. 22-32, abr. 2004.

CAMPOS, MARIA LUIZA DE ALMEIDA. **Elaboração de modelos de domínio em ontologias: a abordagem onomasiológica e a função da definição**. 2017. Disponível em:<http://revista.ibict.br/ciinf/article/view/4016/3716>. Acesso em: 26 jun. 2018.

CAMPOS, MARIA LUIZA DE ALMEIDA; CAMPOS, LINAIR MARIA. **A organização do conhecimento e suas teorias de representação: a ontologia de fundamentação como um modelo teórico para a representação de domínios**. In: Encontro Nacional de Pesquisa em Ciência da Informação, XIII., 2012, Rio de Janeiro/RJ. GT 2 - Organização e Representação do Conhecimento...[S.l.:s.n.], 2012. p. 1-18. v. [s.n.].Disponível em: <http://enancib.ibict.br/ index.php/enancib/xiiienancib/ paper/view/3677/2800>. Acesso em: 26 jun. 2018.

CASTRO, SUSANA DE. **Ontologia** – Filosofia passo-a-passo. Rio de Janeiro: Jorge Zahar Ed., 2008.

COLEMAN, D.; KHANNA, R. **Groupware: Technologies and applications**. Upper Saddle River, NJ: Prentice Hall. 1995.

CHAFFEY, DAVE. **Groupware, Workflow and Intranets, Reengineering the Enterprise with Collaborative Software**. Digital Press. 1998.

CHORLEY, R.; HAGGETT, P. **Modelos, paradigmas e a nova geografia.** In: CHORLEY, R. e HAGGETT, P. Modelos sócios-econômicos em geografia. Rio de Janeiro: Livros Técnicos e Científicos/USP, p.1-22, 1975.

CARRARETTO, R. **Separating ontological and informational concerns**: a model-driven approach for conceptual modeling. 2012. 120 f. Dissertação (Mestrado em Informática) - Universidade Federal do Espírito Santo, Vitória, 2012.

CLARK, PETER. **Some Ongoing KBS/Ontology Projects and Groups**. 2006. Disponível em: http://www.cs.utexas .edu/users/mfkb/related.html#u>. Acesso em: 02 ago. 2018.

CROWSTON, K. **A Taxonomy of Organizational Dependencies and Coordination Mechanisms**. Massachusetts Institute of Technology, Sloan School of Management, 1994.

CROWSTON, K., E OSBORN, C. S. **A Coordination Theory Approach to Process Description and Redesign.** in Organizing 40 Business Knowledge: The MIT Process Handbook, T. W. Malone, K. Crowston, and G. Herman (Eds.), MIT Press, 41 Cambridge, MA, 2003.

CRUZ, CRISTIANO C. **Análise da educação e da informática na sociedade segundo a Teoria da Atividade**. In XXI Congresso da Sociedade Brasileira de Computação. Fortaleza, 2001. Anais XXI Congresso da Sociedade Brasileira de Computação. Fortaleza: SBC, 2001. Disponível na rede: <http://www.dspcom.fee.unicamp .br/ cristia/artigos/SBC2001/artigo_sbc.pdf >. Acesso em: 09 jul. 2018.

CYBULSKI, J. ; REED, K. **Requirements classification and reuse: Crossing domain boundaries**. In Proceedings of the 6th International Conference on Software Reuse (ICSR), pages 190–210, 2000.

DEUTSCH, P. MODELS AND PATTERNS. IN: J. GREENFIELD; K. SHORT; S. COOK; S. KENT (Orgs.).;et al. **Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools**. 2004. Indianapolis: John Wiley & Sons.

DIAS, CÉLIA DA CONSOLAÇÃO. **A análise de domínio, as comunidades discursivas, a garantia da literatura e outras garantias**. 2015. Disponível em: <http://www.ies.ufpb.br/ ojs/index.php/ies/article/ viewFile/7/13 741>. Acesso em: 26 mai. 2017.

DILLENBOURG, P.; SELF, J.A. **A computational approach to socially distributed cognition**. European Journal of Psychology of Education, 3 (4), 353-372. 1992.

DZIEKANIAK, GISELE VASCONCELOS. **Desenvolvimento de uma ontologia sobre componentes de ontologias**. *Perspectivas em Ciência da Informação* 15.1 (2010): 173-184.

ELLIS, CLARENCE A.; GIBBS, SIMON J.; REIN, GAIL. et al. **Groupware: some issues and experiences.** Communi cations of the ACM, v. 34, n. 1, p. 39-58, 1991.

ELLIS, C. A., GIBBS, S. J., REIN, G. L. **Groupware: some Issues and Experience**. Communications of the ACM, v. 34, n. 1, pp. 38-58, Jan. 1991.

EDWARDS, M. **Users offered many options for groupware.** Communication News, 32, 48-49. 1995.

ENGHOLM JUNIOR, HÉLIO. **Analise e Design Orientados a Objetos**, 1. ed. São Paulo: Novatec, 376 p., 2016.

FALBO, R. DE A.; BARCELLOS, M. P.; NARDI, J. C.; GUIZZARDI, G.; et al. **Organizing ontology design patterns as ontology pattern language**. Proceedings of the 10th Extended Semantic Web Conference - ESWC 2013. 2013a. Montpellier, France. Disponível em:<https://eswc-conferences.org/sites/default/files/papers2013/falbo.pdf>. Acesso em: 5 de abr. de 2018.

FALBO, R. DE A.; GUIZZARDI, G.; GANGEMI, A.; PRESUTTI, V. **Ontology patterns: clarifying concepts and terminology**. In: CEUR-WS. ORG. Proceedings of the 4th International Conference on Ontology and Semantic Web Patterns-Volume 1188. 2013b. p. 14–26. Disponível em:<http://ceur-ws.org/Vol-1188/paper_11.pdf>. Acesso em: 5 de abr. de 2018.

FALBO, R. A.; NARDI, J. C. **Evolving a Software Requirements Ontology**. In: XXXIV Conferencia Latino americana de Informática - CLEI´ 2008, Santa Fe. Anales de XXXIV Conferencia Latino americana de Informática, pp. 300-309, 2008.

FALBO, R. A., BARCELLOS, M. P., NARDI, J. C., GUIZZARDI, G. **Ontology and Conceptual Modeling Research Group (NEMO)**. Springer-Verlag Berlin Heidelberg, Computer Science Department, Federal University of Espírito Santo, Vitória, Brazil. p. 1, 2011.

FALBO, R. A. **SABiO: Systematic approach for building ontologies**. In: *ONTO.COM/ODISE@ FOIS*. [s.n.], 2011. Disponível em:<https://pdfs.semanticscholar.org/2660/676b0d784d46d3c407793c0cee0c10e76c5b.pdf>. Acesso em 07 de junho de 2018.

FALBO, R. DE A.; RUY, F. B.; GUIZZARDI, G.; BARCELLOS, M. P.; ALMEIDA, J. P. A.; el al. **Towards an enterprise ontology pattern language**. Proceedings of the 29th Annual ACM Symposium on Applied Computing. p.323–330, 2014. ACM.

FALBO, R. A.; BARCELLOS, M. P.; RUY, FABIANO B.; GUIZZARDI, G.; GUIZZARDI, *R. S. S.* **Ontology Pattern Languages**. In: Hitzler, P.; Gangemi, *A.;* Janowicz, K; Krisnadhi, A; Presutti, *V*. (Org.). Ontology Engineering with Ontology Design Patterns. 1ed.Amsterdam: IOS Press, 2016, *v*. 25, *p*. 133-159.

FENSEL, D; VAN HARMELEN, F; HORROCKS, I; MCGUINNESS, D; PATELSCHNEIDER,P.;el at. **Oil: An ontology infrastructure for the semantic web**. IEEE Intelligent Systems, 16:38–44, 2001.

FINHOLT, T.; SPROULL, L. **Electronic groups at work. Organizational Science**, 1(1),41-64. (1990).

FERREIRA, HIRAN N. M. ; NAVES, THIAGO F. **Reuso de software: suas vantagens, técnicas e práticas**. Enacomp 2011, Faculdade de Computação - Universidade Federal de Uberlândia(UFU) Uberlândia - MG - Brasil, Disponível em: <http://www.enacomp.com.br/2011/anais/trabalhos-aprovados/pdf/enacomp2011 _submission_43.pdf>.Acesso em: 15 mai. 2017.

FUKS,H., RAPOSO, A. B. GEROSA, M. A. **Engenharia de *Groupware*: Desenvolvimento de Aplicações Colaborativas**. XXI Jornada de Atualização em Informática. 2002. Anais de XXII Congresso da Sociedade Brasileira de Computação, V2, Cap. 3, ISBN 85-88442-248, pp. 89-128.

FUKS, H.; GEROSA, M. A.; LUCENA, C. J. P. **The Development and Application of Distance Learning on the Internet**. The Journal of Open and Distance Learning, Carfaz Publishing, Vol. 17. N. 1, p.23-38. ISSN 0268-0513, UK, February, 2002.

GADELHA, B. F. **Uma Abordagem de Desenvolvimento de Groupware Baseada em Linha de Produto de Software e Modelo 3C de Colaboração**. Tese (Tese de Doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Dezembro 2011. Disponível em:<http://groupware.les.inf.puc-rio.br/public/papers/0812625_2011 _completa.pdf> . Acesso em 28 de set. de 2014.

GAMMA, ERICH; HELM, RICHARD; JOHNSON, RALPH; VLISSIDES, JOHN.; et al. **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison-Wesley. 1994.

