

# Uma Técnica Orientada por Objetivos para a Construção de Ontologias de Aplicação

Luis Eduardo Santos<sup>1</sup>, Rosario Girardi<sup>1</sup>

<sup>1</sup>DEINF/GESEC – Universidade Federal do Maranhão (UFMA)  
Av. dos Portugueses, s/n, Campus Bacanga, CEP 65080-040 – São Luís – MA – Brasil

luis.php89@hotmail.com, rgirardi@deinf.ufma.br

**Abstract.** *Ontologies are knowledge representation structures capable of expressing a set of entities in a domain, their relationships and axioms. For their construction, several techniques have been proposed. However, few of them support the development of application ontologies. This paper proposes GAODT, a goal-oriented technique for the development of application ontologies. Its evaluation was performed through the development of an application ontology, which is the main component of a knowledge-based system to support decision making in the domain of Inheritance Law.*

**Resumo.** *As ontologias são estruturas de representação do conhecimento capazes de expressar um conjunto de entidades em um domínio, seus relacionamentos e axiomas. Para a sua construção, diversas técnicas têm sido propostas. Entretanto, poucas dessas apoiam o desenvolvimento de ontologias de aplicação. Este artigo propõe GAODT, uma técnica orientada por objetivos para o desenvolvimento de ontologias de aplicação. Sua avaliação foi realizada através do desenvolvimento de uma ontologia de aplicação, que é o principal componente de um sistema baseado em conhecimento para apoio a tomada de decisões no domínio do Direito Sucessório.*

## 1. Introdução

As ontologias 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) para representar e compartilhar conhecimento de um determinado domínio de aplicação. As ontologias permitem um processamento semântico da informação e uma interpretação mais precisa dos dados, fornecendo maior efetividade e usabilidade [Girardi 2010].

As ontologias são classificadas quanto a sua generalidade, como: de alto-nível, domínio, tarefa e aplicação [Guarino 1998]. 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. Essa hierarquização 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 [Girardi 2010].

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

Este artigo apresenta GAODT (“Goal-Oriented Application Ontology Development Technique”), uma técnica orientada por objetivos para a construção de ontologias de aplicação. A técnica foi inicialmente avaliada através do desenvolvimento de uma ontologia de aplicação, que representa o conhecimento 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 [Gonçalves 2009].

Este artigo está organizado da seguinte forma: a seção 2 apresenta a técnica proposta, os atores envolvidos, seu ciclo de atividades, as subatividades compreendidas em cada atividade e os artefatos gerados; a aplicação da técnica é ilustrada com exemplos obtidos na construção de uma ontologia de aplicação no domínio do Direito Sucessório; a seção 3 resume os principais trabalhos relacionados com a técnica proposta, abordando alguns critérios comparativos e, por fim, a seção 4 conclui o artigo.

## **2. A Técnica GAODT**

A Figura 1 apresenta uma visão geral da técnica para a construção de ontologias de aplicação. Ela consiste de quatro atividades: “Seleção dos Objetivos e Fatos”, “Representação dos Predicados em Lógica de Primeira Ordem (LPO) [Russel e Norvig 2004]”, “Especificação dos Axiomas em LPO” e “Especificação/Extensão da Ontologia de Aplicação”.

Para as atividades serem desenvolvidas, é necessária a participação do desenvolvedor da ontologia de aplicação e do especialista do domínio. 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. Por exemplo, “Calcular a herança de uma pessoa” seria um objetivo e “Todo bem possui um valor” seria um fato, ou seja, alguma afirmação sobre o domínio. Em posse dessa lista, através da atividade “Seleção dos Objetivos e Fatos”, o desenvolvedor define quais destes são utilizados como entrada para a próxima atividade, a partir do diálogo com o especialista do domínio. Na atividade “Representação dos Predicados em LPO”, o desenvolvedor especifica os objetivos e fatos selecionados anteriormente e que estão em linguagem natural em predicados na LPO. A atividade “Especificação dos Axiomas em LPO”, tem como entrada os predicados especificados na atividade anterior. O desenvolvedor especifica as regras que permitirão alcançar os objetivos do sistema. Essa atividade é iterativa, ou seja, um objetivo (predicado) pode necessitar que outros subobjetivos (predicados) sejam alcançados para que ele possa ser satisfeito. Para isso, esse objetivo é retornado para a primeira atividade “Seleção dos Objetivos e Fatos” a fim de buscar os subobjetivos e fatos que irão satisfazê-lo. Forma-se então um ciclo iterativo de

identificação e especificação de objetivos que vai se desenvolvendo e tendo a sua decomposição final expressa em fatos. Por exemplo, o objetivo “Determinar os ascendentes de uma pessoa” para ser satisfeito, necessita que outros subobjetivos sejam alcançados, como “Determinar o genitor de uma pessoa”, e, pela sua vez, para que este objetivo seja atingido ele precisa de fatos como: “Toda pessoa possui pai” que retornaria para o objetivo “O pai dessa pessoa”. Por fim, a atividade “Especificação/Extensão da Ontologia de Aplicação” recebe como entrada os axiomas da atividade anterior e extrai deles os elementos necessários para compor a ontologia de aplicação. Após a criação da ontologia de aplicação, é possível estendê-la, realizando uma busca semântica por ontologias de aplicação em algum repositório de ontologias. Nas próximas subseções são detalhadas as atividades da técnica GAODT.

