

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE ELETRICIDADE
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE

LUÍS EDUARDO SANTOS

**UMA TÉCNICA E FERRAMENTA ORIENTADAS POR
OBJETIVOS PARA A CONSTRUÇÃO DE ONTOLOGIAS DE
APLICAÇÃO**

São Luís
2013

LUÍS EDUARDO SANTOS

**UMA TÉCNICA E FERRAMENTA ORIENTADAS POR OBJETIVOS PARA A
CONSTRUÇÃO DE ONTOLOGIAS DE APLICAÇÃO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia da Eletricidade da Universidade Federal do Maranhão, para a obtenção do título de mestre em Engenharia da Eletricidade - Área de Concentração: Ciência da Computação.

Orientadora: Profa. Dra. Rosario Girardi

São Luís
2013

Santos, Luís Eduardo.

Uma técnica e ferramenta orientadas por objetivos para a construção de ontologias de aplicação / Luís Eduardo Santos. – São Luís, 2013.

114 f.

Impresso por computador (fotocópia).

Orientador: Rosario Girardi.

Dissertação (Mestrado) – Universidade Federal do Maranhão, Programa de Pós-graduação em Engenharia de Eletricidade, 2013.

1. Sistemas baseados em conhecimento – Ontologias de aplicação. 2. Sucessão legítima. I. Título.

CDU 004.89

**UMA TÉCNICA E FERRAMENTA ORIENTADAS POR OBJETIVOS
PARA A CONSTRUÇÃO DE ONTOLOGIAS DE APLICAÇÃO**

Luis Eduardo Santos

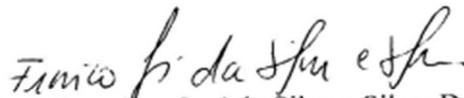
Dissertação aprovada em 29 de janeiro de 2013.



Profa. María del Rosario Girardi Gutiérrez, Ph.D
(Orientadora)



Profa. Renata Vieira, Ph.D.
(Membro da Banca Examinadora)



Prof. Francisco José da Silva e Silva, Dr.
(Membro da Banca Examinadora)

A minha mãe.

AGRADECIMENTOS

Agradeço a todos que contribuíram direta ou indiretamente para a elaboração desta dissertação, em especial:

A Deus, o criador de todas as coisas.

A minha mãe Antônia e a minha namorada Fernanda pelo amor, dedicação e incentivo.

À Professora Rosario, pela orientação segura e presente e pelos ensinamentos, imprescindíveis para a realização deste trabalho.

Aos meus companheiros de trabalho, em especial, Carla Faria, Fabio Vieira, Ivo Serra, Nailson Boaz e Rodrigo Sousa pelas suas inestimáveis contribuições.

Aos meus atuais colegas do grupo de pesquisa GESEC – Adriana Leite, Antonio Phillipi, Giulia Soares, Suzane Carvalho e William Corrêa – próximo dos quais estive trabalhando bastante nos últimos meses e que participaram direta ou indiretamente do andamento da presente dissertação, tanto com ideias quanto com produção, de modo secundário, porém essencial.

A toda Coordenação do Mestrado, coordenador, funcionários e professores, pelos bons serviços oferecidos, que foram fundamentais para a conclusão deste curso.

RESUMO

As ontologias de aplicação são estruturas de representação de conhecimento capazes de expressar um conjunto de entidades de um dado domínio, seus relacionamentos e axiomas, sendo utilizadas pelos modernos Sistemas Baseados em Conhecimento (SBC) no processo de tomada de decisões. Construir ontologias de aplicação exige o claro entendimento dos objetivos que o SBC pretende alcançar. Prover suporte sistematizado a formalização desses objetivos em uma ontologia de aplicação é a ideia fundamental da GAODT.

Proposta nesse trabalho, a GAODT é uma técnica orientada por objetivos para a construção de ontologias de aplicação. Para prover suporte a sua aplicação foi desenvolvida a ferramenta de software GAODTool, utilizada na avaliação da GAODT através de um estudo de caso na construção da “InheritanceLaw”, uma ontologia de aplicação para apoio a tomada de decisões no domínio da Sucessão Legítima, ramo do direito que governa a transferência do patrimônio de alguém, depois de sua morte, em virtude de lei.

Palavras-chave: Ontologias de Aplicação, Sucessão Legítima, Sistemas Baseados em Conhecimento, Bases de Conhecimento.

ABSTRACT

Application ontologies are knowledge representation structures capable of expressing a set of entities from a given domain, their relationships and axioms and has been used by modern Knowledge-Based Systems (KBS) in decision-making. Building application ontologies require a clear understanding of the goals of the KBS under development. Provide support for the formalization of these goals in an application ontology is the fundamental idea of GAODT.

GAODT, the main contribution of this work, is a goal-oriented technique for building application ontologies. To provide support to its application a software tool named GAODTool was developed and used in an evaluation that consisted in building “InheritanceLaw”, an application ontology to support decision making in the domain of Intestate Succession, branch of law that governs the transfer of properties from someone after his death.

Keywords: Application Ontologies, Intestate Succession, Knowledge-Based Systems, Knowledge Bases.

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Motivação	14
1.2	Objetivos	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	15
1.3	Estrutura do trabalho	15
2	O ESTADO DA ARTE	17
2.1	Introdução	17
2.2	Técnica DERONTO	17
2.3	Técnica 101	23
2.4	Técnica de Michael Gruninger e Mark Fox	27
2.5	Técnica de Uschold e King	31
2.6	Methontology	35
2.7	Análise Comparativa	41
2.8	Considerações Finais	43
3	A TÉCNICA GAODT	44
3.1	Introdução	44
3.2	Seleção dos Objetivos e Fatos	45
3.3	Representação dos Predicados em LPO	46
3.4	Especificação dos Axiomas em LPO	49
3.5	Especificação/Extensão da Ontologia de Aplicação	53
3.6	GAODTTool – Uma Ferramenta de Suporte a GAODT	56
3.7	Discussão sobre a GAODT	61
3.8	Considerações Finais	62
4	O ESTUDO DE CASO	63
4.1	Introdução	63
4.2	O Domínio Abordado: A Sucessão Legítima	63
4.3	Construindo a Ontologia de Aplicação “InheritanceLaw”	66
4.4	Considerações Finais	79
5	CONCLUSÕES	80
5.1	Resultados e contribuições da pesquisa	80
5.1.1	Publicações	81
5.2	Trabalhos futuros	82
	REFERÊNCIAS	83
	APÊNDICES	86

LISTA DE FIGURAS

Figura 2.2.1: Ciclo de vida da técnica DERONTO.....	18
Figura 2.2.2: Diagrama DER desenvolvido a partir do enunciado do problema [4].....	19
Figura 2.2.3: Transformação da relação “C_Corrente” em uma entidade associada a novas relações [4].....	19
Figura 2.2.4: DER transformado em novas entidades e relações [4].....	20
Figura 2.2.5: Axioma em Lógica de Descrição [4].....	22
Figura 2.3.1: Ciclo de vida da técnica 101.....	24
Figura 2.3.2: Verificando a existência de hierarquia.....	26
Figura 2.4.1: Ciclo de vida da técnica proposta Michael Gruninger e Mark Fox.....	28
Figura 2.4.2: Questão de competência representada em LPO.....	29
Figura 2.4.3: Restringindo a interpretação do conceito “ViajanteJovem”.....	30
Figura 2.4.4: Exemplo de um Teorema de completude.....	30
Figura 2.4.5: Alguns axiomas, uma consulta e a resposta em uma ontologia.....	30
Figura 2.5.1: Ciclo de vida da técnica de Uschold e King.....	31
Figura 2.5.2: Hierarquia desenvolvida utilizando a abordagem Top-down.....	33
Figura 2.5.3: Hierarquia desenvolvida utilizando a abordagem Bottom-up.....	33
Figura 2.5.4: Hierarquia desenvolvida utilizando a abordagem Middle-out.....	34
Figura 2.6.1: Ciclo de vida da Methontology [14].....	35
Figura 2.6.2: Subatividades da atividade de conceitualização.....	37
Figura 2.6.3: Taxonomia e suas características no domínio de viagens.....	38
Figura 2.6.4: Diagrama de relações ad hoc no domínio de viagens.....	38
Figura 3.1.1: A visão geral da técnica GAODT.....	45
Figura 3.3.1: Atividade “Representação dos Predicados em LPO”.....	46
Figura 3.4.1: Atividade “Especificação dos Axiomas em LPO”.....	49
Figura 3.5.1: Atividade “Especificação/Extensão da Ontologia de Aplicação”.....	53
Figura 3.5.2: Exemplo de um axioma representado em RuleML.....	54
Figura 3.5.3: Relacionamentos taxonômicos da ontologia de aplicação.....	55
Figura 3.5.4: Exemplo de um axioma representado em RuleML.....	55
Figura 3.6.2: Tela principal da ferramenta da GAODTool.....	57
Figura 3.6.3: Área “Adição de Objetivos e Fatos”.....	57
Figura 3.6.4: Lista de objetivos e fatos contidos na área “Seleção dos Objetivos e Fatos”.....	58
Figura 3.6.5: Área “Representação dos Predicados em LPO”.....	58
Figura 3.6.6: Área “Especificação dos Axiomas em LPO”.....	59
Figura 3.6.7: Lista de axiomas desenvolvidos.....	59
Figura 3.6.8: Área “Especificação da Ontologia de Aplicação”.....	60
Figura 3.6.9: Arquivo contendo a ontologia de aplicação gerada.....	60
Figura 4.3.1: Inserção de um objetivo no domínio da Sucessão Legítima.....	66
Figura 4.3.2: Visão parcial dos objetivos e fatos do sistema no domínio da Sucessão Legítima.....	67
Figura 4.3.3: Entidades especificadas através da subárea “Identificação das entidades”.....	68
Figura 4.3.4: Entidades redefinidas através da subárea “Redefinição das entidades”.....	68

Figura 4.3.5: Relacionamentos especificados através da subárea “Identificação dos relacionamentos”	69
Figura 4.3.6: Relacionamentos redefinidos através da subárea “Redefinição dos relacionamentos”	69
Figura 4.3.7: Aridade definida através da subárea “Definição da aridade”	70
Figura 4.3.8: Predicados construídos através da subárea “Definição dos predicados”	70
Figura 4.3.9: Predicados redefinidos através da subárea “Redefinição das entidades dos predicados”	71
Figura 4.3.10: Predicados que compõem a condição e conclusão do axioma definidos através da subárea “Definição da condição e conclusão”	71
Figura 4.3.11: Predicados da condição unidos pela conjunção através da subárea “Definição dos operadores booleanos”	72
Figura 4.3.12: Quantificador definido para cada variável dos predicados, através da subárea “Definição dos quantificadores”	72
Figura 4.3.13: Definindo a implicação do axioma, através da subárea “Definição de implicação ou equivalência”	73
Figura 4.3.14: Axioma desenvolvido após termino da área “Especificação dos Axiomas em LPO”	73
Figura 4.3.15: Objetivos selecionados pelo desenvolvedor para criar um novo axioma em LPO.	74
Figura 4.3.16: Predicados em LPO para formarem o novo axioma	74
Figura 4.3.17: Novo axioma gerado e armazenado na lista de axiomas desenvolvidos.	75
Figura 4.3.18: Axioma gerado e armazenado na lista de axiomas desenvolvidos.	75
Figura 4.3.19: Lista parcial dos axiomas desenvolvidos no domínio da Sucessão Legítima.	75
Figura 4.3.20: Visão parcial dos axiomas representados em RuleML	76
Figura 4.3.21: Definição das classes da ontologia de aplicação “InheritanceLaw”	76
Figura 4.3.22: Visão parcial dos relacionamentos não taxonômicos da “InheritanceLaw”	77
Figura 4.3.23: Relacionamentos taxonômicos definidos.	77
Figura 4.3.24: Propriedades da ontologia	78
Figura 4.3.25: Taxonomia, relacionamentos não taxonômicos e propriedades da ontologia de aplicação “InheritanceLaw”	79

LISTA DE TABELAS

Tabela 2.2.1: Entidades do DER transformadas em conceitos da ontologia.....	20
Tabela 2.2.2: Relações de herança identificadas.....	20
Tabela 2.2.3: Parte da documentação da ontologia.....	21
Tabela 2.2.4: Atributos da ontologia.....	22
Tabela 2.2.5: Relações identificadas da ontologia.....	22
Tabela 2.6.1: Glossário de termos do domínio de viagens.....	37
Tabela 2.6.2: Dicionário de conceitos no domínio de viagens.....	39
Tabela 2.6.3: Relações binárias ad hoc descritas.....	39
Tabela 2.6.4: Alguns atributos de instância descritos.....	39
Tabela 2.6.5: Descrevendo alguns atributos de classe.....	39
Tabela 2.6.6: Uma constante no domínio de viagens.....	40
Tabela 2.6.7: Um axioma no domínio de viagens.....	40
Tabela 2.6.8: Uma regra no domínio de viagens.....	40
Tabela 2.6.9: Algumas instâncias no domínio de viagens.....	41
Tabela 2.7.1: Análise comparativa das técnicas para a construção de ontologias.....	42
Tabela 3.2.1: Lista de objetivos e fatos do sistema.....	46
Tabela 3.3.1: Entradas e saídas da subatividade “Identificação das entidades”.....	47
Tabela 3.3.2: Entradas e saídas da subatividade “Redefinição das entidades”.....	47
Tabela 3.3.3: Entradas e saídas da subatividade “Identificação dos relacionamentos”.....	47
Tabela 3.3.4: Entradas e saídas da subatividade “Redefinição dos relacionamentos”.....	48
Tabela 3.3.5: Entradas e saídas da subatividade “Definição da aridade”.....	48
Tabela 3.3.6: Itens selecionados traduzidos para predicados em LPO.....	48
Tabela 3.3.7: Entradas e saídas da subatividade “Redefinição das entidades dos predicados”.....	49
Tabela 3.4.1: Saída da subatividade “Definição da condição e conclusão”.....	49
Tabela 3.4.2: Objetivos representados como predicados.....	51
Tabela 3.4.3: Axioma desenvolvido na atividade “Especificação dos Axiomas em LPO”.....	51
Tabela 3.4.4: Axioma desenvolvido através do processo iterativo da GAODT.....	51
Tabela 3.4.5: Produto final da atividade “Especificação dos Axiomas em LPO”.....	52
Tabela 3.5.1: Relacionamentos não taxonômicos da ontologia de aplicação.....	56
Tabela 3.5.2: Propriedades da ontologia de aplicação.....	56
Tabela 3.7.1: Classificação da GAODT com relação aos critérios de análise da seção 2.7.....	62

LISTA DE ABREVIATURAS E SIGLAS

CSS	Cascading Style Sheets
DER	Diagrama de Entidade Relacionamento
DERONTO	Método para Construção de Ontologias a partir de diagramas Entidade-Relacionamento
GAODT	Goal-Oriented Application Ontology Development Technique
GAODTool	Goal-Oriented Application Ontology Development Tool
LPO	Lógica de Primeira Ordem
MADAE-Pro	Multiagent Domain and Application Engineering Process
Methontology	Methodology ontology
OWL	Web Ontology Language
PHP	PHP: Hypertext Preprocessor
POSL	Positional-Slotted Language
RuleML	Rule Markup Language
SBC	Sistemas Baseados em Conhecimento
TAILP	Technique for ontology Axioms extraction using Inductive Logic Programming
XHTML	Extensible Hypertext Markup Language

1 INTRODUÇÃO

Uma ontologia é uma especificação formal explícita de uma conceituação compartilhada de um domínio de interesse [15]. Conceituação refere-se a um modelo abstrato de algum fenômeno do mundo. Explícito significa que o tipo de conceitos utilizados e as limitações do seu uso, são explicitamente definidos. Formal refere-se ao fato de que a ontologia deve ser legível por máquina. Compartilhada reflete a noção de que uma ontologia captura o conhecimento consensual, isto é, não é privada de algum indivíduo, mas aceita por um grupo.

As ontologias como estruturas de representação de conhecimento ganharam importância na última década. Atualmente, elas são aplicadas na comunicação de agentes de software, na integração da informação, na composição de Web Services, na descrição do conteúdo para facilitar a sua recuperação, no processamento da linguagem natural, na Web Semântica, na construção de Sistemas Baseados em Conhecimento (SBC) e nas aplicações de gerenciamento de conhecimento.

De acordo com Guarino [17], as ontologias são classificadas quanto a sua generalidade, como: de alto-nível, domínio, tarefa e aplicação. As ontologias de alto-nível descrevem conceitos genéricos como tempo e espaço, independentes de um domínio em particular. As ontologias de domínio explicitam o vocabulário do domínio e suas relações, por exemplo, o domínio jurídico. As ontologias de tarefa descrevem as atividades presentes nesse domínio, por exemplo, o acesso à informação. Finalmente, as ontologias de aplicação são especializações das ontologias de domínio e de tarefa, sendo utilizadas em uma aplicação em particular, por exemplo, em aplicações para o acesso a informação jurídica.

1.1 Motivação

A hierarquização de ontologias proposta por Guarino [17] promove a reutilização de ontologias, ou seja, para construir ontologias de aplicação é necessário estender as ontologias de domínio e de tarefa, e essas por sua vez, estendem as ontologias de alto-nível. Entretanto, na prática observa-se que construir ontologias reusáveis é um processo caro e dispendioso. Portanto, construir primeiramente ontologias de aplicação e depois generalizá-las em ontologias de domínio e de tarefa, apresenta-se como uma alternativa adequada [12].

Várias técnicas foram desenvolvidas para oferecer suporte no processo de construção de ontologias, porém, a maioria aborda o desenvolvimento de ontologias de domínio, deixando de contemplar as ontologias de aplicação.

Por esses motivos, propõem-se a GAODT (“Goal-Oriented Application Ontology Development Technique”), uma técnica orientada por objetivos para a construção de ontologias de aplicação. Sua aplicação é ilustrada com o desenvolvimento de uma ontologia de aplicação no domínio da Sucessão Legítima um dos tipos de sucessão disciplinado pelo Direito Sucessório.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral desse trabalho é contribuir com a qualidade e produtividade no desenvolvimento de ontologias de aplicação, por meio da proposição de uma solução computacional que automatize parcialmente seu desenvolvimento.

1.2.2 Objetivos Específicos

- Realização de uma revisão crítica do estado da arte das técnicas para o desenvolvimento de ontologias, com o intuito de encontrar necessidades ainda não satisfeitas.
- Desenvolvimento de uma técnica para a construção de ontologias de aplicação a qual é guiada pelos objetivos do sistema no qual a ontologia de aplicação será utilizada.
- Construção de uma ferramenta de suporte a técnica proposta.
- Avaliação da técnica e ferramenta propostas através do desenvolvimento de um estudo de caso na construção de uma ontologia de aplicação no domínio da Sucessão Legítima.

1.3 Estrutura do trabalho

Este trabalho, incluindo a introdução, está estruturado em cinco capítulos.

O segundo capítulo apresenta as mais representativas técnicas do estado da arte no desenvolvimento de ontologias, sendo ilustradas por meio de exemplos práticos. Por fim, é realizada uma análise comparativa das técnicas apresentadas.

O terceiro capítulo apresenta a GAODT, uma técnica orientada por objetivos para a construção de ontologias de aplicação. A GAODT traduz os objetivos e fatos em linguagem natural que expressam os requisitos de um SBC para regras e fatos em Lógica de Primeira Ordem, que em seguida são mapeados para uma ontologia de aplicação.

O quarto capítulo apresenta a avaliação da técnica GAODT através do desenvolvimento de um estudo de caso que consistiu na construção de uma ontologia de aplicação no domínio da Sucessão Legítima.

No quinto capítulo são apresentadas as considerações finais do trabalho, destacando os resultados obtidos e os trabalhos futuros que poderão ser desenvolvidos a partir desta pesquisa.

2 O ESTADO DA ARTE

2.1 Introdução

Com a crescente importância das ontologias, surgiram inúmeras técnicas para construí-las. Contudo, alguns desenvolvedores ao construir uma ontologia não utilizam uma técnica, seja ela criada ou adotada. Isso acarreta a passagem direta da etapa de aquisição do conhecimento para a etapa de implementação, gerando problemas, os quais são:

- Modelo conceitual da ontologia fica implícito no código;
- Como a ontologia está intimamente ligada ao código, surgem problemas no reuso;
- Dificuldades na comunicação entre o especialista e os desenvolvedores;
- As chances de haver inconsistência são muito grandes, pois a passagem da aquisição do conhecimento para a implementação se dá de forma brusca;
- Dependendo da linguagem escolhida para a codificação, pode-se limitar a capacidade de descrição conceitual do domínio da ontologia.

Existem muitas técnicas desenvolvidas, porém sua escolha está relacionada aos objetivos finais que se deseja alcançar com o desenvolvimento da ontologia, como o nível de detalhes dos processos e sua manutenção, assim como o tipo de ontologia que se deseja construir para o domínio considerado.

Nas seções seguintes 2.2 a 2.6 são apresentadas cinco técnicas do estado da arte para a construção de ontologias: DERONTO [4], Técnica 101 [26], Técnica de Uschold e King [13] [35], Técnica de Gruninger e Fox [13] [16] e a Methontology [13] [11]. Na seção 2.7 é realizada uma análise comparativa entre as técnicas estudadas. Na seção 2.8 são apresentadas as considerações finais do capítulo.

2.2 Técnica DERONTO

A técnica DERONTO é voltada para a criação de ontologias de domínio e aplicação a partir de um Diagrama de Entidade Relacionamento (DER) que é fornecido pelo usuário com conhecimento em banco de dados ao desenvolvedor da ontologia, e seu objetivo principal é auxiliar os profissionais da área de informática com pouco conhecimento no processo de

desenvolvimento de ontologias [4].

A técnica DERONTO consiste de seis etapas: “Criação do DER”, “Reescrita do DER”, “Definição dos conceitos e subconceitos da ontologia”, “Definição das propriedades da ontologia”, “Definição dos axiomas” e a “Verificação e modificação da ontologia”. A Figura 2.2.1 ilustra as etapas da DERONTO.

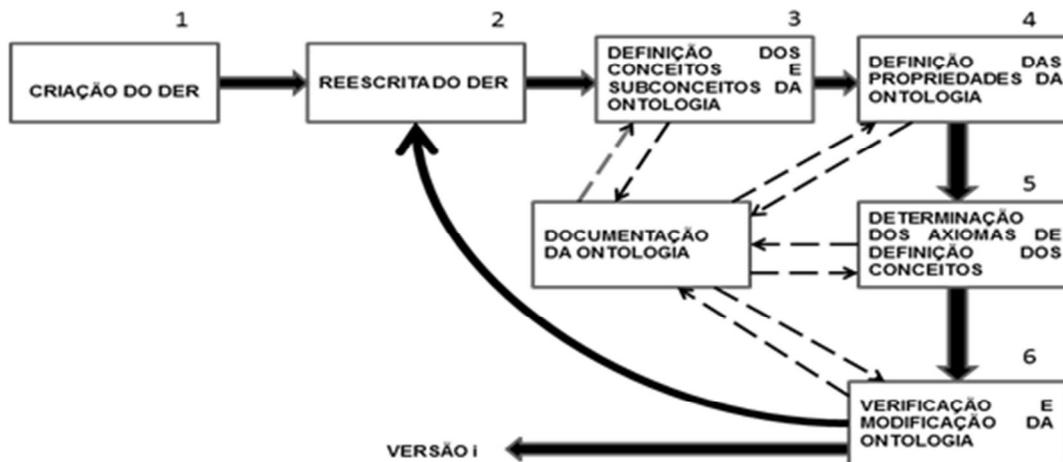


Figura 2.2.1: Ciclo de vida da técnica DERONTO

A etapa “Criação do DER” consiste em desenvolver um modelo conceitual da ontologia por meio de um DER, a partir de um enunciado ou uma lista de perguntas fornecidas ao desenvolvedor pelo usuário. Por exemplo, o usuário fornece o seguinte enunciado ao desenvolvedor: uma agência bancária possui diversos correntistas, estes podem ter diversas contas correntes de diferentes tipos, cada uma localizada em uma agência específica e uma conta pode pertencer a mais de um correntista. A conta possui um número e saldo, o correntista pode ser pessoa física ou jurídica. Se uma conta possuir um saldo superior a R\$ 12.500,00 então ela será considerada uma conta especial. Por fim, a ontologia deve responder as seguintes perguntas elaboradas pelo desenvolvedor:

- Qual o nome do correntista que possui conta especial?
- Qual o CPF do correntista X?
- Qual a data de nascimento do correntista Y?
- Quantas contas correntes possuem a agência X?

A Figura 2.2.2 apresenta o DER construído a partir do enunciado do problema de exemplo apresentado anteriormente, utilizando a notação de Peter Chen [5].

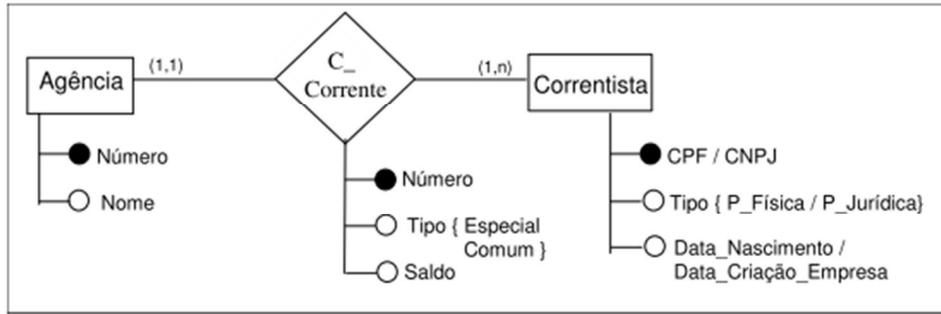


Figura 2.2.2: Diagrama DER desenvolvido a partir do enunciado do problema [4]

A etapa “Reescrita do DER” consiste em transformar um diagrama complexo em um diagrama simplificado, onde as relações associativas são transformadas em entidades; Já as relações com atributos associados são transformadas em entidades com seus respectivos atributos.

A partir do DER desenvolvido (Figura 2.2.2) é realizada a transformação da relação “C_Corrente” em uma entidade, duas novas relações também são criadas e nomeadas pela concatenação do nome das entidades “Agência” e “Correntista” associadas a relação. A Figura 2.2.3 apresenta a transformação da relação “C_Corrente” com atributos associados para uma entidade.

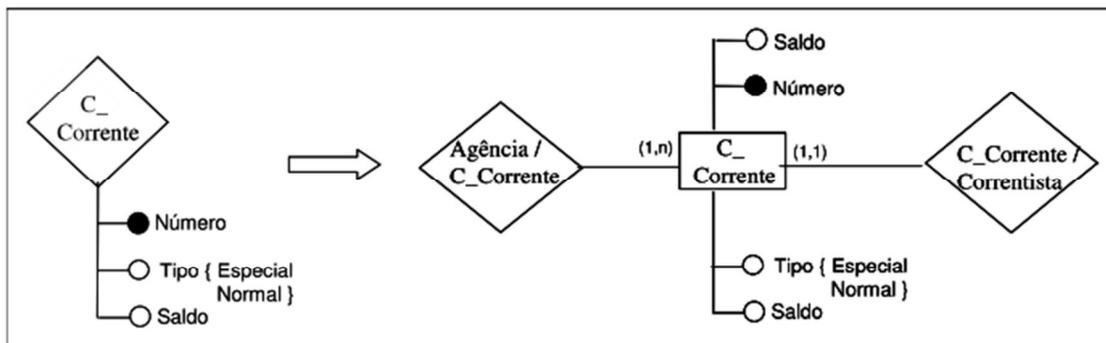


Figura 2.2.3: Transformação da relação “C_Corrente” em uma entidade associada a novas relações [4]

Em seguida, observa-se que os DER das Figuras 2.2.2 e 2.2.3 possuem o atributo “Tipo” presentes nas entidades “Correntista” e “C_Corrente”, demonstrando que há uma relação de especialização. A Figura 2.2.4 apresenta a transformação do atributo “Tipo” em subentidades.

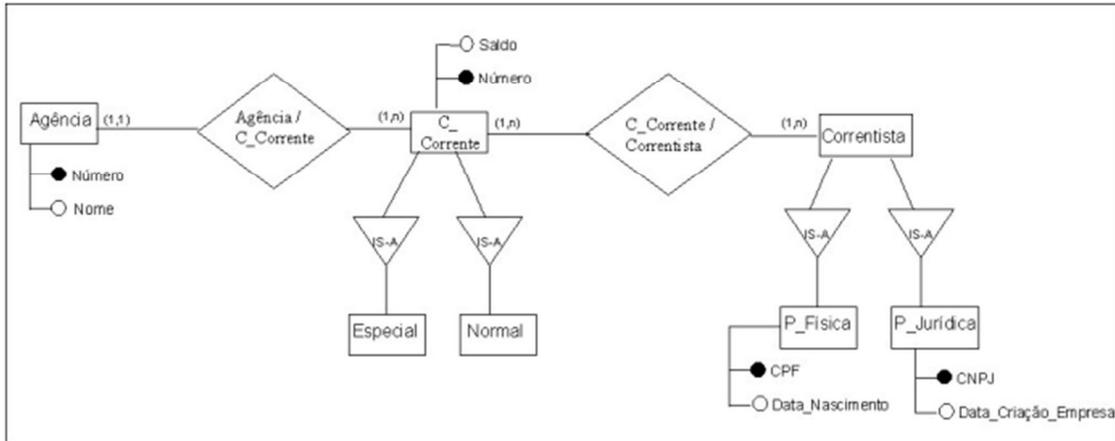


Figura 2.2.4: DER transformado em novas entidades e relações [4]

Inicia-se a construção da ontologia através da etapa “Definição dos conceitos e subconceitos” a qual especifica que as entidades no DER (Figura 2.2.4) são os conceitos da ontologia e as relações “IS-A” são a relações de herança entre as entidades do DER, por fim é realizada a documentação. A Tabela 2.2.1 apresenta as entidades convertidas em conceitos na ontologia, a Tabela 2.2.2 apresenta as relações de herança e a Tabela 2.2.3 apresenta uma parte da documentação desenvolvida.

ENTIDADE	CONCEITO
Agência	AGÊNCIA
C_Corrente	C_CORRENTE
Correntista	CORRENTISTA
Especial	ESPECIAL
Normal	NORMAL
P_Física	P_FÍSICA
P_Jurídica	P_JURÍDICA

Tabela 2.2.1: Entidades do DER transformadas em conceitos da ontologia

ENTIDADE	SUBCONCEITO DE
ESPECIAL	C_CORRENTE
NORMAL	C_CORRENTE
P_FÍSICA	CORRENTISTA
P_JURÍDICA	CORRENTISTA

Tabela 2.2.2: Relações de herança identificadas

Nome do Conceito:	AGÊNCIA
Descrição do Conceito:	Sucursal de uma repartição financeira, banco
Sinônimo:	Banco
Nome do Conceito:	CONTA_CORRENTE
Descrição do Conceito:	É um estilo de conta onde o cliente bancário poderá realizar empréstimos, utilizar limite (caso seja conta especial), pedir cartões de crédito e ter algumas operações que somente o tipo de conta corrente lhe permite fazer.
Sinônimo:	-
Nome do Conceito:	CORRENTISTA
Descrição do Conceito:	É a pessoa física ou jurídica que possui uma conta na agência.
Sinônimo:	Cliente do Banco
Nome do Conceito:	ESPECIAL
Descrição do Conceito:	É um tipo de conta bancária que o correntista não possui limites, portanto são cobrados dela, taxas de manutenção da conta, cartões de crédito, cestas de serviços (se for o caso) e outros serviços vigentes de acordo com cada banco.
Sinônimo:	PREFERENCIAL
Nome do Conceito:	NORMAL
Descrição do Conceito:	É um tipo de conta bancária que o correntista possui limites, portanto são cobrados dela apenas a manutenção da conta bancária, cartões de crédito (se for o caso) e outros serviços vigentes de acordo com cada banco.
Sinônimo:	NORMAL

Tabela 2.2.3: Parte da documentação da ontologia

A etapa “Definição das propriedades” consiste em definir as propriedades da ontologia através dos atributos e relacionamentos presentes no DER (Figura 2.2.4). Dessa forma, todos os atributos contidos no DER são os atributos na ontologia. Além disso, é realizada uma mudança na nomenclatura dos atributos, por exemplo, o atributo “Número” na entidade “Agência”, passa a se chamar “Número_Agência”, isso ocorre devido a entidade “Conta_Corrente” também possuir o atributo “Número”, portanto é necessário diferenciá-los. Outra modificação necessária é em relação às abreviações, por exemplo, o atributo “CPF” deve ser renomeado para sua forma por extenso “Cadastro de Pessoa Física”. A Tabela 2.2.4 apresenta alguns atributos identificados e outras características desses atributos.

NOME DA PROPRIEDADE	TIPO DA PROPRIEDADE	DOMÍNIO	IMAGEM	CARACTERÍSTICA	NOME DA PROPRIEDADE INVERSA
Número_Agência	Datatype	AGÊNCIA	Int	Funcional	-
Número_Conta_Corrente	Datatype	CONTA_CORRENTE	Int	Funcional	-
Cadastro_Pessoa_Fisica	Datatype	PESSOA_FISICA	String	Funcional	-
Cadastro_Nacional_Pessoa_Juridica	Datatype	PESSOA_JURIDICA	String	Funcional	-

Tabela 2.2.4: Atributos da ontologia

A Tabela 2.2.5 apresenta alguns dos relacionamentos da ontologia e o domínio e imagem a que pertencem através das relações contidas no DER (Figura 2.2.4).

NOME DA PROPRIEDADE	TIPO DA PROPRIEDADE	DOMÍNIO	IMAGEM	CARACTERÍSTICA	NOME DA PROPRIEDADE INVERSA
Agencia/ Conta_Corrente	Object	AGÊNCIA	CONTA_CORRENTE	-	-
Conta_Corrente / Correntista	Object	CONTA_ CORRENTE	CORRENTISTA	-	-

Tabela 2.2.5: Relações identificadas da ontologia

A etapa “Determinação dos axiomas de definição dos conceitos” consiste em especificar regras que determinem que para um indivíduo qualquer, seja classificado como indivíduo de um conceito, este deve satisfazer as condições necessárias e suficientes do conceito e para construir essas regras é necessária a intervenção do desenvolvedor. A Figura 2.2.5 apresenta duas regras em Lógica de Descrição que formalizam a seguinte situação: “Se uma conta possuir um saldo superior a R\$ 12.500,00 então ela será considerada uma conta especial”.

$$CONTA_CORRENTE \supseteq ESPECIAL \equiv ESPECIAL.Saldo > 12.500,00$$

$$CONTA_CORRENTE \supseteq NORMAL \equiv NORMAL.Saldo \leq 12.500,00$$

Figura 2.2.5: Axioma em Lógica de Descrição [4]

Finalmente, a etapa “Verificação e modificação” analisa se o mundo real esta representado corretamente com o mundo modelado formalmente. Para fazer isso, o desenvolvedor verifica se a ontologia permite responder às questões iniciais apresentadas na

primeira etapa. É possível ainda realizar verificações formais providas por um motor de inferência, como: consistência dos conceitos, consistência da taxonomia e inferência de indivíduos. Caso o desenvolvedor decida modificar a ontologia, deve fazê-lo no DER voltando a etapa 2. No exemplo apresentado não foram necessárias modificações na ontologia.