GANGEMI, A.; PRESUTTI, V. **Ontology Design Patterns**. In: S. Staab; R. Studer (Orgs.); Handbook on Ontologies. Second ed., p.221 – 243, 2009. Springer.

GEROSA, M.A.; PIMENTEL, M.G.; FILIPPO, D.; BARRETO, C.G.; RAPOSO, A.B.; FUKS, H. e LUCENA, C.J.P.; et al. **Componentes Baseados no Modelo 3C para o Desenvolvimento de Ferramentas Colaborativas**. Anais do 5º Workshop de Desenvolvimento Baseado em Componentes - WDBC 2005, 07-09 de Novembro, Juiz de Fora-MG, ISBN 85-88279-47-9, pp. 109-112. Disponível em <https://webserver2.tecgraf.puc-rio.br/~abraposo/pubs/WCSCW05_WDBC05/WDBC2005.Gerosa.pdf>. Acesso em: 24 jul. 2017.

GÓMEZ-PÉREZ, A. **Knowledge sharing and reuse**. Handbook of AppliedExpert Systems. Liebowitz, editor, CRC Press.1998.

GUTWIN, C., GREENBERG, S. **The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces**. IEEE 9th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000, pp. 98-103.

GLYN-JONES, F. **The groupware grapevine**. Management Today, pp. 83-86. 1995.

GRACAS, A.; GUIZZARDI, G. **Padrões de Modelagem e Regras de Construção de Modelos para a criação de Ontologias de Domínio Bem-Fundamentadas em OntoUML**. 3° SEMINÁRIO DE PESQUISA DE M ONTOLOGIA NO BRASIL. Florianópolis, SC: [s.n]. 2010.

GRENON, P.; SMITH, B. **SNAP and SPAN: Prolegomenon to geodynamic ontology**. In: Spatial Cognition and Computation, 2004.

GROSZ, B. J., **Collaborative Systems (AAAI-94 Presidential Address)**. AI magazine, v. 17, n. 2, p. 67, 1996.

GRUBER, T. Ontolíngua**: A mechanism to support portable ontologies**, 1992.

GRUBER, T., **Ontology**. *Encyclopedia of Database Systems*, Liu. L., Özsu. M. (eds.), Springer-Verlag. 2009.

GRUDIN, J. **Computer-Supported Cooperative Work: History and Focus**. Computer (IEEE). 1994.

GUARINO, NICOLA; GIARETTA, PIERDANIELE. **Ontologies and Knowledge Bases Towards a Terminalogical Classification: Towards Very Large Knowledge Bases**. University Of Twente, Enschede, The Netherlands: N. J. I Mars, 1995. 307 p. Disponível em:<https://books.google.com.br/books?hl=pt-BR&lr=&id=8xiJLQSDGfEC&oi=fnd&pg=PA25&dq=.+Ontologies+and+Knowledge+Bases:+towards+a+terminological+clarification.+1995&ots=hljfeF9V6l&sig=HgjibimKs9dTxDY7HOLewV9rLx0#v=onepage&q&f=false>. Acesso em: 28 mai. 2017.

GUARINO, N. **Understanding, building and using ontologies**. International Journal of Human and Computer Studies, 45(2/3), 1997.

GUARINO, NICOLA. **Formal Ontology in Information Systems**. In: IOS Press, A, editor, FOIS'98, TRENTO, ITALY, p. 3–15, 06 1998. Disponível em: <http://www.loa.istc.cnr.it/old/Papers/FOIS98.pdf>.Acesso em: 22 mai. 2017.

GUARINO, N.; WELTY, C. A. **An overview of OntoClean**. In: STAAB, S.; STUDER, R. (Ed.). Handbook on ontologies. Berlin: Springer, 2004. p. 151-171.

GUERSON, JOHN; SALES, TIAGO PRINCE; GUIZZARDI, GIANCARLO; ALMEIDA, JOÃO PAULO A. **OntoUML Lightweight Editor: A model-based environment to build, evaluate and implement reference ontologies**. 2015. Disponível em:<https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/ontouml_lightweight_editor__a_model_based_environment_to_build__evaluate_and_implement_reference_ontologies_2015.pdf>. Acesso em: 13 abr. 2018.

GUIZZARDI, G. **Desenvolvimento para e com reuso: Um estudo de caso no domínio de vídeo sob demanda**. Master's thesis, Universidade Federal do Espírito Santo, 2000.

GUIZZARDI, G.; WAGNER, G. **Some Applications of a Unified Foundational Ontology in Business Modeling**. Ontologies and Business Systems Analysis, Michael Rosemann and Peter Green (Eds.). IDEA Publisher, 2005.

GUIZZARDI, G., **Ontological foundations for structural conceptual models**. PhD Thesis, Enschede: CTIT, Centre for Telematics and Information Technology, 2005a. Disponível em:<https://www.inf.ufes.br/~gguizzardi/OFSCM.pdf>. Acesso em: 10 de abr. de 2018.

GUIZZARDI, G. **Agent Roles, Qua Individuals and The Counting Problem**.International Workshop on Software Engineering for Large-Scale Multi-agent Systems. **Anais**...Springer Berlin Heidelberg, 2005b

GUIZZARDI, G. **The problem of transitivity of part-whole relations in conceptual modeling revisited**, 21st Intl. Conf. on Advanced Information Systems Engineering (CAISE'09), Amsterdam, The Netherlands, 2009.

GUIZZARDI, G., FALBO, R. A., GUIZZARDI, R. S. S. **The Importance of Foundational Ontologies for Domain Engineering: The case of the Software Process Domain**. IEEE Transactions Latin America, v. 6, n. 3, p. 244-251. 2008a.

GUIZZARDI, G., FALBO, R.A., GUIZZARDI, R.S.S. **Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology**. Anais do XI Worshop Iberoamericano de Ambientes de Software e Engenharia de Requisitos, Recife, Brazil, 2008b.

GUIZZARDI, G., FALBO, R.A., GUIZZARDI, R.S.S. **A importância de Ontologias de Fundamentação para a Engenharia de Ontologias de Domínio: o caso do domínio de Processos de Software**. Revista IEEE América Latina, v. 6, p. 244-251, 2008c.

GUIZZARDI, GIANCARLO; ALMEIDA, JOÃO PAULO ANDRADE; GUIZZARDI, RENATA S.S.; FALBO, RICARDO. **Ontologias de fundamentação e modelagem conceitual**. In: Seminário de Pesquisa em Ontologia no Brasil, 2.,

2009, Rio de Janeiro. Anais eletrônicos...Rio de janeiro:IME 2009. Disponível em:<http://ontobra.comp.ime .eb.br/artigos/Gr17.pdf>. Acesso em: 20 jun. 2018.

GUIZZARDI, G. **On the Representation of Quantities and their Parts in Conceptual Modeling**, 6th International Conference on Formal Ontologies in Information Systems (FOIS 2010), Toronto, 2010.

GUIZZARDI, G., **Ontological foundations for conceptual part-whole relations: the case of collectives and their parts**, 23rd International Conference on Advanced Information System Engineering (CAiSE'11), London, UK, 2011a.

GUIZZARDI, GIANCARLO. ALMEIDA, JOÃO PAULO. GUIZZARDI, RENATA S.S. BARCELLOS, MONALESSA PERINI .FALDO, RICARDO. **Ontologias de Fundamentação, Modelagem Conceitual e Interoperabilidade Semântica**. 2011b. Disponível em : < http://ceur-ws.org/Vol-728/paper6.pdf>. Acesso em: 10 de abr. de 2018.

GUIZZARDI, G., WAGNER, G., FALBO, R.A., GUIZZARDI, R.S.S., ALMEIDA, J.P.A., **Towards Ontological Foundations for the Conceptual Modeling of Events**, 32nd International Conference on Conceptual Modeling (ER 2013), Hong Kong, 2013.

HOMANS, G. C. **Social Behavior as Exchange**, The American Journal of Sociology, Vol. 63, No. 6, Emile Durkheim-Georg Simmel, 1858-1958 (May, 1958), The University of Chicago Press, pp. 597-606, 1958.

ITU-T. ITU-T **Recommendation G.805: Generic functional architecture of transport networks**. International Telecommunication Union (ITU). [S.l.], p. 58. 2000.

HUNT, C.; BURFORD, A. **Group support systems: A new frontier for amplifying creativity and collaborative learning.** Business Education Forum, 48(3), 31-34. 1994.

KOTTER, J. **Power, Dependence and Effective Management**., Harvard Business Review 55, nº4, 1977.

KHOSHAFIAN, S.; BUCKIEWICZ, M. **Introduction to Groupware, Workflow and Workgroup Computing**. New York: John Wiley & Sons, Inc, 1995

LAGNI, DELIA TEREZINHA. **Teoria da atividade e a psicologia histórico-cultural: um estudo a partir da Proposta Curricular de Santa Catarina**. Dissertação (Mestrado em Educação) - UFSC, Florianópolis, 2004.

LAUDON, KENNETH C.; LAUDON, JANE PRICE. **Sistemas de informação gerenciais: administrando a empresa digital**. Tradução Arlete Simille Marques; revisão técnica Erico Veras Marques, Belmiro João. São Paulo: Prentice Hall, 2004.

LENAT, D. Cyc: **A large-scale investment in knowledge infrastructure**. Communications of the ACM, 38(11):33–38, 1995.

LIPNACK, J. e STAMPS, J. **Virtual Teams: People Working Across Boundaries with Technolog**., Second Edition. 2nd. John Wiley & Sons, Inc, 2000.