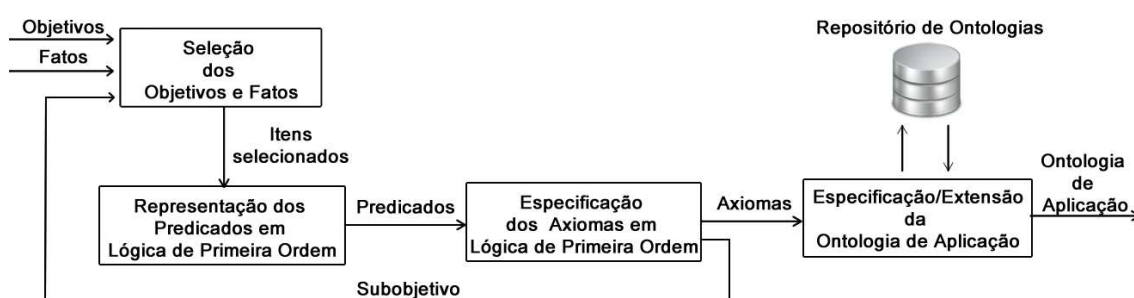


Figura 1. Visão geral da técnica GAODT

## 2.1. Seleção dos Objetivos e Fatos

Essa atividade recebe como entrada a lista de todos os objetivos e fatos do sistema, fornecidos pelo especialista do domínio. A partir dessa lista, o desenvolvedor junto ao especialista define quais objetivos e fatos serão dados como entrada para a próxima atividade. Para uma melhor compreensão da técnica, apresenta-se um exemplo de sua aplicação, considerando-se como entrada uma lista de objetivos e fatos (Tabela 1).

Primeiramente, deve ser definido o objetivo geral e os principais objetivos específicos do sistema. As próximas interações nessa atividade ocorrem quando a atividade “Especificação dos Axiomas em LPO” retornar um subobjetivo onde o desenvolvedor buscará na lista de objetivos e fatos aqueles que irão satisfazê-lo.

Tabela 1. Lista de objetivos e fatos da ontologia exemplo desenvolvida

1	Calcular a herança de uma pessoa	13	Toda pessoa possui mãe
2	Determinar os herdeiros de uma pessoa	14	Toda pessoa possui pai
3	Determinar a herança dos herdeiros	15	Determinar os bens de uma pessoa
4	Determinar os descendentes de uma pessoa	16	Determinar a meação dos herdeiros
5	Determinar os ascendentes de uma pessoa	17	Determinar o quinhão de uma pessoa
6	Determinar o cônjuge de uma pessoa	18	Determinar os bens de uma pessoa
7	Determinar os colaterais de uma pessoa	19	Determinar os bens particulares de uma pessoa
8	Determinar os filhos de uma pessoa	20	Determinar os bens comuns de uma pessoa
9	Determinar os avós de uma pessoa	21	Determinar o regime de uma pessoa
10	Determinar o genitor de uma pessoa	22	Determinar os tios de uma pessoa
11	Determinar os irmãos de uma pessoa	23	Todos os bens comuns têm valor
12	Determinar os primos de uma pessoa	24	Todos os bens particulares têm valor

Como mencionado, na primeira interação desta atividade, o desenvolvedor e o especialista de domínio devem definir o objetivo geral e os objetivos específicos principais do sistema. A partir da lista definida, o objetivo geral selecionado é o item 1: “Calcular a herança de uma pessoa”. E os objetivos específicos principais são os itens 2: “Determinar os herdeiros de uma pessoa” e 3: “Determinar a herança dos herdeiros”.

## 2.2. Representação dos Predicados em LPO

Esta atividade consiste na tradução dos itens selecionados na atividade anterior, expressos em linguagem natural para predicados em LPO. É composta de sete subatividades: “Identificação das entidades”, “Redefinição das entidades”, “Identificação dos relacionamentos”, “Redefinição dos relacionamentos”, “Definição da aridade”, “Definição dos predicados”, e por fim, “Redefinição das entidades do predicado”.