Alguns aspectos sobre DERONTO que merecem ser destacados são:

- O nome atribuído às relações não é representativo, portanto, prejudica a semântica. Por exemplo, a relação “Agência/C_Corrente” entre as classes “Agência” e “C_Corrente”. Uma possível solução seria realizar uma busca na web por relações entre os conceitos “Agência” e “C_Corrente”.
- A técnica não trata do povoamento. Uma possível solução seria utilizar um DER vindo de um ambiente de produção, pois o mesmo já teria um conjunto de tuplas (linhas), dessa forma a ontologia já poderia ser povoada.
- A técnica poderia utilizar um banco de dados já povoado e aplicar técnicas de mineração de dados com o intuito de descobrir conhecimentos os quais poderiam ser axiomas na ontologia.

2.3 Técnica 101

A técnica 101 possui o ciclo de vida iterativo e foi desenvolvida a partir da experiência das autoras Natalya F. Noy e a Deborah L. McGuinness na construção de ontologias de domínio [26], ambas as autoras são da universidade de Stanford, sendo esta universidade a desenvolvedora do software Protégè [29], uma ferramenta de código fonte aberto para construção de ontologias. Esta técnica destaca algumas regras que são consideradas dogmas na construção de ontologias:

- Não há uma única forma correta de modelar um domínio. A melhor forma sempre depende do propósito que se tem em mente.
- O desenvolvimento de ontologias é necessariamente um processo iterativo.
- Conceitos da ontologia devem ser próximos a objetos (físicos ou lógicos) e relacionamentos em um domínio de interesse. Estes conceitos são mais prováveis de serem substantivos (objetos) ou verbos (relacionamentos) em sentenças que descrevem o domínio.

A Figura 2.3.1 apresenta a técnica 101 composta de sete etapas: “Determinar o domínio e o escopo”, “Considerar o reuso”, “Enumerar termos importantes”, “Definir classes e hierarquia de classes”, “Definir as propriedades”, “Definir as restrições das propriedades” e “Criar as instâncias”.

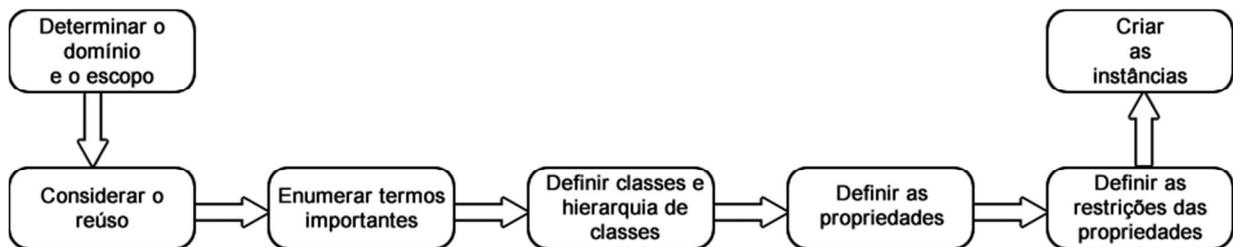


Figura 2.3.1: Ciclo de vida da técnica 101

Para iniciar o desenvolvimento da ontologia, a primeira etapa “Determinar o domínio e o escopo da ontologia” especifica que o desenvolvedor deve responder a algumas perguntas básicas:

- Qual é o domínio que a ontologia irá cobrir?
- Para o que a ontologia será utilizada?
- Quais questões a ontologia deve responder?
- Quem vai utilizar e manter a ontologia?

Uma forma de responder a essas perguntas é criar uma lista de questões que a ontologia deve ser capaz de responder. A ideia da elaboração de questões para as ontologias responderem foi originalmente propostas por Gruninger e Fox [16], nomeada de “Questões de Competência”. Para exemplificar a técnica, as seguintes questões foram elaboradas:

- Quais características de vinho eu devo considerar quando escolher um vinho?
- Bordeaux é um vinho tinto ou branco?
- Cabernet Sauvignon vai bem com frutos do mar?
- Qual a melhor escolha de vinho para carne grelhada?
- Quais as características de um vinho que afetam sua relação com um prato (comida)?
- O sabor ou corpo de um vinho específico muda com o ano da safra?
- Qual foi a melhor safra do Napa Zinfandel?

A partir dessas questões conclui-se que a ontologia ira cobrir o domínio de “vinhos e comidas”; a ontologia será utilizada para as aplicações que desejem “sugerir boas combinações de vinhos e alimentos”; as questões que a ontologia deve responder serão as questões de

competência destacadas anteriormente, por fim, será utilizada por clientes de um restaurante e será mantida pelos proprietários desse restaurante.

A etapa “Considerar o reuso de ontologias” consiste em averiguar se alguém já desenvolveu ontologias a cerca do domínio que se deseja trabalhar, a fim de não se “reinventar a roda”. Existem diversas bibliotecas de ontologias reutilizáveis na Web e um local adequado para encontrar ontologias para se reutilizar é o Swoogle [34], um motor de busca para acesso a ontologias desenvolvido pela Google, e conta com cerca de 10.000 ontologias armazenadas em seu repositório.

Em seguida, a etapa “Enumerar termos importantes” especifica que o desenvolvedor deve criar uma lista de todos os termos possíveis presentes no domínio. É importante ressaltar que no desenvolvimento dessa lista, o desenvolvedor não deve se preocupar com possíveis redundâncias entre os conceitos que eles possam representar, ou que tipo de relações existem entre os termos, ou que características eles possuem ou se estes termos são classes ou propriedades. No domínio de vinhos e comidas os seguintes termos então presentes: uva, vinícola, localização, cor do vinho, corpo do vinho, sabor do vinho, quantidade de açúcar, vinho branco, vinho tinto, vinho Bordeaux, frutos do mar, peixe, carne, vegetais, queijo, fabricante, região de origem, coloração e tipo de comidas.

Nesse momento, é realizada a etapa “Definir classes e hierarquia de classes” que consiste na definição das classes e subclasses da ontologia, e para o seu desenvolvimento os seguintes itens devem ser satisfeitos:

1. Selecionar na lista de termos aqueles que descrevem um conjunto de objetos.
2. Para definir a hierarquia, faz-se a seguinte pergunta: uma instância de uma classe será necessariamente uma instância de outra classe.

Dessa forma, a partir dos termos definidos anteriormente define-se, por exemplo, Vinho Branco, Vinho Rosé e Vinho como classes e para a construção da hierarquia faz-se a seguinte pergunta, uma instância de Vinho Branco é uma instância de Vinho Rosé? Não; Uma instância de Vinho Branco é uma instância de Vinho? Sim. Portanto, essa pergunta é realizada para todas as classes da ontologia com o intuito de construir a hierarquia. A Figura 2.3.2 ilustra uma simples hierarquia desenvolvida.

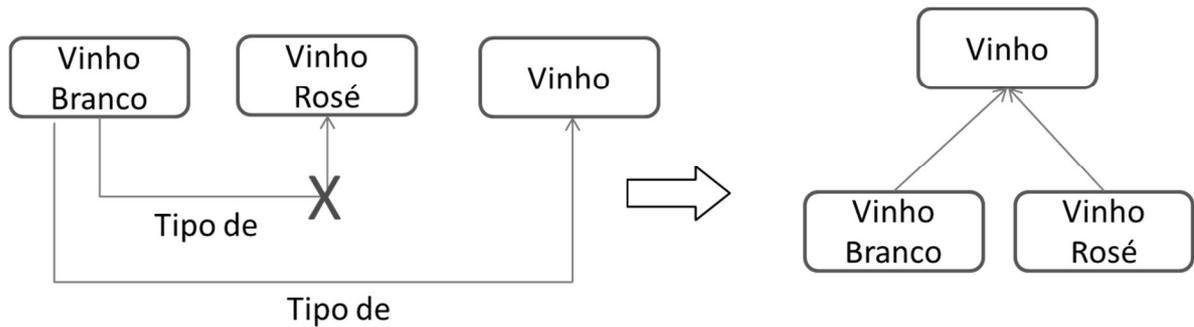


Figura 2.3.2: Verificando a existência de hierarquia

Em seguida, na etapa “Definir as propriedades das classes” são definidos os atributos e relacionamentos a partir da lista de termos definidos na etapa “Enumerar termos importantes”. Por exemplo, os atributos da classe vinho seriam: cor, quantidade de açúcar, preço e idade. Já as relações seriam entre vinho e região, através da relação “feito em”.

Vale ressaltar que é mais apropriado colocar as propriedades identificadas na classe mais genérica que possui essa propriedade, ou seja, utilizando-se da característica de herança. Por exemplo, as classes Vinho Tinto e Vinho Rosé possuem a propriedade Idade, é mais adequado colocar a propriedade Idade na classe Vinho, pois ambas a herdam.

A etapa “Definir as restrições das propriedades” consiste em definir as restrições de cada propriedade, como:

- **Cardinalidade:** define quantos valores uma propriedade pode ter. Por exemplo, um Vinho pode ter no máximo certa quantidade de açúcar.
- **Tipo do valor:** que tipo de valor pode ter uma propriedade. Os tipos mais comuns são: string, integer, float, boolean e tipo instância (permite a definição de relacionamentos entre indivíduos), por exemplo, nome do vinho é do tipo string. Propriedades do tipo instância devem definir uma lista de classes autorizadas, ou seja, a que classes uma instância da propriedade pode pertencer, por exemplo, a relação “feito em”, pertence a classe vinho.
- **Domínio e imagem:** as propriedades do tipo instância precisam ter um domínio e uma imagem, ou seja, o domínio é a classe onde a propriedade se encontra e a imagem é a classe que estabelece a relação. Por exemplo, a relação “feito em” é do domínio (classe) Vinho e tem a imagem(classe) Região.

Por fim, é realizada a etapa “Criar instâncias” que consiste em definir os indivíduos (instâncias) das classes e para desenvolver essa etapa são necessários três passos:

1. Escolher a classe;
2. Criar uma instância desta classe;
3. Preencher os valores das propriedades.

Para exemplificar, podemos criar uma instância da classe Chateau-Morgon-Beaujolais para representar um tipo específico de vinho Beaujolais. Esta instância tem os valores dos slots definidos a seguir: Cor: vermelho, Sabor: delicado, Composto de: Uva Gamay, Feito em: Região Beaujolais.

Alguns aspectos sobre a técnica 101 que merecem ser destacados são:

- É uma técnica bem simples para a construção de ontologias de domínio e muito útil para desenvolvedores que estão iniciando no processo de construção de ontologias.
- Define de forma linguística como encontrar conceitos através dos substantivos e os relacionamentos através de verbos.
- Não define um procedimento de como identificar as instâncias do domínio.

2.4 Técnica de Michael Gruninger e Mark Fox

Desenvolvida a partir da experiência obtida no projeto TOVE em 1995, cujos princípios teóricos e metodológicos encontram-se na Inteligência Artificial [16]. O objetivo do projeto TOVE é criar um modelo de senso comum sobre empresas, isto é, um conhecimento compartilhado sobre o negócio que conduza a deduções de respostas sobre questões acerca do domínio. Para tal, ontologias são criadas no sentido de especificar modelos para organizações públicas e privadas, levando em consideração as seguintes etapas: “Definição dos cenários motivadores”, “Definição informal das questões de competência”, “Especificação da terminologia em Lógica de Primeira Ordem”, “Definição formal das questões de competência”, “Especificação dos axiomas em Lógica de Primeira Ordem” e “Verificação através dos teoremas de completude”, conforme ilustrada na Figura 2.4.1.

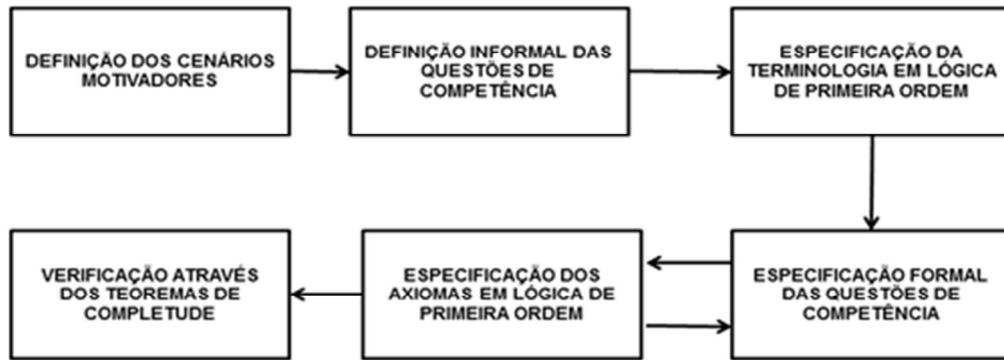


Figura 2.4.1: Ciclo de vida da técnica proposta Michael Gruninger e Mark Fox

A primeira etapa “Definição dos cenários motivadores” consiste em descrever os motivos pela qual a ontologia está sendo construída e definir o cenário na qual a ontologia será desenvolvida. Por exemplo, construir uma ontologia no domínio de viagens para fornecer um modelo de conhecimento consensual desse domínio que poderia ser usado por agências de viagens.

Em seguida, a etapa “Definição informal das questões de competência” define um conjunto de perguntas que necessitam ser respondidas pela ontologia. Essas perguntas são as questões de competência da ontologia e não são expressas em linguagem formal. Por exemplo, no domínio de viagens seriam as seguintes perguntas:

1. Dado as preferências de um viajante (viagens culturais, viagens a montanhas, viagens a praias, etc.) com certa idade e algumas restrições (econômica e odeia viajar de trem) quais os destinos são os mais apropriados?
2. Dado um viajante jovem com certo dinheiro para pagar uma hospedagem, quais hospedagens estão disponíveis?

Uma ontologia não é bem projetada se todas as questões de competência forem consulta simples, ou seja, questões de competências atômicas. Portanto, questões de competência bem formuladas podem ser decompostas. Por exemplo, a questão de competência “1” apresentada anteriormente pode ser decomposta em duas, uma pergunta sobre as restrições econômicas e outra sobre a viagem. A seguir o resultado da decomposição:

- 1.1 Dado as preferências de um viajante (viagens culturais, a montanhas, a praias, etc.) com certa idade e algumas restrições econômicas quais os destinos são os mais apropriados?

1.2 Dados as preferências de uma viajante (viagens culturais, a montanhas, a praias, etc.) com certa idade e algumas restrições sobre a viagem (o viajante odeia viajar de trem) que destinos são os mais apropriados?

Uma vez que foram definidas informalmente as questões de competência, a etapa “Especificação da terminologia em Lógica de Primeira Ordem” extrai a partir dessas questões os conceitos, relações e propriedades da ontologia, sendo então representados em LPO e as respostas a essas questões serão as instâncias da ontologia. Por exemplo, utilizando a questão de competência “1” apresentada anteriormente é possível retirar os seguintes elementos representados em predicados na LPO para compor a ontologia:

- **Classes:** Viajante(X), ViagensCulturais(Y), Montanhas(Z), Praias(W), Trem(T) e Destino(D), Cidade(C);
- **Relação:** preferências(Viajante, Destino), odeiaViajar(Viajante,Trem);
- **Propriedades:** idade(X);
- **Instâncias:** Cidade(“Paris”), Cidade(“Madri”), Cidade(“Londres”) e Cidade(“Nova York”).

Em seguida, as questões de competência definidas informalmente são representadas em regras na LPO através da etapa “Especificação formal das questões de competência”. A Figura 2.4.2 ilustra a questão de competência “2” representada em LPO.

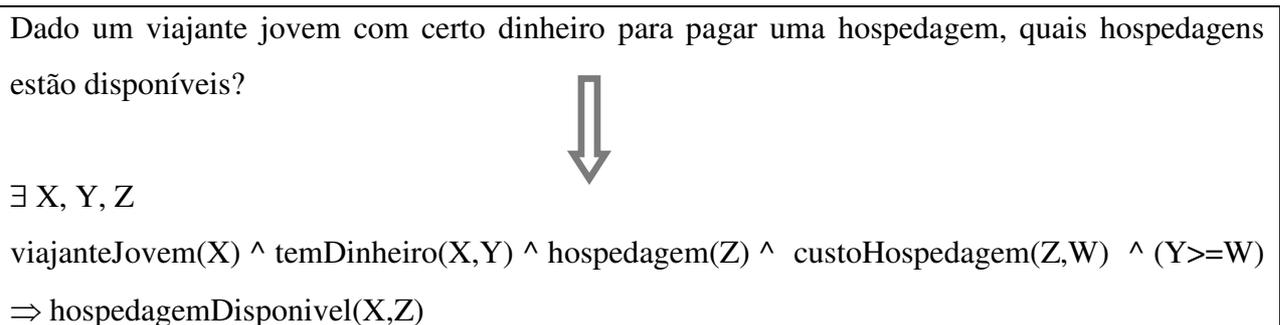


Figura 2.4.2: Questão de competência representada em LPO

A etapa “Especificação dos axiomas em LPO” tem o objetivo de definir através de axiomas como deve ser realizada a interpretação dos conceitos na ontologia, ou seja, atribuir restrições aos conceitos. A Figura 2.4.3, apresenta um exemplo que restringe formalmente a interpretação do conceito “Viajante Jovem”, definindo que ele é um viajante com uma idade inferior a trinta anos.

$$\forall X, Y \text{ viajante}(X) \wedge \text{idade}(X,Y) \wedge (Y < 30) \Rightarrow \text{viajanteJovem}(X)$$

Figura 2.4.3: Restringindo a interpretação do conceito “ViajanteJovem”

Finalmente, a etapa “Verificação através dos teoremas de completude” consiste em definir as condições em que as soluções para as questões de competência sejam consideradas completas. A Figura 2.4.4 apresenta um exemplo de um teorema de completude, que pode ser interpretado da seguinte forma: a ontologia com a união das instâncias satisfaz as questões de competência, se e somente se, a ontologia com a união das instâncias estiver com todos os axiomas. Em outras palavras, este teorema estabelece que a ontologia desenvolvida deve estar completa (ontologia e instâncias) para ser capaz de responder as questões de competências, ou seja, as consultas que a ela serão feitas.

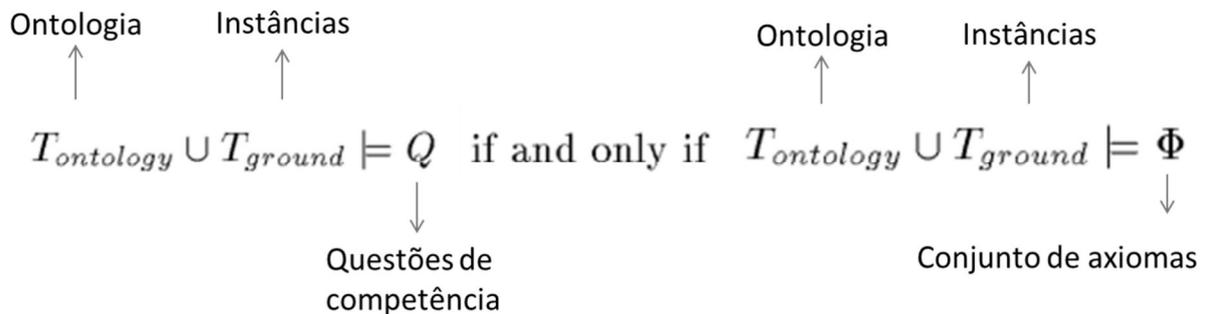


Figura 2.4.4: Exemplo de um Teorema de completude

Para melhor esclarecimento do teorema de completude tem-se o seguinte exemplo: Um dado viajante quer saber o horário do voo do avião “a2” do modelo “boing”. Para responder essa pergunta (questão competência) a ontologia deve estar completa, ou seja, com todos os axiomas e instâncias, se faltar uma regra, a ontologia além de não responder a pergunta, não estará completa. A Figura 2.4.5 apresenta o exemplo proposto.

$$\text{aviao}(X,Z) \wedge \text{horario}(X,W) \Rightarrow \text{vooHorario}(X,Z,W)$$

$$\text{modelo}(X,Z) \Rightarrow \text{aviao}(X,Z)$$

Instâncias:
 modelo(a1,boing).
 modelo(a2,airbus).
 horario(a1,10.30).
 horario(a2,11.30).

Consulta: vooHorario(a1,boing,W).
Resposta: 10:30

Se faltasse essa regra a ontologia não responderia a consulta

Figura 2.4.5: Alguns axiomas, uma consulta e a resposta em uma ontologia

Alguns aspectos sobre a técnica de Michael Gruninger e Mark Fox que merecem ser destacados são:

A partir da etapa que define informalmente as questões de competência, a técnica exige uma codificação em linguagem formal, ou seja, após a aquisição do conhecimento que vai até a segunda etapa, a técnica parte para a geração das regras em LPO. Com todo esse formalismo presente sua grande vantagem é a possível avaliação da ontologia para verificar se a mesma encontra-se com todos os requisitos satisfeitos, evitando assim o sentido dúbio na especificação.

2.5 Técnica de Uschold e King

Proposta em 1995 foi uma das primeiras técnicas para a construção de ontologias e foi desenvolvida a partir da experiência dos autores no projeto Enterprise Ontology, por meio do Instituto de Aplicações em Inteligência Artificial da Universidade de Edinburgh e parceiros como IBM, Unilever e outros [35]. A Figura 2.5.1 apresenta suas cinco etapas: “Identificar o propósito”, “Captura”, “Codifica”, “Integra” e “Avalia”.

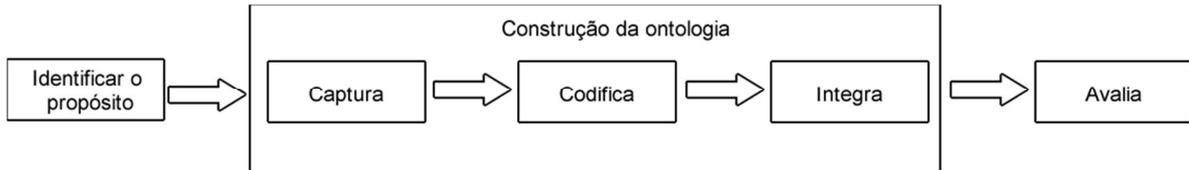


Figura 2.5.1: Ciclo de vida da técnica de Uschold e King

A primeira etapa “Identificar o propósito” consiste em definir os seguintes aspectos sobre a construção da ontologia:

- **Definir o domínio da ontologia.** Por exemplo, o domínio de Viagens;
- **Definição do porque construir a ontologia.** Por exemplo, prover um modelo de conhecimento consensual entre agentes de viagens;
- **Quais os seus usos previstos.** Por exemplo, o compartilhamento de conhecimento;
- **Fazer um levantamento dos termos mais relevantes do domínio.** Por exemplo, locais, tipo de locais, hospedagem, tipo de hospedagens, hotel, pousada, trens, ônibus, ônibus local, metrô, meios de transporte.

A partir dos termos identificados na atividade anterior, a etapa “Captura da

ontologia” define os termos que estão associados a conceitos e a relacionamentos e ainda realiza uma definição textual, precisa e não ambígua dos conceitos e relacionamentos, sendo que estas definições textuais não seguem o estilo clássico de dicionários, sendo construídas criando uma referência a outros termos (destacados em negrito), incluindo noções de classes e relações. A seguir um exemplo:

- Meios de Transporte: é uma classe. Todo meio de transporte tem um **ponto de partida**.
- Ônibus: é uma classe. É um **tipo de meio de transporte**.
- Ônibus local: é uma classe. É um **tipo de ônibus**, que tem **local de partida** e **local de chegada** e as **paradas** na mesma **localização**.

Para desenvolver a hierarquia de classes existem três abordagens “Top-down”, “Bottom-up” e “Middle-out”, descritas a seguir:

- **Top-down:** o processo se inicia pela definição dos conceitos mais genérico do domínio, os quais vão sendo especializados. Por exemplo, defini-se algo bem genérico como “Objeto”. Então se especializa para “Objeto Concreto” e “Objeto Abstrato”, e se especializa mais ainda de “Objeto Concreto” para “Taxi”, “Ônibus” e “Trem”, já “Objeto Abstrato” se especializa para “Transporte por Trem”, “Transporte por Taxi” e “Transporte por Ônibus”. Nessa abordagem é possível se generalizar demais e colocar conceitos desnecessários na ontologia, por exemplo, os conceitos “Transporte por Trem”, “Transporte por Taxi” e “Transporte por Ônibus” apresentados anteriormente não são necessários, pois apenas os conceitos “Taxi”, “Ônibus” e “Trem” são suficientes. A Figura 2.5.2 apresenta a hierarquia desenvolvida com essa abordagem.

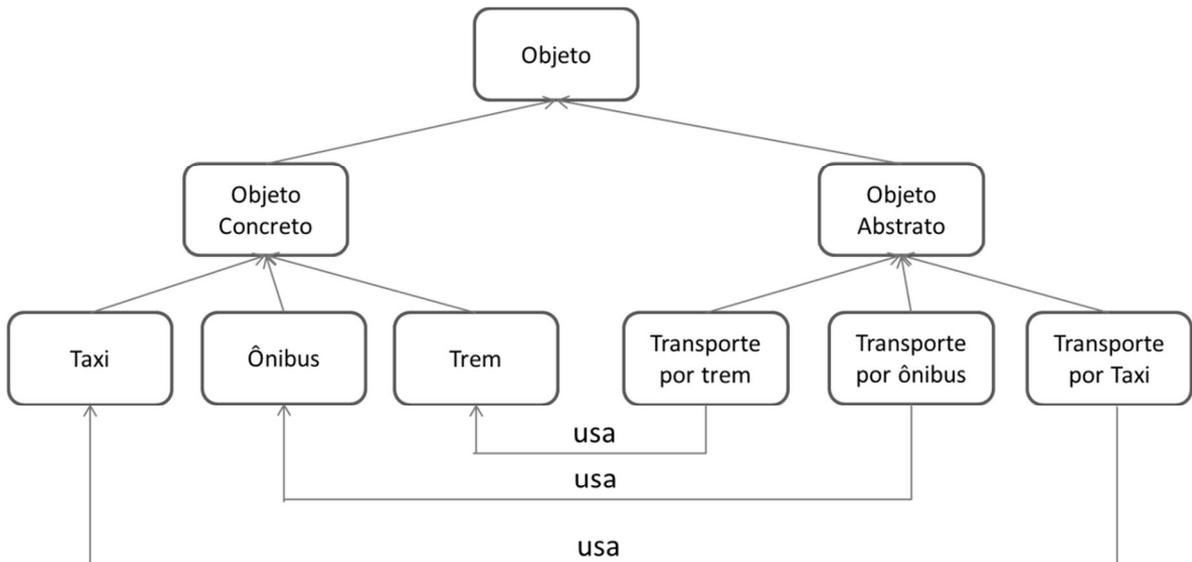


Figura 2.5.2: Hierarquia desenvolvida utilizando a abordagem Top-down

- Bottom-up:** o processo de desenvolvimento começa com a definição dos conceitos mais específicos e depois generaliza-se. Por exemplo, define-se algo bem específico como “Metrô de Londres”, “Ônibus Local de Londres”, “Taxi de Londres”, “Metrô de Madri”, “Ônibus local de Madri” e “Taxi de Madri”, sendo então generalizados para “Metrô”, “Ônibus Local” e “Taxi”, por fim, generaliza-se para “Meios de Transporte”. Assim como a abordagem Top-down é possível generalizar classes desnecessárias, por exemplo, “Meios de transporte de Londres” e “Meios de transporte de Madri”. A Figura 2.5.3 apresenta a hierarquia desenvolvida com essa abordagem.

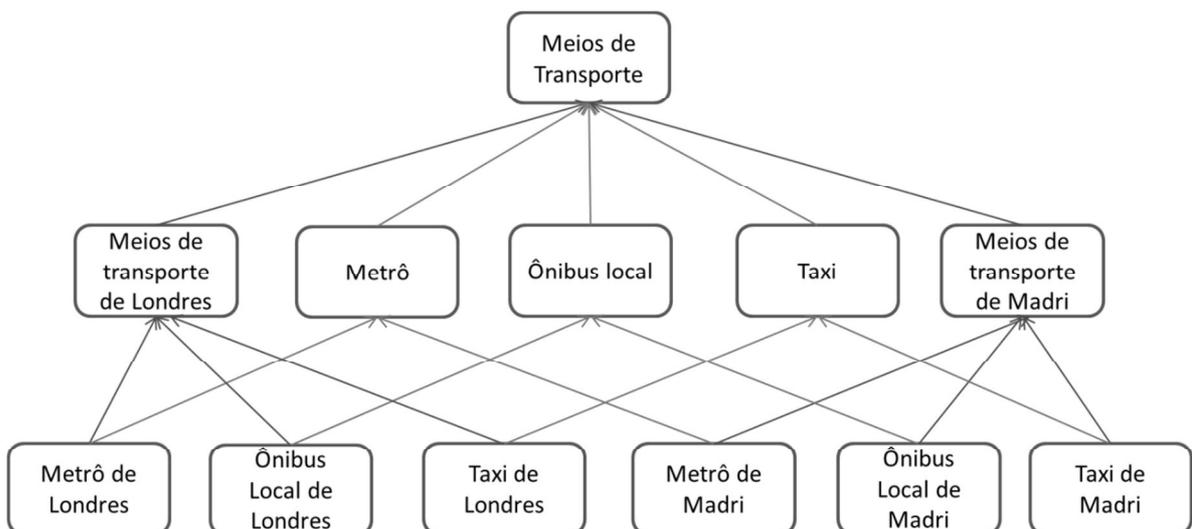


Figura 2.5.3: Hierarquia desenvolvida utilizando a abordagem Bottom-up

- **Middle-out:** é uma combinação das abordagens Top-down e Bottom-up. Definem-se os conceitos mais importantes primeiro e depois estes vão sendo generalizados e especializados. Por exemplo, descobrem-se inicialmente os conceitos “Metrô”, “Ônibus” e “Taxi”, em seguida, generaliza-se para “Meios de Transporte”, por fim, especializa-se “Ônibus” para “Ônibus Local” e “Ônibus de Passeio”. Essa abordagem é considerada a melhor, pois permite um maior controle do nível detalhes da hierarquia. A Figura 2.5.4 apresenta a hierarquia desenvolvida com essa abordagem.

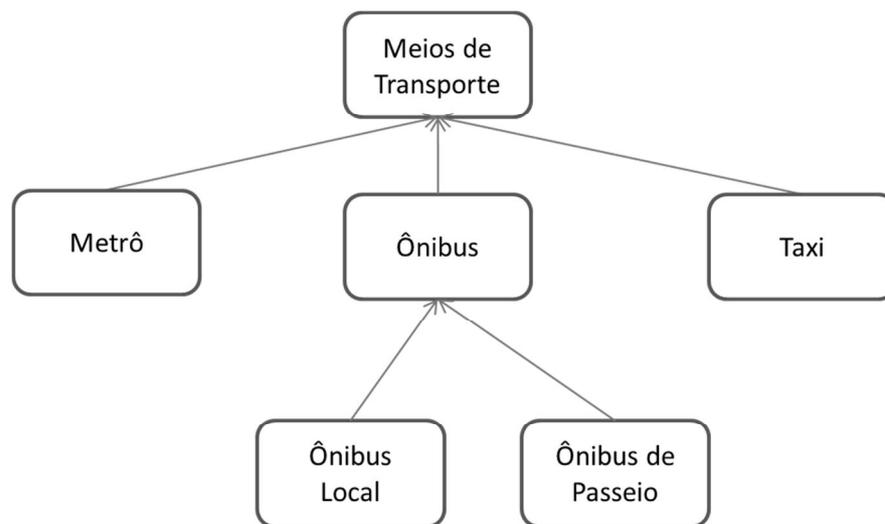


Figura 2.5.4: Hierarquia desenvolvida utilizando a abordagem Middle-out

Os autores não definem como desenvolver a etapa “Codificação”. Entretanto, recomendam ir ao “Plinius Project” onde foram realizadas experiências com varias linguagens como Prolog, L-Lilog, Ontolingua, dentre outras, para definir a melhor linguagem para ser utilizada [32] [33]. Além disso, é definida uma lista de critérios para a escolha da linguagem mais adequada:

- Distância conceitual: é o esforço necessário para passar de um modelo para outro;
- A linguagem esta de acordo com algum padrão, por exemplo, ISO;
- Transportabilidade: ao desenvolver em uma linguagem é fácil migra para outra;
- Existe alguma técnica ou algum guia para a utilização dessa linguagem;
- A base de usuários que usam essa linguagem deve ter um tamanho razoável;

A etapa “Integração” consiste em integrar ontologias existentes como, por exemplo, KIF-Numbers (ontologia sobre números) ou a Standard-Units (ontologia de unidades de medidas (metro, quilômetro, etc.)), podendo ser feita em paralelo com as outras

atividades. Por fim, a etapa de “Avaliação” é realizada através de uma ontologia de referência.

Alguns aspectos sobre a técnica de Uschold e King que merecem ser destacados são:

- Uma grande desvantagem da técnica é que não descreve de uma forma precisa a execução das diferentes etapas do processo de desenvolvimento, só oferecendo princípios gerais muito vagos;
- Não define como identificar propriedades, axiomas e instâncias;
- Define de forma insuficiente a integração de ontologias existentes e a avaliação.

2.6 Methontology

Baseada no padrão IEEE para o desenvolvimento de software, a Methontology foi proposta por Mariano Fernández, Asunción Gómez Pérez e Natalia Juristo no Laboratório de Inteligência Artificial da Universidade Politécnica de Madrid na Espanha em 1997 [11] [13]. Possui um ciclo de vida baseado na evolução de protótipos para o desenvolvimento de ontologias, permitindo adicionar, mudar ou remover termos em cada nova versão, ou seja, novo protótipo da ontologia. A Figura 2.6.1 apresenta as atividades da Methontology.

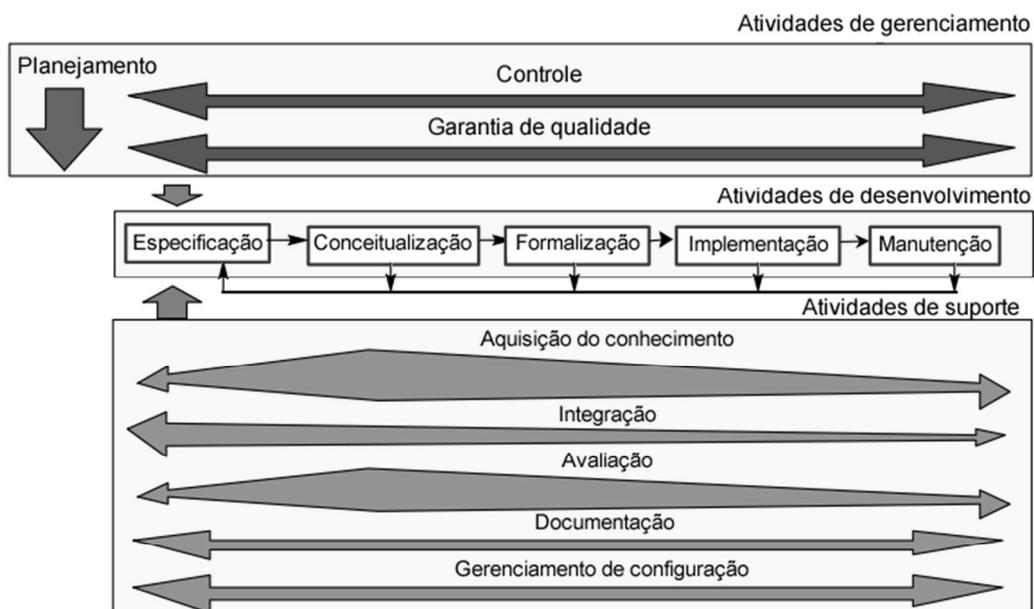


Figura 2.6.1: Ciclo de vida da Methontology [13]

Nas atividades de “Gerenciamento” a primeira atividade a ser desenvolvida é o “Planejamento” que define as principais tarefas, o cronograma, os recursos necessários (pessoas,

software e hardware) e como essas tarefas serão organizadas. A atividade de “Controle” garante a conclusão das tarefas conforme o pretendido, ou seja, verifica se não há desvios indesejáveis no cronograma. Por fim, a atividade de “Garantia de qualidade” verifica a qualidade de cada produto desenvolvido, por exemplo, a ontologia e a documentação.