LIMA, PAULO SERGIO RODRIGUES. **Um Ambiente Colaborativo de Aprendizagem Interdisciplinar Apoiado por Interfaces Adaptativas**. 2006. 259 p. Tese de Doutorado (Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico)- Universidade Federal do Pará, Pará, 2006. Disponível em: <http://ppgee.propesp. ufpa.br/ARQUIVOS/teses/TD10_2006_Paulo%20S%C3%A9rgio%20Rodrigues%20Lima.pdf>. Acesso em: 29 jun. 2018.

LOPES, MAURO MOURA GOMES. **Modelagem conceitual de regras de negócio baseada em ontologia de fundamentação**. 2011. 223 f. Dissertação (Mestrado) - Universidade Federal do Estado do Rio de Janeiro, Departamento de Informática Aplicada, Rio de Janeiro, 2011.

MALUCELLI, A. **Ontologias**. (2002). Disponível em: <https://paginas.fe.up.pt/~eol/TNE/0506/Ontologia.pdf>. Acesso em: 25 mai. 2017.

MARCONDES, CARLOS HENRIQUE. **Análise ontológica de definições de informação: em busca da sua essência.** Transinformação [online]. 2015, vol.27, n.2, pp.105-122. ISSN 0103-3786.

MANDVIWALLA, M. & OLFMAN, L. **What do groups need? A proposed set of generic requirements**. ACM Transactions on Computer-Human Interaction,Vol. 1, No. 3, pp. 245-268. 1994.

MELO, J. M.**Comunicação social: teoria e pesquisa**. – Petrópolis, Vozes, 1973.

MOODY, D. L. **Theoretical and practical issues in evaluating the quality of conceptual models: Current state and future directions**. Data and Knowledge Engineering, v. 55, n. 3, p. 243-276, 2005.

MORAES, I; ZORZO, A. **Uma Arquitetura Genérica para Aplicações Colaborativas**, Relatório Técnico n°. 6. 2000.

MILLER, G. **Wordnet: A on-line lexical database**. International Journal of Lexicography, (3-4), 1990.

MUSEN, M. A. E. A. **PROTEGE-II: computer support for development of intelligent systems from libraries of components**. MEDINFO 95 - World Congress on Medical Informatics. 1995.

NEMO, O. C. M. R. G. **Oled - ontouml lightweight editor**. In: . [s.n.], 2017. Disponível em: <https://nemo.inf. ufes.br/ projects/oled/>. Acessado em 13 abr. 2017.

NEUMANN, V. J.; MORGENSTERN, O. **Theory of games and economic behavior**, Princeton University Press, Morgenstern, Oskar, 1944.

O'BRIEN,JAMES A. **Sistemas de informação e as decisões gerenciais na erra da internet**. São Paulo: Saraiva, 2001.

OLIVEIRA, ROBERTA MERCIA RODRIGUES DE. **COLLABPRO - Metodologia para utilização do Modelo 3C de Colaboração nas disciplinas de Engenharia de Software**. 2011. 57 p. Trabalho de Conclusão de Curso (Título: Bacharel em Ciência da Computação e Curso: Projeto de Computação Supervisionado II do Departamento de Ciências Exatas da UESB)- Universidade Estadual do Sudoeste da Bahia Departamento de Ciências Exatas, Vitória – ES, Brasil, 2011. Disponível em: <http://www2.uesb.br/computacao/wp-content/uploads/2014 /09/COLLABPRO-Metodologia-para-utiliza%C3%A7%C3%A3o-do-Modelo-3C-de-Colabora%C3%A7%C3%A3o-nas-disciplinas.pdf>. Acesso em: 10 abr. 2018.

OLIVÉ, A., **Conceptual Modeling of Information Systems**. Springer-Verlag New York, Inc, 2007.

OLIVEIRA, FELIPE FRECHIANI DE. **Uma ontologia de colaboração e suas aplicações**. 128 p. Dissertação submetida ao Programa de Pós-Graduação em Informática (Programa de Pós-Graduação em Informática)- Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Informática., Universidade Federal do Espírito Santo, VITÓRIA – ES, Brasil, 2009. Disponível em:< https://nemo .inf.ufes.br/wp-content/papercite-data/pdf/uma_ontologia_de_colaboracao_e_suas_aplicacoes_2009.pdf>. Acesso em: 06 mai. 2018.

OMG, OCL Specification v2.4, 2014. Disponível em: <http://www.omg.org/spec/OCL/2.4/>.Acesso em: 13 abr. 2018.

PADILHA, NATALIA F.; BAIÃO, FERNANDA; REVOREDO, KATE. **Ontologias de fundamentação apoiando o alinhamento de ontologias de domínio**. In: Simpósio Brasileiro de Sistemas de Informação, VIII., 2012, São

Paulo, SP. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/wtdsi/2012/002.pdf>. Acesso em: 04 jul. 2018.

PÁDUA, SÍLVIA INÊS DALLAVALLE DE; CAZARINI, EDSON WALMIR; INAMASU, RICARDO YASSUSHI. **Modelagem organizacional: captura dos requisitos organizacionais no desenvolvimento de sistemas de informação**. 2004. Disponível em: <http://www.scielo.br/pdf/gp/v11n2/a06v11n2>. Acesso em: 09 jul. 2018.

PAULA FILHO, WILSON DE PÁDUA. **Engenharia de Software: fundamentos, métodos e padrões**. São Paulo: LTC Editora, 2000.

PENTEADO, J. R. W. **A Técnica da Comunicação Humana**., São Paulo: Pioneira, 1993.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M.; et al. **Systematic Mapping Studies in Software Engineering**. 12th International Conference on Evaluation and Assessment in *Software* Engineering (EASE). University of Bari, Italy, 26 - 27 June 2008.

PETROVIC, O.; KRICKL, O. **Traditionally moderated versus computer sup-ported brainstorming: A comparative study**. Information & Management, 27(4), 233-243. 1994.

PIMENTEL, MARIANO**. RUP-3C-Groupware: um processo de desenvolvimento de groupware baseado no Modelo 3C de Colaboração**", Tese de Mestrado, PUC, Rio, Brazil, 2006.

PIMENTEL, MARIANO; FUKS, HUGO (Org.). **Sistema Colaborativos. Local de publicação: Rio de Janeiro**. Editora. Elsevier/Campos/SBC, 2011. V1, ISBN 978-85-352-4669-8.

PICHILIANI, MAURO CARLOS. **Mapeamento de software para permitir a colaboração síncrona**. 170 f. Tese de doutorado - Pós-Graduação e Pesquisa do Instituto Tecnológico de Aeronáutica, Campo Montenegro - São José dos Campos, SP - Brasil, 2006.

PRESSMAN, ROGER S. **Engenharia de Software**, 6 ed. São Paulo: McGraw Hill/Nacional, 2006.

PRESUTTI, VALENTINA; DAGA, ENRICO; GANGEMI, ALDO; BLOMQVIST, EVA.; et al. **Extreme Design with Content Ontology Design Patterns**. In Proceedings of the Workshop on Ontology Patterns. pp. 83-97. 2009

POLI, R. **Framing ontology**. 2001. Disponível em: <http://www.Formalontology.it/Framing first.htm>.Acesso em: 22 mai. 2017.

QUATERMAN, J. **The matrix: computer networks and conferencing systems worldwide**. Bedford, MA: Digital Press. 1990.

QUIRINO, GLAICE KELLY DA SILVA. **Uma Notação Visual para Representação de Linguagens de Padrões Ontológicos**. 2016. 152 p. Dissertação de Mestrado (Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática)- Universidade Federal do Espírito Santo, Vitória (ES) - Brasil, 2016. Disponível em: <https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/notacao_visual_para_linguagens_de_padroes_ontologicos_2016.pdf>. Acesso em: 02 jul. 2018.

QUIRINO, GLAICE KELLY DA SILVA; BARCELLOS, MONALESSA P.; FALBO, RICARDO DE ALMEIDA. **OPL-ML: A Modeling Language for Representing Ontology Pattern Languages**. in: Advances in Conceptual Modeling: ER 2017 Workshops AHA, MoBiD, MREBA, OntoCom, and QMMQ, Valencia, Spain, November 6–9, Proceedings (pp.187-201), 2017.

REICH, JACQUELINE RENÉE. **Ontological design patterns: Metadata of molecular biological ontologies, information and knowledge**. Proceedings of the 11th international conference on database & expert systems applications, p. 698–709, London, 2000.

RODDEN, T.;BLAIR, G. **CSCW and Distributed Systems: The Problem of Control**. In Proceedings of the Second European Conference on Computer-Supported Cooperative Work - ECSCW'91 (Sept. 24-27, Amsterdam, NL). Kluwer Academic Publishers, Dortrecht, NL, 1991. pp. 49-64.

RUY, FABIANO BORGES; REGINATO, CÁSSIO CHAVES; SANTOS, VICTOR AMORIM; FALBO, RICARDO DE ALMEIDA; GUIZZARDI, GIANCARLO. **Ontology Engineering by Combining Ontology Patterns**. 34th International Conference on Conceptual Modeling, Springer, p. 173—186, 2015.

RUY, F. B.; FALBO, R. DE A.; BARCELLOS, M. P.; GUIZZARDI, G.; QUIRINO, G. K. **An ISO-based software process ontology pattern language and its application for harmonizing standards**. ACM SIGAPP Applied Computing Review, v. 15, n. 2, p. 27–40, 2015.

SAYÃO, L. F. **Modelos teóricos em ciência da informação: abstração e método científico**. Ciência da Informação, Brasília, v. 30, n. 1, p. 82-91, jan./abr. 2001.