Na subatividade “Identificação das entidades”, a partir dos itens selecionados na Tabela 1 é realizada a identificação das entidades presentes. São consideradas entidades todos os sujeitos e objetos, implícitos ou não. O resultado desta subatividade é apresentado na Tabela 2.

**Tabela 2. Entradas e saídas da subatividade “Identificação das entidades”**

Nº	Itens selecionados	Entidades
1	Calcular a herança de uma pessoa	Herança, Pessoa
2	Determinar os herdeiros de uma pessoa	Herdeiros, Pessoa
3	Determinar a herança dos herdeiros	Herdeiros, Herança

A subatividade “Redefinição das entidades” leva em consideração as entidades identificadas na Tabela 2, para verificar se aquilo identificado como entidade é na verdade uma relação. Verifica-se que a entidade “Herdeiros” na verdade é uma relação entre “Pessoas”, ou seja, “Uma pessoa é herdeiro de outra pessoa”. Então, “Herdeiros” é redefinida como “Pessoa”, considerando as entidades integrantes da relação. Entretanto, a palavra “Herdeiros” não é descartada, pois será útil na subatividade de “Redefinição dos relacionamentos”. A Tabela 3 apresenta o resultado desta subatividade.

**Tabela 3. Entradas e saídas da subatividade “Redefinição das entidades”**

Nº	Entidades	Redefinição das entidades
1	Herança, Pessoa	Herança, Pessoa
2	Herdeiros, Pessoa	Pessoa, Pessoa
3	Herdeiros, Herança	Pessoa, Herança

A subatividade “Identificação dos relacionamentos” utiliza-se dos itens selecionados na Tabela 1. Para identificar os relacionamentos contidos, consideram-se os verbos presentes. A Tabela 4 apresenta os relacionamentos identificados.

**Tabela 4. Entradas e saídas da subatividade “Identificação dos relacionamentos”**

Nº	Itens selecionados	Relacionamentos
1	Calcular a herança de uma pessoa	Calcular
2	Determinar os herdeiros de uma pessoa	Determinar
3	Determinar a herança dos herdeiros	Determinar

A subatividade “Redefinição dos relacionamentos” leva em consideração os relacionamentos identificados na Tabela 4, a fim de verificar se esses relacionamentos

são verbos transitivos, pois os mesmos necessitam de um complemento para terem sentido. Por exemplo, o relacionamento “Determinar”, necessita de um complemento, logo, acrescenta-se como complemento aquilo que lhe der mais sentido, utilizando-se as suas respectivas entidades identificadas na Tabela 3 ou as palavras que eram consideradas entidades na primeira subatividade, por exemplo, a palavra “Herdeiros”. A Tabela 5 apresenta o resultado desta subatividade aplicada a cada relacionamento.

**Tabela 5. Entradas e saídas da subatividade “Redefinição dos relacionamentos”**

Nº	Relacionamentos	Entidades/Palavras redefinidas	Redefinição dos relacionamentos
1	Calcular	Herança	calcularHerança
2	Determinar	Herdeiros	determinarHerdeiros
3	Determinar	Herança	determinarHerança

A subatividade “Definição da aridade” consiste em definir a quantidade de entidades que os relacionamentos identificados anteriormente englobam. Essa quantidade é definida pelo número de entidades identificadas em cada item selecionado. A Tabela 6 apresenta a aridade identificada nos itens 1, 2 e 3.

**Tabela 6. Entradas e saídas da subatividade “Definição da aridade”**

Nº	Relacionamentos	Entidades	Aridade
1	calcularHerança	Herança, Pessoa	2
2	determinarHerdeiros	Pessoa, Pessoa	2
3	determinarHerança	Pessoa, Herança	2

A subatividade “Definição dos predicados” realiza a representação das entidades e relacionamentos identificados nas Tabelas 3 e 5, para cada um dos itens selecionados. A Tabela 7 apresenta os predicados definidos em LPO resultantes da realização desta subatividade.

**Tabela 7. Representação dos itens selecionados em predicados em LPO**

Nº	Itens selecionados	Predicados
1	Calcular a herança de uma pessoa	calcularHerança(Pessoa,Herança)
2	Determinar os herdeiros de uma pessoa	determinarHerdeiros(Pessoa,Pessoa)
3	Determinar a herança dos herdeiros	determinarHerança(Pessoa,Herança)

A subatividade “Redefinição das entidades do predicado” utiliza-se dos predicados definidos na Tabela 7. Quando um predicado possui em seus argumentos entidades com o mesmo nome, por exemplo, “determinarHerdeiros(Pessoa,Pessoa)”, essas duas entidades são consideradas variáveis. Logo, por se tratarem de “Pessoas” distintas devem ser representadas por variáveis diferentes. O predicado é redefinido para “determinarHerdeiros(PessoaX,PessoaY)” sendo essa alteração também propagada para todos os outros predicados na Tabela 7. A Tabela 8 apresenta o resultado desta subatividade e o produto final dessa atividade.