Dentre as atividades de “Desenvolvimento” a atividade de “Especificação” é responsável por elaborar um documento descrevendo o propósito e o escopo da ontologia, por que está sendo construída, qual será sua utilização e os usuários finais. A atividade de “Conceitualização” é considerada a atividade mais relevante e consiste na estruturação do conhecimento do domínio em um modelo conceitual, ou seja, são descobertos os elementos (classes, relações, axiomas, etc) que compõe a ontologia e são estruturados através de gráficos e tabelas. A atividade de “Formalização” utiliza o modelo conceitual desenvolvido anteriormente e o representa em uma linguagem formal, por exemplo, em Lógica de Primeira Ordem ou Lógica de Descrição. A atividade de “Implementação” utiliza o modelo desenvolvido anteriormente e realiza sua tradução para alguma linguagem de ontologias, por exemplo, OWL (Web Ontology Language) [30]. Por fim, a atividade de “Manutenção” consiste nas alterações quando necessárias, para possíveis melhorias ou correções.

Nas atividades de “Suporte” a atividade de “Aquisição de conhecimento” adquire o conhecimento sobre o domínio da ontologia, através de entrevistas com o especialista do domínio, análise de livros do domínio, brainstorming, etc. A atividade de “Integração” procura integrar o máximo possível de ontologias existentes à nova ontologia, evitando o trabalho repetitivo. A atividade de “Avaliação” realiza revisões no que esta sendo desenvolvido e faz a execução de testes. A atividade de “Documentação” registra os detalhes de cada etapa concluída e o produto gerado. Por fim, a atividade de “Gerenciamento” arquiva as versões da ontologia e da documentação para controlar as mudanças.

Com o intuito de esclarecer a Methontology, a seguir são apresentados exemplos utilizando apenas as atividades de “Especificação” e “Conceitualização” que são consideradas as mais importantes.

Inicialmente o desenvolvedor ao realizar a atividade de “Especificação” deve elaborar um documento descrevendo as seguintes questões:

- **Definir o domínio da ontologia:** por exemplo, viagens.
- **Definir porque está sendo construída:** por exemplo, estabelecer um modelo de conhecimento consensual entre agência de viagens.
- **Quais os seus usos previstos:** por exemplo, compartilhamento de conhecimento.
- **Usuários finais:** por exemplo, viajantes.

Em seguida, a atividade de “Conceitualização” consiste na estruturação do conhecimento do domínio em um modelo conceitual estruturado através de gráficos e tabelas. A Figura 2.6.2 apresenta suas onze subatividades.

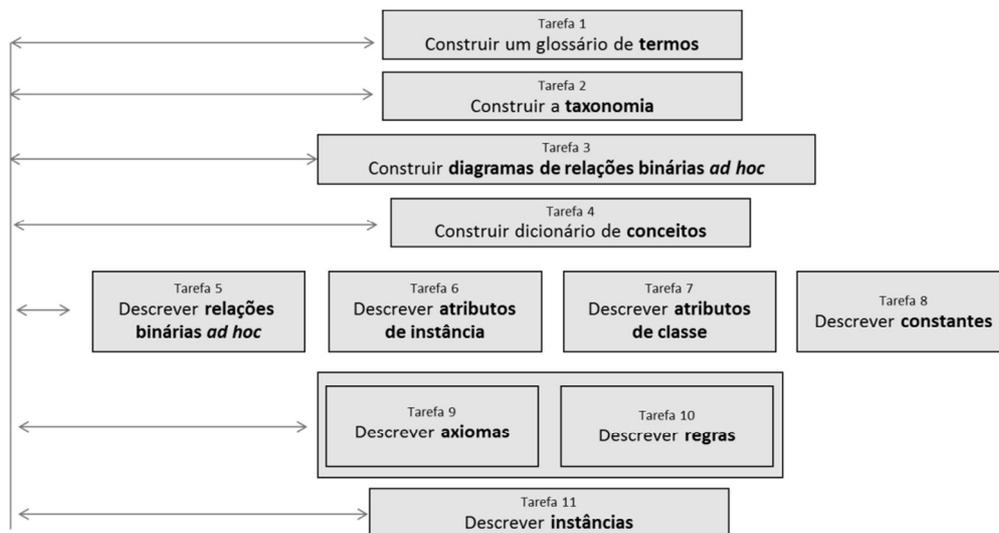


Figura 2.6.2: Subatividades da atividade de conceitualização

A subatividade “Construir glossário de termos” consiste em elaborar de forma clara e simples, um glossário contendo todos os termos relevantes do domínio (conceitos, instâncias, propriedades, entre outros) e suas devidas descrições. A Tabela 2.6.1 apresenta um pequeno glossário de termos contendo seu sinônimo, acrônimo, descrição e tipo.

Nome	Sinônimo	Acrônimo	Descrição	Tipo
América Airlines Flight	--	A A Flight	Voo operado por América Airlines.	Conceito
Número máximo de passageiros no avião	--	--	O número máximo de passageiros num avião, ao mesmo tempo.	Constante
Preço final	--	--	O preço final de um pacote para um passageiro	Atributo de instância
Localização	Lugar	--	Uma posição ou assento disponível para ocupar ou marcado por uma característica	Conceito
Origem	Destino	--	Local onde o voo...	Relação

Tabela 2.6.1: Glossário de termos do domínio de viagens

Nesse momento, é realizada a subatividade “Construir a taxonomia” que tem por objetivo construir a hierarquia de classes, por meio da seguinte pergunta: a instância de uma classe é instância de outra. Define-se ainda se as classes são disjuntas ou possuem decomposição exaustiva, ou seja, quando classes podem ter elementos em comum, ou ainda, se essas classes são do tipo partição, a qual uma superclasse não possui instâncias, apenas suas subclasses. A Figura 2.6.3 ilustra um exemplo.

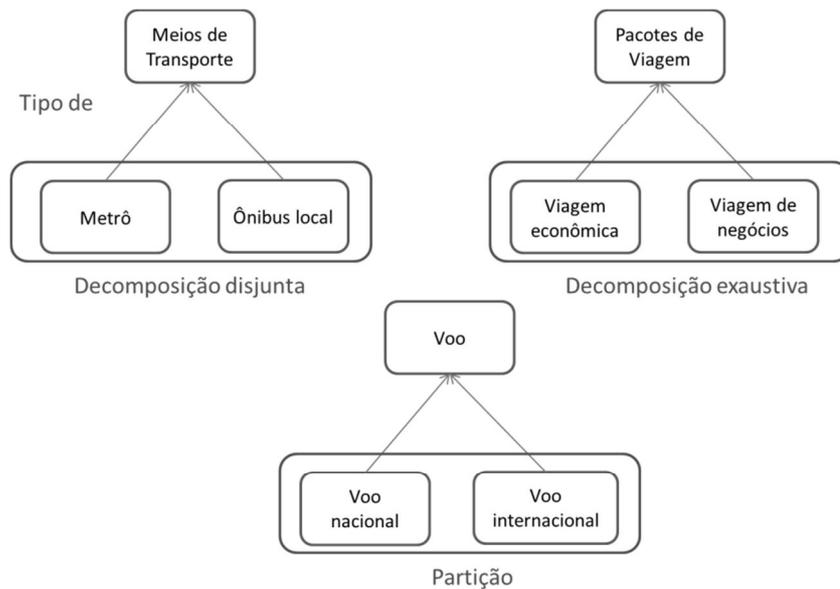


Figura 2.6.3: Taxonomia e suas características no domínio de viagens

Em seguida, é necessário desenvolver a subatividade “Construir diagramas de relações binárias ad hoc” que consiste em definir de forma gráfica, as relações entre as classes e suas relações inversas. A Figura 2.6.4 apresenta um exemplo de um diagrama.

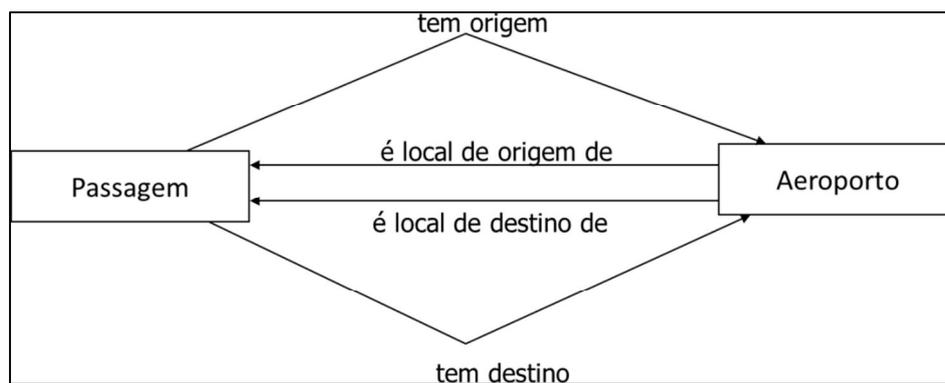


Figura 2.6.4: Diagrama de relações ad hoc no domínio de viagens

A subatividade “Construir dicionário de conceitos” é responsável por construir um dicionário contendo os conceitos, relações, atributos e instâncias. A Tabela 2.6.2 ilustra um exemplo de um dicionário contendo alguns conceitos.

Nome do Conceito	Atributo de Classe	Atributo de Instância	Relações
Aeroporto	--	Nome, tamanho	É local de origem de É local de destino de
America Airlines Flight	Nome da companhia	--	--
Hotel 05 estrelas	Número de estrelas	--	--
AA7462	--	--	O mesmo que voo
Voo	--	--	O mesmo que voo

Tabela 2.6.2: Dicionário de conceitos no domínio de viagens

Em seguida, a subatividade “Descrever relações binárias ad hoc” detalha as relações definidas no diagrama ad hoc desenvolvido na subatividade “Construir diagramas de relações binárias ad hoc”. A Tabela 2.6.3 apresenta como exemplo duas relações sendo descritas.

Nome	Conceito de origem	Cardinalidade origem (máximo)	Conceito destino	Propriedades matemáticas	Relação inversa
Mesmo que voo	Voo	N	Voo	Transitividade simétrica	--
Tem origem	Passagem	1	Aeroporto	--	É local de, Origem de

Tabela 2.6.3: Relações binárias ad hoc descritas

A subatividade “Descrever atributos de instância” é responsável por descrever os atributos de instância, o conceito a que pertence, o seu tipo de valor, a unidade de medida e a cardinalidade. A Tabela 2.6.4 ilustra um exemplo descrevendo alguns atributos de instância.

Nome do atributo	Nome do conceito	Tipo do valor	Unidade de medida	Cardinalidade
Data de chegada	Passagem	Date	--	(0,1)
Nome da Companhia	Passagem	String	--	(0,N)
Data de saída	Passagem	Date	--	(0,1)
Preço ida	Passagem	Float	Moeda	(0,1)
Preço ida-volta	Passagem	Float	Moeda	(0,1)

Tabela 2.6.4: Alguns atributos de instância descritos

Nesse momento, é desenvolvida a subatividade “Descrever atributos de classe” que consiste em descrever os atributos das classes que não são herdados pelas subclasses. A Tabela 2.6.5 apresenta um exemplo contendo o atributo, o conceito a que pertence, o seu tipo de valor, a unidade de medida e a cardinalidade.

Nome do atributo	Conceito	Tipo de valor	Unidade de Medida	Cardinalidade
Nome da companhia	American Airlines Flight	String	--	(1,1)
Nome da companhia	Iberia Flight	String	--	(1,1)
Número de estrelas	Hotel 05 estrelas	Integer	Estrela	(1,1)

Tabela 2.6.5: Descrevendo alguns atributos de classe

O desenvolvedor por meio da subatividade “Descrever constantes” deve definir os valores que nunca mudam, ou seja, são sempre constantes no domínio. A Tabela 2.6.6 apresenta um exemplo contendo uma constante, o seu tipo de valor, o valor e a sua unidade de medida.

Nome	Tipo valor	Valor	Unidade de medida
Número máximo de passageiros no avião	Integer	200	Pessoas

Tabela 2.6.6: Uma constante no domínio de viagens

A subatividade “Descrever axiomas” é responsável por definir como deve ser a interpretação dos conceitos da ontologia, ou seja, define as restrições necessárias para que um indivíduo pertença a um conceito. A Tabela 2.6.7 ilustra um axioma, contendo sua descrição, sua representação formal, os conceitos, atributos, relações e variáveis associadas.

Nome axioma	Trem dentro da Europa
Descrição	Todo trem que sai de um local na Europa deve chegar a outro local na Europa
Expressão	Forall(?X,?Y,?Z) ([Train Travel] (?X) and [departure Place] (?X,?Y) and [arrival Place] (?X, ?Z) and [European Location] (?Y) \Rightarrow [European Location] (?Z))
Conceitos	Train Travel European Location
Atributos referidos	--
Relações Binárias	Departure Place Arrival Place
Variáveis	?X, ?Y, ?Z

Tabela 2.6.7: Um axioma no domínio de viagens

Em seguida, a subatividade “Descrever regras” consiste em descrever as regras responsáveis por inferir novos conhecimentos, ou seja, regras para serem submetidas a um mecanismo de inferência para gerarem novas informações ou conhecimentos. A Tabela 2.6.8 apresenta um exemplo de uma regra.

Nome regra	Costa Cruises
Descrição	Todo navio que parte da Europa é disponibilizado para a companhia Costa Cruises
Expressão	If [European Location] (?Y) and Ship(?X) and [departure Place] (?X, ?Y) Then [company Name] (?X, “Costa Cruises”)
Conceitos	Ship, European Location
Atributos referidos	Company Name
Relações Binárias	Departure Place
Variáveis	?X, ?Y

Tabela 2.6.8: Uma regra no domínio de viagens

Por fim, a subatividade “Descrever instâncias” consiste em definir os indivíduos, ou seja, os seres únicos do domínio. A Tabela 2.6.9 apresenta um exemplo contendo duas instâncias, os atributos e os seus valores.

Nome da Instância	Nome do Conceito	Atributos	Valores
AA7462_Feb08_2002	AA7462	Nome companhia	American Airlines
		Data saída	02/08/2002
		Data chegada	02/08/2002
		Preço ida	300
AA7462_Feb16_2002	AA7462	Nome companhia	American Airlines
		Data saída	16/08/2002
		Data chegada	16/08/2002
		Preço ida	300

Tabela 2.6.9: Algumas instâncias no domínio de viagens

Alguns aspectos sobre a técnica Methontology que merecem ser destacados são:

Dentre as técnicas apresentadas a Methontology demonstrou ser mais apropriada a auxiliar o processo de desenvolvimento de ontologias, pois facilita a integração dos profissionais pertencentes à equipe, definindo e alocando as atividades em três grupos para construção da ontologia, tais como: atividades de gerenciamento de ontologias, atividades ligadas ao desenvolvimento de ontologias e atividades de suporte.

2.7 Análise Comparativa

A Tabela 2.7.1 apresenta o resultado de uma análise comparativa entre as mais representativas técnicas para a construção de ontologias de acordo com os seguintes critérios: o tipo da ontologia desenvolvida, a ordem na qual os elementos da ontologia são descobertos, o tipo do ciclo de vida (clássico ou iterativo), o reuso de ontologias existentes e a utilização de conhecimento linguístico para encontrar classes e relacionamentos.

As ontologias podem ser quatro tipos: alto-nível, domínio, tarefa e aplicação [17]. A técnica escolhida deve ser apropriada pra a construção de um tipo particular de ontologia. Dentre as técnicas do estado da arte analisadas neste trabalho somente DERONTO afirma desenvolver ontologias de aplicação. Entretanto, DERONTO não apresenta um procedimento para a construção dos axiomas, sendo este, um elemento fundamental que compõem uma ontologia de aplicação.

Uma ontologia de aplicação é composta de seis elementos: classes, taxonomia, relacionamentos, propriedades, axiomas e instâncias [12]. A ordem na qual esses elementos são descobertos influencia o desenvolvimento das subatividades da técnica. Por exemplo, todas as

técnicas analisadas iniciam com a identificação de termos relevantes do domínio que correspondem as classes da ontologia.

O processo de desenvolvimento de ontologias deve ser preferivelmente incremental, permitindo ao engenheiro do conhecimento adicionar novos elementos para a ontologia a cada iteração do processo. DERONTO, Técnica 101 e Methontology suportam um processo de desenvolvimento iterativo. Já as técnicas proposta por Gruninger e Fox e Uschold e King possuem um ciclo de vida clássico, o qual não permite alterações na ontologia ao término do seu desenvolvimento.

Boas técnicas devem considerar o reuso de ontologias existentes como, ontologias de domínio, tarefa e aplicação. Das técnicas analisadas, apenas Methontology e 101 abordam o reuso, sendo que ainda informam boas fontes de ontologias reusáveis; já DERONTO, Gruninger e Fox e Uschold e King não abordam o reuso, sendo que está última, menciona a necessidade do reuso, entretanto, não aborda em maiores detalhes como o fazer.

Somente a Técnica 101 define regras para encontrar classes e relacionamentos através de substantivos e verbos respectivamente. Esta é uma vantagem por permitir a fácil identificação dos elementos da ontologia através de estratégias linguísticas. Outra vantagem do uso de conhecimento linguístico é a possível automação da extração destes elementos usando técnicas de processamento da linguagem natural [9] [10].

Critério comparativo	DERONTO [4]	Uschold e King [35]	Técnica 101 [26]	Methontology[11]	Gruninger e Fox [16]
O tipo de ontologia desenvolvida	Ontologias de domínio e aplicação	Ontologias de domínio	Ontologias de domínio	Ontologias de domínio	Ontologias de domínio
A ordem na qual os elementos da ontologia são descobertos	Classes Taxonomia Propriedades Relações Axiomas	Classes Taxonomia Relações	Classes Taxonomia Propriedades Relações Axiomas Instâncias	Classes Taxonomia Relações Propriedades Axiomas Instâncias	Classes Relações Propriedades Axiomas Instâncias Taxonomia
O tipo do ciclo de vida	Iterativo	Clássico	Iterativo	Iterativo	Clássico
O reuso de ontologias existentes	Não	Sim	Sim	Sim	Não
Usa conhecimento linguístico	Não	Não	Sim	Não	Não

Tabela 2.7.1: Análise comparativa das técnicas para a construção de ontologias.

2.8 Considerações Finais

Neste capítulo foram apresentadas as principais técnicas para o desenvolvimento de ontologias com a finalidade de obter embasamento teórico e referencial bibliográfico para a elaboração da técnica e ferramenta propostas.

Inicialmente explicou-se a necessidade de se utilizar técnicas para guiar o processo de desenvolvimento de ontologias e os problemas que acarretam na sua não utilização.

Em seguida, tendo realizado a análise, constatou-se a necessidade de técnicas que detalhem a construção dos axiomas, sendo este, um elemento fundamental que compõem uma ontologia de aplicação, a qual é alvo deste trabalho.

Finalmente, realizou-se uma análise comparativa entre as técnicas analisadas destacando cinco critérios, os quais serão úteis para a delimitação da técnica e ferramenta propostas no capítulo 3.

3 A TÉCNICA GAODT

3.1 Introdução

As ontologias de aplicação são utilizadas pelos modernos SBC no processo de tomada de decisões, isso é possível, pois o principal componente de uma ontologia de aplicação são os axiomas. Para construir uma ontologia de aplicação é necessário o claro entendimento dos requisitos do SBC, ou seja, os objetivos que ele pretende alcançar. Esses objetivos representam as regras do domínio e correspondem aos axiomas em uma ontologia de aplicação. Prover suporte sistematizado a formalização desses objetivos em uma ontologia de aplicação é a ideia fundamental da GAODT (“Goal-Oriented Application Ontology Development Technique”).

A Figura 3.1.1 ilustra a visão geral da técnica GAODT juntamente com suas quatro atividades: “Seleção dos Objetivos e Fatos”, “Representação dos Predicados em LPO”, “Especificação dos Axiomas em LPO” e “Especificação/Extensão da Ontologia de Aplicação”.

O desenvolvedor da ontologia de aplicação e o especialista do domínio participam na execução destas atividades. O desenvolvedor é o engenheiro do conhecimento responsável pela construção da ontologia de aplicação. O especialista do domínio é alguém que detém informações específicas daquela área de conhecimento.

A técnica tem como entrada uma lista com todos os objetivos e fatos do sistema fornecidos pelo especialista do domínio. Os objetivos são os requisitos que o SBC tem que alcançar e os fatos são afirmações que complementam esses objetivos. Por exemplo, o objetivo “Determinar o genitor de uma pessoa” para ser alcançado necessita dos fatos “Uma pessoa tem pai” e “Uma pessoa tem mãe”.

Na atividade “Seleção dos Objetivos e Fatos”, o desenvolvedor em consenso com o especialista de domínio, selecionam os mais representativos objetivos e fatos para serem usados como entrada para a próxima atividade.

Na atividade “Representação dos Predicados em LPO” o desenvolvedor traduz os objetivos e fatos em linguagem natural para predicados em LPO.

A atividade “Especificação dos Axiomas em LPO” recebe como entrada os predicados especificados na atividade anterior e especifica em LPO as regras necessárias para alcançar os objetivos do sistema. Esta atividade é iterativa, isto é, um predicado objetivo pode necessitar de outros subobjetivos ou fatos para satisfazê-lo. Por exemplo, para satisfazer o objetivo “Determinar os ascendentes de uma pessoa” o subobjetivo “Determinar o genitor de uma pessoa” deve ser alcançado. Todo o processo é executado iterativamente até que todos os

objetivos tenham sido satisfeitos.

Finalmente, a atividade “Especificação/Extensão da Ontologia de Aplicação” utiliza os axiomas gerados na atividade anterior e extrai a partir deles os elementos necessários para compor a ontologia de aplicação. A ontologia de aplicação criada pode ser ainda estendida através da realização de uma busca semântica em um repositório de ontologias de aplicação.

A execução da técnica GAODT será ilustrada com exemplos no domínio da Sucessão Legítima, o qual será melhor explicado no estudo caso no capítulo quatro. Nas seções 3.2 a 3.5 as atividades da GAODT são descritas detalhadamente.

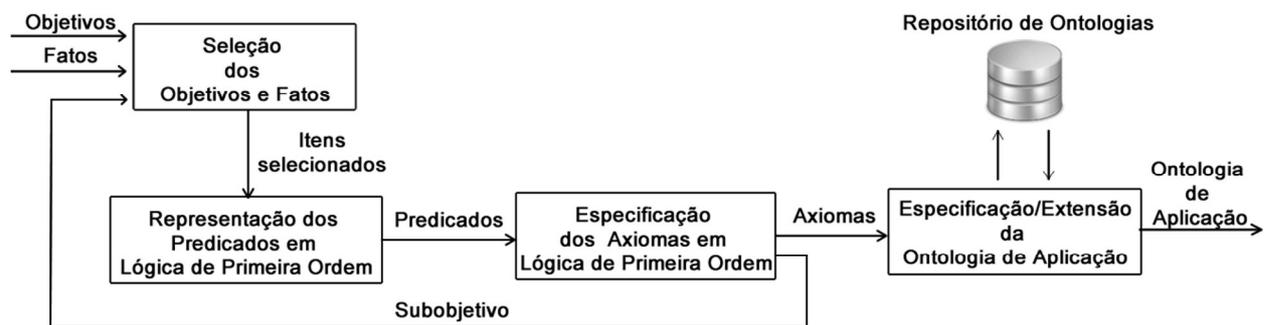


Figura 3.1.1: A visão geral da técnica GAODT.

3.2 Seleção dos Objetivos e Fatos

Esta atividade recebe como entrada uma lista de objetivos e fatos do sistema providos pelo especialista do domínio. A partir desta lista, o desenvolvedor e o especialista definem qual deles serão dados como entrada para a próxima atividade. Como um exemplo, a Tabela 3.2.1 apresenta uma lista de objetivos e fatos de um SBC de apoio as decisões no domínio da Sucessão Legítima.

Inicialmente deve ser definido o objetivo geral e os principais objetivos específicos do sistema. Por exemplo, para o objetivo geral 1: “Calcular a herança de uma pessoa”, os principais objetivos específicos 2: “Identificar os herdeiros de uma pessoa” e 3: “Determinar a herança dos herdeiros”. Para satisfazer esses subobjetivos, outros objetivos ou fatos poderão ser definidos em iterações subsequentes em um processo executado recursivamente para todos os objetivos da lista.

1	Calcular a herança de uma pessoa	Objetivo
2	Identificar os herdeiros de uma pessoa	Objetivo
3	Determinar a herança dos herdeiros	Objetivo
4	Identificar o cônjuge de uma pessoa	Objetivo
5	Definir o quinhão dos descendentes no regime universal	Objetivo
6	Identificar os ascendentes de uma pessoa	Objetivo
7	Identificar os descendentes de uma pessoa	Objetivo
8	Determinar a meação do cônjuge no regime universal	Objetivo
9	Identificar os colaterais de uma pessoa	Objetivo
10	Determinar a quantidade de descendentes de uma pessoa	Objetivo
11	Determinar a quantidade de ascendentes de uma pessoa	Objetivo
12	Determinar a meação dos descendentes no regime universal	Objetivo
13	Dividir a meação pela quantidade de descendentes	Objetivo
14	Uma pessoa tem descendentes	Fato
15	Uma pessoa tem cônjuge	Fato
16	Uma pessoa tem ascendentes	Fato
17	Uma pessoa tem colaterais	Fato
18	Uma pessoa tem bens comuns	Fato
19	Uma pessoa tem bens particulares	Fato
20	Uma pessoa tem regime universal	Fato
21	Os bens comuns tem valor	Fato
22	Os bens particulares tem valor	Fato
23	Dividir a herança para o cônjuge em concorrência com os descendentes no regime universal	Objetivo
24	Partilhar os bens de uma pessoa para os descendentes e o cônjuge no regime universal	Objetivo
25	Somar os bens comuns de uma pessoa	Objetivo
26	Somar os bens particulares de uma pessoa	Objetivo
27	Somar os bens de uma pessoa	Objetivo
28	Verificar a existência de ascendentes de uma pessoa	Objetivo
29	Verificar a existência do cônjuge de uma pessoa	Objetivo
30	Verificar a existência de descendentes de uma pessoa	Objetivo
31	Verificar a existência do regime universal de uma pessoa	Objetivo

Tabela 3.2.1: Lista de objetivos e fatos do sistema

3.3 Representação dos Predicados em LPO

Esta atividade consiste na tradução dos objetivos e fatos selecionados anteriormente, expressos em linguagem natural para predicados em LPO. A Figura 3.3.1 ilustra esta atividade composta de sete subatividades: “Identificação das entidades”, “Redefinição das entidades”, “Identificação dos relacionamentos”, “Redefinição dos relacionamentos”, “Definição da aridade”, “Definição dos predicados” e “Redefinição das entidades dos predicados”.

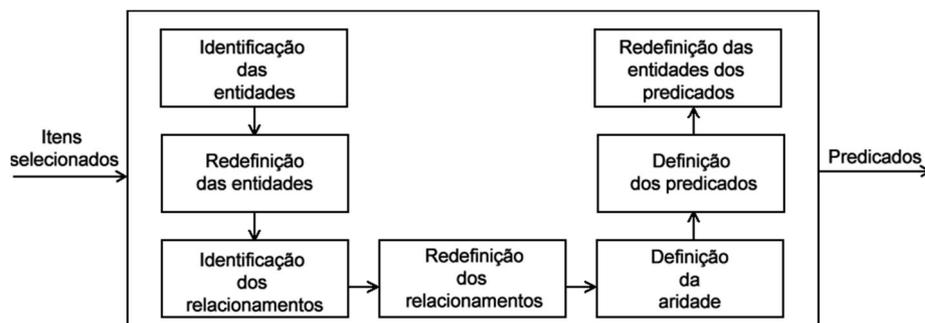


Figura 3.3.1: Atividade “Representação dos Predicados em LPO”

Na subatividade “Identificação das entidades” todos os sujeitos e objetos implícitos ou explícitos são identificados a partir dos itens selecionados na atividade anterior. O resultado desta subatividade é apresentado na Tabela 3.3.1.

Itens Selecionados	Entidades
Calcular a herança de uma pessoa	Herança, Pessoa
Identificar os herdeiros de uma pessoa	Herdeiros, Pessoa
Determinar a herança dos herdeiros	Herança, Herdeiros

Tabela 3.3.1: Entradas e saídas da subatividade “Identificação das entidades”

A subatividade “Redefinição das entidades” leva em consideração as entidades identificadas na subatividade anterior, e para cada uma é verificado se esta é uma entidade ou um relacionamento. Por exemplo, a entidade “Herdeiros” presente na Tabela 3.3.1 é atualmente um relacionamento entre duas “Pessoas”, ou seja, “Uma pessoa é herdeira de outra pessoa”. Portanto, “Herdeiros” é redefinido como “Pessoa”, considerando as entidades que integram o relacionamento. Entretanto, a palavra “Herdeiros” não é descartada, pois será utilizada na subatividade “Redefinição dos relacionamentos”. A Tabela 3.3.2 apresenta os resultados desta subatividade.

Entidades	Redefinição das Entidades
Herança, Pessoa	Herança, Pessoa
Herdeiros, Pessoa	Pessoa, Pessoa
Herança, Herdeiros	Herança, Pessoa

Tabela 3.3.2: Entradas e saídas da subatividade “Redefinição das entidades”

A subatividade “Identificação dos relacionamentos” utiliza os itens selecionados na atividade “Seleção dos Objetivos e Fatos”, para identificar os verbos presentes nas frases na qual representam os relacionamentos a serem extraídos. Por exemplo, no item selecionado “Calcular a herança de uma pessoa”, o verbo “Calcular” é identificado como um relacionamento. A Tabela 3.3.3 apresenta os relacionamentos identificados.

Itens selecionados	Relacionamentos
Calcular a herança de uma pessoa	Calcular
Identificar os herdeiros de uma pessoa	Identificar
Determinar a herança dos herdeiros	Determinar

Tabela 3.3.3: Entradas e saídas da subatividade “Identificação dos relacionamentos”

A subatividade “Redefinição dos relacionamentos” leva em consideração os relacionamentos identificados na atividade anterior (exemplificado na Tabela 3.3.3) e verifica se estes relacionamentos são verbos transitivos, pois eles necessitam de um complemento para terem sentido. Por exemplo, o relacionamento “identificar” precisa de um complemento para ter sentido, logo, utiliza-se sua respectiva entidade na Tabela 3.3.2 ou as palavras que foram

consideradas entidades na primeira subatividade, por exemplo, a palavra “Herdeiros”. A Tabela 3.3.4 apresenta o resultado desta subatividade utilizando os exemplos na Tabela 3.3.2 e 3.3.3.

Relacionamentos	Entidades/Palavras redefinidas	Redefinição dos relacionamentos
Calcular	Herança	calcularHerança
Identificar	Herdeiros	identificarHerdeiros
Determinar	Herança	determinarHerança

Tabela 3.3.4: Entradas e saídas da subatividade “Redefinição dos relacionamentos”

A subatividade “Definição da aridade” define o número de entidades envolvidas nos relacionamentos identificados anteriormente. Esta quantidade é determinada de acordo com o número de entidades identificadas em cada item selecionado. A Tabela 3.3.5 apresenta a aridade identificada para as entidades na Tabela 3.3.2.

Relacionamentos	Entidades	Aridade
calcularHerança	Herança, Pessoa	2
identificarHerdeiros	Pessoa, Pessoa	2
determinarHerança	Herança, Pessoa	2

Tabela 3.3.5: Entradas e saídas da subatividade “Definição da aridade”

Na subatividade “Definição dos predicados” as entidades e relacionamentos identificados (por exemplo, Tabela 3.3.2 e 3.3.4) são representados em LPO. A Tabela 3.3.6 apresenta os predicados resultantes da realização desta subatividade.

Itens selecionados	Predicados
Calcular a herança de uma pessoa	calcularHerança(Herança, Pessoa)
Identificar os herdeiros de uma pessoa	identificarHerdeiros(Pessoa, Pessoa)
Determinar a herança dos herdeiros	determinarHerança(Herança, Pessoa)

Tabela 3.3.6: Itens selecionados traduzidos para predicados em LPO

A subatividade “Redefinição das entidades dos predicados” tem o objetivo de renomear os argumentos dos predicados definidos na subatividade “Definição dos predicados”, quando esses argumentos possuírem o mesmo nome. Por exemplo, as entidades presentes no predicado “identificarHerdeiros(Pessoa, Pessoa)” são consideradas variáveis diferentes uma vez que elas representam pessoas distintas. Assim, o predicado é redefinido para “identificarHerdeiros(PessoaX, PessoaY)” sendo essa mudança propagada para todos os outros predicados na Tabela 3.3.6. A Tabela 3.3.7 apresenta os resultados desta subatividade e o produto final desta atividade.

Predicados	Predicados redefinidos
calcularHerança(Herança, Pessoa)	calcularHerança(Herança, PessoaX)
identificarHerdeiros(Pessoa, Pessoa)	identificarHerdeiros(PessoaX, PessoaY)
determinarHerança(Herança, Pessoa)	determinarHerança(Herança, PessoaY)

Tabela 3.3.7: Entradas e saídas da subatividade “Redefinição das entidades dos predicados”

3.4 Especificação dos Axiomas em LPO

O propósito desta atividade é especificar as regras que permitem alcançar os objetivos do sistema, na qual são representados como predicados em LPO. O processo é iterativo, por que há uma iteração com a atividade “Seleção dos Objetivos e Fatos”. Para cada objetivo contido em uma regra uma busca é realizada na lista de objetivos e fatos para recuperar os subobjetivos ou fatos que o satisfazem.

Esta atividade consiste de quatro subatividades: “Definição da condição e conclusão”, “Definição dos operadores booleanos”, “Definição dos quantificadores” e “Definição de implicação ou equivalência”. A Figura 3.4.1 apresentam as subatividades desta atividade.

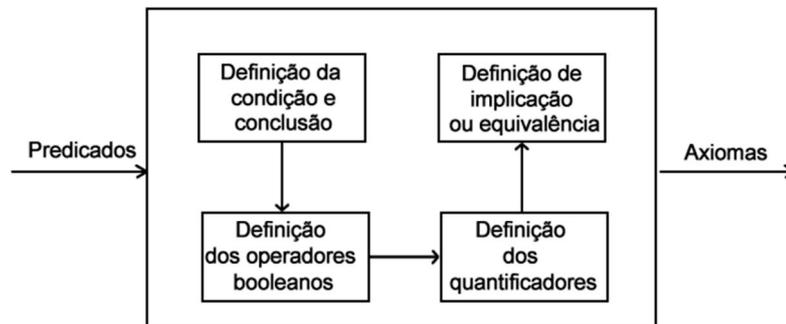


Figura 3.4.1: Atividade “Especificação dos Axiomas em LPO”

A subatividade “Definição da condição e conclusão” determina os predicados que compõe a condição e a conclusão da regra. A conclusão é o objetivo principal que deve ser alcançado e a condição pode ser considerada como um conjunto de premissas ou subobjetivos que permitem alcançar o objetivo principal. Esta subatividade recebe como entrada os predicados identificados na atividade anterior (por exemplo, Tabela 3.3.7). A Tabela 3.4.1 apresenta o resultado desta subatividade.