SALES T.P., GUIZZARDI G. **Detection, Simulation and Elimination of Semantic Anti-Patterns in Ontology-Driven Conceptual Models**. In Conceptual Modeling (ER), pp. 363-376, 2014.

SEARLE, R. J. **Intentionality: An Essay in the Philosophy of Mind**. Cambridge University Press, 1983.

SILVA, MAURÍCIO SEVERO DA. **Collabcode: ferramenta para apoio ao desenvolvimento distribuído e colaborativo de software**. 2013. 109 f. Trabalho de Conclusão de Curso (Bacharel em Sistemas de Informação)- Título e curso Centro Universitário - UNIVATES, Lajeado, 2013.

SILVA, DANIELA LUCAS DA; SOUZA, RENATO ROCHA; ALMEIDA, MAURÍCIO BARCELLOS. **Ontologias e vocabulários controlados: comparação de metodologias para construção**. Ci. Inf 37.3 (2008): 60-75.

SOUZA, ANDRÉ KAPICH. **Automatizado para desenvolvimento de ontologias utilizando padrões ontológicos de domínio**. 2017. 163 f. Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática (Mestre em Informática)- Universidade Federal do Espírito Santo, Universidade Federal do Espírito Santo, VITÓRIA, ES. Brasil, 2017.

SOUZA, ANDRE; GUIZZARDI, RENATA; CAMPOS, MARIA LUIZA M.; GUIZZARDI, GIANCARLO. et al. **Extending an Ontology Editor for Domain-related Ontology Patterns Reuse: An Application in the Collaboration Domain**. In: ONTOBRAS – Seminário de Pesquisa em Ontologias do Brasil, 4., 2016, Brasil. Anais. Curitiba, 2016.

SOMMERVILLE, IAN., **Engenharia de Software**. 6ª ed., São Paulo: Addison Wesley, 2003.

_____, IAN. **Engenharia de Software**, 8ª ed., Rio de Janeiro: Prentice-Hall, 2008.

SULLIVAN, C. **Preferences for electronic mail in organizational communication tasks**. Journal of Business Communication, 32(1), 49-64. 1995.

SHANNON, C. **The Mathematical Theory of Communication**. Urbana, Il.: Univ. of Illinois Press, 1949.

SCHNEIDERMAN, B., **Designing the User Interface: Strategies for Effective Human-Computer Interaction**. Addison-Weslwy, 1992.

SCHEIDEGGER, PATRICIA MERLIM LIMA.**Análise ontológica na construção de definições: diretrizes baseadas em ontologia de fundamentação**. 2016. 105 p. Dissertação de Mestrado (Programa de Pós-Graduação em

Informática) - Instituto de Matemática e Instituto Tércio Pacciti, UFRJ - Universidade Federal do Rio de Janeiro., Rio de Janeiro, 2016. Disponível em: <https://www.researchgate.net/publication/301747038>. Acesso em: 30 mai. 2017.

SCHEIDEGGER, PATRICIA MERLIM L.; CAMPOS, MARIA LUIZA MACHADO. **Uma abordagem baseada em análise ontológica para alinhamento entre conceitos de negócio e modelos conceituais. In:** II **Workshop de Teses e Dissertação em Sistemas de Informação**, 2015, Goiânia, GO, Brasil. 2015. p. 34-36. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/wtdsi/2015/009.pdf>. Acesso em: 08 jul. 2018.

SCHRAMM, W. **Mecanismo de la comunicación**. In Processo y effectos de la comunicación humana. Quito, CIESPAL, 1964.

SCHMIDT, KJELD; BANNON, LIAM. **Constructing CSCW: The first quarter century, Computer Supported Cooperative Work (CSCW): The Journal of Collaborative Computing and Work Practices**, vol. 22, no. 4-6, Aug. 2013, pp. 345–372. Disponível em:< https://www.researchgate.net/profile/Kjeld_Schmidt/publication/257561758_CSCW_The_first_quarter_century_Special_Jubilee_Issue/links/00b7d527a2474d4628000000/CSCW-The-first-quarter-century-Special-Jubilee-Issue.pdf>. Acesso em: 25 jan. 2018.

SMITH, B.; WELTY, C. **Ontology: Towards a new synthesis**. In: SMITH B., e WELTY, C. (Eds.). Proceedings of the International Conference on Formal Ontology in Information Systems (pp. 3–9). New York: ACM Press, 2001.

TIETZE, D.A. **A Framework for Developing Component-based Co-operative Applications**. PhD Dissertation, Computer Science, Technischen Universität Darmstadt, Germany, 2001.

THE, L. **Beta not wait for groupware**. Datamation, 4/(13), 69-75. 1995.

THIERAUF, R. **Group decision support systems for effective decision making**. A guide for MIS practitioners and end users. New York: Quorum Books. 1989.

TRAVASSOS, G.; GUROV, D. AMARAL, E. **Introdução à Engenharia de Software Experimental**. Relatório Técnico ES-590/02 Programa de Engenharia de Sistemas e Computação COPPEUFRJ, p. 52, 2002.

VELAYO, R. **Supplementary classroom instruction via computer conferencing**. Educational Technology, 34(5), 20-26. 1994.

VILLELA, MARIA LÚCIA; OLIVEIRA, ALCIONE DE PAIVA; BRAGA, JOSÉ LUÍS. ; et al **Modelagem Ontológica no Apoio à Modelagem Conceitual**. In: Simpósio Brasileiro de Engenharia de Software, XVIII., 2004, Brasília, DF, Brasil. BDBComp - Biblioteca Digital Brasileira de Computação... Brasília, DF, Brasil: [s.n.], 2004. p. 241-256. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/sbes/2004/017.pdf>. Acesso em: 02 mai. 2018.

WAND, Y.; WEBER, R. **Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda**. Information Systems Research, 13, p. 363-376, 2002.

WILD, R.H.; WINNIFORD, M. **Remote collaboration among students using electronic mail**. Computers and Education 21, 193–203, 1993.

ZUBOFF, SHOSHANA. **In the Age of the Smart Machine: The Future of Work and Power**. New York: Basic, 1988. Print. ISBN 9780465032112

APÊNDICE A

LISTA DE PUBLICAÇÕES INCLUÍDAS NA REVISÃO SISTEMÁTICA

[ET001]		MACHADO, R. G.; BORGES, M. R. S. ; GOMES, JOSÉ ORLANDO. SBSC – III Simpósio Brasileiro De Sistemas Colaborativos. **Método Colaborativo de Observação: Entendendo coletivamente os ambientes complexos**. 9p. 2006, Natal - RN - Brasil. Digital; Série: 1; ISSN/ISBN: 8576690977.Qualis N-C. Disponível em: <http://comissoes.sbc.org.br/ce-sc/sbsc/anais SBSC2006.pdf>. Acesso em: 02 mar. 2017.
[ET002]		PIMENTEL, MARIANO; FUKS, HUGO (Org.).(2011). **Sistema Colaborativos.** Local de publicação: Rio de Janeiro. Editora Elsevier/Campos/SBC, 2011. pp. 17-49.V1, Cap. 2, ISBN 978-85-352-4669-8. 2011.
[ET003]		SCHEIDEGGER, PATRICIA MERLIM LIMA. **Análise ontológica na construção de definições: diretrizes baseadas em ontologia de fundamentação**. 104 f. Dissertação de Mestrado (PPGI - Programa de Pós-Graduação em Informática, Instituto de Matemática e Instituto Tércio Pacciti) - UFRJ, Universidade Federal do Rio de Janeiro, RJ, 2016.
[ET004]		BARCELOS, P. P. F. **Análise Arquitetural, Ontológica e Proposta de Modelo de Referência para a Recomendação ITU-T G**. 805. Diss. Dissertação (Mestrado) - Universidade Federal do Espírito Santo, 2011.
[ET005]		CARRARETTO, R. **Separating ontological and informational concerns: a model-driven approach for conceptual modeling**. 2012. 120 f. Dissertação (mestrado em informática)- universidade federal do espírito santo, vitória, 2012.
[ET006]		SOUZA, ANDRE; GUIZZARDI, RENATA; CAMPOS, MARIA LUIZA M.; GUIZZARDI, GIANCARLO. **Extending an Ontology Editor for Domain-related Ontology Patterns Reuse: An Application in the Collaboration Domain**. In: ONTOBRAS – Seminário de Pesquisa em Ontologias do Brasil, 4., 2016, Brasil. Anais...Curitiba, 2016.
[ET007]		DOING-HARRIS, K., LIVNAT, Y., AND MEYSTRE, S. (2015). **Automated concept and relationship extraction for the semi-automated ontology management system (SEAM)**. Journal of Biomedical Semantics, Vol. 6, N. 15.
[ET008]		SILVA, MAURÍCIO SEVERO DA. **Collabcode: ferramenta para apoio ao desenvolvimento distribuído e colaborativo de software**. 2013. 109 f. Trabalho de Conclusão de Curso (Bacharel em Sistemas de Informação)- Título e curso Centro Universitário - UNIVATES, Lajeado, 2013.
[ET009]		FALBO, R. A., BARCELLOS, M.P., NARDI, J.C., GUIZZARDI, G. et al. **Organizing Ontology Design Patterns as Ontology Pattern Languages**, 10th Extended Semantic Web Conference, Montpellier, France, 2013.
[ET010]		SANTOS, VICTOR AMORIM DOS. **Uma Abordagem Baseada Em Padrões Para Construção De Modelos Conceituais Em Ontouml.** 2015. 107 f. Dissertação (Programa de Pós-Graduação em Informática)- Universidade Federal do Espírito Santo, Universidade Federal do Espírito Santo, VITÓRIA – ES, Brasil, 2015.