**Tabela 8. Entradas e saídas da subatividade “Redefinição das entidades do predicado”**

Nº	Predicado	Predicado redefinido
1	calcularHerança(Pessoa,Herança)	calcularHerança(PessoaX,Herança)
2	determinarHerdeiros(Pessoa,Pessoa)	determinarHerdeiros(PessoaX,PessoaY)
3	determinarHerança(Pessoa,Herança)	determinarHerança(PessoaY,Herança)

## 2.2. Especificação dos Axiomas em LPO

O objetivo desta atividade é especificar as regras que permitirão alcançar os objetivos do sistema, objetivos estes expressos como predicados em LPO resultantes da atividade anterior “Representação dos Predicados em LPO”. O processamento é iterativo, pois, há uma iteração com a atividade “Seleção dos Objetivos e Fatos”. Essa iteração é necessária porque um objetivo (predicado) contido em uma regra precisa buscar na lista apresentada na Tabela 1 os objetivos e fatos necessários para satisfazê-lo.

Esta atividade é composta de quatro subatividades: “Definição da condição e conclusão”, “Definição dos operadores booleanos”, “Definição dos quantificadores”, por fim, “Definição de implicação ou equivalência”.

A subatividade “Definição da condição e conclusão” consiste na definição de quais predicados é a condição e a conclusão. A conclusão é o objetivo principal que se pretende alcançar e a condição pode ser considerada como um conjunto de premissas ou subobjetivos para alcançar o objetivo principal. Essa subatividade recebe como entrada os predicados identificados na Tabela 8. A Tabela 9 apresenta o resultado desta subatividade.

**Tabela 9. Saída da subatividade “Definição da condição e conclusão do axioma”**

Condição e predicados que a integram	Conclusão
determinarHerdeiro(PessoaX,PessoaY)	calcularHerança(PessoaX,Herança)
determinarHerança(PessoaY,Herança)	

A subatividade “Definição dos operadores booleanos” recebe como entrada os predicados que representam as premissas integrantes da condição (Tabela 9), e tem como objetivo especificar a condição da regra de inferência, ligando as diferentes premissas através de operadores booleanos. Os operadores utilizados são a conjunção representada pelo símbolo ( $\wedge$ ), a disjunção representada pelo símbolo ( $\vee$ ) e a negação representada pelo símbolo ( $\neg$ ).

Para a escolha dos operadores é necessário analisar se um predicado necessita do outro para satisfazer seu objetivo, quando isso ocorrer, utiliza-se a conjunção, quando não, utiliza-se a disjunção, e a negação quando se deseja tornar inválido algum predicado. Por exemplo, para atingir o objetivo de “Calcular a herança de uma pessoa” (calcularHerança(PessoaX,Herança)), é preciso que se satisfaçam as premissas “Determinar os herdeiros de uma pessoa” (determinarHerdeiros(PessoaX,PessoaY)) e também “Determinar a herança dos herdeiros” (determinarHerança(PessoaY,Herança)). Portanto, utiliza-se a conjunção para estes predicados.

A subatividade “Definição dos quantificadores” recebe como entrada o axioma definido na subatividade anterior e define os quantificadores presentes no axioma. Os quantificadores utilizados são o universal ( $\forall$ ) e o existencial ( $\exists$ ). O quantificador universal é utilizado para indicar que um predicado é verdadeiro para todos os elementos de um certo conjunto e o quantificador existencial é utilizado para indicar que um predicado é verdadeiro para pelo menos um elemento num dado conjunto. Por exemplo, a variável “PessoaX” trata de “Pelo menos uma pessoa que faleceu”, sendo definido para essa variável o quantificador existencial. As variáveis “PessoaY” e “Herança” seguem o mesmo princípio, sendo definido a elas o quantificador existencial.

A subatividade “Definição de implicação ou equivalência” recebe como entrada um conjunto de predicados como aqueles do exemplo da Tabela 9, e tem como objetivo determinar se o axioma a ser criado trata-se de uma implicação ou equivalência. A implicação é utilizada quando a satisfação da condição leva a conclusão. Já a equivalência ocorre quando há simetria entre a condição e a conclusão, ou seja, a condição implica na conclusão e a conclusão implica na condição. Assim, no exemplo é utilizada a implicação para formar o seguinte axioma: “ $\exists$  PessoaX, PessoaY, Herança determinarHerdeiros(PessoaX,PessoaY)  $\wedge$  determinarHerança(PessoaY,Herança)  $\Rightarrow$  calcularHerança(PessoaX,Herança)”.