Condição e predicados do axioma	Conclusão
identificarHerdeiros(PessoaX, PessoaY)	calcularHerança(Herança, PessoaX)
determinarHerança(Herança, PessoaY)	

Tabela 3.4.1: Saída da subatividade “Definição da condição e conclusão”

A subatividade “Definição dos operadores booleanos” especifica os operadores booleanos na qual integram os predicados da condição do axioma. Os operadores utilizados são a

conjunção representada pelo símbolo (\wedge) e a disjunção representada pelo símbolo (\vee).

Os predicados na condição são unidos pela conjunção quando todos eles são necessários para alcançar a conclusão; já o operador de disjunção quando eles são predicados alternativos para alcançar a conclusão. Por exemplo, para alcançar o objetivo “Calcular a herança de uma pessoa” ($\text{calcularHeranca}(\text{Heranca}, \text{PessoaX})$), é necessário satisfazer todos os objetivos “Identificar os herdeiros de uma pessoa” ($\text{identificarHerdeiros}(\text{PessoaX}, \text{PessoaY})$) e “Determinar a herança dos herdeiros” ($\text{determinarHeranca}(\text{Heranca}, \text{PessoaY})$). Portanto, a conjunção é utilizada para integrar estes dois predicados.

A subatividade “Definição dos quantificadores” define os quantificadores apropriados para associar as entidades presentes no axioma. Os quantificadores podem ser o universal (\forall) ou existencial (\exists). O primeiro é utilizado para indicar que um predicado é verdadeiro para todos os elementos de um dado conjunto, enquanto o último é utilizado para indicar que um predicado é verdadeiro para pelo menos um elemento em um dado conjunto. Por exemplo, a entidade “PessoaX” refere-se a “Pelo menos uma pessoa que faleceu”, assim o quantificador existencial é associado a esta entidade. A entidade “PessoaY” e “Herança” seguem o mesmo princípio, sendo definido a elas o quantificador existencial.

A subatividade “Definição de implicação ou equivalência” recebe como entrada os predicados que compõem a condição e conclusão do axioma, identificados na subatividade “Definição da condição e conclusão” para determinar se o axioma a ser criado é uma implicação ou uma equivalência. A implicação é utilizada quando a satisfação da condição leva a conclusão. A equivalência ocorre quando há uma simetria entre a condição e a conclusão. Por exemplo, uma implicação é usada para formar o seguinte axioma: “ $\exists \text{ PessoaX}, \text{ PessoaY}, \text{ Heranca} \mid \text{identificarHerdeiros}(\text{PessoaX}, \text{ PessoaY}) \wedge \text{determinarHeranca}(\text{Heranca}, \text{ PessoaY}) \Rightarrow \text{calcularHeranca}(\text{Heranca}, \text{ PessoaX})$ ”.

Como ilustrado na Figura 3.1.1 as atividades da GAODT são executadas iterativamente. Portanto, para construir um novo axioma, cada um dos predicados objetivo na condição do corrente axioma são submetidos para a atividade “Seleção dos Objetivos e Fatos” onde os itens contidos na lista de objetivos e fatos (por exemplo, Tabela 3.2.1) que satisfazem a condição são selecionados pelo especialista de domínio e representados em predicados em LPO através da atividade “Representação dos Predicados em LPO”, por fim, esses predicados retornam a atividade “Especificação dos Axiomas em LPO” formando o novo axioma.

Por exemplo, o predicado objetivo “ $\text{identificarHerdeiros}(\text{PessoaX}, \text{ PessoaY})$ ” na qual é parte da condição no axioma de exemplo apresentado anteriormente é submetido para a

atividade “Seleção dos Objetivos e Fatos” onde o especialista de domínio informa que os itens “Identificar os descendentes de uma pessoa”, “Identificar os ascendentes de uma pessoa”, “Identificar o cônjuge de uma pessoa” e “Identificar os colaterais de uma pessoa” satisfazem este objetivo. Então, esses itens em linguagem natural são representados em predicados em LPO através da atividade “Representação dos Predicados em LPO” (Tabela 3.4.2).

Objetivos em Linguagem Natural	Predicados
Identificar os descendentes de uma pessoa	identificarDescendentes(PessoaX, PessoaY)
Identificar os ascendentes de uma pessoa	identificarAscendentes(PessoaX, PessoaY)
Identificar o cônjuge de uma pessoa	identificarCônjuge(PessoaX, PessoaY)
Identificar os colaterais de uma pessoa	identificarColaterais(PessoaX, PessoaY)

Tabela 3.4.2: Objetivos representados como predicados

Estes predicados são dados como entrada para a atividade “Especificação dos Axiomas em LPO” para gerarem o novo axioma apresentado na Tabela 3.4.3 e um novo ciclo da técnica GAODT se inicia.

Variáveis	Condição	Implicação
\exists PessoaX \exists PessoaY	identificarDescendentes(PessoaX, PessoaY) \vee identificarAscendentes(PessoaX, PessoaY) \vee identificarCônjuge(PessoaX, PessoaY) \vee identificarColaterais(PessoaX, PessoaY)	identificarHerdeiros (PessoaX, PessoaY)

Tabela 3.4.3: Axioma desenvolvido na atividade “Especificação dos Axiomas em LPO”.

O predicado “determinarHerança(Herança, PessoaY)” que também faz parte da condição do axioma passa pelo mesmo processo de especificação e representação em LPO na qual o predicado “identificarHerdeiros(PessoaX, PessoaY)” foi submetido, gerando o novo axioma apresentando na Tabela 3.4.4.

Variáveis	Condição	Implicação
\exists Bens \exists PessoaX \exists PessoaY \exists PessoaZ \exists RegimeUniversal \exists Herança	verificarDescendentes(PessoaX, PessoaY) \wedge verificarCônjuge(PessoaX, PessoaY) \wedge verificarRegimeUniversal(RegimeUniversal, PessoaX) \wedge partilharBensDescendentesCônjugeRegimeUniversal (Bens, PessoaX, PessoaY, PessoaZ, RegimeUniversal)	determinarHerança (Herança, PessoaY)

Tabela 3.4.4: Axioma desenvolvido através do processo iterativo da GAODT.

O processo é então executado novamente para cada um dos subobjetivos, até que todos os objetivos dados como entrada para a técnica (como ilustrado na Tabela 3.2.1) tenham sido satisfeitos. O produto desta atividade é um conjunto de axiomas especificados em LPO e apresentados na Tabela 3.4.5.

Variáveis	Condição	Implicação
\exists PessoaX \exists PessoaY \exists Herança	$\text{identificarHerdeiros}(\text{PessoaX}, \text{PessoaY}) \wedge$ $\text{determinarHerança}(\text{PessoaX}, \text{Herança})$	calcularHeranca $(\text{PessoaX}, \text{Herança})$
\exists PessoaX \exists PessoaY	$\text{identificarDescendentes}(\text{PessoaX}, \text{PessoaY}) \vee$ $\text{identificarAscendentes}(\text{PessoaX}, \text{PessoaY}) \vee$ $\text{identificarCônjuge}(\text{PessoaX}, \text{PessoaY}) \vee$ $\text{identificarColaterais}(\text{PessoaX}, \text{PessoaY})$	$\text{identificarHerdeiros}$ $(\text{PessoaX}, \text{PessoaY})$
\exists Bens \exists PessoaX \exists PessoaY \exists PessoaZ \exists RegimeUniversal \exists Herança	$\text{verificarDescendentes}(\text{PessoaX}, \text{PessoaY}) \wedge$ $\text{verificarCônjuge}(\text{PessoaX}, \text{PessoaY}) \wedge$ $\text{verificarRegimeUniversal}$ $(\text{RegimeUniversal}, \text{PessoaX}) \wedge$ $\text{partilharBensDescendentes}$ $\text{CônjugeRegimeUniversal}$ $(\text{Bens}, \text{PessoaX}, \text{PessoaY}, \text{PessoaZ},$ $\text{RegimeUniversal})$	determinarHerança $(\text{Herança}, \text{PessoaY})$
\exists PessoaX \exists PessoaY	temDescendentes $(\text{PessoaX}, \text{PessoaY})$	$\text{verificarDescendentes}$ $(\text{PessoaX}, \text{PessoaY})$
\exists PessoaX \exists PessoaY	temCônjuge $(\text{PessoaX}, \text{PessoaY})$	verificarConjuge $(\text{PessoaX}, \text{PessoaY})$
\exists PessoaX \exists PessoaY	temAscendente $(\text{PessoaX}, \text{PessoaY})$	$\text{verificarAscendentes}$ $(\text{PessoaX}, \text{PessoaY})$
\exists Bens \exists PessoaY \exists Quinhão \exists Meação \exists RegimeUniversal	$\text{determinarQuinhãoDescendentesRegimeUniversal}$ $(\text{Quinhão}, \text{PessoaX}, \text{PessoaY}, \text{RegimeUniversal}) \wedge$ $\text{determinarMeaçãoConjugeRegimeUniversal}$ $(\text{Meação}, \text{PessoaX}, \text{PessoaY}, \text{RegimeUniversal})$	$\text{partilharBensDescendentes}$ $\text{CônjugeRegimeUniversal}$ $(\text{Bens}, \text{PessoaX}, \text{PessoaY}, \text{PessoaZ},$ $\text{RegimeUniversal})$
\exists Bens \exists PessoaY \exists Quinhão \exists Meação \exists RegimeUniversal	$\text{determinarMeaçãoConjugeRegimeUniversal}$ $(\text{Meação}, \text{PessoaX}, \text{PessoaY}, \text{RegimeUniversal}) \wedge$ $\text{determinarQuantidadeDescendentes}$ $(\text{PessoaY}, \text{Quantidade}, \text{PessoaY}, \text{PessoaX}) \wedge$ $\text{dividirMeaçãoQuantidadeDescendentes}$ $(\text{Meação}, \text{Quantidade}, \text{PessoaY}, \text{PessoaX})$	determinarQuinhão $\text{DescendentesRegimeUniversal}$ $(\text{PessoaX}, \text{Quinhão},$ $\text{RegimeUniversal})$
\exists Bens \exists PessoaY \exists Quinhão \exists Meação	$\text{somarBens}(\text{PessoaX}, \text{Bens}) \wedge$ $\text{dividirHerançaBensCônjuge}$ $\text{DescendentesRegimeUniversal}$ $(\text{PessoaX}, \text{Herança}, \text{Bens}, \text{PessoaY}, \text{PessoaZ},$ $\text{RegimeUniversal})$	determinarMeação $\text{ConjugeRegimeUniversal}$ $(\text{PessoaX}, \text{Meação},$ $\text{RegimeUniversal})$
\exists PessoaX \exists BensComuns \exists BensParticulares	$\text{somarBensComuns}(\text{PessoaX}, \text{BensComuns}) \wedge$ $\text{somarBensParticulares}(\text{PessoaX}, \text{BensParticulares})$	somarBens $(\text{PessoaX}, \text{Bens})$
\exists PessoaX \exists BensParticulares \exists Valor	$\text{temBensParticulares}(\text{PessoaX}, \text{BensParticulares}) \wedge$ $\text{temValor}(\text{BensParticulares}, \text{Valor})$	$\text{somarBensParticulares}$ $(\text{BensParticulares}, \text{PessoaX})$
\exists PessoaX \exists BensComuns \exists Valor	$\text{temBensComuns}(\text{PessoaX}, \text{BensComuns}) \wedge$ $\text{temValor}(\text{BensComuns}, \text{Valor})$	somarBensComuns $(\text{BensComuns}, \text{PessoaX})$
\exists PessoaX \exists PessoaY \exists Quantidade	$\text{temDescendentes}(\text{PessoaX}, \text{PessoaY})$	$\text{determinarQuantidadeDescendentes}$ $(\text{Quantidade}, \text{PessoaX}, \text{PessoaY})$

Tabela 3.4.5: Produto final da atividade “Especificação dos Axiomas em LPO”.

3.5 Especificação/Extensão da Ontologia de Aplicação

O objetivo desta atividade é extrair os elementos constituintes dos axiomas especificados na atividade anterior para a construção da ontologia de aplicação. Esta atividade consiste de seis subatividades: “Tradução dos axiomas”, “Definição das classes”, “Definição dos relacionamentos não taxonômicos”, “Definição dos relacionamentos taxonômicos”, “Definição das propriedades” e “Recuperação de ontologia de aplicação” (Figura 3.5.1).

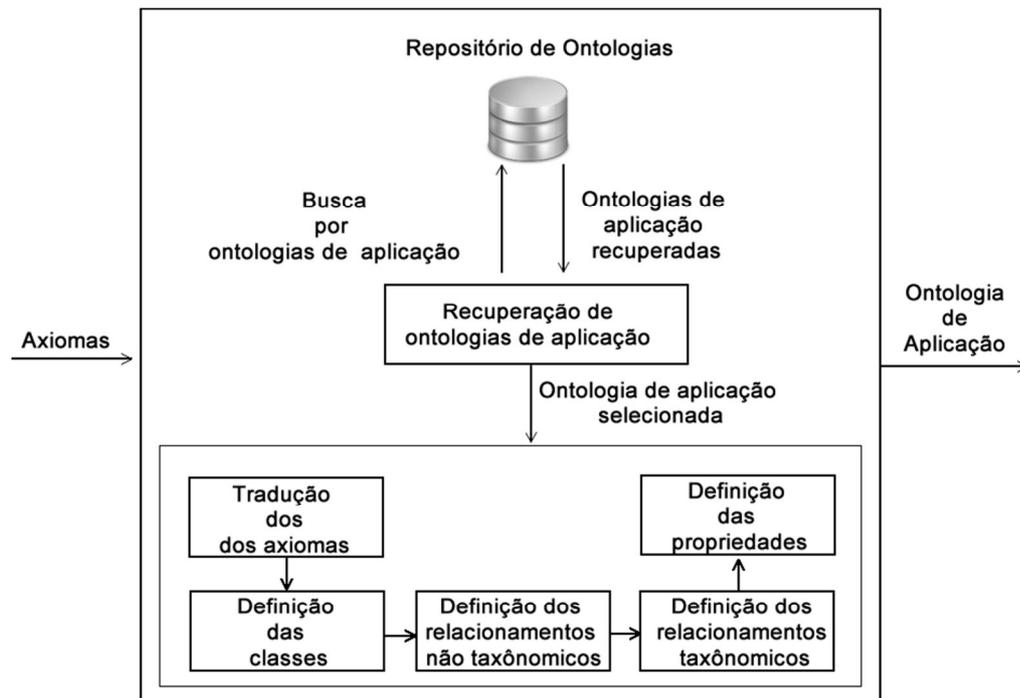


Figura 3.5.1: Atividade “Especificação/Extensão da Ontologia de Aplicação”.

A subatividade “Tradução dos axiomas” converte os axiomas definidos na atividade anterior (“Especificação dos Axiomas em LPO”) expressos em LPO para uma linguagem de regras para ontologias, como RuleML [18]. A linguagem RuleML foi utilizada por causa de sua expressividade, ou seja, possui recursos suficientes para representar regras em LPO.

Para realizar esta tradução, as seguintes heurísticas são aplicadas. Primeiro, expressões regulares [20] [23] são utilizadas para extrair os predicados dos axiomas. Por exemplo, para a regra “ \exists PessoaX, PessoaY, Herança | identificarHerdeiros(PessoaX, PessoaY) \wedge determinarHerança(Herança, PessoaY) \Rightarrow calcularHerança(Herança, PessoaX)”, a seguinte expressão regular foi utilizada “ $\wedge(\wedge+\backslash(.*)\wedge) \wedge (\wedge+\backslash(.*)\wedge) \Rightarrow (\wedge+\backslash(.*)\wedge)$ ”. Então, os predicados dos axiomas são especificados em POSL [3] e automaticamente traduzidos para axiomas em RuleML (Figura 3.5.2).

```

<Assert>
  <Rulebase mapClosure='universal'>
<Implies>
<And>
<Atom>
  <Rel>identificarHerdeiros</Rel>
  <Var>PessoaX</Var>
  <Var>PessoaY</Var>
</Atom>
<Atom>
  <Rel>determinarHerança</Rel>
  <Var>Herança</Var>
  <Var>PessoaY</Var>
</Atom>
</And>
<Atom>
  <Rel>calcularHerança</Rel>
  <Var>Herança</Var>
  <Var>PessoaX</Var>
</Atom>
</Implies>
  </Rulebase>
</Assert>

```

Figura 3.5.2: Exemplo de um axioma representado em RuleML

A subatividade “Definição das classes” extrai as variáveis dos predicados presentes nos axiomas desenvolvidos na atividade “Especificação dos Axiomas em LPO” e ilustrados na Tabela 3.4.5. Por exemplo, o predicado “identificarHerdeiros(PessoaX, PessoaY)” tem as variáveis “PessoaX” e “PessoaY” ambas correspondendo a classe “Pessoa”.

A subatividade “Definição dos relacionamentos não taxonômicos” extrai os relacionamentos não taxonômicos da ontologia de aplicação a partir dos predicados na lista de axiomas desenvolvidos na atividade “Especificação dos Axiomas em LPO” (por exemplo, Tabela 3.4.5). Portanto, são extraídos os predicados e as variáveis que eles relacionam. Por exemplo, em relação ao predicado “calcularHerança(PessoaX, Herança)” o relacionamento não taxonômico identificado é o predicado “calcularHerança”, que define a relação entre as classes “Pessoa” e “Herança”.

A subatividade “Definição dos relacionamentos taxonômicos” extrai um conjunto de relacionamentos taxonômicos baseado nas relações hierárquicas entre as classes definidas na subatividade “Definição das classes”. Por exemplo, há uma relação hierárquica entre as classes, “Bens”, “BensComuns” e “BensParticulares”.

A subatividade “Definição das propriedades” extrai os predicados contidos nos axiomas desenvolvidos na atividade anterior que descrevem os atributos das classes. Por exemplo, “temValor(BensParticulares, Valor)” e “temValor(BensComuns, Valor)” descreve que as classes “BensParticulares” e “BensComuns” tem a propriedade “temValor”.

Se existir a necessidade de estender a ontologia de aplicação desenvolvida, a subatividade “Recuperação de ontologias de aplicação” realiza uma busca semântica por ontologias de aplicação em algum repositório de ontologias. Algumas medidas de similaridade [6] [24] podem ser usadas para ranquear as ontologias recuperadas.

A Figura 3.5.3, Figura 3.5.4, Tabela 3.5.1 e Tabela 3.5.2 apresentam respectivamente a taxonomia, um exemplo de um axioma, os relacionamentos não taxonômicos e a propriedades da ontologia de aplicação desenvolvida na qual constituem o produto final da GAODT.

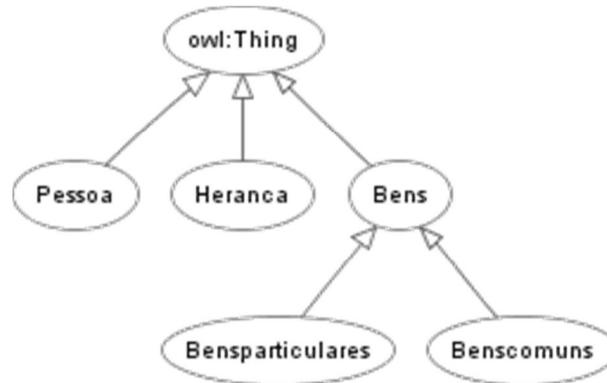


Figura 3.5.3: Relacionamentos taxonômicos da ontologia de aplicação.

```

<Assert>
<Rulebase mapClosure='universal'>
<Implies>
<And>
<Atom>
  <Rel>temValor</Rel>
  <Var>BensComuns</Var>
  <Var>Valor</Var>
</Atom>
<Atom>
  <Rel>somarBensComuns</Rel>
  <Var>PessoaX</Var>
  <Var>BensComuns </Var>
</Atom>
</Implies>
</Rulebase>
</Assert>
  
```

Figura 3.5.4: Exemplo de um axioma representado em RuleML.

Domínios	Relacionamentos	Imagens
Pessoa	calcularHerança	Herança
Pessoa	identificarAscendentes	Pessoa
Pessoa	identificarCônjuge	Pessoa
Pessoa	identificarDescendentes	Pessoa
Pessoa	identificarColaterais	Pessoa
Pessoa	determinarHerança	Herança
Pessoa	identificarHerdeiros	Pessoa
Pessoa	temDescendentes	Pessoa
Pessoa	temBensComuns	BensComuns
Pessoa	temBensParticulares	BensParticulares
Pessoa	partilharBensDescendentes	Bens
Pessoa	partilharBensDescendentesConjugeRegimeUniversal	Bens

Tabela 3.5.1: Relacionamentos não taxonômicos da ontologia de aplicação.

Domínios	Propriedades
Pessoa	verificarDescendentes
Pessoa	temRegimeUniversal
Pessoa	verificarRegimeUniversal
Pessoa	verificarCônjuge
Pessoa	determinarQuantidadeAscendentes
Pessoa	determinarQuantidadeDescendentes
Pessoa	somarBensComuns
Pessoa	somarBensParticulares
Pessoa	somarBens
Pessoa	determinarQuinhãoDescendentesRegimeUniversal
Pessoa	determinarMeaçãoConjugeRegimeUniversal
BensParticulares	temValor
BensComuns	temValor
Pessoa	dividirMeaçãoQuantidadeDescendentes
Bens	dividirHerançaCônjugeDescendentesRegimeUniversal

Tabela 3.5.2: Propriedades da ontologia de aplicação.

3.6 GAODTool – Uma Ferramenta de Suporte a GAODT

Para auxiliar o desenvolvimento de ontologias de aplicação com a GAODT desenvolveu-se uma ferramenta de software que provê suporte semi-automatizado a sua aplicação. Para a especificação da interface utilizou-se as linguagens XHTML [36], CSS [8] e JavaScript [22]. Já PHP [28] e Java [21] estão sendo empregadas para gerar a ontologia de aplicação.

A Figura 3.6.1 apresenta a tela inicial da ferramenta GAODTool. Já a Figura 3.6.2 apresenta a tela principal contendo os quatro títulos “Seleção dos Objetivos e Fatos”, “Representação dos Predicados em LPO”, “Especificação dos Axiomas em LPO” e “Especificação da Ontologia de Aplicação” a qual correspondem a cada uma das atividades da GAODT. Para cada um deles há uma tela específica as quais serão apresentadas a seguir, na ordem especificada pela GAODT.



Figura 3.6.1: Tela inicial da ferramenta GAODTool

Adição de Objetivos e Fatos
Selecione o item a ser inserido ▾
Seleção dos Objetivos e Fatos
Representação dos Predicados em LPO
Especificação dos Axiomas em LPO
Especificação da Ontologia de Aplicação

Figura 3.6.2: Tela principal da ferramenta da GAODTool

A Figura 3.6.3 corresponde a tela que permite ao engenheiro do conhecimento informar os objetivos e fatos do sistema, o que corresponde as informações de entrada da GAODT, conforme definido na seção 3.1.

Adição de Objetivos e Fatos
Objetivo ▾
Informe o objetivo
<input type="text"/>
ADICIONAR OBJETIVO

Figura 3.6.3: Área “Adição de Objetivos e Fatos”

Os objetivos e fatos informados são automaticamente adicionados a área “Seleção dos Objetivos e Fatos” formando uma lista com os objetivos e fatos do sistema. A partir dessa lista o engenheiro do conhecimento especifica quais desses serão dados como entrada para a área “Representação dos Predicados em LPO” da ferramenta. É possível ainda excluir e editar os

objetivos ou fatos contidos na lista. Na Figura 3.6.4 pode-se observar um objetivo e um fato adicionados.

Seleção dos Objetivos e Fatos			
Lista de Objetivos e Fatos			Tipo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Determinar os ...	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Todos os ...	fato

Figura 3.6.4: Lista de objetivos e fatos contidos na área “Seleção dos Objetivos e Fatos”

Os objetivos e fatos em linguagem natural selecionados na área anterior “Seleção dos Objetivos e Fatos” são automaticamente inseridos na área “Representação dos Predicados em LPO” para representá-los em predicados em LPO.

A área “Representação dos Predicados em LPO” (Figura 3.6.5) possui sete subáreas: “Identificação das entidades”, “Redefinição das entidades”, “Identificação dos relacionamentos”, “Redefinição dos relacionamentos”, “Definição da aridade”, “Definição dos predicados”, e “Redefinição das entidades dos predicados” a qual correspondem as subatividades especificadas na sessão 3.3. Cada subárea é acessada através dos botões “Anterior” e “Próxima”.

A área “Representação dos Predicados em LPO” possui um compartimento contendo uma numeração de “1” a “7” referente às sete subáreas e a cada numeração é associada uma cor: “laranja”, representa a subárea desenvolvida atualmente; “amarelo”, a subárea ainda não desenvolvida e “verde”, a subárea já desenvolvida.

Representação dos Predicados em LPO							
<div style="display: flex; justify-content: space-around;"> 1 2 3 4 5 6 7 </div>							
<div style="border: 1px solid black; padding: 2px; margin-bottom: 10px;"> Identificação dos Relacionamentos </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Itens Selecionados</th> <th style="width: 50%;">Relacionamentos</th> </tr> </thead> <tbody> <tr> <td>Determinar os ...</td> <td>Informe aqui os relacionamentos</td> </tr> <tr> <td>Todos os ...</td> <td>Informe aqui os relacionamentos</td> </tr> </tbody> </table>		Itens Selecionados	Relacionamentos	Determinar os ...	Informe aqui os relacionamentos	Todos os ...	Informe aqui os relacionamentos
Itens Selecionados	Relacionamentos						
Determinar os ...	Informe aqui os relacionamentos						
Todos os ...	Informe aqui os relacionamentos						
<div style="display: flex; justify-content: space-around;"> ANTERIOR PRÓXIMA </div>							

Figura 3.6.5: Área “Representação dos Predicados em LPO”

Ao finalizar a área “Representação dos Predicados em LPO”, o conjunto de predicados desenvolvidos são automaticamente submetidos a área “Especificação dos Axiomas em LPO” para formar os axiomas.

A área “Especificação dos Axiomas em LPO” (Figura 3.6.6) possui quatro subáreas:

“Definição da condição e conclusão”, “Definição dos operadores booleanos”, “Definição dos quantificadores” e “Definição de implicação ou equivalência” a qual estão em conformidade com as subatividades especificadas na sessão 3.4. Cada subárea é acessada através dos botões “Anterior” e “Próxima”.

Assim como a área “Representação dos Predicados em LPO” a área “Especificação dos Axiomas em LPO” também possui um compartimento contendo uma numeração de “1” a “4” referente às quatro subáreas e a cada numeração é associada uma cor (laranja, amarelo ou verde) que representa o seu status.

Figura 3.6.6: Área “Especificação dos Axiomas em LPO”

Como especificado na sessão 3.4 a GAODT possui um ciclo iterativo, logo, ao finalizar a área “Especificação dos Axiomas em LPO” o axioma construído é armazenado em uma lista de axiomas desenvolvidos (Figura 3.6.7) e as informações contidas nas áreas “Representação dos Predicados em LPO” e “Especificação dos Axiomas em LPO” são removidas para permitir que o engenheiro do conhecimento possa desenvolver um novo axioma e assim o ciclo se repete. É possível ainda editar, excluir e trocar a ordem dos axiomas desenvolvidos.

Figura 3.6.7: Lista de axiomas desenvolvidos

Os axiomas desenvolvidos são submetidos para área “Especificação da Ontologia de Aplicação” o qual a partir deles são extraídos os elementos que compõe a ontologia de aplicação.

A área “Especificação da Ontologia de Aplicação” (Figura 3.6.8) possui cinco subáreas: “Tradução dos axiomas”, “Definição das classes”, “Definição dos relacionamentos não taxonômicos”, “Definição dos relacionamentos taxonômicos” e “Definição das propriedades” a qual estão em conformidade com as subatividades especificadas na sessão 3.5. Cada subárea é

acessada através dos botões “Anterior” e “Próxima”.

Assim como as áreas “Representação dos Predicados em LPO” e “Especificação dos Axiomas em LPO” a área “Especificação da Ontologia de Aplicação” também possui um compartimento contendo uma numeração de “1” a “5” referente às cinco subáreas e a cada numeração é associada uma cor (laranja, amarelo ou verde) que representa o seu status.

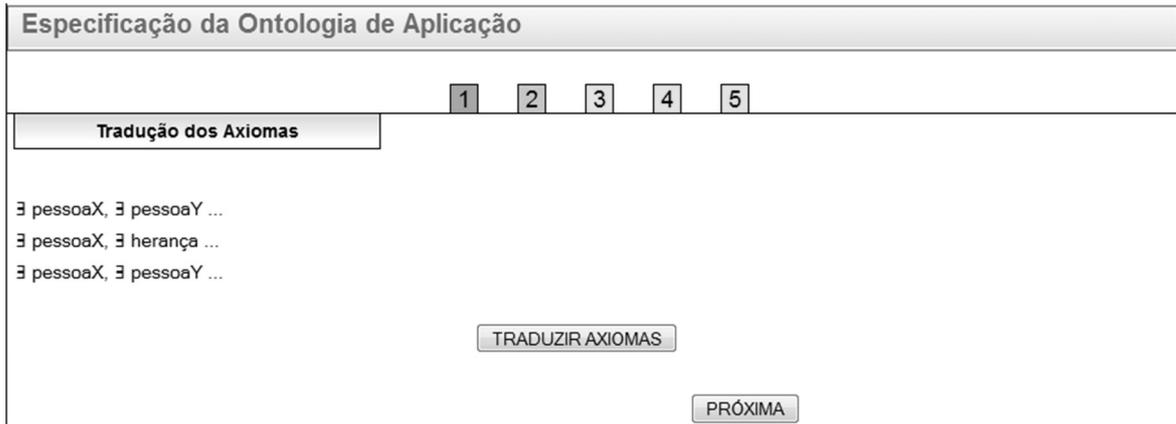


Figura 3.6.8: Área “Especificação da Ontologia de Aplicação”

Por fim, após a utilização da área “Especificação da Ontologia de Aplicação” a ferramenta disponibiliza ao engenheiro do conhecimento a opção de gerar a ontologia de aplicação desenvolvida. A Figura 3.6.9 apresenta um arquivo de exemplo contendo uma ontologia de aplicação desenvolvida e disponibilizada ao desenvolvedor para download.

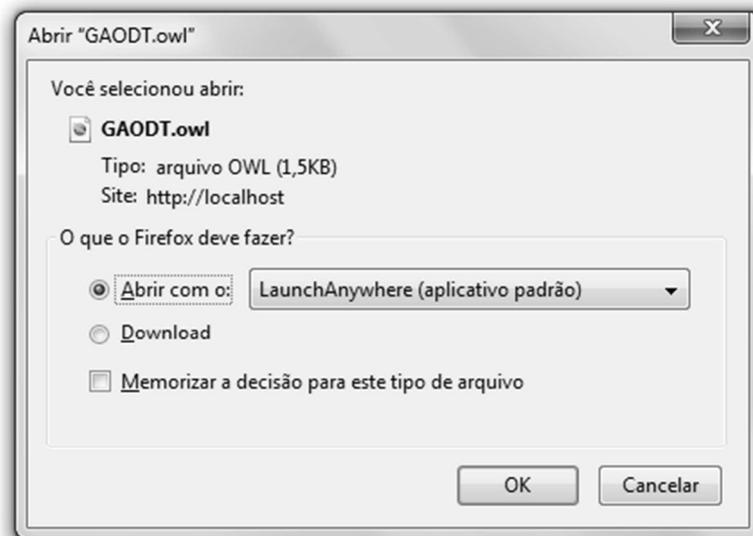


Figura 3.6.9: Arquivo contendo a ontologia de aplicação gerada.

3.7 Discussão sobre a GAODT

A GAODT é uma técnica para a construção de ontologias de aplicação que por meio de um ciclo de desenvolvimento orientado por objetivos, facilita a construção incremental de ontologias de aplicação.

A técnica também fornece ao desenvolvedor uma forma bem definida de traduzir o conhecimento expresso em linguagem natural, para uma forma computacional, utilizando-se inicialmente a LPO.

No desenvolvimento de um SBC uma grande quantidade de informação é produzida e precisa ser gerenciada, sendo latente, dessa forma, a necessidade de suporte automatizado. Para tanto foi desenvolvida a GAODTool, uma ferramenta de interface intuitiva que provê suporte a todas as atividades da GAODT.

Construir ontologias reusáveis é um processo caro e dispendioso. Dentre os quatro tipos de ontologias definidas por Guarino [17], as ontologias de aplicação são menos reusáveis uma vez que elas são desenvolvidas para aplicações de software específicas. Entretanto, são mais rápidas e baratas de se desenvolver.

Desenvolver ontologias de aplicação e então generalizar seus elementos para ontologias de domínio e tarefa é uma boa abordagem alternativa para o desenvolvimento de artefatos reusáveis. Neste contexto, GAODT consiste da primeira etapa nesta direção por definir um caminho sistematizado para a construção de ontologias de aplicação.

Na Tabela 3.7.1, a técnica GAODT é classificada com relação aos critérios adotados na seção 2.7: Dentre os quatro tipos de ontologias definidos por Guarino [17], a GAODT permite o desenvolvimento de ontologias de aplicação. Na definição da ordem de descoberta dos elementos da ontologia, a GAODT inicia pela descoberta dos axiomas considerando que eles representam os requisitos do SBC. O seu ciclo de vida é iterativo, pois permite ao engenheiro do conhecimento adicionar novos elementos para a ontologia a cada iteração do processo. Além disso, a GAODT permite o reuso de ontologias através da subatividade “Recuperação de ontologias de aplicação” e define regras para encontrar classes através de substantivos e relacionamentos através de verbos.

Critério comparativo	GAODT
O tipo de ontologia desenvolvida	Ontologias de aplicação
A ordem na qual os elementos da ontologia são descobertos	Axiomas (Objetivos e Fatos) Classes Relações Taxonomia Propriedades
O tipo do ciclo de vida	Iterativo
O reuso de ontologias existentes	Sim
Usa conhecimento linguístico	Sim

Tabela 3.7.1: Classificação da GAODT com relação aos critérios de análise da seção 2.7

3.8 Considerações Finais

Este capítulo contemplou a parte principal da presente dissertação, posto que nele foram descritas a técnica GAODT para o desenvolvimento de ontologia de aplicação, por meio de um ciclo de desenvolvimento orientado por objetivos e a GAODTool uma ferramenta semi-automatizada que suporta o desenvolvimento das atividades da GAODT.

Algo que merece especial destaque é o nível de detalhes que a GAODT fornece ao engenheiro do conhecimento para formalizar o conhecimento expresso em linguagem natural, para uma forma computacional, utilizando-se inicialmente a LPO.

Assim, foram descritas todas as atividades, subatividades e produtos da técnica GAODT, a qual foi ilustrada através de exemplos no domínio da Sucessão Legítima, que será melhor explicado no estudo de caso apresentado no capítulo 4.

A ferramenta de suporte GAODTool, por meio de uma interface intuitiva provê suporte a todas as atividades da GAODT, permitindo assim, que o engenheiro de conhecimento construa ontologias de aplicação de qualidade.