[ET011]		QUIRINO, GLAICE KELLY DA SILVA. **Uma Notação Visual para Representação de Linguagens de Padrões Ontológicos**. 2016. Dissertação de Mestrado - Universidade Federal Do Espírito Santo Centro Tecnológico Departamento De Informática Programa De Pós-Graduação Em Informática, Universidade de Vitória.
[ET012]		FALBO, RICARDO; BARCELLOS, MONALESSA; RUY, FABIANO; GUIZZARDI, GIANCARLO; GUIZZARDI, RENATA.; et al. **Ontology Pattern Languages**. 2016. Disponível em: <https://www.researchgate. net/publication/305808689>. Acesso em: 01 ago. 2017.
[ET013]		QUIRINO, G. K.; NARDI, J. C.; BARCELLOS, M. P. et al. **Towards an Service Ontology Pattern Language**. Proceedings of the 34th International Conference on Conceptual Modeling (ER 2015). 2015. Stockolm, Sweden.
[ET014]		RUY, F. B.; FALBO, R. DE A.; BARCELLOS, M. P. et al. **An ISO-based software process ontology pattern language and its application for harmonizing standards**. ACM SIGAPP Applied Computing Review, v. 15, n. 2, p. 27–40, 2015.
[ET015]		FALBO, R. DE A.; QUIRINO, G. K.; NARDI, J. et al. **An Ontology Pattern Language for Service Modeling**. The 31st ACM/SIGAPP Symposium on Applied Computing. 2016. Pisa, Italy.

APÊNDICE B

CHECKLIST DE AVALIAÇÃO DOS REQUISITOS DE GROUPWARE

Logo Empresa	CHECKLIST DE AVALIAÇÃO DOS REQUISITOS	Código: 0000
		Revisão: 0000
Setor:		Data:
Empresa:		

	ITENS VISTORIADOS	BOM	AJUSTE		N.A	OBSERVAÇÕES
			SIM	NÃO		
1	Averiguar os **REQUISITOS DE CLIENTE** para entender o que o produto deve fazer;	☐	☐	☐	☐	
2	Verificar a existência dos **REQUISITOS GERAIS** do groupware;	☐	☐	☐	☐	
	2.1 Os requisitos de identidade individual dos membros (papéis e responsabilidades de cada um);	☐	☐	☐	☐	
	2.2 Os requisitos de posição do usuário (nomes e/ou ícones visíveis na tela);	☐	☐	☐	☐	
	2.3 Os requisitos de visão pessoal ou padrão, particular, pública e convergente (visão especificada por cada usuário para um problema);	☐	☐	☐	☐	
	2.4 Os requisitos de conhecimento de outros integrantes, a fim de saber da presença de um usuário no grupo;	☐	☐	☐	☐	
	2.5 Os requisitos de controle da palavra (algumas situações necessitam de somente um usuário manipulando um aplicativo ou emitindo opiniões);	☐	☐	☐	☐	
	2.6 Controle de acesso (direitos de acesso para cada usuário);	☐	☐	☐	☐	
	2.7 Atualização e sincronismo (para não haver divergências na edição e/ou na entrada e saída de dados);	☐	☐	☐	☐	
	2.8 Coordenação da informação (coordenação de atividade ou grupo de atividades, aumentando a eficiência).	☐	☐	☐	☐	
3	Verificar a existência dos **REQUISITOS DE USUÁRIO** do groupware;	☐	☐	☐	☐	
	3.1 Acesso aos objetos compartilhados e às ferramentas de colaboração;	☐	☐	☐	☐	
	3.2 Auxílio na escolha das ferramentas apropriadas;	☐	☐	☐	☐	
	3.3 Fornecimento de informações de percepção;	☐	☐	☐	☐	
	3.4 Colaboração síncrona e assíncrona;	☐	☐	☐	☐	
	3.5 Acesso ao ambiente independente da estação de trabalho;	☐	☐	☐	☐	
	3.6 Fornecimento de espaço privativo e público e transição entre eles ;	☐	☐	☐	☐	
	3.7 Extensão dinâmica do ambiente;	☐	☐	☐	☐	
	3.8 Sincronização entre ferramentas diferentes;	☐	☐	☐	☐	
	3.9 Mobilidade;	☐	☐	☐	☐	
	3.10 Agrupamento de ferramentas;	☐	☐	☐	☐	
	3.11 Alta performance.	☐	☐	☐	☐	
4	Verificar a existência dos **REQUISITOS DE DESENVOLVEDOR** do groupware;	☐	☐	☐	☐	
	4.1 Reúso da experiência e conhecimento anteriores;	☐	☐	☐	☐	

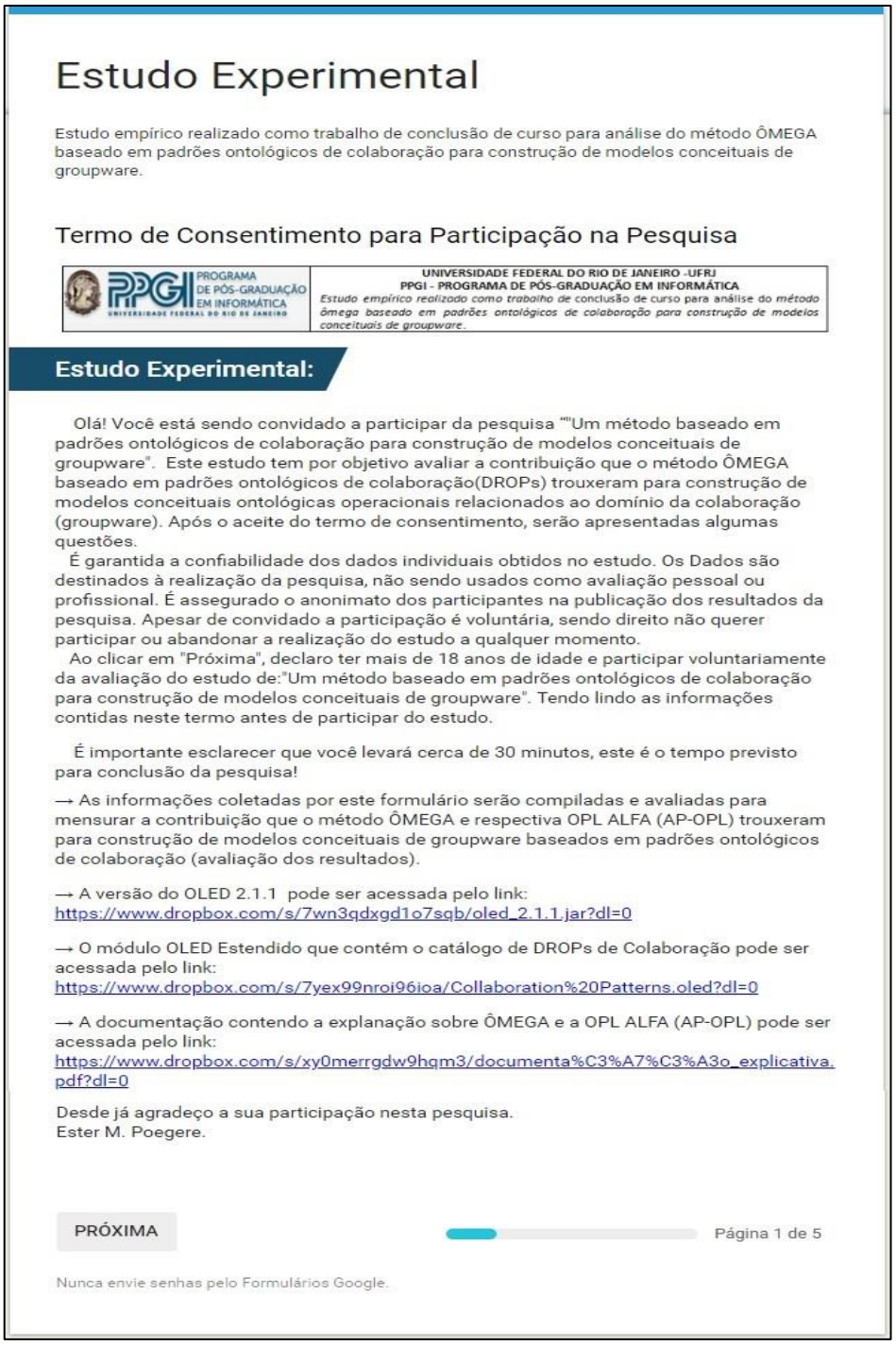
	4.2 Aproveitamento do modelo de dados;	☐	☐	☐	☐	
	4.3 Compartilhamento transparente de dados;	☐	☐	☐	☐	
	4.4 Suporte a dados locais e compartilhados;	☐	☐	☐	☐	
	4.5 Acesso às informações de percepção;	☐	☐	☐	☐	
	4.6 Disponibilização de novas ferramentas;	☐	☐	☐	☐	
	4.7 Integração com ferramentas externas;	☐	☐	☐	☐	
	4.8 Escalabilidade;	☐	☐	☐	☐	
	4.9 Suporte a ferramentas localizadas no servidor.	☐	☐	☐	☐	
5	Averiguar as restrições do projeto;	☐	☐	☐	☐	
6	Negociar com os envolvidos, caso haja requisitos conflitantes;	☐	☐	☐	☐	
7	Elaborar um plano para fazer as alterações (se houverem).	☐	☐	☐	☐	

CONCLUSÃO DA AVALIAÇÃO:

APÊNDICE C

FORMULÁRIO UTILIZADO NO ESTUDO EXPERIMENTAL

A1. Termo de Consentimento utilizado nos Estudos Experimentais



Estudo Experimental

Estudo empírico realizado como trabalho de conclusão de curso para análise do método ÔMEGA baseado em padrões ontológicos de colaboração para construção de modelos conceituais de groupware.