Como mencionado, essa atividade é iterativa, por exemplo, para que o objetivo “determinarHerdeiros(PessoaX,PessoaY)” seja satisfeito, o desenvolvedor da ontologia de aplicação retorna a atividade “Seleção de Objetivos e Fatos” e pergunta ao especialista do domínio quais itens definidos na Tabela 1 satisfazem esse objetivo. O especialista do domínio informa que são os itens 4: “Determinar os descendentes de uma pessoa”, 5: “Determinar os ascendentes de uma pessoa”, 6: “Determinar o cônjuge de uma pessoa” e 7: “Determinar os colaterais de uma pessoa”, esses objetivos são traduzidos para predicados em LPO através da atividade “Representação dos predicados em LPO”. Por fim, esses predicados (subobjetivos) retornam para esta atividade “Especificação dos Axiomas em LPO” formando-se um axioma composto do objetivo principal a ser satisfeito e os subobjetivos que o satisfazem. Dessa forma, a iteração dessa atividade termina até que todos os objetivos sejam satisfeitos.

O produto dessa atividade apresentado na Figura 2 é um conjunto de axiomas que podem ser representados graficamente como uma hierarquia de objetivos, predicados em LPO, desenvolvida utilizando todos os itens da Tabela 1.

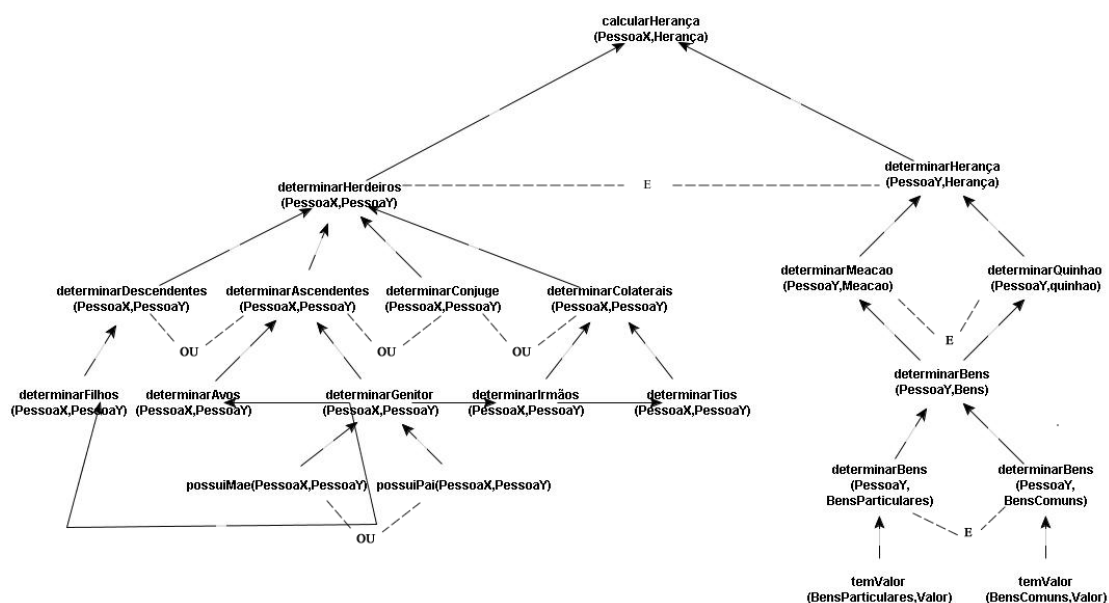


Figura 2. Hierarquia de objetivos e os correspondentes axiomas desenvolvidos

### 2.3. Especificação/Extensão da Ontologia de Aplicação

A partir dos axiomas especificados na atividade anterior, é realizada a extração dos elementos desses axiomas para a construção da ontologia de aplicação. Tendo-a construído, se houver necessidade ainda é possível estendê-la. Esta atividade é

composta de cinco subatividades: “Tradução dos axiomas”, “Definição das classes”, “Definição dos relacionamentos taxonômicos”, “Definição dos relacionamentos não taxonômicos”, “Definição das propriedades”, por fim, “Recuperação de ontologias de aplicação”.

A subatividade “Tradução dos axiomas” consiste em utilizar os axiomas da atividade anterior e traduzi-los em alguma linguagem de regras para ontologias, por exemplo, RuleML [Harold 2001]. A tradução dos axiomas em LPO é feita para RuleML, a escolha dessa linguagem foi feita com base em sua expressividade, ou seja, possui mais recursos pra representar regras em LPO.