4 O ESTUDO DE CASO

4.1 Introdução

Este capítulo apresenta um estudo caso que consistiu na utilização da GAODT na construção da “InheritanceLaw”, uma ontologia de aplicação para apoio a tomada de decisões na Sucessão Legítima, um tipo de sucessão disciplinado pelo domínio do Direito Sucessório. O Direito Sucessório compreende o conjunto de normas que disciplinam a transferência do patrimônio de alguém, depois de sua morte, em virtude de lei ou testamento. O desenvolvimento da “InheritanceLaw” foi assistido pela ferramenta GAODTool. O domínio da Sucessão Legítima foi escolhido por ser um ramo do direito que é fonte de estudos do Grupo de Pesquisa em Engenharia de Software e Engenharia do Conhecimento - GESEC.

Na seção 4.2 é descrito o domínio utilizado para a construção da ontologia de aplicação. Na seção 4.3 é apresentado o desenvolvimento da ontologia de aplicação “InheritanceLaw” com a utilização da ferramenta de suporte GAODTool. Na seção 4.4 são apresentadas as considerações finais do capítulo.

4.2. O Domínio Abordado: A Sucessão Legítima

A Sucessão Legítima é um instituto do direito que governa a transferência do patrimônio de alguém, em função de sua morte, na ausência de um testamento. Neste caso, há um conjunto de regras utilizadas para determinar quem herdará o patrimônio. A seguinte ordem é aplicada: descendentes (filhos, netos, dentre outros) concorrem com o cônjuge sobrevivente; na ausência dos descendentes, os ascendentes (pai, mãe, avô, avó, dentre outros) também concorrem com o cônjuge sobrevivente; se não existem ascendentes, o cônjuge e finalmente, na ausência do cônjuge, os bens são transmitidos para os colaterais (irmãos, tios e primos) [1].

A concorrência do cônjuge com os descendentes depende do regime matrimonial. O regime matrimonial consiste no sistema de bens entre os cônjuges, que prevê a criação ou a ausência de bens conjugais, como e por quem eles são geridos, e como ele será dividido e herdado ao fim do casamento, os quais são de quatro tipos:

- O Regime Universal consiste na união de todos os bens particulares e comuns do casal, e no momento da partilha o cônjuge sobrevivente não concorre pela herança, pois a metade (meação) de todos os bens já lhe pertence. Por exemplo,

considere que o falecido João deixou um patrimônio avaliado em R\$ 100,000; desse valor, metade já pertence à Maria sua esposa viva, o restante da importância será dividida entre os demais herdeiros.

- No Regime Parcial o cônjuge sobrevivente possui a metade dos bens comuns e ainda concorre na herança dos bens particulares com os descendentes. Por exemplo, o falecido Pedro deixou um patrimônio particular de R\$ 200,000 e um patrimônio constituído com Luísa avaliado em R\$ 100,000; desses valores, Luísa já possui a metade do patrimônio comum e ainda irá concorrer com os descendentes pelo patrimônio particular.
- No Regime de Separação de Bens não há comunicação de bens, estes permanecem sob a administração exclusiva de cada um dos cônjuges que poderão livremente dispor deles. Esse regime pode ocorrer de forma voluntária ou obrigatória. No regime de separação voluntária, os nubentes estipulam livremente em contrato (pacto antenupcial) antes de celebrado o casamento; já no regime de separação obrigatória, é obrigatória a separação dos bens, decorrendo esta da vontade da lei. Por exemplo, o falecido Lucas casado com Ana no regime de separação voluntária, possui dois filhos e um patrimônio avaliado em R\$ 900,000; desse valor, os filhos e a esposa receberão partes iguais.
- A Participação nos Aquestos é um regime híbrido entre a comunhão parcial e a separação total de bens, permitindo a separação dos bens individuais durante a união conjugal e, após o término desta, permite aos ex-cônjuges a participação final nesses bens (aquestos), com a divisão proporcional dos bens adquiridos durante a constância do casamento. É uma espécie de regime de bens que confere grande autonomia ao longo da união, útil para casais em determinadas profissões que exigem autonomia em relação a bens imóveis, tais como corretores, que para vender e comprar imóveis não necessita da autorização do outro cônjuge.

A concorrência dos ascendentes com o cônjuge independe de regime matrimonial, uma vez que existem duas regras específicas para essa situação. A primeira consiste em reservar

um terço de todo o patrimônio do falecido ao seu cônjuge, caso ele concorra com os dois ascendentes de primeiro grau (pai e mãe) do falecido. Por exemplo, o falecido Marcos é casado com Julia e filho de João e Maria, ambos vivos. Marcos possui um patrimônio avaliado em R\$ 600,000; desse valor, tanto sua esposa Julia, como João e Maria receberão R\$ 200,000 cada um. A segunda regra prevê a reserva de metade do patrimônio do falecido para o seu cônjuge, caso ele concorra com apenas um ascendente de primeiro grau (pai ou mãe) ou ascendentes de maior grau (avô, avó, bisavô, bisavó, dentre outros). Por exemplo, João é casado com Ana, possui dois avós e faleceu deixando um patrimônio de R\$ 500,000; desse valor, metade pertence à Ana e o restante será dividido igualmente entre os dois avós.

Na ausência dos descendentes e ascendentes o cônjuge herdará todos os bens comuns e particulares do falecido. Por exemplo, Ana é casada com o falecido João que deixou um patrimônio avaliado em R\$ 100,000 e por não possuir descendentes e nem ascendentes, ela receberá toda a herança.

Na ausência dos descendentes, ascendentes e cônjuge, os colaterais até quarto grau (irmãos, tios e primos) herdam todo o patrimônio do falecido. Por exemplo, o falecido Pedro não possui descendentes, ascendentes e cônjuge, porém, possui dois irmãos germanos (mesmo pai e mãe) que no momento da partilha receberão cada um a metade da herança.

A Sucessão Legítima também define regras para a União Estável, que consiste na relação de convivência duradoura entre o homem (companheiro) e a mulher (companheira) com o objetivo de constituição familiar. A companheira ou o companheiro participará da sucessão do outro, quanto aos bens adquiridos onerosamente (bens comuns) na vigência da união estável, nas seguintes condições:

- Se concorrer com filhos comuns, fará jus a uma quota equivalente à que legalmente couber a eles. Por exemplo, Lucas tinha um patrimônio particular de R\$ 100,000; uniu-se com Ana, sendo que durante a convivência os dois tiveram dois filhos e formaram um patrimônio comum de R\$ 600,000. Lucas morreu. Portanto, Ana tem direito à meação (R\$ 300,000). O patrimônio particular de Lucas será partilhado apenas entre seus dois filhos. Já os outros R\$ 300,000 restantes do patrimônio comum serão divididos em partes iguais para Ana e seus dois filhos.
- Se concorrer com os descendentes só do falecido terá direito à metade do que couber a cada um deles. Utilizando-se o exemplo anterior, suponha que os

filhos são apenas de Lucas. O patrimônio particular será partilhado entre seus dois filhos e sua companheira Ana receberá a metade do que os filhos de Lucas receberão em relação à divisão do patrimônio comum.

- Se concorrer com outros parentes sucessíveis (ascendentes ou colaterais) terá direito a um terço da totalidade da herança. Por exemplo, somando o patrimônio particular de João, com o patrimônio adquirido a partir da constância da união com Maria, obtêm-se R\$ 300,000. Caso João venha a falecer e não possua filhos, R\$ 100,000 pertencem a Maria e o restante será dividido entre os demais herdeiros.
- Não havendo parentes sucessíveis (descendentes, ascendentes e colaterais) o companheiro terá direito a totalidade da herança.

4.3 Construindo a Ontologia de Aplicação “InheritanceLaw”

Esta seção apresenta a construção da “InheritanceLaw”, uma ontologia de aplicação no domínio da Sucessão Legítima, cujo desenvolvimento foi assistido pela ferramenta GAODTool.

Inicialmente o desenvolvedor em consenso com o especialista de domínio informam os objetivos e fatos do sistema no domínio da Sucessão Legítima, através da área “Adição de Objetivos e Fatos”, a Figura 4.3.1 ilustra a inserção do objetivo “Calcular a herança de uma pessoa”.

A imagem mostra uma interface de usuário com o título "Adição de Objetivos e Fatos". No topo, há um campo de seleção rotulado "Objetivo". Abaixo dele, há uma seção intitulada "Informe o objetivo" com um campo de texto contendo o texto "Calcular a herança de uma pessoa". Na base da interface, há um botão com o rótulo "ADICIONAR OBJETIVO".

Figura 4.3.1: Inserção de um objetivo no domínio da Sucessão Legítima

Os objetivos e fatos informados são automaticamente adicionados a área “Seleção dos Objetivos e Fatos” formando uma lista com os objetivos e fatos do sistema. A Figura 4.3.2 apresenta a visão parcial da lista desenvolvida. A lista completa dos objetivos e fatos encontra-se no Apêndice A.

Seleção dos Objetivos e Fatos				
Lista de Objetivos e Fatos			Tipo	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Calcular a herança de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Identificar os herdeiros de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Determinar a herança dos herdeiros	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Verificar a existência de descendentes de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Identificar os descendentes de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Verificar a existência do cônjuge de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Identificar o cônjuge de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Verificar a existência de ascendentes de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Identificar os ascendentes de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Verificar a existência dos colaterais de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Identificar os colaterais de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Uma pessoa tem descendentes	fato
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Uma pessoa tem cônjuge	fato
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Uma pessoa tem ascendentes	fato
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Uma pessoa tem bens comuns	fato
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Uma pessoa tem bens particulares	fato

Figura 4.3.2: Visão parcial dos objetivos e fatos do sistema no domínio da Sucessão Legítima

A partir da lista de objetivos e fatos (Figura 4.3.2) o engenheiro do conhecimento especifica quais desses serão dados como entrada para a área “Representação dos Predicados em LPO”, com o intuito de traduzir os objetivos e fatos em linguagem natural para predicados em LPO. No contexto da Sucessão Legítima foi definido como objetivo geral “Calcular a herança de uma pessoa” e os principais objetivos específicos são: “Identificar os herdeiros de uma pessoa” e “Determinar a herança dos herdeiros”, os quais são dados como entrada para a primeira subárea “Identificação das entidades”.

Como definido na seção 3.3 a subárea “Identificação das entidades” visa definir todos os sujeitos e objetos implícitos ou explícitos a partir dos itens selecionados na atividade anterior. Então, o desenvolvedor especifica na ferramenta as entidades presentes em cada objetivo, como apresentado na Figura 4.3.3.

Representação dos Predicados em LPO							
	1	2	3	4	5	6	7
Identificação das entidades							
	Itens Selecionados			Entidades			
	Calcular a herança de uma pessoa			Herança, Pessoa			
	Identificar os herdeiros de uma pessoa			Herdeiros, Pessoa			
	Determinar a herança dos herdeiros			Herdeiros, Herança			
PRÓXIMA							

Figura 4.3.3: Entidades especificadas através da subárea “Identificação das entidades”

Tendo especificado as entidades, o próximo passo é a “Redefinição das entidades” o qual tem por objetivo verificar se aquilo identificado como entidade (Figura 4.3.3) é na verdade uma relação, como definido na seção 3.3. A Figura 4.3.4 apresenta a palavra “Herdeiros” que era definida como entidade sendo redefinida para “Pessoa”.

Representação dos Predicados em LPO							
	1	2	3	4	5	6	7
Redefinição das Entidades							
	Entidades			Redefinição das entidades			
	Herança, Pessoa			Redefina aqui as entidades			
	Herdeiros, Pessoa			Pessoa, Pessoa			
	Herdeiros, Herança			Pessoa, Herança			
ANTERIOR				PRÓXIMA			

Figura 4.3.4: Entidades redefinidas através da subárea “Redefinição das entidades”

Como definido na seção 3.3, tendo redefinidas as entidades, o próximo passo é a “Identificação dos relacionamentos” a qual tem o propósito de identificar os verbos presentes nos itens selecionados, pois eles representam os relacionamentos a serem extraídos. A Figura 4.3.5 apresenta os relacionamentos especificados na GAODTool pelo desenvolvedor.

Representação dos Predicados em LPO							
	1	2	3	4	5	6	7
Identificação dos Relacionamentos							
	Itens Seleccionados			Relacionamentos			
	Calcular a herança de uma pessoa			calcular			
	Identificar os herdeiros de uma pessoa			identificar			
	Determinar a herança dos herdeiros			determinar			
ANTERIOR				PRÓXIMA			

Figura 4.3.5: Relacionamentos especificados através da subárea “Identificação dos relacionamentos”

Em seguida, é necessário verificar se os relacionamentos identificados (Figura 4.3.5) são verbos transitivos, pois eles necessitam de um complemento para terem sentido. Por exemplo, o relacionamento “calcular” precisa de um complemento para ter sentido, logo, utiliza-se sua respectiva entidade “Herança” definida na Figura 4.3.3. Sendo assim, a Figura 4.3.6 apresenta os relacionamentos redefinidos pelo desenvolvedor, através da subárea “Redefinição dos relacionamentos”.

Representação dos Predicados em LPO							
	1	2	3	4	5	6	7
Redefinição dos Relacionamentos							
	Relacionamentos	Entidades/Palavras redefinidas		Redefinição dos relacionamentos			
	calcular	Pessoa,Herança		calcularHerança			
	identificar	Herdeiros,Pessoa		identificarHerdeiros			
	determinar	Herdeiros,Herança,Pessoa		determinarHerança			
ANTERIOR				PRÓXIMA			

Figura 4.3.6: Relacionamentos redefinidos através da subárea “Redefinição dos relacionamentos”.

Nesse momento é necessário que o desenvolvedor através da subárea “Definição da aridade” defina o número de entidades (Figura 4.3.4) envolvidas nos relacionamentos (Figura 4.3.6) identificados. Esta quantidade é determinada de acordo com o número de entidades identificadas em cada item seleccionado. A Figura 4.3.7 apresenta a aridade “2” definida pelo desenvolvedor para todos os predicados.

Representação dos Predicados em LPO									
	1 2 3 4 5 6 7								
Redefinição das Entidades do Predicado									
	<table border="1"> <thead> <tr> <th>Predicados</th> <th>Redefinição dos predicados</th> </tr> </thead> <tbody> <tr> <td>calcularHerança(Pessoa, Herança)</td> <td>calcularHerança(PessoaX, Herança)</td> </tr> <tr> <td>identificarHerdeiros(Pessoa, Pessoa)</td> <td>identificarHerdeiros(PessoaX, PessoaY)</td> </tr> <tr> <td>determinarHerança(Pessoa, Herança)</td> <td>determinarHerança(PessoaY, Herança)</td> </tr> </tbody> </table>	Predicados	Redefinição dos predicados	calcularHerança(Pessoa, Herança)	calcularHerança(PessoaX, Herança)	identificarHerdeiros(Pessoa, Pessoa)	identificarHerdeiros(PessoaX, PessoaY)	determinarHerança(Pessoa, Herança)	determinarHerança(PessoaY, Herança)
Predicados	Redefinição dos predicados								
calcularHerança(Pessoa, Herança)	calcularHerança(PessoaX, Herança)								
identificarHerdeiros(Pessoa, Pessoa)	identificarHerdeiros(PessoaX, PessoaY)								
determinarHerança(Pessoa, Herança)	determinarHerança(PessoaY, Herança)								
	<input type="button" value="ANTERIOR"/> <input type="button" value="FINALIZAR"/>								

Figura 4.3.9: Predicados redefinidos através da subárea “Redefinição das entidades dos predicados”.

Ao término do preenchimento da área “Representação dos Predicados em LPO”, os predicados construídos são dados como entrada para a próxima área “Especificação dos Axiomas em LPO”, a qual permite ao desenvolvedor especificar as regras que permitirão alcançar os objetivos do sistema.

A primeira subárea a ser desenvolvida é a “Definição da condição e conclusão” na qual o desenvolvedor informa os predicados que compõe a condição e a conclusão do axioma. A Figura 4.3.10 apresenta a conclusão definida pelo predicado “calcularHerança(PessoaX, Herança)” e a condição pelos predicados “identificarHerdeiros(PessoaX, PessoaY)” e “determinarHerança(PessoaY, Herança)”.

Especificação dos Axiomas em LPO									
	1 2 3 4								
Definição da condição e conclusão									
	<table border="1"> <thead> <tr> <th colspan="2">Predicados</th> </tr> </thead> <tbody> <tr> <td>Conclusão</td> <td>calcularHerança(PessoaX, Herança)</td> </tr> <tr> <td>Condição</td> <td>identificarHerdeiros(PessoaX, PessoaY)</td> </tr> <tr> <td>Condição</td> <td>determinarHerança(PessoaY, Herança)</td> </tr> </tbody> </table>	Predicados		Conclusão	calcularHerança(PessoaX, Herança)	Condição	identificarHerdeiros(PessoaX, PessoaY)	Condição	determinarHerança(PessoaY, Herança)
Predicados									
Conclusão	calcularHerança(PessoaX, Herança)								
Condição	identificarHerdeiros(PessoaX, PessoaY)								
Condição	determinarHerança(PessoaY, Herança)								
	<input type="button" value="PRÓXIMA"/>								

Figura 4.3.10: Predicados que compõem a condição e conclusão do axioma definidos através da subárea “Definição da condição e conclusão”.

Em seguida o desenvolvedor especifica os operadores booleanos que integram os predicados da condição do axioma, através da subárea “Definição dos operadores booleanos”. Os operadores utilizados são a conjunção representada pelo símbolo (\wedge) e a disjunção representada pelo símbolo (\vee). A Figura 4.3.11 apresenta a conjunção utilizada para unir os predicados

“identificarHerdeiros(PessoaX, PessoaY)” e “determinarHerança(PessoaY, Herança)”.

Especificação dos Axiomas em LPO	
	<div style="display: flex; justify-content: space-around;"> 1 2 3 4 </div>
Definição dos operadores booleanos	<p>OPERADORES BOOLEANOS</p> <p style="text-align: center;">^</p> <p style="text-align: center;">v</p>
	<p>CONDIÇÃO</p> <p>identificarHerdeiros(PessoaX, PessoaY)</p> <p style="text-align: center;">^</p> <p>determinarHerança(PessoaY, Herança)</p>
	<div style="display: flex; justify-content: space-around;"> ANTERIOR PRÓXIMA </div>

Figura 4.3.11: Predicados da condição unidos pela conjunção através da subárea “Definição dos operadores booleanos”.

Nesse momento o desenvolvedor define os quantificadores apropriados para associar as entidades presentes no axioma. Os quantificadores podem ser o universal (\forall) ou existencial (\exists). Para as entidade “PessoaX” , “PessoaY” e “Herança” é definido o quantificador existencial, conforme ilustrado na Figura 4.3.12.

Especificação dos Axiomas em LPO													
	<div style="display: flex; justify-content: space-around;"> 1 2 3 4 </div>												
Definição dos quantificadores	<table border="1"> <thead> <tr> <th colspan="2"></th> <th>Variáveis</th> </tr> </thead> <tbody> <tr> <td>Existencial (\exists)</td> <td style="text-align: center;">▼</td> <td>PessoaX</td> </tr> <tr> <td>Existencial (\exists)</td> <td style="text-align: center;">▼</td> <td>Herança</td> </tr> <tr> <td>Existencial (\exists)</td> <td style="text-align: center;">▼</td> <td>PessoaY</td> </tr> </tbody> </table>			Variáveis	Existencial (\exists)	▼	PessoaX	Existencial (\exists)	▼	Herança	Existencial (\exists)	▼	PessoaY
		Variáveis											
Existencial (\exists)	▼	PessoaX											
Existencial (\exists)	▼	Herança											
Existencial (\exists)	▼	PessoaY											
	<div style="display: flex; justify-content: space-around;"> ANTERIOR PRÓXIMA </div>												

Figura 4.3.12: Quantificador definido para cada variável dos predicados, através da subárea “Definição dos quantificadores”.

Ao término do preenchimento da área “Especificação dos Axiomas em LPO”, a subárea “Definição de implicação ou equivalência” permite ao desenvolvedor determinar se o axioma a ser criado é uma implicação ou uma equivalência. A implicação é utilizada quando a satisfação da condição leva a conclusão. A equivalência ocorre quando há uma simetria entre a condição e a conclusão. A Figura 4.3.13 apresenta a implicação definida pelo desenvolvedor.

Especificação dos Axiomas em LPO	
	<div style="display: flex; justify-content: space-around;"> 1 2 3 4 </div>
Definição de implicação ou equivalência	$\exists \text{ PessoaX}, \exists \text{ Herança}, \exists \text{ PessoaY}$ $\text{identificarHerdeiros}(\text{PessoaX}, \text{PessoaY}) \wedge \text{determinarHerança}(\text{PessoaY}, \text{Herança})$ <p style="text-align: center;">Implicação (\Rightarrow) <input type="button" value="v"/></p> $\text{calcularHerança}(\text{PessoaX}, \text{Herança})$ <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <input type="button" value="ANTERIOR"/> <input type="button" value="FINALIZAR"/> </div>

Figura 4.3.13: Definindo a implicação do axioma, através da subárea “Definição de implicação ou equivalência”.

Após o desenvolvedor ter definido os “predicados”, a “condição e conclusão”, os “operadores booleanos”, os “quantificadores” e a “implicação ou equivalência” o axioma “ $\exists \text{ Herança}, \exists \text{ PessoaX}, \exists \text{ PessoaY} \mid \text{identificarHerdeiros}(\text{PessoaX}, \text{PessoaY}) \wedge \text{determinarHerança}(\text{PessoaY}, \text{Herança}) \Rightarrow \text{calcularHerança}(\text{PessoaX}, \text{Herança})$ ” é criado e armazenado no local “Axiomas Desenvolvidos”, conforme apresentado na Figura 4.3.14.

Especificação dos Axiomas em LPO	
Axiomas Desenvolvidos	
$\otimes + \exists \text{ Herança}, \exists \text{ PessoaX}, \exists \text{ PessoaY} \mid \text{identificarHerdeiros}(\text{PessoaX}, \text{PessoaY}) \wedge \text{determinarHerança}(\text{PessoaY}, \text{Herança}) \Rightarrow \text{calcule}$	
<input type="button" value="FINALIZAR"/>	

Figura 4.3.14: Axioma desenvolvido após termino da área “Especificação dos Axiomas em LPO”.

Como definido na seção 3.4, as atividades da GAODT são executadas iterativamente, ou seja, um novo ciclo da técnica se inicia para criar um novo axioma. Para o desenvolvedor construir o novo axioma que satisfaça o predicado objetivo “ $\text{identificarHerdeiros}(\text{PessoaX}, \text{PessoaY})$ ” contido na condição do axioma desenvolvido (Figura 4.3.14), ele em consenso com o especialista de domínio selecionam na lista de objetivos e fatos (Figura 4.3.1) os objetivos: “Identificar os descendentes de uma pessoa”, “Identificar o cônjuge de uma pessoa”, “Identificar os ascendentes de uma pessoa” e “Identificar os colaterais de uma pessoa” como aqueles que satisfazem esse objetivo. A Figura 4.3.15 apresenta esses objetivos selecionados na lista de objetivos e fatos.

Seleção dos Objetivos e Fatos				
Lista de Objetivos e Fatos			Tipo	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Calcular a herança de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Identificar os herdeiros de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Determinar a herança dos herdeiros	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Verificar a existência de descendentes	objetivo
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Identificar os descendentes de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Verificar a existência do cônjuge	objetivo
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Identificar o cônjuge de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Verificar a existência de ascendentes	objetivo
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Identificar os ascendentes de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Verificar a existência dos colaterais	objetivo
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Identificar os colaterais de uma pessoa	objetivo
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Uma pessoa tem descendentes	fato
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Uma pessoa tem cônjuge	fato
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Uma pessoa tem ascendentes	fato
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Uma pessoa tem bens comuns	fato

Figura 4.3.15: Objetivos selecionados pelo desenvolvedor para criar um novo axioma em LPO.

Com os objetivos selecionados (Figura 4.3.15), todo o processo de representação em predicados em LPO na área “Representação dos Predicados em LPO” é realizado novamente. A Figura 4.3.16 apresenta todos os objetivos representados em forma de predicados.

Representação dos Predicados em LPO	
	1 2 3 4 5 6 7
Redefinição das Entidades do Predicado	
Predicados	Redefinição dos predicados
identificarDescendentes(Pessoa, Pessoa)	identificarDescendentes(PessoaX, PessoaY)
identificarConjuge(Pessoa, Pessoa)	identificarConjuge(PessoaX, PessoaY)
identificarAscendentes(Pessoa, Pessoa)	identificarAscendentes(PessoaX, PessoaY)
identificarColaterais(Pessoa, Pessoa)	identificarColaterais(PessoaX, PessoaY)
<input type="button" value="ANTERIOR"/>	<input type="button" value="FINALIZAR"/>

Figura 4.3.16: Predicados em LPO para formarem o novo axioma.

Os predicados desenvolvidos (Figura 4.3.16) são dados como entrada para área “Especificação dos Axiomas em LPO” para gerarem o novo axioma: “ \exists PessoaX, \exists PessoaY | identificarDescendentes(PessoaX, PessoaY) \vee identificarAscendentes(PessoaX, PessoaY) \vee

$\text{identificarC\u00f4njuge}(\text{PessoaX}, \text{PessoaY}) \vee \text{identificarColaterais}(\text{PessoaX}, \text{PessoaY}) \Rightarrow \text{identificarHerdeiros}(\text{PessoaX}, \text{PessoaY})$ ” apresentado na Figura 4.3.17 e um novo ciclo da t\u00e9cnica GAODT se inicia.

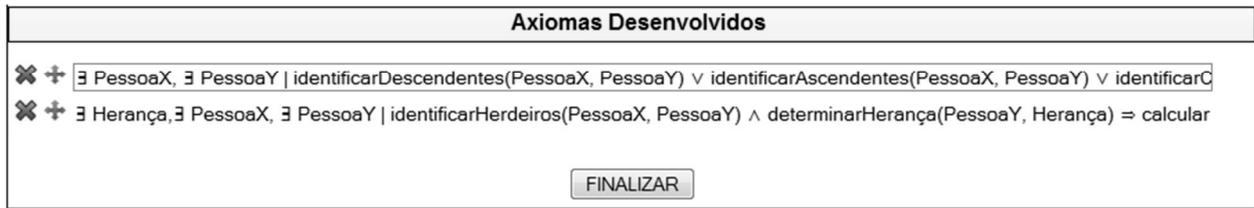


Figura 4.3.17: Novo axioma gerado e armazenado na lista de axiomas desenvolvidos.

O predicado “determinarHeran\u00e7a(PessoaY, Heran\u00e7a)” que tamb\u00e9m faz parte da condi\u00e7\u00e3o do axioma passa pelo mesmo processo de especifica\u00e7\u00e3o e representa\u00e7\u00e3o em LPO a qual o predicado “identificarHerdeiros(PessoaX, PessoaY)” foi submetido, gerando um novo axioma (Figura 4.3.18).

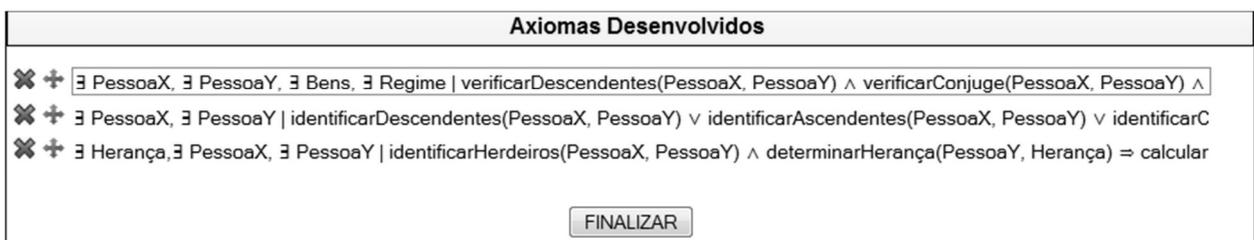


Figura 4.3.18: Axioma gerado e armazenado na lista de axiomas desenvolvidos.

O processo \u00e9 ent\u00e3o executado novamente para cada um dos subobjetivos, at\u00e9 que todos os objetivos da Figura 4.3.2 dados como entrada para a GAODTool tenham sido satisfeitos. A Figura 4.3.19 apresenta a vis\u00e3o parcial dos axiomas desenvolvidos na ferramenta. A lista completa dos axiomas encontra-se no Ap\u00eandice B.

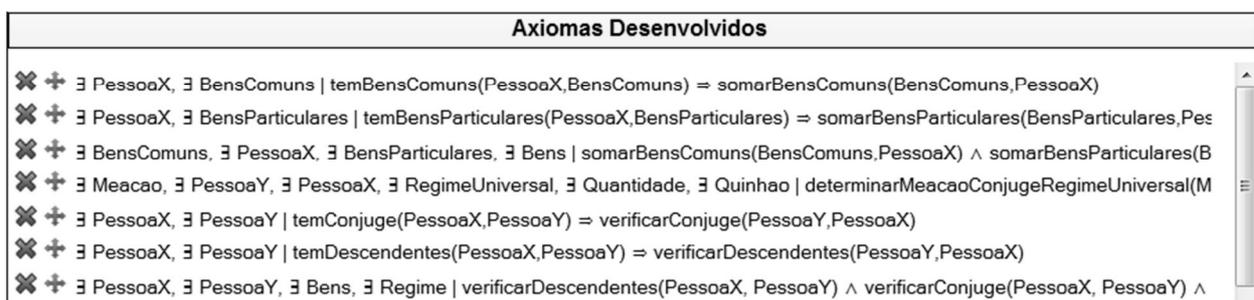


Figura 4.3.19: Lista parcial dos axiomas desenvolvidos no dom\u00ednio da Sucess\u00e3o Leg\u00edtima.

Em seguida, na \u00e1rea “Especifica\u00e7\u00e3o da Ontologia de Aplica\u00e7\u00e3o” \u00e9 realizada a extra\u00e7\u00e3o dos elementos constituintes dos axiomas contidos na lista de axiomas desenvolvidos (Figura 4.3.19).

A primeira subárea a ser utilizada é a “Tradução dos axiomas” que permite ao desenvolvedor converter automaticamente os axiomas desenvolvidos para regras expressas na linguagem RuleML [18]. A Figura 4.3.20 apresenta uma visão parcial dos axiomas traduzidos para RuleML. A lista completa dos axiomas encontra-se no Apêndice C.

```

<Implies>
  <Atom>
    <Rel>temBensComuns</Rel>
    <Var>PessoaX</Var>
    <Var>BensComuns</Var>
  </Atom>
  <Atom>
    <Rel>somarBensComuns</Rel>
    <Var>BensComuns</Var>
    <Var>PessoaX</Var>
  </Atom>
</Implies>
<Implies>
  <Atom>
    <Rel>temBensParticulares</Rel>
    <Var>PessoaX</Var>
    <Var>BensParticulares</Var>
  </Atom>
  <Atom>
    <Rel>somarBensParticulares</Rel>
    <Var>BensParticulares</Var>
    <Var>PessoaX</Var>
  </Atom>
</Implies>

```

Figura 4.3.20: Visão parcial dos axiomas representados em RuleML.

Tendo traduzidos os axiomas, é realizada a definição das classes através da extração das variáveis dos predicados presentes nos axiomas desenvolvidos (Figura 4.3.19). Por exemplo, dos predicados “identificarHerdeiros(PessoaX, PessoaY)” e calcularHerança(PessoaX, Herança) extrai-se as classes “Pessoa” e “Herança”. A Figura 4.3.21 apresenta as classes especificadas.

Especificação da Ontologia de Aplicação				
1 2 3 4 5				
Definição das Classes				
Variáveis				
<input checked="" type="checkbox"/> PessoaX	<input checked="" type="checkbox"/> PessoaY	<input type="checkbox"/> Quantidade	<input checked="" type="checkbox"/> PessoaB	<input checked="" type="checkbox"/> PessoaV
<input checked="" type="checkbox"/> BensComuns	<input checked="" type="checkbox"/> BensParticulares	<input checked="" type="checkbox"/> Bens	<input type="checkbox"/> RegimeUniversal	<input type="checkbox"/> RegimeParcial
<input type="checkbox"/> RegimeParticipacao	<input type="checkbox"/> RegimeSeparacaoVoluntaria	<input type="checkbox"/> RegimeSeparacaoObrigatoria	<input type="checkbox"/> Metade	<input type="checkbox"/> Meacao
<input type="checkbox"/> Quinhao	<input checked="" type="checkbox"/> PessoaZ	<input type="checkbox"/> Soma	<input type="checkbox"/> RegimeParticipacaoAquestos	<input checked="" type="checkbox"/> PessoaW
Classes				
Pessoa	BensComuns	BensParticulares	Bens	Heranca

Figura 4.3.21: Definição das classes da ontologia de aplicação “InheritanceLaw”.

Em seguida, é realizada a definição dos relacionamentos não taxonômicos a partir dos predicados na lista de axiomas desenvolvidos (Figura 4.3.19). Por exemplo, a relação não taxonômica “calcularHerança” define a relação entre as classes “Pessoa” e “Herança”. A Figura 4.3.22 apresenta uma visão parcial dos relacionamentos não taxonômicos definidos na subárea “Definição dos relacionamentos não taxonômicos”. A lista completa dos relacionamentos não taxonômicos encontra-se no Apêndice D.

Especificação da Ontologia de Aplicação			
1 2 3 4 5			
Definição dos Relacionamentos Não Taxonômicos			
	Domínio	Relação	Imagem
<input checked="" type="checkbox"/>	Pessoa ▾	identificarHerdeiros	Pessoa ▾
<input checked="" type="checkbox"/>	Pessoa ▾	determinarHeranca	Heranca ▾
<input checked="" type="checkbox"/>	Pessoa ▾	calcularHeranca	Heranca ▾
<input checked="" type="checkbox"/>	Pessoa ▾	determinarGenitor	Pessoa ▾
<input checked="" type="checkbox"/>	Pessoa ▾	determinarAvos	Pessoa ▾
<input checked="" type="checkbox"/>	Pessoa ▾	determinarBiSavos	Pessoa ▾

Figura 4.3.22: Visão parcial dos relacionamentos não taxonômicos da “InheritanceLaw”.

Nesse momento é realizada a definição dos relacionamentos taxonômicos baseado nas relações hierárquicas entre as classes definidas na subárea “Definição das classes” (Figura 4.3.21). Por exemplo, uma relação hierárquica encontrada ocorre entre as classes, “Bens”, “BensComuns” e “BensParticulares. A Figura 4.3.23 apresenta os relacionamentos taxonômicos definidos através da subárea “Definição dos relacionamentos taxonômicos”.

Especificação da Ontologia de Aplicação			
1 2 3 4 5			
Definição dos Relacionamentos Taxonômicos			
CLASSES ▾		Sub-Classe de	CLASSES ▾
<input type="button" value="ADICIONAR"/>			
	Classe		Classe
<input checked="" type="checkbox"/>	BensComuns	Sub-Classe de	Bens
<input checked="" type="checkbox"/>	BensParticulares	Sub-Classe de	Bens

Figura 4.3.23: Relacionamentos taxonômicos definidos.

Logo após, são definidas as propriedades através dos predicados contidos nos axiomas desenvolvidos (Figura 4.3.19). Por exemplo, uma propriedade definida é “determinarQuantidadeDescendentes” associada a classe “Pessoa”. A Figura 4.3.24 apresenta uma visão parcial das propriedades definidas na subárea “Definição das propriedades”. A lista completa das propriedades encontra-se no Apêndice E.