Termo de Consentimento para Participação na Pesquisa

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO -UFRJ
PPGI - PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA
Estudo empírico realizado como trabalho de conclusão de curso para análise do método ômega baseado em padrões ontológicos de colaboração para construção de modelos conceituais de groupware.

Estudo Experimental:

Olá! Você está sendo convidado a participar da pesquisa ""Um método baseado em padrões ontológicos de colaboração para construção de modelos conceituais de groupware". Este estudo tem por objetivo avaliar a contribuição que o método ÔMEGA baseado em padrões ontológicos de colaboração(DROPs) trouxeram para construção de modelos conceituais ontológicas operacionais relacionados ao domínio da colaboração (groupware). Após o aceite do termo de consentimento, serão apresentadas algumas questões.

É garantida a confiabilidade dos dados individuais obtidos no estudo. Os Dados são destinados à realização da pesquisa, não sendo usados como avaliação pessoal ou profissional. É assegurado o anonimato dos participantes na publicação dos resultados da pesquisa. Apesar de convidado a participação é voluntária, sendo direito não querer participar ou abandonar a realização do estudo a qualquer momento.

Ao clicar em "Próxima", declaro ter mais de 18 anos de idade e participar voluntariamente da avaliação do estudo de:"Um método baseado em padrões ontológicos de colaboração para construção de modelos conceituais de groupware". Tendo lindo as informações contidas neste termo antes de participar do estudo.

É importante esclarecer que você levará cerca de 30 minutos, este é o tempo previsto para conclusão da pesquisa!

→ As informações coletadas por este formulário serão compiladas e avaliadas para mensurar a contribuição que o método ÔMEGA e respectiva OPL ALFA (AP-OPL) trouxeram para construção de modelos conceituais de groupware baseados em padrões ontológicos de colaboração (avaliação dos resultados).

→ A versão do OLED 2.1.1 pode ser acessada pelo link:
https://www.dropbox.com/s/7wn3qdxgd1o7sqb/oled_2.1.1.jar?dl=0

→ O módulo OLED Estendido que contém o catálogo de DROPs de Colaboração pode ser acessada pelo link:
https://www.dropbox.com/s/7yex99nroi96ioa/Collaboration%20Patterns.oled?dl=0

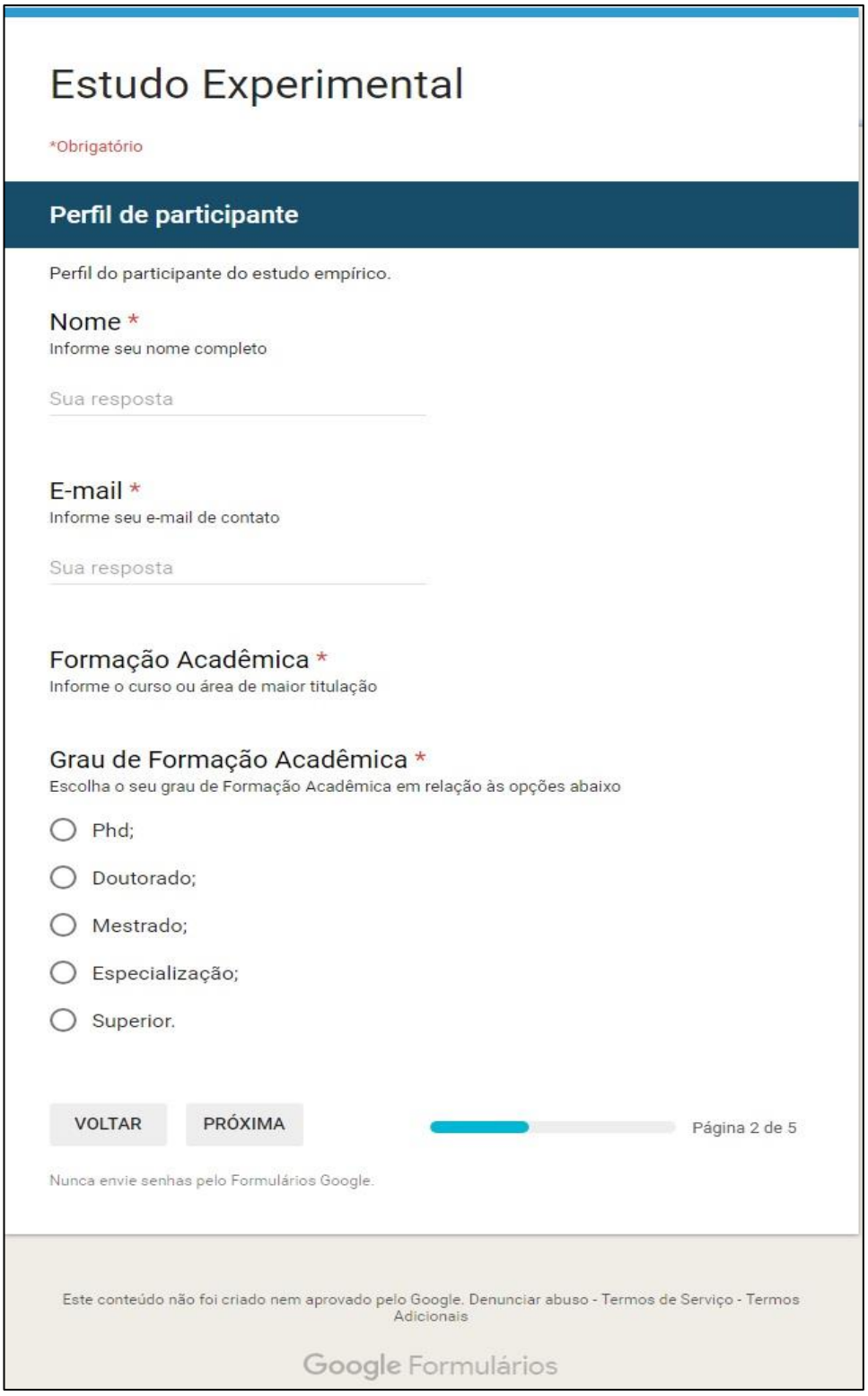
→ A documentação contendo a explanação sobre ÔMEGA e a OPL ALFA (AP-OPL) pode ser acessada pelo link:
https://www.dropbox.com/s/xy0merrgdw9hqm3/documenta%C3%A7%C3%A3o_explicativa.pdf?dl=0

Desde já agradeço a sua participação nesta pesquisa.
Ester M. Poegere.

PRÓXIMA

Página 1 de 5

Nunca envie senhas pelo Formulários Google.



Estudo Experimental

*Obrigatório

Perfil de participante

Perfil do participante do estudo empírico.

Nome *
Informe seu nome completo

Sua resposta

E-mail *
Informe seu e-mail de contato

Sua resposta

Formação Acadêmica *
Informe o curso ou área de maior titulação

Grau de Formação Acadêmica *
Escolha o seu grau de Formação Acadêmica em relação às opções abaixo

- ◯ Phd;
- ◯ Doutorado;
- ◯ Mestrado;
- ◯ Especialização;
- ◯ Superior.

VOLTAR PRÓXIMA Página 2 de 5

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. Denunciar abuso - Termos de Serviço - Termos Adicionais

Google Formulários

Estudo Experimental

*Obrigatório

Experiência em Modelagem Conceitual

Qual seu grau de experiência em Modelagem Conceitual ? *
Exemplo de experiências: uso UML, ONTOLOGIA, ONTOUML e OPLs