Para demonstrar essa subatividade, leva-se em consideração o axioma desenvolvido na atividade anterior “ $\exists$  PessoaX, PessoaY, Herança determinarHerdeiros(PessoaX,PessoaY)  $\wedge$  determinarHerança(PessoaY,Herança)  $\Rightarrow$  calcularHerança(PessoaX,Herança)”, em posse dessa regra utiliza-se Expressões Regulares [Jargas 2001], por exemplo, “ $\wedge(\backslash w+\backslash(.*)\backslash) \wedge (\backslash w+\backslash(.*)\backslash) \Rightarrow (\backslash w+\backslash(.*)\backslash)$ ” que realiza o casamento e captura do padrão e dos subpadrões da regra. Isso é necessário para converter a regra em LPO para a linguagem POSL [Boley 2004], resultando na seguinte regra “calcularHerança(PessoaX,Herança) :- determinarHerdeiros(PessoaX, PessoaY), determinarHerança(PessoaY,Herança)” muito semelhante com a sintaxe em Prolog, por fim, utiliza-se uma API que converte de POSL para a regra em RuleML apresentada na Figura 3 .

<pre> &lt;Assert&gt;   &lt;Rulebase mapClosure='universal'&gt; &lt;Implies&gt; &lt;And&gt; &lt;Atom&gt;   &lt;Rel&gt;determinarHerdeiros&lt;/Rel&gt;   &lt;Var&gt;PessoaX&lt;/Var&gt;   &lt;Var&gt;PessoaY&lt;/Var&gt; &lt;/Atom&gt; &lt;Atom&gt;   &lt;Rel&gt;determinarHerança&lt;/Rel&gt;   &lt;Var&gt;PessoaX&lt;/Var&gt; </pre>	<pre>     &lt;Var&gt;PessoaY&lt;/Var&gt;   &lt;/Atom&gt; &lt;/And&gt; &lt;Atom&gt;   &lt;Rel&gt;calcularHerança&lt;/Rel&gt;   &lt;Var&gt;PessoaX&lt;/Var&gt;   &lt;Var&gt;PessoaY&lt;/Var&gt; &lt;/Atom&gt; &lt;/Implies&gt; &lt;/Rulebase&gt; &lt;/Assert&gt; </pre>
--	---

**Figura 3. Regra representada em RuleML**

A subatividade “Definição das classes” consiste em extrair as variáveis contidas nos predicados que formam os axiomas da atividade anterior. Por exemplo, o predicado “determinarHerdeiros(PessoaX,PessoaY)” possui as variáveis “PessoaX” e “PessoaY” ambas correspondem a classe “Pessoa”.

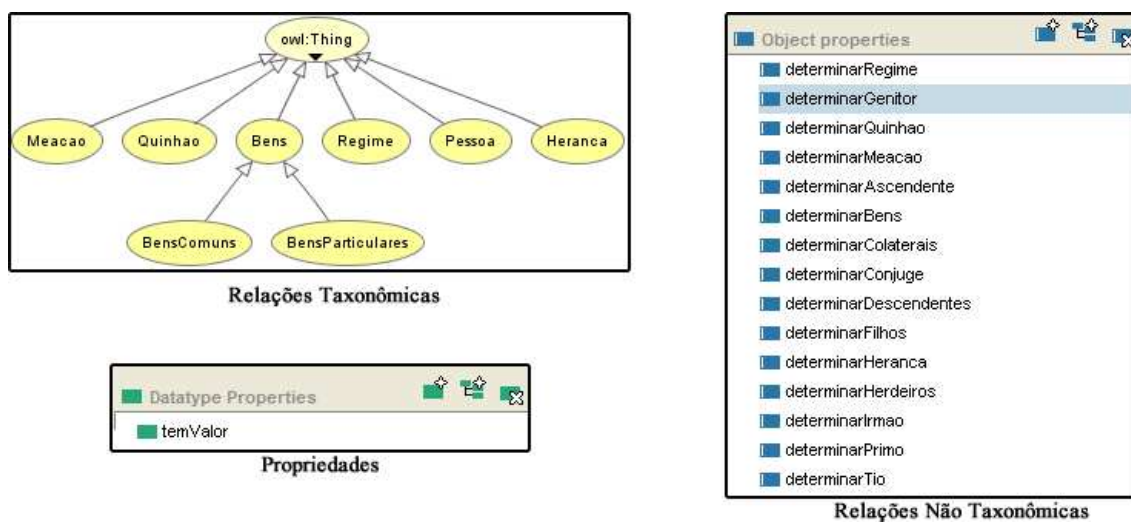
A subatividade “Definição dos relacionamentos taxonômicos” consiste em extrair dos axiomas definidos anteriormente, os predicados que são iguais e que possuem a mesma aridade. Por exemplo, os predicados “determinarBens(PessoaX,Bens)”, “determinarBens(PessoaX,BensComuns)” e “determinarBens(PessoaX,BensParticulares)” possuem o mesmo predicado e aridade “determinarBens/2”. Observa-se que existe uma relação de hierarquia entre as classes, “Bens”, “BensComuns” e “BensParticulares”. A escolha da superclasse, é definida através da classe mais genérica, a classe “Bens”, e as subclasses são definidas a partir das classes mais específicas, “BensComuns” e “BensParticulares”.