Especificação da Ontologia de Aplicação			
1 2 3 4 5			
Definição das Propriedades			
	Domínio	Propriedade	Imagem
<input checked="" type="checkbox"/>	BensComuns	somarBensComuns	float
<input checked="" type="checkbox"/>	BensParticulares	somarBensParticulares	float
<input checked="" type="checkbox"/>	Bens	somarBens	float
<input checked="" type="checkbox"/>	Pessoa	determinarQuinhaoDescendentesRegimeParcial	float
<input checked="" type="checkbox"/>	Pessoa	determinarMeacaoConjugeRegimeUniversal	float
<input checked="" type="checkbox"/>	Pessoa	determinarQuantidadeDescendentes	int
<input checked="" type="checkbox"/>	Pessoa	dividirQuantidadeDescendentes	float

Figura 4.3.24: Propriedades da ontologia.

Por fim, um arquivo OWL [27] contendo a ontologia de aplicação “InheritanceLaw” desenvolvida é fornecido. A Figura 4.3.25 apresenta a taxonomia e uma visão parcial das relações não taxonômicas e propriedades da ontologia de aplicação “InheritanceLaw” no ambiente gráfico Protégè [29]. Os axiomas não são apresentados, devido ao fato do Protégè não possuir um plugin para sua visualização. Entretanto, a lista completa desses axiomas encontra-se no Apêndice C.

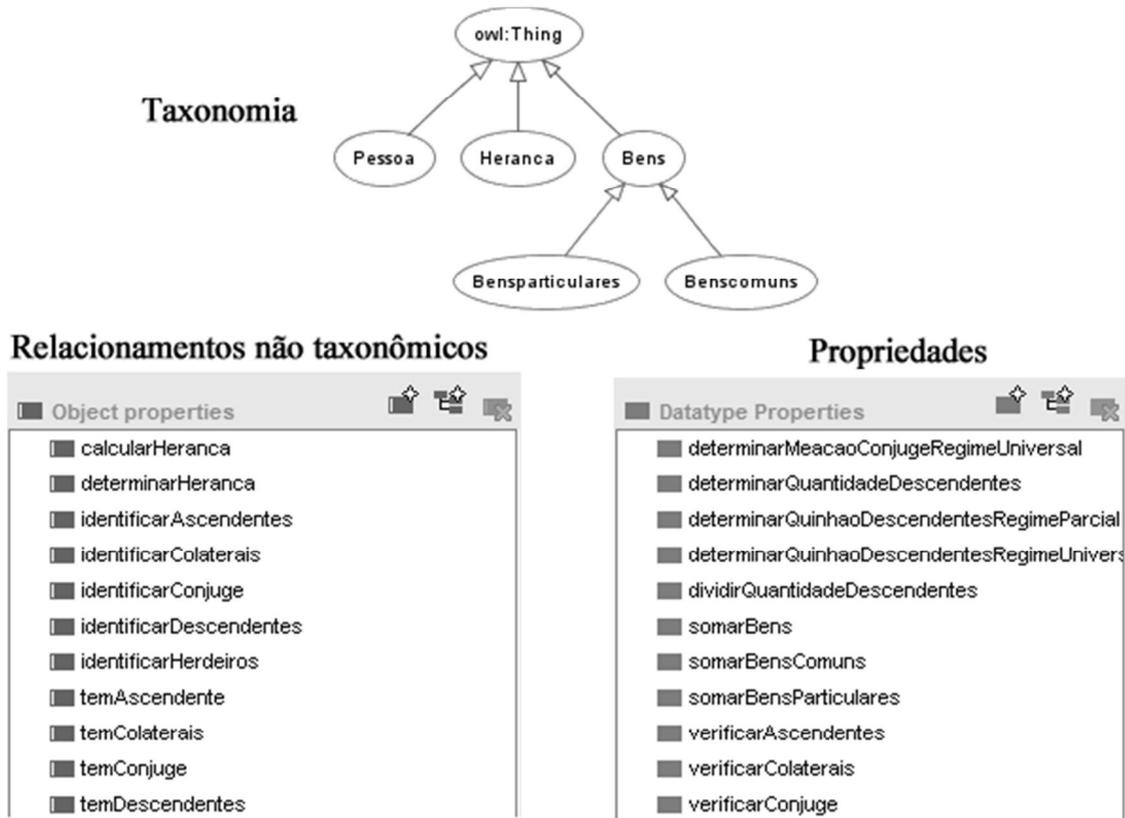


Figura 4.3.25: Taxonomia, relacionamentos não taxonômicos e propriedades da ontologia de aplicação “InheritanceLaw”.

4.4 Considerações Finais

Neste capítulo foi descrito o estudo de caso elaborado com o intuito de avaliar a técnica GAODT e a ferramenta GAODTool. Tal estudo de caso consistiu no desenvolvimento da ontologia de aplicação “InheritanceLaw” no domínio da Sucessão Legítima com o suporte da ferramenta GAODTool.

A partir desse estudo de caso, constatou-se a facilidade proporcionada ao engenheiro do conhecimento em construir ontologias de aplicação, devido a interface intuitiva e de fácil navegação que a GAODTool proporciona.

Outra característica da GAODTool é a automatização na tradução das regras em LPO para a linguagem RuleML, e a geração do arquivo OWL contendo a ontologia de aplicação desenvolvida.

5 CONCLUSÕES

Construir ontologias reusáveis é um processo custoso. Dentre os quatro tipos de ontologias definidos por Guarino [17], ontologias de aplicação são menos reusáveis uma vez que elas são desenvolvidas para aplicações de software específicas. Portanto, elas são geralmente mais rápidas e baratas de se desenvolver. Construir ontologias de aplicação e então generalizar seus elementos para ontologias de domínio e de tarefa é uma abordagem adequada para o desenvolvimento de artefatos reusáveis.

Neste trabalho foi apresentada a GAODT, uma técnica para a construção de ontologias de aplicação que utiliza um ciclo de desenvolvimento guiado pelos objetivos do sistema, facilitando assim a construção incremental de ontologias de aplicação.

A GAODTool é uma ferramenta semi-automatizada de interface intuitiva e fácil navegação que além de prover suporte as atividades da GAODT, automatiza a criação dos axiomas em RuleML e do arquivo OWL contendo a ontologia de aplicação desenvolvida, permitindo assim que o tempo de desenvolvimento seja consideravelmente reduzido.

Para avaliar a GAODT foi desenvolvido um estudo de caso, que consistiu na utilização da ferramenta GAODTool no desenvolvimento da “InheritanceLaw”, uma ontologia de aplicação no domínio da Sucessão Legítima que poderá ser utilizada em um SBC para apoio a tomada de decisões nesse domínio.

5.1 Resultados e contribuições da pesquisa

As principais contribuições desta pesquisa foram:

- Análise do estado da arte de algumas das mais bem documentadas técnicas para o desenvolvimento de ontologias.
- Uma análise comparativa das técnicas estudadas, ressaltando seus pontos fortes e deficiências.
- Desenvolvimento da técnica GAODT para a construção de ontologias de aplicação por meio de um ciclo de desenvolvimento guiado pelos objetivos do sistema.
- A GAODT proporciona ao desenvolvedor uma forma bem definida de traduzir o conhecimento expresso em linguagem natural para uma representação computacional.

- Desenvolvimento da ferramenta de software GAODTool permitindo assim que o tempo de desenvolvimento de ontologias de aplicação reduza consideravelmente.
- Aplicação da GAODTool no domínio da Sucessão Legítima para a construção da ontologia de aplicação “InheritanceLaw” que pode ser utilizada em um SBC para apoio a tomada de decisões.
- Maior qualidade no desenvolvimento de ontologias de aplicação, uma vez que a GAODT especifica uma forma sistematizada de traduzir o conhecimento expresso em linguagem natural para uma representação computacional. Já a GAODTool permite maior produtividade na construção de ontologias de aplicação, uma vez que possui uma interface de fácil navegação, automatiza a tradução dos axiomas em RuleML além de gerar o arquivo OWL contendo a ontologia de aplicação desenvolvida.

5.1.1 Publicações

- A publicação **“Uma Técnica Orientada por Objetivos para a Construção de Ontologias de Aplicação**, Luís Eduardo e Rosario Girardi, VIII Simpósio Brasileiro de Sistemas de Informação - SBSI, Ed. USP-Leste, pp. 13-24. São Paulo - Brasil. 16 a 18 de maio de 2012”, consistiu na primeira publicação sobre a GAODT, na qual descreveu-se como o seu ciclo orientado por objetivos facilita a construção de ontologias de aplicação.
- A publicação **“Building Application Ontologies through Knowledge System Goals**. Luís Eduardo e Rosario Girardi. KEOD. Barcelona, Espanha. 04 a 07 de outubro de 2012”, apresentou uma versão aprimorada da GAODT e as principais técnicas do estado da arte para a construção de ontologias.
- A publicação **“A Case Study on the Construction of Application Ontologies**. Luís Eduardo e Rosario Girardi. ITNG. Las Vegas, Nevada, USA. 15 a 17 de abril de 2013”, apresentou um estudo de caso que consistiu em utilizar a ferramenta GAODTool na construção de uma ontologia de aplicação no domínio da Sucessão Legítima.
- A publicação **“A Goal-Driven Technique for Building Application Ontologies**. Luís Eduardo e Rosario Girardi, Revista Brasileira de Sistemas de Informação - iSys, agosto de 2012”, apresentou em detalhes um estudo de caso que consistiu em utilizar

a ferramenta GAODTool na construção da InheritanceLaw, uma ontologia de aplicação no domínio do Direito Sucessório. **(Submetido)**

5.2 Trabalhos futuros

Vários trabalhos podem ser abordados a partir dos resultados obtidos neste:

- Aprimoramento da subatividade “Recuperação de ontologias de aplicação” através do desenvolvimento de uma técnica que permita o reuso de ontologias por meio de uma busca semântica em um repositório.
- Integração da técnica GAODT ao MADAE-Pro, um processo baseado no conhecimento para engenharia de domínio e de aplicações multiagente [7]. O objetivo é utilizar as ontologias de aplicação construídas com a GAODT, como a base de conhecimento dos agentes deliberativos desenvolvidos com este processo.
- Desenvolvimento de uma técnica que integre as regras em LPO desenvolvidas pela GAODT, com as regras em LPO geradas a partir da TAILP, uma técnica para a extração de axiomas de ontologias utilizando a programação em lógica indutiva [2]. O objetivo é aperfeiçoar o processo de construção de ontologias de aplicação, uma vez que TAILP pode construir regras não desenvolvidas pela GAODT, por se utilizar de uma ontologia já validada pelo especialista de domínio para construir essas regras.

REFERÊNCIAS

- [1] ARTIGO Nº 1.829. **Código Civil Brasileiro**. Legislação Federal, 2002.
- [2] BOAZ N. **Aquisição Automática de Axiomas de Ontologias utilizando Programação em Lógica Indutiva**. Trabalho de Iniciação Científica. Universidade Federal do Maranhão. 2012
- [3] BOLEY H. **POSL: An Integrated Positional-Slotted Language for Semantic Web Knowledge**. W3C, May 11, 2004.
- [4] CALIARI, FÁBIO M. **DERONTO: Método para Construção de Ontologias a partir de Diagramas Entidade-Relacionamento**. Dissertação de Mestrado na Universidade Tecnológica Federal do Paraná. Curitiba, 2007.
- [5] CHEN, P. **The Entity-Relationship Model - toward a unified view of data**, ACM Transactions on Database Systems, Vol. 1, nº 1, pp. 9-36, 1976.
- [6] CLAUDIA, A., FANIZZI, N., ESPOSITO, F. **A semantic similarity measure for expressive description logics**. In Proceedings of Italian Conference on Computational Logic, Roma, June, 2005.
- [7] COSTA, A. **MADAE-Pro Um processo baseado no conhecimento para Engenharia de Domínio e de Aplicações**. Dissertação de Mestrado em Engenharia de Eletricidade. Universidade Federal do Maranhão, 2009.
- [8] CSS. **Cascading Style Sheets**. Disponível em: <http://www.w3schools.com/css>. Acessado em agosto, 2012.
- [9] CUNNINGHAM, H., MAYNARD, D., BONTCHEVA, K., TABLAN, V. **GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications**, In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, July, 2002
- [10] DALE, R., MOISL, H., SOMERS, H. **Handbook of natural language processing**, CRC, 2000.
- [11] FERNÁNDEZ, M. GÓMEZ-PÉREZ, A. JURISTO, N. **METHONTOLOGY: From Ontological Art Towards Ontological Engineering**. Spring Symposium Series. Stanford. 33-40, April, 1997.
- [12] GIRARDI, R. **Guiding Ontology Learning and Population by Knowledge System Goals**. In: Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Ed. INSTIIC, France, Valence, p. 480 – 484, May, 2010

- [13] GÓMEZ-PÉREZ, A. ET AL. **Ontological Engineering: with examples from the areas of knowledge management, e-commerce and the semantic web.** London: Springer-Verlag, 2004.
- [14] GONÇALVES, C. **Direito Civil brasileiro: Direito das Sucessões**, volume VII São Paulo, Ed. Saraiva, 2009.
- [15] GRUBER, T. **Toward principles for the design of ontologies used for knowledge sharing.** International Journal of Human-Computer Studies, Vol 5, p. 907-928, 1995.
- [16] GRÜNINGER M, FOX MS. **Methodology for the design and evaluation of ontologies.** In: IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing. Montreal, Canada, 1995.
- [17] GUARINO, N. **Formal Ontology and Information Systems**, Proceedings of FOIS'98, Trento, Italy. Amsterdam, IOS Press, pp. 3-15, 6-8 June, 1998.
- [18] HAROLD B. **The Rule Markup Language: RDF-XML Data Model, XML Schema Hierarchy, and XSL Transformations.** In Proc. 14th International Conference on Applications of Prolog (INAP2001). The University of Tokyo, LNAI 2543, Springer-Verlag, October 2002.
- [19] HORROCKS I, PATEL-SCHNEIDER PF, BOLEY H, TABELT S, GROSOFF B, DEAN M. **SWRL: A Semantic Web Rule Language Combining OWL and RuleML.** W3C, May 21, 2004.
- [20] JARGAS, A. **Expressões Regulares: uma abordagem divertida.** Novatec, 2001.
- [21] JAVA. **Linguagem de programação JAVA.** Disponível em: <http://www.oracle.com/technetwork/java/index.html>. Acessado em julho, 2012.
- [22] JAVASCRIPT. **Scripting Language of the Web.** Disponível em: <http://www.w3schools.com/js/default.asp>. Acessado em agosto, 2012.
- [23] JEFFREY E. F. **Mastering Regular Expressions.** O'Reilly Media, 3rd Edition, 2006.
- [24] LEE, W., SHAH, N., SUNDLASS, K., MUSEN, M. **Comparison of ontology-based semantic-similarity measures.** AMIA Symposium, p. 384–388, 2008.
- [25] NECHES, R., FIKES, R., FININ, T., GRUBER, T., PATIL, R., SENATOR, T. AND SWARTOUT, W. R. **Enabling technology for knowledge sharing.** AI Magazine, 12(3):16-36, 1991.
- [26] NOY, N. F., MCGUINNESS, D. L. **Ontology Development 101: A Guide to Creating Your First Ontology.** Disponível em: <http://www.ksl.stanford.edu/people/dlm/papers/>

ontology-tutorial-noy-mcguinness.pdf, acessado em outubro, 2011.

- [27] OWL. **Web Ontology Language Overview**. Disponível em: <http://www.w3.org/TR/owl-features>, acessado em julho, 2012.
- [28] PHP. **PHP: Hypertext Preprocessor**. Disponível em: <http://www.php.net>. Acessado em agosto, 2012.
- [29] PROTÉGÉ. **Ontology Editor and Knowledge Acquisition System**. Disponível em: <http://protege.stanford.edu>, acessado em julho, 2012.
- [30] RUSSEL, S., NORVIG, P. **Inteligência Artificial**. Rio de Janeiro: Elsevier, 2004.
- [31] SOMMERVILLE, Ian. **Engenharia de Software**. 6. Ed. São Paulo: Addison Wesley, 2002.
- [32] SPELL, P. **Ontologies for very large knowledge bases in materials science: a case study**. Towards Very Large Knowledge Bases - Knowledge Building and Knowledge Sharing, pages 73-83. IOS Press, Amsterdam, 1995.
- [33] SPELL, P. **Scalability of the performance of knowledge representation systems**. In Towards Very Large Knowledge Bases – Knowledge Building and Knowledge Sharing, pages 173-183. IOS Press, Amsterdam, 1995.
- [34] SWOOGLE. **Semantic Web Search**. Disponível em: <http://swoogle.umbc.edu>. Acessado em outubro, 2012.
- [35] USCHOLD M, KING M. **Towards a Methodology for Building Ontologies**. In: IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing. Montreal, Canada, 1995.
- [36] XHTML. **Extensible Hypertext Markup Language**. Disponível em: http://www.w3schools.com/html/html_xhtml.asp. Acessado em agosto, 2012.

APÊNDICES

APÊNDICE A: Objetivos e fatos de um SBC no domínio da Sucessão Legítima.

Objetivos e Fatos do Sistema	Tipo
Calcular a herança de uma pessoa	Objetivo
Determinar os avós de uma pessoa	Objetivo
Determinar os bens de uma pessoa para o seu companheiro	Objetivo
Determinar os bens de uma pessoa para o seu cônjuge	Objetivo
Determinar os bisavós de uma pessoa	Objetivo
Determinar o genitor de uma pessoa	Objetivo
Determinar a herança dos herdeiros	Objetivo
Determinar a meação do cônjuge de uma pessoa no regime parcial sobre os bens comuns	Objetivo
Determinar a meação do cônjuge de uma pessoa no regime de participação nos aquestos sobre os bens comuns	Objetivo
Determinar a meação do cônjuge de uma pessoa no regime de participação nos aquestos sobre os bens comuns	Objetivo
Determinar a meação do cônjuge de uma pessoa no regime universal dos aquestos sobre os bens comuns	Objetivo
Determinar a quantidade de ascendentes de uma pessoa	Objetivo
Determinar a quantidade de ascendentes de primeiro grau de uma pessoa	Objetivo
Determinar a quantidade de ascendentes de segundo grau de uma pessoa	Objetivo
Determinar a quantidade de ascendentes de terceiro grau de uma pessoa	Objetivo
Determinar a quantidade de colaterais de uma pessoa	Objetivo
Determinar a quantidade de colaterais germanos de uma pessoa	Objetivo
Determinar a quantidade de colaterais unilaterais de uma pessoa	Objetivo
Determinar a quantidade de descendentes de uma pessoa	Objetivo
Determinar a quantidade de filhos comuns de uma pessoa	Objetivo
Determinar a quantidade de filhos incomuns de uma pessoa	Objetivo
Determinar o quinhão dos ascendentes em concorrência com o companheiro de uma pessoa	Objetivo
Determinar o quinhão dos ascendentes de primeiro grau em concorrência com o companheiro de uma pessoa	Objetivo
Determinar o quinhão dos ascendentes de segundo grau em concorrência com o companheiro de uma pessoa	Objetivo
Determinar o quinhão dos ascendentes de terceiro grau em concorrência com o companheiro de uma pessoa	Objetivo
Determinar o quinhão dos colaterais em concorrência com o companheiro de uma pessoa	Objetivo
Determinar o quinhão dos colaterais germanos de uma pessoa	Objetivo
Determinar o quinhão do companheiro em concorrência com os seus filhos comuns sobre os bens comuns de uma pessoa	Objetivo
Determinar o quinhão dos ascendentes de primeiro grau de uma pessoa	Objetivo
Determinar o quinhão dos ascendentes de segundo grau de uma pessoa	Objetivo
Determinar o quinhão dos ascendentes de terceiro grau de uma pessoa	Objetivo
Determinar o quinhão do cônjuge em concorrência com o companheiro de uma pessoa	Objetivo
Determinar o quinhão do cônjuge em concorrência com os descendentes no regime parcial sobre os bens particulares de uma pessoa	Objetivo
Determinar o quinhão do cônjuge em concorrência com os descendentes no regime de participação nos aquestos sobre os bens particulares de uma pessoa	Objetivo
Determinar o quinhão dos descendentes em concorrência com o cônjuge de uma pessoa no regime de separação voluntária	Objetivo
Determinar o quinhão dos descendentes no regime parcial sobre os bens comuns	Objetivo

Determinar o quinhão dos descendentes no regime de participação nos aquestos sobre os bens comuns de uma pessoa	Objetivo
Determinar o quinhão dos descendentes no regime de separação obrigatória sobre os bens comuns de uma pessoa	Objetivo
Determinar o quinhão dos descendentes no regime de separação obrigatória sobre os bens particulares de uma pessoa	Objetivo
Determinar o quinhão dos descendentes no regime universal	Objetivo
Determinar o quinhão dos filhos comuns em concorrência com o companheiro sobre os bens comuns de uma pessoa	Objetivo
Determinar o quinhão dos filhos comuns em concorrência com o companheiro sobre os bens particulares de uma pessoa	Objetivo
Determinar o quinhão dos filhos comuns em concorrência com o companheiro sobre os bens comuns de uma pessoa	Objetivo
Determinar o quinhão dos filhos incomuns em concorrência com o companheiro sobre os bens particulares de uma pessoa	Objetivo
Determinar a quota do companheiro em concorrência com os ascendentes, cônjuge ou colaterais de uma pessoa	Objetivo
Determinar a quota do companheiro em concorrência com os filhos incomuns sobre os bens comuns de uma pessoa	Objetivo
Dividir os bens de uma pessoa para os ascendentes e o cônjuge	Objetivo
Dividir os bens comuns de uma pessoa para os filhos comuns em concorrência com o companheiro	Objetivo
Dividir os bens comuns pela metade	Objetivo
Dividir os bens comuns pela quantidade de filhos incomuns de uma pessoa	Objetivo
Dividir os bens de uma pessoa pela metade	Objetivo
Dividir os bens particulares pela quantidade de descendentes	Objetivo
Dividir os bens particulares pela quantidade de filhos comuns	Objetivo
Dividir os bens particulares pela quantidade de filhos incomuns	Objetivo
Dividir os bens particulares de uma pessoa pela soma do número de descendentes e cônjuge	Objetivo
Dividir os bens de uma pessoa pela quantidade de colaterais germanos	Objetivo
Dividir os bens de uma pessoa pela quantidade de descendentes e cônjuge	Objetivo
Dividir os bens subtraídos pela quantidade de ascendentes de uma pessoa	Objetivo
Dividir os bens subtraídos pela quantidade de colaterais de uma pessoa	Objetivo
Dividir a meação do cônjuge pela quantidade de descendentes	Objetivo
Dividir pela metade os bens de uma pessoa	Objetivo
Dividir pela metade os bens do cônjuge	Objetivo
Dividir a metade dos bens de uma pessoa pela quantidade de ascendentes	Objetivo
Dividir a metade do quinhão dos filhos comuns entre os filhos incomuns	Objetivo
Identificar os herdeiros de uma pessoa	Objetivo
Obter a terça parte dos bens de uma pessoa	Objetivo
Partilhar os bens de uma pessoa entre os ascendentes de primeiro grau e o cônjuge	Objetivo
Partilhar os bens de uma pessoa entre os ascendentes de segundo grau e o cônjuge	Objetivo
Partilhar os bens de uma pessoa entre os ascendentes de terceiro grau e o cônjuge	Objetivo
Partilhar os bens de uma pessoa entre os colaterais germanos	Objetivo
Partilhar os bens de uma pessoa para o cônjuge	Objetivo
Partilhar os bens de uma pessoa para os descendentes e o cônjuge sobre o regime parcial	Objetivo
Partilhar os bens de uma pessoa para os descendentes e o cônjuge sobre o regime de participação nos aquestos	Objetivo
Partilhar os bens de uma pessoa para os descendentes e o cônjuge sobre o regime de separação obrigatória	Objetivo

Partilhar os bens de uma pessoa para os descendentes e o cônjuge sobre o regime de separação voluntária	Objetivo
Partilhar os bens de uma pessoa para os descendentes e o cônjuge sobre o regime de participação nos aquestos	Objetivo
Partilhar os bens de uma pessoa para o companheiro em concorrência com os ascendentes, cônjuge ou colaterais	Objetivo
Partilhar os bens de uma pessoa para o companheiro em concorrência com os filhos comuns	Objetivo
Partilhar os bens de uma pessoa para o companheiro em concorrência com os filhos incomuns	Objetivo
Somar os bens de uma pessoa	Objetivo
Somar os bens comuns de uma pessoa	Objetivo
Somar os bens particulares de uma pessoa	Objetivo
Somar a quantidade de ascendentes com o cônjuge de uma pessoa	Objetivo
Somar a quantidade de descendentes com o cônjuge de uma pessoa	Objetivo
Somar a quantidade de filhos comuns com o cônjuge de uma pessoa	Objetivo
Somar os bens comuns com os bens particulares	Objetivo
Subtrair dos bens de uma pessoa a quota referente ao companheiro	Objetivo
Armazenar os ascendentes de uma pessoa	Objetivo
Armazenar o cônjuge de uma pessoa	Objetivo
Armazenar os descendentes de uma pessoa	Objetivo
Armazenar os colaterais de uma pessoa	Objetivo
Uma pessoa tem pai	Fato
Uma pessoa tem mãe	Fato
Uma pessoa tem avô	Fato
Uma pessoa tem avó	Fato
Uma pessoa tem bisavô	Fato
Uma pessoa tem bisavó	Fato
Uma pessoa tem descendentes	Fato
Uma pessoa tem ascendentes	Fato
Uma pessoa tem cônjuge	Fato
Uma pessoa tem regime de bens	Fato
Uma pessoa tem regime universal	Fato
Uma pessoa tem regime parcial	Fato
Uma pessoa tem regime de participação nos aquestos	Fato
Uma pessoa tem regime de separação obrigatória	Fato
Uma pessoa tem regime de separação voluntária	Fato
Uma pessoa tem bens comuns	Fato
Uma pessoa tem bens particulares	Fato
Uma pessoa tem colaterais	Fato
Uma pessoa tem colaterais germanos	Fato
Uma pessoa tem colaterais unilaterais	Fato
Uma pessoa tem filhos comuns	Fato
Uma pessoa tem filhos incomuns	Fato
Uma pessoa tem companheiro	Fato
Verificar a existência de ascendentes de uma pessoa	Objetivo
Verificar a existência dos avôs de uma pessoa	Objetivo
Verificar a existência dos avós de uma pessoa	Objetivo
Verificar a existência dos bisavôs de uma pessoa	Objetivo
Verificar a existência das bisavós de uma pessoa	Objetivo
Verificar a existência do cônjuge de uma pessoa	Objetivo

Verificar a existência dos descendentes de uma pessoa	Objetivo
Verificar a existência do regime parcial de uma pessoa	Objetivo
Verificar a existência do regime universal de uma pessoa	Objetivo
Verificar a existência dos ascendentes de primeiro grau de uma pessoa	Objetivo
Verificar a existência dos ascendentes de segundo grau de uma pessoa	Objetivo
Verificar a existência dos ascendentes de terceiro grau de uma pessoa	Objetivo
Verificar a existência dos colaterais de uma pessoa	Objetivo
Verificar a existência dos colaterais germanos de uma pessoa	Objetivo
Verificar a existência do companheiro de uma pessoa	Objetivo
Verificar a existência do cônjuge de uma pessoa	Objetivo
Verificar a existência dos filhos comuns de uma pessoa	Objetivo
Verificar a existência dos filhos incomuns de uma pessoa	Objetivo
Verificar a existência do regime de participação nos aquestos de uma pessoa	Objetivo
Verificar a existência do regime de separação obrigatória de uma pessoa	Objetivo
Verificar a existência do regime de separação voluntária de uma pessoa	Objetivo

APÊNDICE B: Axiomas desenvolvidos no domínio da Sucessão Legítima.

\exists PessoaX, \exists PessoaY | temPai(PessoaX,PessoaY) \Rightarrow determinarGenitor(PessoaX,PessoaY)

\exists PessoaX, \exists PessoaY | temMae(PessoaX,PessoaY) \Rightarrow determinarGenitor(PessoaX,PessoaY)

\exists PessoaX, \exists PessoaY | temAvo(PessoaX,PessoaY) \Rightarrow determinarAvos(PessoaX,PessoaY)

\exists PessoaX, \exists PessoaY | temAvoh(PessoaX,PessoaY) \Rightarrow determinarAvos(PessoaX,PessoaY)

\exists PessoaX, \exists PessoaY | temBiSavo(PessoaX,PessoaY) \Rightarrow determinarBiSavos(PessoaX,PessoaY)

\exists PessoaX, \exists PessoaY | temBiSavoh(PessoaX,PessoaY) \Rightarrow determinarBiSavos(PessoaX,PessoaY)

\exists PessoaX, \exists PessoaY, \exists Quantidade | temDescendentes(PessoaX,PessoaY) \Rightarrow determinarQuantidadeDescendentes(Quantidade,PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY, \exists Quantidade | temAscendentes(PessoaX,PessoaY) \Rightarrow determinarQuantidadeAscendentes(Quantidade,PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY, \exists Quantidade | determinarGenitor(PessoaX,PessoaY) \Rightarrow determinarQuantidadeAscendentesPrimeiroGrau(Quantidade,PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY, \exists Quantidade, \exists PessoaB | determinarAvos(PessoaX,PessoaY) \Rightarrow determinarQuantidadeAscendentesSegundoGrau(Quantidade,PessoaB,PessoaX)

\exists PessoaX, \exists PessoaY, \exists Quantidade, \exists PessoaV | determinarBiSavos(PessoaX,PessoaY) \Rightarrow determinarQuantidadeAscendentesTerceiroGrau(Quantidade,PessoaV,PessoaX)

\exists PessoaX, \exists PessoaY, \exists Quantidade | temColaterais(PessoaX,PessoaY) \Rightarrow determinarQuantidadeColaterais(Quantidade,PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY, \exists Quantidade | temColateraisGermanos(PessoaX,PessoaY) \Rightarrow determinarQuantidadeColateraisGermanos(Quantidade,PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY, \exists Quantidade | temColateraisUnilaterais(PessoaX,PessoaY) \Rightarrow determinarQuantidadeColateraisUnilaterais(Quantidade,PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY, \exists Quantidade, \exists PessoaV | temFilhoComum(PessoaX,PessoaY) \Rightarrow determinarQuantidadeFilhosComuns(Quantidade,PessoaV,PessoaX)

\exists PessoaX, \exists PessoaY, \exists Quantidade, \exists PessoaV | temFilhoIncomum(PessoaX,PessoaY) \Rightarrow determinarQuantidadeFilhosIncomuns(Quantidade,PessoaV,PessoaX)

\exists PessoaX, \exists BensComuns | temBensComuns(PessoaX,BensComuns) \Rightarrow somarBensComuns(BensComuns,PessoaX)

\exists PessoaX, \exists BensParticulares | temBensParticulares(PessoaX,BensParticulares) \Rightarrow somarBensParticulares(BensParticulares,PessoaX)

\exists BensComuns, \exists PessoaX, \exists BensParticulares, \exists Bens | somarBensComuns(BensComuns,PessoaX) \wedge somarBensParticulares(BensParticulares,PessoaX) \wedge somarTodosBensComunsBensParticulares(BensComuns,BensParticulares) \Rightarrow somarBens(Bens,PessoaX)

\exists PessoaX, \exists PessoaY | temDescendentes(PessoaX,PessoaY) \Rightarrow verificarDescendentes(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | temConjuge(PessoaX,PessoaY) \Rightarrow verificarConjuge(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | temAscendentes(PessoaX,PessoaY) \Rightarrow verificarAscendentes(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | determinarGenitor(PessoaX,PessoaY) \Rightarrow verificarAscendentesPrimeiroGrau(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | determinarAvos(PessoaX,PessoaY) \Rightarrow verificarAscendentesSegundoGrau(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | determinarBiSavos(PessoaX,PessoaY) \Rightarrow verificarAscendentesTerceiroGrau(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | temColateraisGermanos(PessoaX,PessoaY) \Rightarrow verificarColateraisGermanos(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | temFilhoComum(PessoaX,PessoaY) \Rightarrow verificarFilhosComuns(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | temFilhoIncomum(PessoaX,PessoaY) \Rightarrow verificarFilhosIncomuns(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | temColaterais(PessoaX,PessoaY) \Rightarrow verificarColaterais(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | temCompanheiro(PessoaX,PessoaY) \Rightarrow verificarCompanheiro(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | temAvo(PessoaX,PessoaY) \Rightarrow verificarAvo(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | temAvoh(PessoaX,PessoaY) \Rightarrow verificarAvoh(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | temBiSavo(PessoaX,PessoaY) \Rightarrow verificarBiSavo(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | temBiSavoh(PessoaX,PessoaY) \Rightarrow verificarBiSavoh(PessoaY,PessoaX)

\exists PessoaX, \exists RegimeUniversal | temRegimeUniversal(PessoaX,RegimeUniversal) \Rightarrow verificarRegimeUniversal(PessoaY,PessoaX)

\exists PessoaX, \exists RegimeParcial | temRegimeParcial(PessoaX,RegimeParcial) \Rightarrow verificarRegimeParcial(PessoaY,PessoaX)

\exists PessoaX, \exists RegimeParticipacao | temRegimeParticipacaoAquestos(PessoaX,RegimeParticipacao) \Rightarrow verificarRegimeParticipacaoAquestos(RegimeParticipacaoAquestos,PessoaX)

\exists PessoaX, \exists RegimeSeparacaoVoluntaria | temRegimeSeparacaoVoluntaria(PessoaX,RegimeSeparacaoVoluntaria) \Rightarrow verificarRegimeSeparacaoVoluntaria(RegimeSeparacaoVoluntaria,PessoaX)

\exists PessoaX, \exists RegimeSeparacaoObrigatoria | temRegimeSeparacaoObrigatoria(PessoaX,RegimeSeparacaoObrigatoria) \Rightarrow
verificarRegimeSeparacaoObrigatoria(RegimeSeparacaoObrigatoria,PessoaX)

\exists Bens, \exists PessoaX, \exists Metade, \exists Meacao, \exists PessoaY, \exists RegimeUniversal | somarBens(Bens,PessoaX) \wedge dividirBensMetade(Bens,Metade) \Rightarrow
determinarMeacaoConjugeRegimeUniversal(Meacao,PessoaY,PessoaX,RegimeUniversal)

\exists Meacao, \exists PessoaY, \exists PessoaX, \exists RegimeUniversal, \exists Quantidade, \exists Quinhao | determinarMeacaoConjugeRegimeUniversal(Meacao,PessoaY,PessoaX,RegimeUniversal) \wedge
determinarQuantidadeDescendentes(Quantidade,PessoaY,PessoaX) \wedge dividirMeacaoDescendentes(Meacao,Quantidade) \Rightarrow
determinarQuinhaoDescendentesRegimeUniversal(Quinhao,PessoaY,PessoaX,RegimeUniversal)

\exists PessoaX, \exists PessoaY, \exists Meacao, \exists RegimeUniversal, \exists Quinhao, \exists Bens, \exists PessoaZ | temConjuge(PessoaX,PessoaY) \wedge
determinarMeacaoConjugeRegimeUniversal(Meacao,PessoaY,PessoaX,RegimeUniversal) \wedge temDescendentes(PessoaX,PessoaY) \wedge
determinarQuinhaoDescendentesRegimeUniversal(Quinhao,PessoaY,PessoaX,RegimeUniversal) \Rightarrow
partilharBensDescendentesConjugeRegimeUniversal(Bens,PessoaX,PessoaY,PessoaZ,RegimeUniversal)