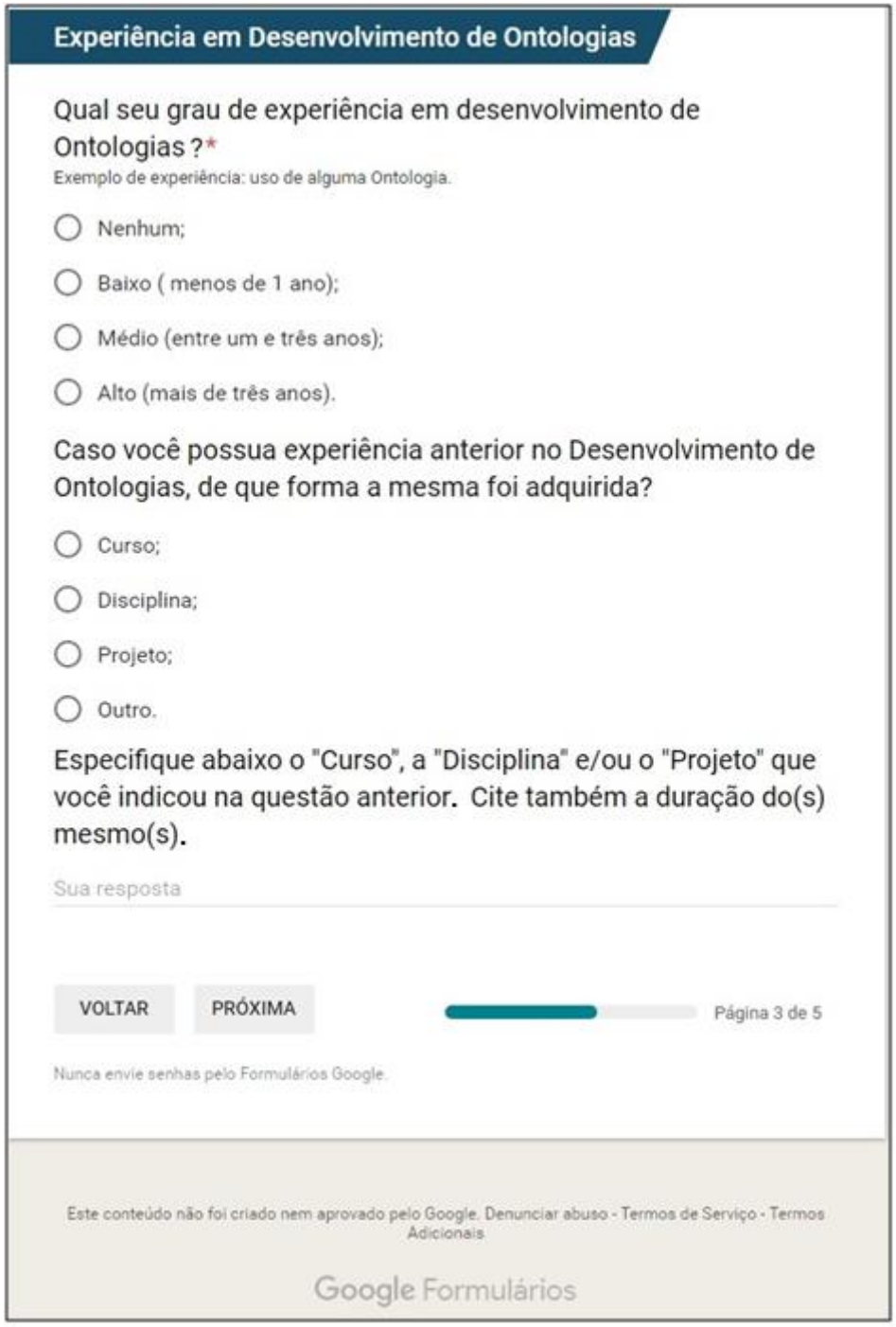
- ◯ Nenhum;
- ◯ Baixo (menos de 1 ano);
- ◯ Médio (entre um e três anos);
- ◯ Alto (mais de três anos).

Caso possua experiência em Modelagem Conceitual, de que forma a mesma foi adquirida? *

- ◯ Curso;
- ◯ Disciplina;
- ◯ Projeto;
- ◯ Outro.

Especifique abaixo o "Curso", a "Disciplina" e/ou o "Projeto" que você indicou na questão anterior. Cite também a duração do(s) mesmo(s).

Sua resposta



Experiência em Desenvolvimento de Ontologias

Qual seu grau de experiência em desenvolvimento de Ontologias ?*

Exemplo de experiência: uso de alguma Ontologia.

- ◯ Nenhum;
- ◯ Baixo (menos de 1 ano);
- ◯ Médio (entre um e três anos);
- ◯ Alto (mais de três anos).

Caso você possua experiência anterior no Desenvolvimento de Ontologias, de que forma a mesma foi adquirida?

- ◯ Curso;
- ◯ Disciplina;
- ◯ Projeto;
- ◯ Outro.

Especifique abaixo o "Curso", a "Disciplina" e/ou o "Projeto" que você indicou na questão anterior. Cite também a duração do(s) mesmo(s).

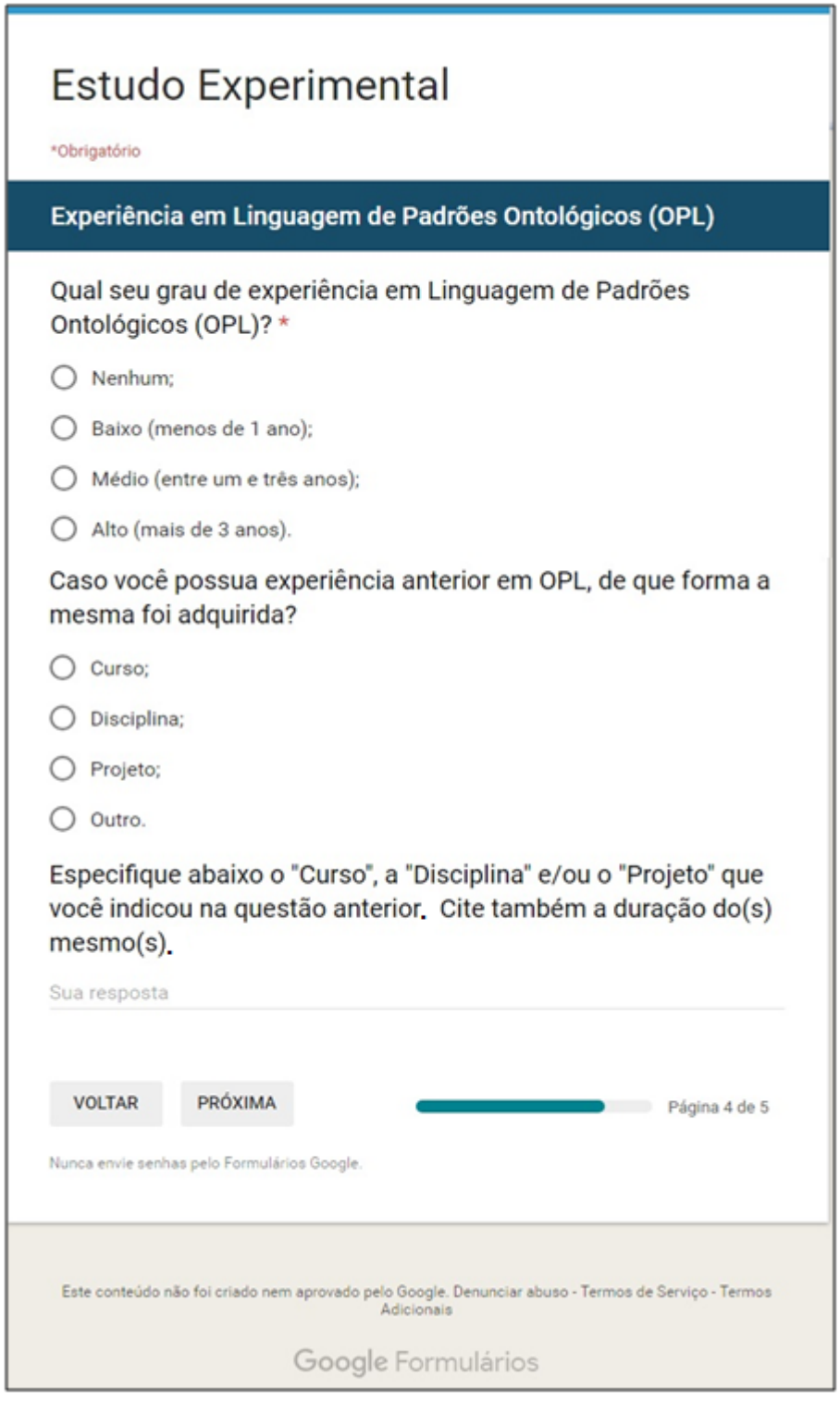
Sua resposta

VOLTAR PRÓXIMA Página 3 de 5

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. Denunciar abuso - Termos de Serviço - Termos Adicionais

Google Formulários



Estudo Experimental

*Obrigatório

Experiência em Linguagem de Padrões Ontológicos (OPL)

Qual seu grau de experiência em Linguagem de Padrões Ontológicos (OPL)? *

- ◯ Nenhum;
- ◯ Baixo (menos de 1 ano);
- ◯ Médio (entre um e três anos);
- ◯ Alto (mais de 3 anos).

Caso você possua experiência anterior em OPL, de que forma a mesma foi adquirida?

- ◯ Curso;
- ◯ Disciplina;
- ◯ Projeto;
- ◯ Outro.

Especifique abaixo o "Curso", a "Disciplina" e/ou o "Projeto" que você indicou na questão anterior. Cite também a duração do(s) mesmo(s).

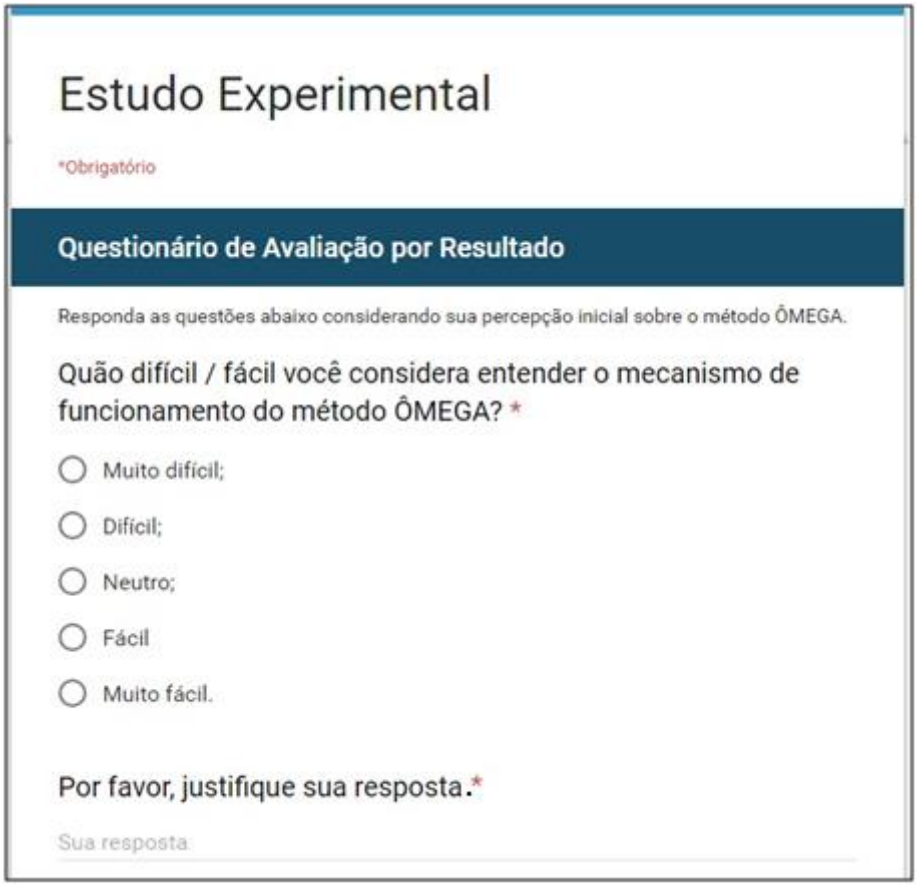
Sua resposta

VOLTAR PRÓXIMA Página 4 de 5

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. Denunciar abuso - Termos de Serviço - Termos Adicionais

Google Formulários



Estudo Experimental

*Obrigatório

Questionário de Avaliação por Resultado

Responda as questões abaixo considerando sua percepção inicial sobre o método ÔMEGA.