A subatividade “Definição dos relacionamentos não taxonômicos” consiste em extrair dos axiomas definidos anteriormente os predicados, sendo ainda necessário definir as variáveis que eles relacionam. Dessa forma, os predicados são as relações não taxonômicas e as variáveis são as classes 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”. Existem predicados que não entram nessa subatividade, para identificá-los é necessário verificar se eles representam características das classes, por exemplo, o predicado “temValor(Bens,Valor)” define que a classe “Bens” possui uma característica que é “temValor”.

A subatividade “Definição das propriedades” consiste em extrair dos axiomas os predicados que descrevem características das classes. Por exemplo, os predicados “temValor(BensComuns,Valor)” e “temValor(BensParticulares,Valor)” descrevem que as classes “BensComuns” e “BensParticulares” possui as características “temValor”.

O produto final dessa atividade e da técnica proposta é apresentado na Figura 4, onde são apresentadas as relações taxonômicas, não taxonômicas e propriedades da ontologia de aplicação desenvolvida como exemplo no domínio do Direito Sucessório. Ressalta-se que as regras não foram apresentadas por serem extensas. Como exemplo, apresenta-se a regra da Figura 3 exibida anteriormente.



**Figura 4. Relações taxonômicas, não taxonômicas e propriedades da ontologia de aplicação desenvolvida**

Por fim, 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. Para realizar a busca semântica será necessário aplicar uma medida de similaridade entre a ontologia de aplicação desenvolvida e as ontologias de aplicação do repositório, com intuito de escolher o melhor casamento. Por fim, realiza-se a integração entre a ontologia de aplicação desenvolvida e a ontologia de aplicação recuperada.

### 3. Trabalhos Relacionados

Varias metodologias e técnicas para a construção de ontologias têm sido apresentadas na literatura, entre elas, a técnica 101 [Noy and McGuinness], DERONTO

[Caliari 2007], Uschold e King [Uschold and King 1995] [Fernández et al. 2004], Gruninger e [Gruniger and Fox 1995] [Fernández et al. 2004] e a metodologia Methontology [Pérez 2004] [Fernández et al. 2004]. Nesta seção apresenta-se uma comparação entre elas e a GAODT (Tabela 10) usando os seguintes critérios:

**O tipo da ontologia que é desenvolvida:** ontologias podem ser de quatro tipos: alto-nível, domínio, tarefa e aplicação. A escolha da técnica deve ser de acordo com o tipo de ontologia a ser desenvolvida. Dentre as técnicas do estado da arte somente GAODT e DERONTO apoiam o desenvolvimento de ontologias de aplicação.

**A ordem de descoberta dos elementos da ontologia:** uma ontologia é composta de seis elementos (classes, taxonomia, relacionamentos, propriedades, axiomas e instâncias) [Girardi 2010]. A ordem na qual esses elementos são descobertos determina a sequência na qual as subatividades devem ser executadas. Por exemplo, GAODT inicia pela descoberta dos axiomas, uma vez que eles representam os requisitos do sistema. Já técnicas como Uschold e King, 101, Gruninger e Fox e Methontology iniciam pela identificação de termos relevantes do domínio os quais correspondem as classes na ontologia.

**O ciclo de vida:** o processo de desenvolvimento de ontologias deve ser preferivelmente incremental, uma vez que o engenheiro do conhecimento pode adicionar novos elementos para a ontologia a cada nova iteração. Exemplos são: DERONTO, 101, Methontology e GAODT.

**Reusabilidade:** boas técnicas e metodologias deveriam considerar o reuso de artefatos existentes. Dentre as abordagens apresentadas apenas DERONTO e Gruninger e Fox não contemplam este aspecto.

**Usa conhecimento linguístico para identificar classes e relacionamentos:** somente GAODT e 101 definem regras para encontrar classes através de substantivos e relacionamentos através de verbos. Esta é uma vantagem por que permite a fácil identificação dos elementos da ontologia através de estratégias linguísticas. Outra vantagem é a possível automatização da extração destes elementos usando técnicas de processamento da linguagem natural.