\exists BensComuns, \exists PessoaX, \exists Metade, \exists Meacao, \exists PessoaY, \exists RegimeParcial | somarBensComuns(BensComuns,PessoaX) \wedge dividirBensComunsMetade(BensComuns,Metade) \Rightarrow
determinarMeacaoConjugeRegimeParcialBensComuns(Meacao,PessoaY,PessoaX,RegimeParcial,BensComuns)

\exists BensParticulares, \exists PessoaX, \exists Quantidade, \exists PessoaY, \exists Soma, \exists PessoaZ, \exists Quinhao, \exists RegimeParcial | somarBensParticulares(BensParticulares,PessoaX) \wedge
determinarQuantidadeDescendentes(Quantidade,PessoaY,PessoaX) \wedge somarQuantidadeDescendentesConjuge(Quantidade,PessoaY) \wedge
dividirBensParticularesSomaDescendentesConjuge(BensParticulares,Soma,PessoaY,PessoaZ) \Rightarrow
determinarQuinhaoConjugeDescendentesRegimeParcialBensParticulares(Quinhao,PessoaY,PessoaZ,RegimeParcial,BensParticulares,PessoaX)

\exists Meacao, \exists PessoaY, \exists PessoaX, \exists RegimeParcial, \exists BensComuns, \exists Quantidade, \exists Quinhao |
determinarMeacaoConjugeRegimeParcialBensComuns(Meacao,PessoaY,PessoaX,RegimeParcial,BensComuns) \wedge determinarQuantidadeDescendentes(Quantidade,PessoaY,PessoaX) \wedge
dividirMeacaoDescendentes(Meacao,Quantidade) \Rightarrow determinarQuinhaoDescendentesRegimeParcialBensComuns(Quinhao,PessoaY,PessoaX,RegimeParcial,BensComuns)

\exists PessoaX, \exists PessoaY, \exists Quinhao, \exists PessoaZ, \exists RegimeParcial, \exists BensParticulares, \exists Meacao, \exists BensComuns, \exists Bens | temDescendentes(PessoaX,PessoaY) \wedge
determinarQuinhaoConjugeDescendentesRegimeParcialBensParticulares(Quinhao,PessoaY,PessoaZ,RegimeParcial,BensParticulares,PessoaX) \wedge temConjuge(PessoaX,PessoaY) \wedge
determinarMeacaoConjugeRegimeParcialBensComuns(Meacao,PessoaY,PessoaX,RegimeParcial,BensComuns) \wedge temDescendentes(PessoaX,PessoaY) \wedge
determinarQuinhaoDescendentesRegimeParcialBensComuns(Quinhao,PessoaY,PessoaX,RegimeParcial,BensComuns) \Rightarrow
partilharBensDescendentesConjugeRegimeParcial(Bens,PessoaX,PessoaY,PessoaZ,RegimeParcial)

\exists BensComuns, \exists PessoaX, \exists Metade, \exists Meacao, \exists PessoaY, \exists RegimeParticipacaoAquestos | somarBensComuns(BensComuns,PessoaX) \wedge dividirBensComunsMetade(BensComuns,Metade) \Rightarrow
determinarMeacaoConjugeRegimeParticipacaoAquestosBensComuns(Meacao,PessoaY,PessoaX,RegimeParticipacaoAquestos,BensComuns)

\exists BensParticulares, \exists PessoaX, \exists Quantidade, \exists PessoaY, \exists Soma, \exists PessoaZ, \exists Quinhao, \exists RegimeParticipacaoAquestos | somarBensParticulares(BensParticulares,PessoaX) \wedge
determinarQuantidadeDescendentes(Quantidade,PessoaY,PessoaX) \wedge somarQuantidadeDescendentesConjuge(Quantidade,PessoaY) \wedge
dividirBensParticularesSomaDescendentesConjuge(BensParticulares,Soma,PessoaY,PessoaZ) \Rightarrow
determinarQuinhaoConjugeDescendentesRegimeParticipacaoAquestosBensParticulares(Quinhao,PessoaY,PessoaZ,RegimeParticipacaoAquestos,BensParticulares,PessoaX)

\exists Meacao, \exists PessoaY, \exists PessoaX, \exists RegimeParticipacaoAquestos, \exists BensComuns, \exists Quantidade, \exists Quinhao |
determinarMeacaoConjugeRegimeParticipacaoAquestosBensComuns(Meacao,PessoaY,PessoaX,RegimeParticipacaoAquestos,BensComuns) \wedge
determinarQuantidadeDescendentes(Quantidade,PessoaY,PessoaX) \wedge dividirMeacaoDescendentes(Meacao,Quantidade) \Rightarrow
determinarQuinhaoDescendentesRegimeParticipacaoAquestosBensComuns(Quinhao,PessoaY,PessoaX,RegimeParticipacaoAquestos,BensComuns)

\exists PessoaX, \exists PessoaY, \exists Quinhao, \exists PessoaZ, \exists RegimeParticipacaoAquestos, \exists BensParticulares, \exists Meacao, \exists BensComuns, \exists Bens | temDescendentes(PessoaX,PessoaY) \wedge
determinarQuinhaoConjugeDescendentesRegimeParticipacaoAquestosBensParticulares(Quinhao,PessoaY,PessoaZ,RegimeParticipacaoAquestos,BensParticulares,PessoaX) \wedge
temConjuge(PessoaX,PessoaY) \wedge determinarMeacaoConjugeRegimeParticipacaoAquestosBensComuns(Meacao,PessoaY,PessoaX,RegimeParticipacaoAquestos,BensComuns) \wedge
temDescendentes(PessoaX,PessoaY) \wedge determinarQuinhaoDescendentesRegimeParticipacaoAquestosBensComuns(Quinhao,PessoaY,PessoaX,RegimeParticipacaoAquestos,BensComuns) \Rightarrow
partilharBensDescendentesConjugeRegimeParticipacaoAquestos(Bens,PessoaX,PessoaY,PessoaZ,RegimeParticipacaoAquestos)

\exists BensComuns, \exists PessoaX, \exists Metade, \exists Meacao, \exists PessoaY, \exists RegimeSeparacaoObrigatoria | somarBensComuns(BensComuns,PessoaX) \wedge dividirBensComunsMetade(BensComuns,Metade) \Rightarrow
determinarMeacaoConjugeRegimeSeparacaoObrigatoriaBensComuns(Meacao,PessoaY,PessoaX,RegimeSeparacaoObrigatoria,BensComuns)

\exists Meacao, \exists PessoaY, \exists PessoaX, \exists RegimeSeparacaoObrigatoria, \exists BensComuns, \exists Quantidade, \exists Quinhao |
determinarMeacaoConjugeRegimeSeparacaoObrigatoriaBensComuns(Meacao,PessoaY,PessoaX,RegimeSeparacaoObrigatoria,BensComuns) \wedge
determinarQuantidadeDescendentes(Quantidade,PessoaY,PessoaX) \wedge dividirMeacaoDescendentes(Meacao,Quantidade) \Rightarrow

determinarQuinhaoDescendentesRegimeSeparacaoObrigatoriaBensComuns(Quinhao,PessoaY,PessoaX,RegimeSeparacaoObrigatoria,BensComuns)

\exists BensParticulares, \exists PessoaX, \exists Quantidade, \exists PessoaY, \exists Quinhao, \exists RegimeSeparacaoObrigatoria | somarBensParticulares(BensParticulares,PessoaX) \wedge determinarQuantidadeDescendentes(Quantidade,PessoaY,PessoaX) \wedge dividirBensParticularesQuantidadeDescendentes(BensParticulares,Quantidade,PessoaY) \Rightarrow determinarQuinhaoDescendentesRegimeSeparacaoObrigatoriaBensParticulares(Quinhao,PessoaY,RegimeSeparacaoObrigatoria,BensParticulares,PessoaX)

\exists PessoaX, \exists PessoaY, \exists Meacao, \exists RegimeSeparacaoObrigatoria, \exists BensComuns, \exists Quinhao, \exists BensParticulares, \exists Bens, \exists PessoaZ | temConjuge(PessoaX,PessoaY) \wedge determinarMeacaoConjugeRegimeSeparacaoObrigatoriaBensComuns(Meacao,PessoaY,PessoaX,RegimeSeparacaoObrigatoria,BensComuns) \wedge temDescendentes(PessoaX,PessoaY) \wedge determinarQuinhaoDescendentesRegimeSeparacaoObrigatoriaBensComuns(Quinhao,PessoaY,PessoaX,RegimeSeparacaoObrigatoria,BensComuns) \wedge temDescendentes(PessoaX,PessoaY) \wedge determinarQuinhaoDescendentesRegimeSeparacaoObrigatoriaBensParticulares(Quinhao,PessoaY,RegimeSeparacaoObrigatoria,BensParticulares,PessoaX) \Rightarrow partilharBensDescendentesConjugeRegimeSeparacaoObrigatoria(Bens,PessoaX,PessoaY,PessoaZ,RegimeSeparacaoObrigatoria)

\exists Quantidade, \exists PessoaY, \exists PessoaX, \exists PessoaZ, \exists Bens, \exists Soma, \exists Quinhao, \exists RegimeSeparacaoVoluntaria | determinarQuantidadeDescendentes(Quantidade,PessoaY,PessoaX) \wedge somarQuantidadeDescendentesConjuge(Quantidade,PessoaY,PessoaZ) \wedge somarBens(Bens,PessoaX) \wedge dividirBensQuantidadeDescendentesConjuge(Bens,Soma,PessoaY,PessoaZ) \Rightarrow determinarQuinhaoDescendentesConjugeRegimeSeparacaoVoluntaria(Quinhao,PessoaY,PessoaZ,PessoaX,RegimeSeparacaoVoluntaria)

\exists Quantidade, \exists PessoaY, \exists PessoaX, \exists PessoaZ, \exists Bens, \exists Soma, \exists Quinhao, \exists RegimeSeparacaoVoluntaria | determinarQuantidadeDescendentes(Quantidade,PessoaY,PessoaX) \wedge somarQuantidadeDescendentesConjuge(Quantidade,PessoaY,PessoaZ) \wedge somarBens(Bens,PessoaX) \wedge dividirBensQuantidadeDescendentesConjuge(Bens,Soma,PessoaY,PessoaZ) \Rightarrow determinarQuinhaoConjugeDescendentesRegimeSeparacaoVoluntaria(Quinhao,PessoaY,PessoaZ,PessoaX,RegimeSeparacaoVoluntaria)

\exists PessoaX, \exists PessoaY, \exists Quinhao, \exists PessoaZ, \exists RegimeSeparacaoVoluntaria, \exists Bens | temDescendentes(PessoaX,PessoaY) \wedge determinarQuinhaoDescendentesConjugeRegimeSeparacaoVoluntaria(Quinhao,PessoaY,PessoaZ,PessoaX,RegimeSeparacaoVoluntaria) \wedge temConjuge(PessoaX,PessoaY) \wedge determinarQuinhaoConjugeDescendentesRegimeSeparacaoVoluntaria(Quinhao,PessoaY,PessoaZ,PessoaX,RegimeSeparacaoVoluntaria) \Rightarrow partilharBensDescendentesConjugeRegimeSeparacaoVoluntaria(Bens,PessoaX,PessoaY,PessoaZ,RegimeSeparacaoVoluntaria)

\exists Quantidade, \exists PessoaW, \exists PessoaX, \exists PessoaZ, \exists Bens, \exists Soma, \exists Quinhao | determinarQuantidadeAscendentesPrimeiroGrau(Quantidade,PessoaW,PessoaX) \wedge somarQuantidadeAscendentesConjuge(Quantidade,PessoaW,PessoaZ) \wedge somarBens(Bens,PessoaX) \wedge dividirBensAscendentesConjuge(Bens,PessoaZ,PessoaW,Soma) \Rightarrow determinarQuinhaoConjugeAscendentesPrimeiroGrau(Quinhao,PessoaZ,PessoaW,PessoaX)

\exists Quantidade, \exists PessoaW, \exists PessoaX, \exists PessoaZ, \exists Bens, \exists Soma, \exists Quinhao | determinarQuantidadeAscendentesPrimeiroGrau(Quantidade,PessoaW,PessoaX) \wedge somarQuantidadeAscendentesConjuge(Quantidade,PessoaW,PessoaZ) \wedge somarBens(Bens,PessoaX) \wedge dividirBensAscendentesConjuge(Bens,PessoaZ,PessoaW,Soma) \Rightarrow determinarQuinhaoAscendentesPrimeiroGrauConjuge(Quinhao,PessoaW,PessoaZ,PessoaX)

\exists Bens, \exists PessoaX, \exists PessoaW, \exists Quinhao, \exists PessoaZ | somarBens(Bens,PessoaX) \wedge dividirMetadeBensConjuge(Bens,PessoaX,PessoaW) \Rightarrow determinarQuinhaoConjugeAscendentesSegundoGrau(Quinhao,PessoaZ,PessoaW,PessoaX)

$\exists \text{ Bens}, \exists \text{ PessoaX}, \exists \text{ Quantidade}, \exists \text{ PessoaW}, \exists \text{ MetadeBens}, \exists \text{ PessoaZ}, \exists \text{ Quinhao}, \exists \text{ PessoaB} \mid \text{somarBens}(\text{Bens}, \text{PessoaX}) \wedge \text{dividirMetadeBens}(\text{Bens}, \text{PessoaX}) \wedge$
 $\text{determinarQuantidadeAscendentesSegundoGrau}(\text{Quantidade}, \text{PessoaW}, \text{PessoaX}) \wedge \text{dividirMetadeBensQuantidadeAscendentes}(\text{MetadeBens}, \text{PessoaZ}, \text{Quantidade}, \text{PessoaW}) \Rightarrow$
 $\text{determinarQuinhaoAscendentesSegundoGrauConjuge}(\text{Quinhao}, \text{PessoaZ}, \text{PessoaB}, \text{PessoaX})$

$\exists \text{ Bens}, \exists \text{ PessoaX}, \exists \text{ PessoaW}, \exists \text{ Quinhao}, \exists \text{ PessoaZ} \mid \text{somarBens}(\text{Bens}, \text{PessoaX}) \wedge \text{dividirMetadeBensConjuge}(\text{Bens}, \text{PessoaX}, \text{PessoaW}) \Rightarrow$
 $\text{determinarQuinhaoConjugeAscendentesTerceiroGrau}(\text{Quinhao}, \text{PessoaZ}, \text{PessoaW}, \text{PessoaX})$

$\exists \text{ Bens}, \exists \text{ PessoaX}, \exists \text{ Quantidade}, \exists \text{ PessoaW}, \exists \text{ MetadeBens}, \exists \text{ PessoaZ}, \exists \text{ Quinhao} \mid \text{somarBens}(\text{Bens}, \text{PessoaX}) \wedge \text{dividirMetadeBens}(\text{Bens}, \text{PessoaX}) \wedge$
 $\text{determinarQuantidadeAscendentesTerceiroGrau}(\text{Quantidade}, \text{PessoaW}, \text{PessoaX}) \wedge \text{dividirMetadeBensQuantidadeAscendentes}(\text{MetadeBens}, \text{PessoaZ}, \text{Quantidade}, \text{PessoaW}) \Rightarrow$
 $\text{determinarQuinhaoAscendentesTerceiroGrauConjuge}(\text{Quinhao}, \text{PessoaZ}, \text{PessoaW}, \text{PessoaX})$

$\exists \text{ PessoaX}, \exists \text{ PessoaY}, \exists \text{ Quinhao}, \exists \text{ PessoaZ}, \exists \text{ Bens} \mid \text{temConjuge}(\text{PessoaX}, \text{PessoaY}) \wedge \text{determinarQuinhaoConjugeAscendentesPrimeiroGrau}(\text{Quinhao}, \text{PessoaY}, \text{PessoaZ}, \text{PessoaX}) \wedge$
 $\text{determinarGenitor}(\text{PessoaX}, \text{PessoaY}) \wedge \text{determinarQuinhaoAscendentesPrimeiroGrauConjuge}(\text{Quinhao}, \text{PessoaY}, \text{PessoaZ}, \text{PessoaX}) \Rightarrow$
 $\text{partilharBensAscendentesPrimeiroGrauConjuge}(\text{Bens}, \text{PessoaX}, \text{PessoaY}, \text{PessoaZ})$

$\exists \text{ PessoaX}, \exists \text{ PessoaY}, \exists \text{ Quinhao}, \exists \text{ PessoaZ}, \exists \text{ Bens} \mid \text{temConjuge}(\text{PessoaX}, \text{PessoaY}) \wedge \text{determinarQuinhaoConjugeAscendentesSegundoGrau}(\text{Quinhao}, \text{PessoaY}, \text{PessoaZ}, \text{PessoaX}) \wedge$
 $\text{determinarAvos}(\text{PessoaX}, \text{PessoaY}) \wedge \text{determinarQuinhaoAscendentesSegundoGrauConjuge}(\text{Quinhao}, \text{PessoaY}, \text{PessoaZ}, \text{PessoaX}) \Rightarrow$
 $\text{partilharBensAscendentesSegundoGrauConjuge}(\text{Bens}, \text{PessoaX}, \text{PessoaY}, \text{PessoaZ})$

$\exists \text{ PessoaX}, \exists \text{ PessoaY}, \exists \text{ Quinhao}, \exists \text{ PessoaZ}, \exists \text{ Bens} \mid \text{temConjuge}(\text{PessoaX}, \text{PessoaY}) \wedge \text{determinarQuinhaoConjugeAscendentesTerceiroGrau}(\text{Quinhao}, \text{PessoaY}, \text{PessoaZ}, \text{PessoaX}) \wedge$
 $\text{determinarBiSavos}(\text{PessoaX}, \text{PessoaY}) \wedge \text{determinarQuinhaoAscendentesTerceiroGrauConjuge}(\text{Quinhao}, \text{PessoaY}, \text{PessoaZ}, \text{PessoaX}) \Rightarrow$
 $\text{partilharBensAscendentesTerceiroGrauConjuge}(\text{Bens}, \text{PessoaX}, \text{PessoaY}, \text{PessoaZ})$

$\exists \text{ Bens}, \exists \text{ PessoaX}, \exists \text{ PessoaY} \mid \text{somarBens}(\text{Bens}, \text{PessoaX}) \Rightarrow \text{determinarBensConjuge}(\text{Bens}, \text{PessoaY}, \text{PessoaX})$

$\exists \text{ PessoaX}, \exists \text{ PessoaY}, \exists \text{ Bens}, \exists \text{ PessoaZ} \mid \text{temConjuge}(\text{PessoaX}, \text{PessoaY}) \wedge \text{determinarBensConjuge}(\text{Bens}, \text{PessoaY}, \text{PessoaX}) \Rightarrow \text{partilharBensConjuge}(\text{Bens}, \text{PessoaX}, \text{PessoaY}, \text{PessoaZ})$

$\exists \text{ Bens}, \exists \text{ PessoaX}, \exists \text{ Quantidade}, \exists \text{ PessoaW}, \exists \text{ Quinhao}, \exists \text{ PessoaY} \mid \text{somarBens}(\text{Bens}, \text{PessoaX}) \wedge \text{determinarQuantidadeColateraisGermanos}(\text{Quantidade}, \text{PessoaW}, \text{PessoaX}) \wedge$
 $\text{dividirBensQuantidadeColateraisGermanos}(\text{Bens}, \text{Quantidade}, \text{PessoaW}) \Rightarrow \text{determinarQuinhaoColateraisGermanos}(\text{Quinhao}, \text{PessoaY}, \text{PessoaX})$

$\exists \text{ Bens}, \exists \text{ PessoaX}, \exists \text{ Quantidade}, \exists \text{ PessoaW}, \exists \text{ Quinhao}, \exists \text{ PessoaY} \mid \text{somarBens}(\text{Bens}, \text{PessoaX}) \wedge \text{determinarQuantidadeColateraisGermanos}(\text{Quantidade}, \text{PessoaW}, \text{PessoaX}) \wedge$
 $\text{dividirBensQuantidadeColateraisGermanos}(\text{Bens}, \text{Quantidade}, \text{PessoaW}) \Rightarrow \text{determinarQuinhaoColateraisGermanos}(\text{Quinhao}, \text{PessoaY}, \text{PessoaX})$

\exists PessoaX, \exists PessoaY, \exists Quinhao, \exists Bens, \exists PessoaZ | temColateraisGermanos(PessoaX,PessoaY) \wedge determinarQuinhaoColateraisGermanos(Quinhao,PessoaY,PessoaX) \Rightarrow partilharBensColateraisGermanos(Bens,PessoaX,PessoaY,PessoaZ)

\exists BensComuns, \exists PessoaX, \exists Quantidade, \exists PessoaY, \exists PessoaZ, \exists Soma, \exists Quinhao | somarBensComuns(BensComuns,PessoaX) \wedge determinarQuantidadeFilhosComuns(Quantidade,PessoaY,PessoaX) \wedge somarQuantidadeFilhosComunsCompanheiro(Quantidade,PessoaY) \wedge dividirBensComunsFilhosComunsCompanheiro(BensComuns,PessoaZ,PessoaY,Soma) \Rightarrow determinarQuinhaoCompanheiroFilhosComunsBensComuns(Quinhao,PessoaZ,PessoaY,PessoaX,BensComuns)

\exists BensComuns, \exists PessoaX, \exists Quantidade, \exists PessoaY, \exists PessoaZ, \exists Soma, \exists Quinhao | somarBensComuns(BensComuns,PessoaX) \wedge determinarQuantidadeFilhosComuns(Quantidade,PessoaY,PessoaX) \wedge somarQuantidadeFilhosComunsCompanheiro(Quantidade,PessoaY) \wedge dividirBensComunsFilhosComunsCompanheiro(BensComuns,PessoaZ,PessoaY,Soma) \Rightarrow determinarQuinhaoFilhosComunsCompanheiroBensComuns(Quinhao,PessoaY,PessoaZ,PessoaX,BensComuns)

\exists BensParticulares, \exists PessoaX, \exists Quantidade, \exists PessoaW, \exists Quinhao, \exists PessoaY, \exists PessoaZ | somarBensParticulares(BensParticulares,PessoaX) \wedge determinarQuantidadeFilhosComuns(Quantidade,PessoaW,PessoaX) \wedge dividirBensParticularesQuantidadeFilhosComuns(BensParticulares,Quantidade,PessoaW) \Rightarrow determinarQuinhaoFilhosComunsCompanheiroBensParticulares(Quinhao,PessoaY,PessoaZ,PessoaX,BensParticulares)

\exists PessoaX, \exists PessoaY, \exists Quinhao, \exists PessoaZ, \exists BensComuns, \exists BensParticulares, \exists Bens | temCompanheiro(PessoaX,PessoaY) \wedge determinarQuinhaoCompanheiroFilhosComunsBensComuns(Quinhao,PessoaZ,PessoaY,PessoaX,BensComuns) \wedge temFilhoComum(PessoaX,PessoaY) \wedge determinarQuinhaoFilhosComunsCompanheiroBensComuns(Quinhao,PessoaY,PessoaZ,PessoaX,BensComuns) \wedge temFilhoComum(PessoaX,PessoaY) \wedge determinarQuinhaoFilhosComunsCompanheiroBensParticulares(Quinhao,PessoaY,PessoaZ,PessoaX,BensParticulares) \Rightarrow partilharBensCompanheiroFilhosComuns(Bens,PessoaX,PessoaY,PessoaZ)

\exists BensComuns, \exists PessoaX, \exists Quantidade, \exists PessoaY, \exists PessoaZ, \exists Quinhao | somarBensComuns(BensComuns,PessoaX) \wedge determinarQuantidadeFilhosIncomuns(Quantidade,PessoaY,PessoaX) \wedge dividirBensComunsQuantidadeFilhosInComuns(BensComuns,PessoaZ,Quantidade,PessoaY) \Rightarrow determinarQuinhaoFilhosIncomunsCompanheiroBensComuns(Quinhao,PessoaY,PessoaZ,PessoaX,BensComuns)

\exists Quinhao, \exists PessoaY, \exists PessoaZ, \exists PessoaX, \exists BensComuns, \exists FilhosIncomuns, \exists Quota | determinarQuinhaoFilhosIncomunsCompanheiroBensComuns(Quinhao,PessoaY,PessoaZ,PessoaX,BensComuns) \wedge dividirMetadeQuinhaoFilhosIncomuns(Quinhao,PessoaX,FilhosIncomuns) \Rightarrow determinarQuotaCompanheiroFilhosIncomunsBensComuns(Quota,PessoaY,PessoaZ,PessoaX,BensComuns)

\exists BensParticulares, \exists PessoaX, \exists Quantidade, \exists PessoaY, \exists PessoaZ, \exists Quinhao | somarBensParticulares(BensParticulares,PessoaX) \wedge determinarQuantidadeFilhosIncomuns(Quantidade,PessoaY,PessoaX) \wedge dividirBensParticularesQuantidadeFilhosInComuns(BensParticulares,PessoaZ,Quantidade,PessoaY) \Rightarrow determinarQuinhaoFilhosIncomunsCompanheiroBensParticulares(Quinhao,PessoaY,PessoaZ,PessoaX,BensParticulares)

\exists PessoaX, \exists PessoaY, \exists Quinhao, \exists PessoaZ, \exists BensComuns, \exists BensParticulares, \exists Quota, \exists Bens | temFilhoIncomum(PessoaX,PessoaY) \wedge
determinarQuinhaoFilhosIncomunsCompanheiroBensComuns(Quinhao,PessoaY,PessoaZ,PessoaX,BensComuns) \wedge temFilhoIncomum(PessoaX,PessoaY) \wedge
determinarQuinhaoFilhosIncomunsCompanheiroBensParticulares(Quinhao,PessoaY,PessoaZ,PessoaX,BensParticulares) \wedge temCompanheiro(PessoaX,PessoaY) \wedge
determinarQuotaCompanheiroFilhosIncomunsBensComuns(Quota,PessoaY,PessoaZ,PessoaX,BensComuns) \Rightarrow partilharBensCompanheiroFilhosInComuns(Bens,PessoaX,PessoaY,PessoaZ)

\exists Bens, \exists PessoaX, \exists PessoaY, \exists Quota | somarBens(Bens,PessoaX) \wedge obterTercaBens(Bens,PessoaY,PessoaX) \Rightarrow
determinarQuotaCompanheiroAscendentesConjugeColateraisBens(Quota,PessoaX,PessoaY,Bens)

\exists Bens, \exists PessoaX, \exists Quota, \exists PessoaY, \exists Quantidade, \exists BensSubtraidos, \exists Quinhao | somarBens(Bens,PessoaX) \wedge
determinarQuotaCompanheiroAscendentesConjugeColateraisBens(Quota,PessoaX,PessoaY,Bens) \wedge subtrairBensQuotaCompanheiro(Bens,Quota,PessoaX,PessoaY) \wedge
determinarQuantidadeAscendentes(Quantidade,PessoaY,PessoaX) \wedge dividirBensSubtraidosQuantidadeAscendentes(BensSubtraidos,Quantidade,PessoaY) \Rightarrow
determinarQuinhaoAscendentesCompanheiroBens(Quinhao,PessoaX,PessoaY,Bens)

\exists Bens, \exists PessoaX, \exists Quota, \exists PessoaY, \exists Quinhao | somarBens(Bens,PessoaX) \wedge determinarQuotaCompanheiroAscendentesConjugeColateraisBens(Quota,PessoaX,PessoaY,Bens) \wedge
subtrairBensQuotaCompanheiro(Bens,Quota,PessoaX,PessoaY) \Rightarrow determinarQuinhaoConjugeCompanheiroBens(Quinhao,PessoaX,PessoaY,Bens)

\exists Bens, \exists PessoaX, \exists Quota, \exists PessoaY, \exists Quantidade, \exists BensSubtraidos, \exists Quinhao | somarBens(Bens,PessoaX) \wedge
determinarQuotaCompanheiroAscendentesConjugeColateraisBens(Quota,PessoaX,PessoaY,Bens) \wedge subtrairBensQuotaCompanheiro(Bens,Quota,PessoaX,PessoaY) \wedge
determinarQuantidadeColaterais(Quantidade,PessoaY,PessoaX) \wedge dividirBensSubtraidosQuantidadeColaterais(BensSubtraidos,Quantidade,PessoaY) \Rightarrow
determinarQuinhaoColateraisCompanheiroBens(Quinhao,PessoaX,PessoaY,Bens)

\exists PessoaX, \exists PessoaY, \exists Quota, \exists Bens, \exists PessoaZ, \exists PessoaW | temCompanheiro(PessoaX,PessoaY) \wedge
determinarQuotaCompanheiroAscendentesConjugeColateraisBens(Quota,PessoaX,PessoaY,Bens) \wedge temAscendentes(PessoaX,PessoaY) \wedge
determinarQuinhaoAscendentesCompanheiroBens(Quinhao,PessoaX,PessoaY,Bens) \wedge temConjuge(PessoaX,PessoaY) \wedge
determinarQuinhaoConjugeCompanheiroBens(Quinhao,PessoaX,PessoaY,Bens) \wedge temColaterais(PessoaX,PessoaY) \wedge
determinarQuinhaoColateraisCompanheiroBens(Quinhao,PessoaX,PessoaY,Bens) \Rightarrow partilharBensCompanheiroAscendentesConjugeColaterais(Bens,PessoaX,PessoaY,PessoaZ,PessoaW)

\exists PessoaX, \exists PessoaY, \exists Bens | temCompanheiro(PessoaX,PessoaY) \wedge somarBens(Bens,PessoaX) \Rightarrow determinarBensCompanheiro(Bens,PessoaX,PessoaY)

\exists PessoaX, \exists PessoaY | temDescendentes(PessoaX,PessoaY) \Rightarrow armazenarDescendentes(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | temConjuge(PessoaX,PessoaY) \Rightarrow armazenarConjuge(PessoaY,PessoaX)

\exists PessoaX, \exists PessoaY | temAscendentes(PessoaX,PessoaY) \Rightarrow armazenarAscendentes(PessoaY,PessoaX)

\exists PessoaY, \exists PessoaX | verificarDescendentes(PessoaY,PessoaX) \wedge armazenarDescendentes(PessoaY,PessoaX) \wedge verificarConjuge(PessoaY,PessoaX) \wedge armazenarConjuge(PessoaY,PessoaX) \vee verificarAscendentes(PessoaY,PessoaX) \wedge armazenarAscendentes(PessoaY,PessoaX) \Rightarrow identificarHerdeiros(PessoaY,PessoaX)

\exists PessoaY, \exists PessoaX, \exists Heranca | identificarHerdeiros(PessoaY,PessoaX) \wedge determinarHeranca(Heranca,PessoaY,PessoaX) \Rightarrow calcularHeranca(Heranca,PessoaX)

\exists PessoaY, \exists PessoaX, \exists RegimeUniversal, \exists Bens, \exists PessoaZ, \exists RegimeParcial, \exists RegimeParticipacaoAquestos, \exists RegimeSeparacaoVoluntaria, \exists RegimeSeparacaoObrigatoria, \exists PessoaW, \exists Heranca | verificarDescendentes(PessoaY,PessoaX) \wedge verificarConjuge(PessoaY,PessoaX) \wedge verificarRegimeUniversal(RegimeUniversal,PessoaX) \wedge partilharBensDescendentesConjugeRegimeUniversal(Bens,PessoaX,PessoaY,PessoaZ,RegimeUniversal) \vee verificarDescendentes(PessoaY,PessoaX) \wedge verificarConjuge(PessoaY,PessoaX) \wedge verificarRegimeParcial(RegimeParcial,PessoaX) \wedge partilharBensDescendentesConjugeRegimeParcial(Bens,PessoaX,PessoaY,PessoaZ,RegimeParcial) \vee verificarDescendentes(PessoaY,PessoaX) \wedge verificarConjuge(PessoaY,PessoaX) \wedge verificarRegimeParticipacaoAquestos(RegimeParticipacaoAquestos,PessoaX) \wedge partilharBensDescendentesConjugeRegimeParticipacaoAquestos(Bens,PessoaX,PessoaY,PessoaZ,RegimeParticipacaoAquestos) \vee verificarDescendentes(PessoaY,PessoaX) \wedge verificarConjuge(PessoaY,PessoaX) \wedge verificarRegimeSeparacaoVoluntaria(RegimeSeparacaoVoluntaria,PessoaX) \wedge partilharBensDescendentesConjugeRegimeSeparacaoVoluntaria(Bens,PessoaX,PessoaY,PessoaZ,RegimeSeparacaoVoluntaria) \vee verificarDescendentes(PessoaY,PessoaX) \wedge verificarConjuge(PessoaY,PessoaX) \wedge verificarRegimeSeparacaoObrigatoria(RegimeSeparacaoObrigatoria,PessoaX) \wedge partilharBensDescendentesConjugeRegimeSeparacaoObrigatoria(Bens,PessoaX,PessoaY,PessoaZ,RegimeSeparacaoVoluntaria) \vee verificarAscendentesPrimeiroGrau(PessoaY,PessoaX) \wedge verificarConjuge(PessoaY,PessoaX) \wedge partilharBensAscendentesPrimeiroGrauConjuge(Bens,PessoaX,PessoaY,PessoaZ) \vee verificarAscendentesSegundoGrau(PessoaY,PessoaX) \wedge verificarConjuge(PessoaY,PessoaX) \wedge partilharBensAscendentesSegundoGrauConjuge(Bens,PessoaX,PessoaY,PessoaZ) \vee verificarAscendentesTerceiroGrau(PessoaY,PessoaX) \wedge verificarConjuge(PessoaY,PessoaX) \wedge partilharBensAscendentesTerceiroGrauConjuge(Bens,PessoaX,PessoaY,PessoaZ) \vee verificarConjuge(PessoaY,PessoaX) \wedge partilharBensConjuge(Bens,PessoaX,PessoaY,PessoaZ) \vee verificarColateraisGermanos(PessoaY,PessoaX) \wedge partilharBensColateraisGermanos(Bens,PessoaX,PessoaY,PessoaZ) \vee verificarCompanheiro(PessoaY,PessoaX) \wedge verificarFilhosComuns(PessoaY,PessoaX) \wedge partilharBensCompanheiroFilhosComuns(Bens,PessoaX,PessoaY,PessoaZ) \vee verificarCompanheiro(PessoaY,PessoaX) \wedge verificarFilhosIncomuns(PessoaY,PessoaX) \wedge partilharBensCompanheiroFilhosInComuns(Bens,PessoaX,PessoaY,PessoaZ) \vee verificarCompanheiro(PessoaY,PessoaX) \wedge verificarAscendentes(PessoaY,PessoaX) \wedge partilharBensCompanheiroAscendentesConjugeColaterais(Bens,PessoaX,PessoaY,PessoaZ,PessoaW) \vee verificarCompanheiro(PessoaY,PessoaX) \wedge verificarColaterais(PessoaY,PessoaX) \wedge partilharBensCompanheiroAscendentesConjugeColaterais(Bens,PessoaX,PessoaY,PessoaZ,PessoaW) \vee verificarCompanheiro(PessoaY,PessoaX) \wedge verificarConjuge(PessoaY,PessoaX) \wedge partilharBensCompanheiroAscendentesConjugeColaterais(Bens,PessoaX,PessoaY,PessoaZ,PessoaW) \Rightarrow determinarHeranca(Heranca,PessoaY,PessoaX)

APÊNDICE C: Objetivos e fatos com os seus respectivos axiomas representados em RuleML.