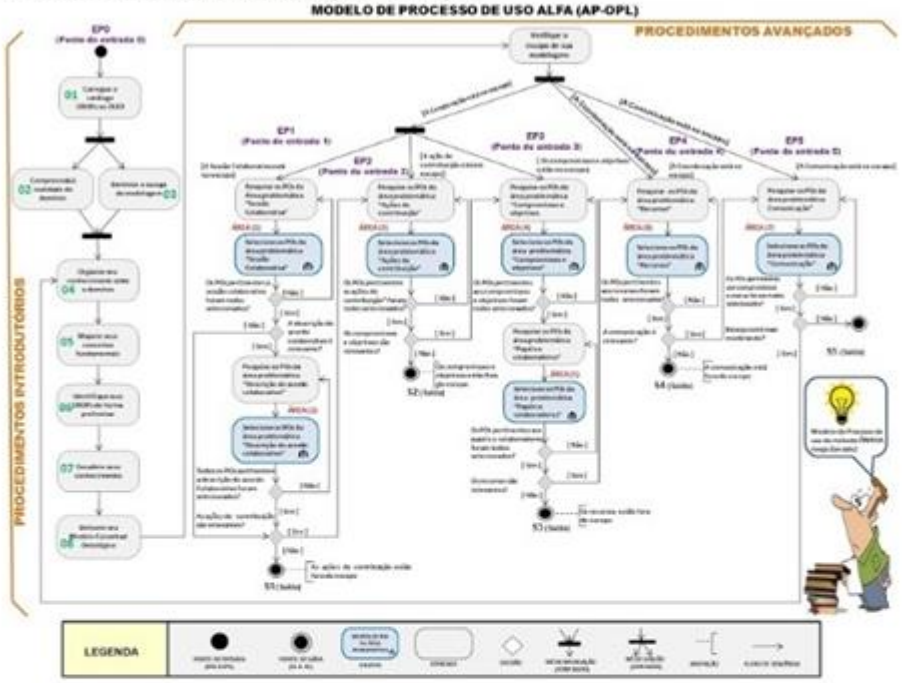
Quão difícil / fácil você considera entender o mecanismo de funcionamento do método ÔMEGA? *

- ◯ Muito difícil;
- ◯ Difícil;
- ◯ Neutro;
- ◯ Fácil
- ◯ Muito fácil.

Por favor, justifique sua resposta.*

Sua resposta

Quão difícil você considera entender o mecanismo de funcionamento da OPL que corresponde ao "Modelo de Processo de Uso ALFA (AP-OPL)"? *

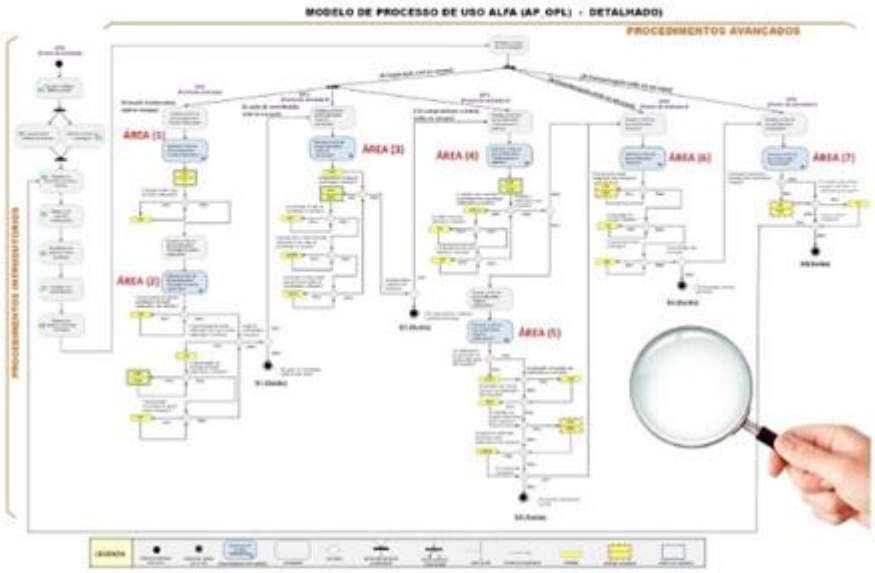


- ◯ Muito difícil;
- ◯ Difícil;
- ◯ Neutro;
- ◯ Fácil
- ◯ Muito fácil.

Por favor, justifique sua resposta.*

Sua resposta

Quão difícil / fácil você considera entender os elementos da OPL ALFA (AP-OPL)? *



- ◯ Muito difícil;
- ◯ Difícil;
- ◯ Neutro;
- ◯ Fácil
- ◯ Muito fácil.

Por favor, justifique sua resposta.*

Sua resposta

Dentre os elementos listados anteriormente, qual deles você teve maior dificuldade para entender? Justifique sua resposta. *

Sua resposta

Você teria alguma sugestão para melhorar a forma como os padrões são conectados, ou seja, para melhorar apresentação do processo da OPL? *

Sua resposta

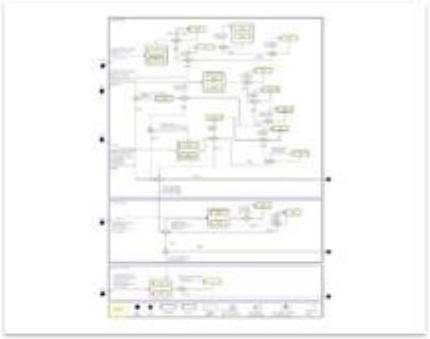
Você considera que essa versão ALFA (AP-OPL) tornou o processo mais claro? *

- ◯ Muito menos claro;
- ◯ Menos claro;
- ◯ Mantém a mesma clareza;
- ◯ Mais claro;
- ◯ Bem mais claro.

Por favor, justifique sua resposta.*

Sua resposta

Em relação à facilidade de entendimento do processo da OPL, qual representação você julga melhor?



A antiga que permitia apenas mostrar alguns padrões de colaboração;

A nova que contempla diferentes níveis de abstração, permitindo trabalhar com uma visão geral do processo, com mais detalhes que podem ser trabalhados no todo.

Por favor, justifique sua resposta.*

Sua resposta

Em sua avaliação, no geral, o método ÔMEGA contribui com o processo de Modelagem Conceitual Ontológica (Por exemplo, vide modelo conceitual do Brainstorming online - abaixo descrito)? *

- ◯ Indiferente;
- ◯ Sim;
- ◯ Não.

Por favor, justifique sua resposta.*

Sua resposta

Você teria alguma sugestão?

Sua resposta

VOLTAR ENVIAR Página 5 de 5

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. Denunciar abuso - Termos de Serviço - Termos Adicionais

Google Formulários

APÊNDICE D

QUESTIONÁRIO UTILIZADO NAS ENTREVISTAS DO ESTUDO EXPERIMENTAL

A2. Questionário usado na entrevista do estudo experimental:

Entrevista:

1. Qual o seu entendimento sobre sistemas colaborativos? Modelagem conceitual? UML? Ontologias?
2. O uso dos aportes teóricos de colaboração permite explicitar os conceitos centrais?
3. Qual é o seu entendimento sobre Padrões ontológicos (*ontology patterns*) e linguagens de padrões?
4. Ao conectar padrões ontológicos estamos proporcionando a oportunidade de sua validação sistêmica?
5. Para garantir o entendimento não ambíguo dos padrões de ontologia, propõe-se modelá-los com base em uma ontologia de fundamentação?
6. Para representar os padrões ontológicos propostos é fundamental o entendimento?
7. Existem diversas categorias de padrões ontológicos ? Qual é uso em nosso método?
8. Quantas visões e procedimentos o método ÕMEGA tem para construir modelos?
9. Em que momento você começam a usar a OPL? E como é realizada a integração dos padrões?
10. Entre os elementos da OPL, qual lhe gera dúvidas?
11. Você tem dúvidas em relação aos estereótipos utlizados em Padrões ontológicos relacionados ao domínio (*Domain-Related Ontology Patterns*)?
12. Em relação às questões de competência e intenções dos padrões como, você as define?
13. Qual é o principal motivo da criação do Método OMEGA e respectiva OPL ALFA (AP-OPL)?
14. Você possue experiência em desenvolvimento de modelos conceituais em ONTOUML?
15. Em seu ponto de vista, quais informações poderiam ter sido perpassadas para facilitar a construção de modelos através da OPL?