**Tabela 10. Resultados da análise comparativa entre metodologias e técnicas para a construção de ontologias**

	DERONTO	Uschold e King	101	Methontology	Gruninger e Fox	GAODT
<b>O tipo de ontologia que é desenvolvido</b>	Domínio e Aplicação	Domínio	Domínio	Domínio	Domínio	Aplicação
<b>A ordem de descoberta dos elementos da ontologia</b>	Classes Taxonomia Propriedades Relações Axiomas	Classes Taxonomia Relações	Classes Taxonomia Propriedades Relações Axiomas Instâncias	Classes Taxonomia Relações Propriedades Axiomas Instâncias	Classes Relações Propriedades Instâncias Axiomas Taxonomia	Axiomas Classes Taxonomia Relações Propriedades
<b>O ciclo de vida</b>	Iterativo	Clássico	Iterativo	Evolutivo	Clássico	Iterativo
<b>Reusabilidade</b>	Não	Sim	Sim	Sim	Não	Sim
<b>Usa conhecimento linguístico</b>	Não	Não	Sim	Não	Não	Sim

#### **4. Conclusões e Trabalhos Futuros**

Este artigo apresentou a GAODT, uma técnica para a construção de ontologias de aplicação através de um ciclo de desenvolvimento guiado por objetivos, facilitando assim 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. Esta característica não é abordada por nenhuma das outras técnicas e metodologias apresentadas no estudo comparativo com a técnica proposta.

Como discutido, construir ontologias reusáveis é um processo demorado e oneroso. Por serem utilizadas por aplicações específicas, as ontologias de aplicação são menos reusáveis, entretanto, são mais rápidas e baratas de se desenvolver. Isso revela que construir ontologias de aplicação e depois tornar seus elementos reutilizáveis, generalizando-os para ontologias de domínio e de tarefa, apresenta-se como uma solução bem adequada. Neste contexto, a técnica GAODT contribui com a construção de ontologias de aplicação, oferecendo o primeiro passo para o desenvolvimento desta solução.

GAODT está sendo avaliada através da sua aplicação na construção de uma ontologia de aplicação no domínio do Direito Sucessório, a ser utilizada na construção de um SBC para apoio aos operadores do Direito no processo da Sucessão Legítima.

Entre os trabalhos futuros previstos, planeja-se um maior detalhamento da subatividade “Recuperação de ontologias de aplicação” e a incorporação da técnica GAODT ao processo MADAE-Pro [Costa 2009], um processo baseado no conhecimento para construção de sistemas multiagentes. O propósito é utilizar GAODT na construção das bases de conhecimento dos agentes deliberativos participantes de uma sociedade.

#### **Agradecimentos**

Este trabalho é apoiado pelo CNPq, CAPES e FAPEMA.

#### **Referências**

- Boley H. (2004) “POSL: An Integrated Positional-Slotted Language for Semantic Web Knowledge”. W3C.
- Caliari, F. (2007) “DERONTO: Método para Construção de Ontologias a partir de Diagramas Entidade-Relacionamento”. Dissertação de Mestrado - Universidade Tecnológica Federal do Paraná.
- Costa, A. (2009) “MADAE-Pro Um processo baseado no conhecimento para Engenharia de Domínio e de Aplicações”. Dissertação de Mestrado - Universidade Federal do Maranhão.
- Fernández, M., Pérez, A. and Juristo, N. (1997) “Methontology: From Ontological Art Towards Ontological Engineering”. Spring Symposium Series. Stanford.
- Girardi, R. (2010) “Guiding Ontology Learning and Population by Knowledge System Goals”. In: Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Ed. INSTIIC, Valence, pp. 480 – 484.

- Pérez, A. et al. (2004) "Ontological Engineering: with examples from the areas of knowledge management, e-commerce and the semantic web". London, Springer-Verlag.
- Gonçalves, C. (2009) "Direito Civil brasileiro: Direito das Sucessões", São Paulo, Saraiva.
- Grüninger M, Fox M. (1995) "Methodology for the design and evaluation of ontologies". In: IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing", Montreal, Canada.
- Guarino, N. (1998) "Formal Ontology in Information Systems". Proceedings of the 1st International Conference, Trento, Italy, IOS Press.
- Harold B. (2001) "The Rule Markup Language: RDF-XML Data Model, XML Schema Hierarchy, and XSL Transformations". In Proc. 14th International Conference on Applications of Prolog.
- Jargas, A. (2001) "Expressões Regulares: uma abordagem divertida". Novatec.
- Noy, N. and McGuinness, D. (2001) "Ontology Development 101: A Guide to Creating Your First Ontology".
- Russel, S., Norvig, P. (2004) "Inteligência Artificial". Rio de Janeiro, Elsevier.
- Uschold M and King M. (1995) "Towards a Methodology for Building Ontologies". In: IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing. Montreal, Canada.