<ul style="list-style-type: none"> • Calcular a herança de uma pessoa • Identificar os herdeiros de uma pessoa • Determinar a herança dos herdeiros
<pre> <Implies> <And> <Atom> <Rel>identificarHerdeiros</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> <Atom> <Rel>determinarHeranca</Rel> <Var>Heranca</Var> <Var>PessoaX</Var> </Atom> </And> <Atom> <Rel>calcularHeranca</Rel> <Var>Heranca</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Determinar o genitor de uma pessoa • Determinar a quantidade de ascendentes de primeiro grau de uma pessoa
<pre> <Implies> <Atom> <Rel>determinarGenitor</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>determinarQuantidadeAscendentesPrimeiroGrau</Rel> <Var>Quantidade</Var> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>

<ul style="list-style-type: none"> • Determinar os avos de uma pessoa • Determinar a quantidade de ascendentes de segundo grau de uma pessoa
<pre> <Implies> <Atom> <Rel>determinarAvos</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>determinarQuantidadeAscendentesSegundoGrau</Rel> <Var>Quantidade</Var> <Var>PessoaB</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Determinar os bisavos de uma pessoa • Determinar a quantidade de ascendentes de terceiro grau de uma pessoa
<pre> <Implies> <Atom> <Rel>determinarBiSavos</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>determinarQuantidadeAscendentesTerceiroGrau</Rel> <Var>Quantidade</Var> <Var>PessoaV</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem colaterais • Determinar a quantidade de colaterais de uma pessoa
<pre> <Implies> <Atom> <Rel>temColaterais</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> </pre>

<pre> </Atom> <Atom> <Rel>determinarQuantidadeColaterais</Rel> <Var>Quantidade</Var> <Var>Pessoa Y</Var> <Var>Pessoa X</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem colaterais germanos • Determinar a quantidade de colaterais germanos de uma pessoa
<pre> <Implies> <Atom> <Rel>temColateraisGermanos</Rel> <Var>Pessoa X</Var> <Var>Pessoa Y</Var> </Atom> <Atom> <Rel>determinarQuantidadeColateraisGermanos</Rel> <Var>Quantidade</Var> <Var>Pessoa Y</Var> <Var>Pessoa X</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem colaterais unilaterais • Determinar a quantidade de colaterais unilaterais de uma pessoa
<pre> <Implies> <Atom> <Rel>temColateraisUnilaterais</Rel> <Var>Pessoa X</Var> <Var>Pessoa Y</Var> </Atom> <Atom> <Rel>determinarQuantidadeColateraisUnilaterais</Rel> <Var>Quantidade</Var> <Var>Pessoa Y</Var> <Var>Pessoa X</Var> </Atom> </Implies> </pre>

<ul style="list-style-type: none"> • Uma pessoa tem filho comum • Determinar a quantidade de filhos comuns de uma pessoa
<pre> <Implies> <Atom> <Rel>temFilhoComum</Rel> <Var>Pessoa X</Var> <Var>Pessoa Y</Var> </Atom> <Atom> <Rel>determinarQuantidadeFilhosComuns</Rel> <Var>Quantidade</Var> <Var>Pessoa V</Var> <Var>Pessoa X</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem filho incomum • Determinar a quantidade de filhos incomuns de uma pessoa
<pre> <Implies> <Atom> <Rel>temFilhoIncomum</Rel> <Var>Pessoa X</Var> <Var>Pessoa Y</Var> </Atom> <Atom> <Rel>determinarQuantidadeFilhosIncomuns</Rel> <Var>Quantidade</Var> <Var>Pessoa V</Var> <Var>Pessoa X</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem bens comuns • Somar os bens comuns de uma pessoa
<pre> <Implies> <Atom> <Rel>temBensComuns</Rel> <Var>Pessoa X</Var> <Var>BensComuns</Var> </Atom> <Atom> </pre>

<pre> <Rel>somarBensComuns</Rel> <Var>BensComuns</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem bem particular • Somar os bens particulares de uma pessoa
<pre> <Implies> <Atom> <Rel>temBensParticulares</Rel> <Var>PessoaX</Var> <Var>BensParticulares</Var> </Atom> <Atom> <Rel>somarBensParticulares</Rel> <Var>BensParticulares</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Somar os bens comuns de uma pessoa • Somar os bens particulares de uma pessoa • Somar os bens comuns e bens particulares de uma pessoa • Somar os bens de uma pessoa
<pre> <Implies> <And> <Atom> <Rel>somarBensComuns</Rel> <Var>BensComuns</Var> <Var>PessoaX</Var> </Atom> <Atom> <Rel>somarBensParticulares</Rel> <Var>BensParticulares</Var> <Var>PessoaX</Var> </Atom> <Atom> <Rel>somarBensComunsBensParticulares</Rel> <Var>BensComuns</Var> <Var>BensParticulares</Var> </Atom> </pre>

<pre> </And> <Atom> <Rel>somarBens</Rel> <Var>Bens</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem descendentes • Verificar a existência dos descendentes de uma pessoa
<pre> <Implies> <Atom> <Rel>temDescendentes</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarDescendentes</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem cônjuge • Verificar a existência do cônjuge de uma pessoa
<pre> <Implies> <Atom> <Rel>temConjuge</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarConjuge</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>

<ul style="list-style-type: none"> • Uma pessoa tem descendentes • Verificar a existência dos descendentes de uma pessoa
<pre> <Implies> <Atom> <Rel>temAscendentes</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarAscendentes</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Determinar o genitor de uma pessoa • Verificar a existência dos ascendentes de primeiro grau de uma pessoa
<pre> <Implies> <Atom> <Rel>determinarGenitor</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarAscendentesPrimeiroGrau</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Determinar os avos de uma pessoa • Verificar a existência dos ascendentes de segundo grau de uma pessoa
<pre> <Implies> <Atom> <Rel>determinarAvos</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarAscendentesSegundoGrau</Rel> <Var>PessoaY</Var> </pre>

<pre> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Determinar os bisavos de uma pessoa • Verificar a existência dos ascendentes de terceiro grau de uma pessoa
<pre> <Implies> <Atom> <Rel>determinarBiSavos</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarAscendentesTerceiroGrau</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem colaterais germanos • Verificar a existência dos colaterais germanos de uma pessoa
<pre> <Implies> <Atom> <Rel>temColateraisGermanos</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarColateraisGermanos</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem filho comum • Verificar a existência dos filhos comuns de uma pessoa
<pre> <Implies> <Atom> <Rel>temFilhoComum</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> </pre>

<pre> <Atom> <Rel>verificarFilhosComuns</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem filho incomum • Verificar a existência dos filhos incomuns de uma pessoa
<pre> <Implies> <Atom> <Rel>temFilhoIncomum</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarFilhosIncomuns</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem colaterais • Verificar a existência dos colaterais de uma pessoa
<pre> <Implies> <Atom> <Rel>temColaterais</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarColaterais</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>

<ul style="list-style-type: none"> • Uma pessoa tem companheiro • Verificar a existência do companheiro de uma pessoa
<pre> <Implies> <Atom> <Rel>temCompanheiro</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarCompanheiro</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem avô • Verificar a existência dos avôs de uma pessoa
<pre> <Implies> <Atom> <Rel>temAvo</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarAvo</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem avó • Verificar a existência das avós de uma pessoa
<pre> <Implies> <Atom> <Rel>temAvoh</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarAvoh</Rel> <Var>PessoaY</Var> </pre>

<pre><Var>PessoaX</Var> </Atom> </Implies></pre>
<ul style="list-style-type: none"> • Uma pessoa tem bisavô • Verificar a existência dos bisavôs de uma pessoa
<pre><Implies> <Atom> <Rel>temBiSavo</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarBiSavo</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies></pre>
<ul style="list-style-type: none"> • Uma pessoa tem bisavó • Verificar a existência das bisavós de uma pessoa
<pre><Implies> <Atom> <Rel>temBiSavoh</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>verificarBiSavoh</Rel> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> </Implies></pre>
<ul style="list-style-type: none"> • Uma pessoa tem regime universal • Verificar a existência do regime universal de uma pessoa
<pre><Implies> <Atom> <Rel>temRegimeUniversal</Rel> <Var>PessoaX</Var> <Var>RegimeUniversal</Var> </Atom></pre>

<pre><Atom> <Rel>verificarRegimeUniversal</Rel> <Var>RegimeUniversal</Var> <Var>PessoaX</Var> </Atom> </Implies></pre>
<ul style="list-style-type: none"> • Uma pessoa tem regime parcial • Verificar a existência do regime parcial de uma pessoa
<pre><Implies> <Atom> <Rel>temRegimeParcial</Rel> <Var>PessoaX</Var> <Var>RegimeParcial</Var> </Atom> <Atom> <Rel>verificarRegimeParcial</Rel> <Var>RegimeParcial</Var> <Var>PessoaX</Var> </Atom> </Implies></pre>
<ul style="list-style-type: none"> • Uma pessoa tem regime de participação nos aquestos • Verificar a existência do regime de participação nos aquestos de uma pessoa
<pre><Implies> <Atom> <Rel>temRegimeParticipacaoAquestos</Rel> <Var>PessoaX</Var> <Var>RegimeParticipacao</Var> </Atom> <Atom> <Rel>verificarRegimeParticipacaoAquestos</Rel> <Var>RegimeParticipacaoAquestos</Var> <Var>PessoaX</Var> </Atom> </Implies></pre>
<ul style="list-style-type: none"> • Uma pessoa tem regime de separação voluntária • Verificar a existência do regime de separação voluntária de uma pessoa
<pre><Implies> <Atom> <Rel>temRegimeSeparacaoVoluntaria</Rel></pre>

<pre> <Var>PessoaX</Var> <Var>RegimeSeparacaoVoluntaria</Var> </Atom> <Atom> <Rel>verificarRegimeSeparacaoVoluntaria</Rel> <Var>RegimeSeparacaoVoluntaria</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem regime de separação obrigatória • Verificar a existência do regime de separação obrigatória de uma pessoa
<pre> <Implies> <Atom> <Rel>temRegimeSeparacaoObrigatoria</Rel> <Var>PessoaX</Var> <Var>RegimeSeparacaoObrigatoria</Var> </Atom> <Atom> <Rel>verificarRegimeSeparacaoObrigatoria</Rel> <Var>RegimeSeparacaoObrigatoria</Var> <Var>PessoaX</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Somas os bens de uma pessoa • Dividir os bens de uma pessoa pela metade • Determinar a meação do cônjuge de uma pessoa no regime universal
<pre> <Implies> <And> <Atom> <Rel>somarBens</Rel> <Var>Bens</Var> <Var>PessoaX</Var> </Atom> <Atom> <Rel>dividirBensMetade</Rel> <Var>Bens</Var> <Var>Metade</Var> </Atom> </And> </pre>

<pre> <Atom> <Rel>determinarMeacaoConjugeRegimeUniversal</Rel> <Var>Meacao</Var> <Var>PessoaY</Var> <Var>PessoaX</Var> <Var>RegimeUniversal</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Determinar a meação do cônjuge de uma pessoa no regime universal • Determinar a quantidade de descendentes de uma pessoa • Dividir a meação pela quantidade de descendentes • Determinar o quinhão dos descendentes de uma pessoa no regime universal
<pre> <Implies> <And> <Atom> <Rel>determinarMeacaoConjugeRegimeUniversal</Rel> <Var>Meacao</Var> <Var>PessoaY</Var> <Var>PessoaX</Var> <Var>RegimeUniversal</Var> </Atom> <Atom> <Rel>determinarQuantidadeDescendentes</Rel> <Var>Quantidade</Var> <Var>PessoaY</Var> <Var>PessoaX</Var> </Atom> <Atom> <Rel>dividirMeacaoDescendentes</Rel> <Var>Meacao</Var> <Var>Quantidade</Var> </Atom> </And> <Atom> <Rel>determinarQuinhaoDescendentesRegimeUniversal</Rel> <Var>Quinhao</Var> <Var>PessoaY</Var> <Var>PessoaX</Var> <Var>RegimeUniversal</Var> </pre>

<pre> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Uma pessoa tem cônjuge • Determinar a meação do cônjuge de uma pessoa no regime universal • Uma pessoa tem descendentes • Determinar o quinhão dos descendentes de uma pessoa no regime universal • Partilhar os bens de uma pessoa entre os descendentes e o cônjuge sobre o regime universal
<pre> <Implies> <And> <Atom> <Rel>temConjuge</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>determinarMeacaoConjugeRegimeUniversal</Rel> <Var>Meacao</Var> <Var>PessoaY</Var> <Var>PessoaX</Var> <Var>RegimeUniversal</Var> </Atom> <Atom> <Rel>temDescendentes</Rel> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> <Atom> <Rel>determinarQuinhaoDescendentesRegimeUniversal</Rel> <Var>Quinhao</Var> <Var>PessoaY</Var> <Var>PessoaX</Var> <Var>RegimeUniversal</Var> </Atom> </And> <Atom> <Rel>partilharBensDescendentesConjugeRegimeUniversal</Rel> <Var>Bens</Var> <Var>PessoaX</Var> <Var>PessoaY</Var> </Atom> </Implies> </pre>

<pre> <Var>PessoaZ</Var> <Var>RegimeUniversal</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Somar os bens comuns de uma pessoa • Dividir os bens comuns de uma pessoa pela metade • Determinar a meação do cônjuge no regime parcial sobre os benscomuns
<pre> <Implies> <And> <Atom> <Rel>somarBensComuns</Rel> <Var>BensComuns</Var> <Var>PessoaX</Var> </Atom> <Atom> <Rel>dividirBensComunsMetade</Rel> <Var>BensComuns</Var> <Var>Metade</Var> </Atom> </And> <Atom> <Rel>determinarMeacaoConjugeRegimeParcialBensComuns</Rel> <Var>Meacao</Var> <Var>PessoaY</Var> <Var>PessoaX</Var> <Var>RegimeParcial</Var> <Var>BensComuns</Var> </Atom> </Implies> </pre>
<ul style="list-style-type: none"> • Somar os bens particulares de uma pessoa • Determinar a quantidade de descendentes de uma pessoa • Somar a quantidade de ascendentes com o cônjuge de uma pessoa • Dividir os bens particulares de uma pessoa pela soma do número de descendentes e cônjuge • Determinar o quinhão do cônjuge em concorrência com os descendentes no regime parcial sobre os bens particulares de uma pessoa
<pre> <Implies> <And> <Atom> </pre>

```

    <Rel>somarBensParticulares</Rel>
    <Var>BensParticulares</Var>
    <Var>PessoaX</Var>
  </Atom>
  <Atom>
    <Rel>determinarQuantidadeDescendentes</Rel>
    <Var>Quantidade</Var>
    <Var>PessoaY</Var>
    <Var>PessoaX</Var>
  </Atom>
  <Atom>
    <Rel>somarQuantidadeDescendentesConjuge</Rel>
    <Var>Quantidade</Var>
    <Var>PessoaY</Var>
  </Atom>
  <Atom>
    <Rel>dividirBensParticularesSomaDescendentesConjuge</Rel>
    <Var>BensParticulares</Var>
    <Var>Soma</Var>
    <Var>PessoaY</Var>
    <Var>PessoaZ</Var>
  </Atom>
</And>
<Atom>
  <Rel>determinarQuinhaoConjugeDescendentesRegimeParcialBensParticulares</Rel>
  <Var>Quinhao</Var>
  <Var>PessoaY</Var>
  <Var>PessoaZ</Var>
  <Var>RegimeParcial</Var>
  <Var>BensParticulares</Var>
  <Var>PessoaX</Var>
</Atom>
</Implies>

```

- Determinar a meação do cônjuge de uma pessoa no regime parcial sobre os bens comuns
- Determinar a quantidade de descendentes de uma pessoa
- Dividir a meação do cônjuge pela quantidade de descendentes
- Determinar o quinhão dos descendentes no regime parcial sobre os bens comuns

```

<Implies>
<And>

```

```

  <Atom>
    <Rel>determinarMeacaoConjugeRegimeParcialBensComuns</Rel>
    <Var>Meacao</Var>
    <Var>PessoaY</Var>
    <Var>PessoaX</Var>
    <Var>RegimeParcial</Var>
    <Var>BensComuns</Var>
  </Atom>
  <Atom>
    <Rel>determinarQuantidadeDescendentes</Rel>
    <Var>Quantidade</Var>
    <Var>PessoaY</Var>
    <Var>PessoaX</Var>
  </Atom>
  <Atom>
    <Rel>dividirMeacaoDescendentes</Rel>
    <Var>Meacao</Var>
    <Var>Quantidade</Var>
  </Atom>
</And>
<Atom>
  <Rel>determinarQuinhaoDescendentesRegimeParcialBensComuns</Rel>
  <Var>Quinhao</Var>
  <Var>PessoaY</Var>
  <Var>PessoaX</Var>
  <Var>RegimeParcial</Var>
  <Var>BensComuns</Var>
</Atom>
</Implies>

```

- Uma pessoa tem descendentes
- Determinar o quinhão do cônjuge em concorrência com os descendentes no regime parcial sobre os bens particulares de uma pessoa
- Uma pessoa tem cônjuge
- Determinar a meação do cônjuge de uma pessoa no regime parcial sobre os bens comuns
- Uma pessoa tem descendentes
- Determinar o quinhão dos descendentes no regime parcial sobre os bens comuns

```

<Implies>
<And>
  <Atom>

```

```

    <Rel>temDescendentes</Rel>
    <Var>PessoaX</Var>
    <Var>PessoaY</Var>
  </Atom>
  <Atom>
    <Rel>determinarQuinhaoConjugeDescendentesRegimeParcialBensParticulares</Rel>
    <Var>Quinhao</Var>
    <Var>PessoaY</Var>
    <Var>PessoaZ</Var>
    <Var>RegimeParcial</Var>
    <Var>BensParticulares</Var>
    <Var>PessoaX</Var>
  </Atom>
  <Atom>
    <Rel>temConjuge</Rel>
    <Var>PessoaX</Var>
    <Var>PessoaY</Var>
  </Atom>
  <Atom>
    <Rel>determinarMeacaoConjugeRegimeParcialBensComuns</Rel>
    <Var>Meacao</Var>
    <Var>PessoaY</Var>
    <Var>PessoaX</Var>
    <Var>RegimeParcial</Var>
    <Var>BensComuns</Var>
  </Atom>
  <Atom>
    <Rel>temDescendentes</Rel>
    <Var>PessoaX</Var>
    <Var>PessoaY</Var>
  </Atom>
  <Atom>
    <Rel>determinarQuinhaoDescendentesRegimeParcialBensComuns</Rel>
    <Var>Quinhao</Var>
    <Var>PessoaY</Var>
    <Var>PessoaX</Var>
    <Var>RegimeParcial</Var>
    <Var>BensComuns</Var>
  </Atom>
</And>

```

```

  <Atom>
    <Rel>partilharBensDescendentesConjugeRegimeParcial</Rel>
    <Var>Bens</Var>
    <Var>PessoaX</Var>
    <Var>PessoaY</Var>
    <Var>PessoaZ</Var>
    <Var>RegimeParcial</Var>
  </Atom>
</Implies>


- Somar os bens comuns de uma pessoa
- Dividir os bens de uma pessoa pela metade
- Determinar a meação do cônjuge de uma pessoa no regime de participação nos aquestos sobre os bens comuns


  <Implies>
    <And>
      <Atom>
        <Rel>somarBensComuns</Rel>
        <Var>BensComuns</Var>
        <Var>PessoaX</Var>
      </Atom>
      <Atom>
        <Rel>dividirBensComunsMetade</Rel>
        <Var>BensComuns</Var>
        <Var>Metade</Var>
      </Atom>
    </And>
    <Atom>
      <Rel>determinarMeacaoConjugeRegimeParticipacaoAquestosBensComuns</Rel>
      <Var>Meacao</Var>
      <Var>PessoaY</Var>
      <Var>PessoaX</Var>
      <Var>RegimeParticipacaoAquestos</Var>
      <Var>BensComuns</Var>
    </Atom>
  </Implies>

```

- Somar os bens particulares de uma pessoa
- Determinar a quantidade de descendentes de uma pessoa
- Somar a quantidade de descendentes com o cônjuge de uma pessoa
- Dividir os bens particulares de uma pessoa pela soma do número de descendentes e cônjuge

```

<Implies>
  <And>
    <Atom>
      <Rel>somarBensParticulares</Rel>
      <Var>BensParticulares</Var>
      <Var>PessoaX</Var>
    </Atom>
    <Atom>
      <Rel>determinarQuantidadeDescendentes</Rel>
      <Var>Quantidade</Var>
      <Var>PessoaY</Var>
      <Var>PessoaX</Var>
    </Atom>
    <Atom>
      <Rel>somarQuantidadeDescendentesConjuge</Rel>
      <Var>Quantidade</Var>
      <Var>PessoaY</Var>
    </Atom>
    <Atom>
      <Rel>dividirBensParticularesSomaDescendentesConjuge</Rel>
      <Var>BensParticulares</Var>
      <Var>Soma</Var>
      <Var>PessoaY</Var>
      <Var>PessoaZ</Var>
    </Atom>
  </And>
  <Atom>
    <Rel>determinarQuinhaoConjugeDescendentesRegimeParticipacaoAquestosBensParticul
ares</Rel>
    <Var>Quinhao</Var>
    <Var>PessoaY</Var>
    <Var>PessoaZ</Var>
    <Var>RegimeParticipacaoAquestos</Var>
    <Var>BensParticulares</Var>
    <Var>PessoaX</Var>
  </Atom>

```

```

</Atom>
</Implies>

```

- Determinar a meação do cônjuge de uma pessoa no regime de participação nos aquestos sobre os bens comuns
- Determinar a quantidade de descendentes de uma pessoa
- Dividir a meação do cônjuge pela quantidade de descendentes
- Determinar o quinhão dos descendentes no regime de participação nos aquestos sobre os bens comuns de uma pessoa

```

<Implies>
  <And>
    <Atom>
      <Rel>determinarMeacaoConjugeRegimeParticipacaoAquestosBensComuns</Rel>
      <Var>Meacao</Var>
      <Var>PessoaY</Var>
      <Var>PessoaX</Var>
      <Var>RegimeParticipacaoAquestos</Var>
      <Var>BensComuns</Var>
    </Atom>
    <Atom>
      <Rel>determinarQuantidadeDescendentes</Rel>
      <Var>Quantidade</Var>
      <Var>PessoaY</Var>
      <Var>PessoaX</Var>
    </Atom>
    <Atom>
      <Rel>dividirMeacaoDescendentes</Rel>
      <Var>Meacao</Var>
      <Var>Quantidade</Var>
    </Atom>
  </And>
  <Atom>
    <Rel>determinarQuinhaoDescendentesRegimeParticipacaoAquestosBensComuns</Rel>
    <Var>Quinhao</Var>
    <Var>PessoaY</Var>
    <Var>PessoaX</Var>
    <Var>RegimeParticipacaoAquestos</Var>
    <Var>BensComuns</Var>
  </Atom>
</Implies>

```

- Uma pessoa tem descendentes
- Determinar o quinhão do cônjuge em concorrência com os descendentes no regime de participação nos aquestos sobre os bens particulares de uma pessoa
- Uma pessoa tem cônjuge
- Determinar a meação do cônjuge de uma pessoa no regime de participação nos aquestos sobre os bens comuns
- Determinar o quinhão dos descendentes no regime de participação nos aquestos sobre os bens comuns de uma pessoa
- Partilhar os bens de uma pessoa para os descendentes e o cônjuge sobre o regime de participação nos aquestos

<Implies>

<And>

<Atom>

<Rel>**temDescendentes**</Rel>

<Var>**PessoaX**</Var>

<Var>**PessoaY**</Var>

</Atom>

<Atom>

<Rel>**determinarQuinhaoConjugeDescendentesRegimeParticipacaoAquestosBensParticulares**</Rel>

<Var>**Quinhao**</Var>

<Var>**PessoaY**</Var>

<Var>**PessoaZ**</Var>

<Var>**RegimeParticipacaoAquestos**</Var>

<Var>**BensParticulares**</Var>

<Var>**PessoaX**</Var>

</Atom>

<Atom>

<Rel>**temConjuge**</Rel>

<Var>**PessoaX**</Var>

<Var>**PessoaY**</Var>

</Atom>

<Atom>

<Rel>**determinarMeacaoConjugeRegimeParticipacaoAquestosBensComuns**</Rel>

<Var>**Meacao**</Var>

<Var>**PessoaY**</Var>

<Var>**PessoaX**</Var>

<Var>**RegimeParticipacaoAquestos**</Var>

<Var>**BensComuns**</Var>

</Atom>

<Atom>

<Rel>**temDescendentes**</Rel>

<Var>**PessoaX**</Var>

<Var>**PessoaY**</Var>

</Atom>

<Atom>

<Rel>**determinarQuinhaoDescendentesRegimeParticipacaoAquestosBensComuns**</Rel>

<Var>**Quinhao**</Var>

<Var>**PessoaY**</Var>

<Var>**PessoaX**</Var>

<Var>**RegimeParticipacaoAquestos**</Var>

<Var>**BensComuns**</Var>

</Atom>

</And>

<Atom>

<Rel>**partilharBensDescendentesConjugeRegimeParticipacaoAquestos**</Rel>

<Var>**Bens**</Var>

<Var>**PessoaX**</Var>

<Var>**PessoaY**</Var>

<Var>**PessoaZ**</Var>

<Var>**RegimeParticipacaoAquestos**</Var>

</Atom>

</Implies>

- Somar os bens comuns de uma pessoa
- Dividir os bens comuns pela metade
- Determinar a meação do cônjuge de uma pessoa no regime de separação obrigatória sobre os bens comuns

<Implies>

<And>

<Atom>

<Rel>**somarBensComuns**</Rel>

<Var>**BensComuns**</Var>

<Var>**PessoaX**</Var>

</Atom>

<Atom>

<Rel>**dividirBensComunsMetade**</Rel>

<Var>**BensComuns**</Var>

<Var>**Metade**</Var>

</Atom>

</And>

```
<Atom>  
  <Rel>determinarMeacaoConjugeRegimeSeparacaoObrigatoriaBensComuns</Rel>  
  <Var>Meacao</Var>  
  <Var>PessoaY</Var>  
  <Var>PessoaX</Var>  
  <Var>RegimeSeparacaoObrigatoria</Var>  
  <Var>BensComuns</Var>  
</Atom>  
</Implies>
```

APÊNDICE D: Relacionamentos não taxonômicos da ontologia de aplicação “InheritanceLaw”.

Domínio	Relação	Imagem
Pessoa	calcularHeranca	Heranca
Pessoa	determinarAvos	Pessoa
Pessoa	determinarBensCompanheiro	Bens
Pessoa	determinarBensConjuge	Bens
Pessoa	determinarBiSavos	Pessoa
Pessoa	determinarGenitor	Pessoa
Pessoa	determinarHeranca	Heranca
Pessoa	identificarHerdeiros	Pessoa
Pessoa	partilharBensAscendentesPrimeiroGrauConjuge	Bens
Pessoa	partilharBensAscendentesSegundoGrauConjuge	Bens
Pessoa	partilharBensAscendentesTerceiroGrauConjuge	Bens
Pessoa	partilharBensColateraisGermanos	Bens
Pessoa	partilharBensCompanheiroAscendentesConjugeColaterais	Bens
Pessoa	partilharBensCompanheiroFilhosComuns	Bens
Pessoa	partilharBensCompanheiroFilhosInComuns	Bens
Pessoa	partilharBensConjuge	Bens
Pessoa	partilharBensDescendentesConjugeRegimeParcial	Bens
Pessoa	partilharBensDescendentesConjugeRegimeParticipacaoAquestos	Bens
Pessoa	partilharBensDescendentesConjugeRegimeSeparacaoObrigatoria	Bens
Pessoa	partilharBensDescendentesConjugeRegimeSeparacaoVoluntaria	Bens
Pessoa	partilharBensDescendentesConjugeRegimeUniversal	Bens
Pessoa	temAscendentes	Pessoa
Pessoa	temAvo	Pessoa
Pessoa	temAvoh	Pessoa
Pessoa	temBensComuns	BensComuns
Pessoa	temBensParticulares	BensParticulares
Pessoa	temBiSavo	Pessoa
Pessoa	temBiSavoh	Pessoa
Pessoa	temColaterais	Pessoa
Pessoa	temColateraisGermanos	Pessoa
Pessoa	temColateraisUnilaterais	Pessoa
Pessoa	temCompanheiro	Pessoa
Pessoa	temConjuge	Pessoa
Pessoa	temDescendentes	Pessoa
Pessoa	temFilhoComum	Pessoa
Pessoa	temFilhoIncomum	Pessoa
Pessoa	temMae	Pessoa
Pessoa	temPai	Pessoa

APÊNDICE E: Propriedades da ontologia de aplicação “InheritanceLaw”.

Domínio	Propriedade	Imagem
Pessoa	determinarMeacaoConjugeRegimeParcialBensComuns	Float
Pessoa	determinarMeacaoConjugeRegimeParticipacaoAquestosBensComuns	Float
Pessoa	determinarMeacaoConjugeRegimeSeparacaoObrigatoriaBensComuns	Float
Pessoa	determinarMeacaoConjugeRegimeUniversal	Float
Pessoa	determinarQuantidadeAscendentes	Int
Pessoa	determinarQuantidadeAscendentesPrimeiroGrau	Int
Pessoa	determinarQuantidadeAscendentesSegundoGrau	Int
Pessoa	determinarQuantidadeAscendentesTerceiroGrau	Int
Pessoa	determinarQuantidadeColaterais	Int
Pessoa	determinarQuantidadeColateraisGermanos	Int
Pessoa	determinarQuantidadeColateraisUnilaterais	Int
Pessoa	determinarQuantidadeDescendentes	Int
Pessoa	determinarQuantidadeFilhosComuns	Int
Pessoa	determinarQuantidadeFilhosIncomuns	Int
Pessoa	determinarQuinhaoAscendentesCompanheiroBens	Float
Pessoa	determinarQuinhaoAscendentesPrimeiroGrauConjuge	Float
Pessoa	determinarQuinhaoAscendentesSegundoGrauConjuge	Float
Pessoa	determinarQuinhaoAscendentesTerceiroGrauConjuge	Float
Pessoa	determinarQuinhaoColateraisCompanheiroBens	Float
Pessoa	determinarQuinhaoColateraisGermanos	Float
Pessoa	determinarQuinhaoCompanheiroFilhosComunsBensComuns	Float
Pessoa	determinarQuinhaoConjugeAscendentesPrimeiroGrau	Float
Pessoa	determinarQuinhaoConjugeAscendentesSegundoGrau	Float
Pessoa	determinarQuinhaoConjugeAscendentesTerceiroGrau	Float
Pessoa	determinarQuinhaoConjugeCompanheiroBens	Float
Pessoa	determinarQuinhaoConjugeDescendentesRegimeParcialBensParticulares	Float
Pessoa	determinarQuinhaoConjugeDescendentesRegimeParticipacaoAquestosBensParticulares	Float
Pessoa	determinarQuinhaoConjugeDescendentesRegimeSeparacaoVoluntaria	Float
Pessoa	determinarQuinhaoDescendentesConjugeRegimeSeparacaoVoluntaria	Float
Pessoa	determinarQuinhaoDescendentesRegimeParcialBensComuns	Float
Pessoa	determinarQuinhaoDescendentesRegimeParticipacaoAquestosBensComuns	Float
Pessoa	determinarQuinhaoDescendentesRegimeSeparacaoObrigatoriaBensComuns	Float
Pessoa	determinarQuinhaoDescendentesRegimeSeparacaoObrigatoriaBensParticulares	Float
Pessoa	determinarQuinhaoDescendentesRegimeUniversal	Float
Pessoa	determinarQuinhaoFilhosComunsCompanheiroBensComuns	Float
Pessoa	determinarQuinhaoFilhosComunsCompanheiroBensParticulares	Float
Pessoa	determinarQuinhaoFilhosIncomunsCompanheiroBensComuns	Float
Pessoa	determinarQuinhaoFilhosIncomunsCompanheiroBensParticulares	Float
Pessoa	determinarQuotaCompanheiroAscendentesConjugeColateraisBens	Float
Pessoa	determinarQuotaCompanheiroFilhosIncomunsBensComuns	Float
Pessoa	dividirBensAscendentesConjuge	Float
Pessoa	dividirBensComunsFilhosComunsCompanheiro	Float
Pessoa	dividirBensComunsMetade	Float
Pessoa	dividirBensComunsQuantidadeFilhosInComuns	Float
Pessoa	dividirBensMetade	Float

Pessoa	dividirBensParticularesQuantidadeDescendentes	Float
Pessoa	dividirBensParticularesQuantidadeFilhosComuns	Float
Pessoa	dividirBensParticularesQuantidadeFilhosInComuns	Float
Pessoa	dividirBensParticularesSomaDescendentesConjuge	Float
Pessoa	dividirBensQuantidadeColateraisGermanos	Float
Pessoa	dividirBensQuantidadeDescendentesConjuge	Float
Pessoa	dividirBensSubtraidosQuantidadeAscendentes	Float
Pessoa	dividirBensSubtraidosQuantidadeColaterais	Float
Pessoa	dividirMeacaoDescendentes	Float
Pessoa	dividirMetadeBensConjuge	Float
Pessoa	dividirMetadeBensQuantidadeAscendentes	Float
Pessoa	dividirMetadeQuinhaoFilhosIncomuns	Float
Pessoa	obterTercaParteBens	Float
Pessoa	somarBens	Float
Pessoa	somarBensComuns	Float
Pessoa	somarBensParticulares	Float
Pessoa	somarQuantidadeAscendentesConjuge	Int
Pessoa	somarQuantidadeDescendentesConjuge	Int
Pessoa	somarQuantidadeFilhosComunsCompanheiro	Int
Pessoa	somarTodosBensComunsBensParticulares	Float
Pessoa	subtrairBensQuotaCompanheiro	Float
Pessoa	temRegimeParcial	Boolean
Pessoa	temRegimeParticipacaoAquestos	Boolean
Pessoa	temRegimeSeparacaoObrigatoria	Boolean
Pessoa	temRegimeSeparacaoVoluntaria	Boolean
Pessoa	temRegimeUniversal	Boolean
Pessoa	verificarAscendentes	Boolean
Pessoa	verificarAscendentesPrimeiroGrau	Boolean
Pessoa	verificarAscendentesSegundoGrau	Boolean
Pessoa	verificarAscendentesTerceiroGrau	Boolean
Pessoa	verificarAvo	Boolean
Pessoa	verificarAvoh	Boolean
Pessoa	verificarBiSavo	Boolean
Pessoa	verificarBiSavó	Boolean
Pessoa	verificarColaterais	Boolean
Pessoa	verificarColateraisGermanos	Boolean
Pessoa	verificarCompanheiro	Boolean
Pessoa	verificarConjuge	Boolean
Pessoa	verificarDescendentes	Boolean
Pessoa	verificarFilhosComuns	Boolean
Pessoa	verificarFilhosIncomuns	Boolean
Pessoa	verificarRegimeParcial	Boolean
Pessoa	verificarRegimeParticipacaoAquestos	Boolean
Pessoa	verificarRegimeSeparacaoObrigatoria	Boolean
Pessoa	verificarRegimeSeparacaoVoluntaria	Boolean
Pessoa	verificarRegimeUniversal	Boolean