Building Application Ontologies through Knowledge System Goals

Luis Eduardo Santos and Rosario Girardi
Federal University of Maranhão, Computer Science Departament, São Luiz, Maranhão, Brazil

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

Abstract: This article presents a case study in building an application ontology for the development of a knowledge-based system in the field of Inheritance Law, developed with GAODT, a goal oriented technique. GAODT proposes the translation of the goals necessary to build a knowledge-based system expressed in natural language to rules in First-order logic, from which the elements that constitute the application ontology (classes, relations, properties and axioms) are extracted. It is also presented a comparative evaluation between GAODT and other state of the art techniques.

1 INTRODUCTION

Ontologies are knowledge representation structures capable of expressing a set of entities in a given domain, their relationships and axioms, being used by modern knowledge-based systems (KBS) as knowledge bases to represent and share knowledge of a particular application domain. They allow semantic processing of information and a more precise interpretation of data, providing greater effectiveness and usability than traditional information systems (Girardi, 2010).

An ontology is classified according to its generality, as high-level, domain, task or application ontology (Guarino, 1998). High-level ontologies describe generic concepts like time and space, independently of a particular domain. Domain ontologies make explicit concepts of a domain and their relationships, for example, the concepts "client", "legal-case" are the relationship "has(client, legal-case)" in the legal field. Task ontologies describe the activities of a domain, for instance, similarity analysis in the information retrieval related activities. Finally, application ontologies are specializations of domain and task ontologies, being used in a particular application, for example, the task relationship "similarity analysis" between the concepts "old legal case" and "new legal case" in a legal information retrieval system. According to Guarino (Guarino, 1998), this hierarchy promotes the reuse of ontologies, i.e., to build application ontologies it is necessary to extend both domain and task ontologies, and these in turn, extend high-level ontologies. However, in practice, building reusable ontologies is a costly process. Therefore, building application ontologies first and then generalizing them to domain and task ontologies is a suitable alternative (Girardi, 2010).

Several techniques have been developed to support the process of ontology construction. However, most of them focus just on the development of domain and task ontologies. Appropriate techniques for the development of application ontologies are needed and the GAODT ("Goal-Oriented Application Ontology Development Technique") technique described in this paper contributes to this goal.

GAODT translates the goals in language natural expressing the requirement of a KBS to rules and facts in First-order logic (FOL) (Russell and Norvig, 2004) and then extracts the elements that constitute the application ontology.

This article describes a case study in building an application ontology using GAODT to support decision making in Intestate Succession, a type of succession disciplined by Inheritance Law. Intestate Succession comprises the set of rules governing the transfer of assets to someone after his death according to the law (Gonçalves, 2009).

The paper is organized as follows. Section 2 presents the case study, emphasizing the advantages of the goal-oriented development cycle adopted by the GAODT technique. Section 3 discusses a comparative evaluation between GAODT and some representative state of the art techniques. Section 4 concludes the paper and points out future work.

Santos L. and Girardi R..
Building Application Ontologies through Knowledge System Goals.
DOI: 10.5220/0004143501150124
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2012), pages 115-124
ISBN: 978-989-8565-30-3

2 A CASE STUDY ABOUT INTESTATE SUCCESSION

The main concepts about Intestate Succession considered in this work are following described.

Intestate Succession is a legal institute that governs the transfer of property of a person by the reason of his/her death without a will. In that case there is set of rules used to determine who will inherit the property.

In the Brazilian law, the following order is applied: First, the descendants (children, grandchildren, and so on) concurring with the spouse; in the absence of descendants, the ascendants (father, mother, grandfather, grandmother, etc.) also concurring with the spouse; if there aren't ascendants, the spouse and finally, in the absence of a spouse, the properties are transmitted to the collaterals (article 1829 of Brazilian Civil Code).

The concurrency of the spouse with descendants or ascendants depends on the matrimonial regime. Matrimonial regime, are systems of property ownership between spouses providing for the creation or absence of a marital estate, and if created, what properties are included in that estate, how and by whom it is managed, and how it will be divided and inherited at the end of the marriage, which can be of four types: Universal Community Regime, Limited Community Regime, Accrual Regime and Separation of Property.

The Universal Community Regime is the union of all pre-marital and marital property of a couple, and when sharing, the surviving spouse does not compete for the inheritance, since half (moiety) of all properties belong to him/her. For example, consider that the late "John" left properties valued at R$ 100,000. Half of this amount belongs to "Mary", his alive wife, and the remaining money will be divided among the other heirs.

In Limited Community Regime, the surviving spouse owns half of the marital properties and still competes for the inheritance of pre-marital properties with the descendants. For example, the late "Peter" left pre-marital properties equivalent to R$ 200,000 and a patrimony made with "Louise" valued at R$ 100,000. The spouse already owns half of the marital properties and will also compete with the descendants for pre-marital properties.

In the present case study only these two regimes will be considered.

2.1 An Overview of the GAODT Technique

Figure 1 illustrates the GAODT process along with its four activities: "Selection of Goals and Facts", "Representation of Predicates in FOL", "Specification of Axioms in FOL" and "Specification/Extension of the Application Ontology".

The developer of the application ontology and the domain expert participate in the execution of the activities. The developer is the knowledge engineer responsible for building the application ontology. The domain expert is someone who has expertise in an area of knowledge.

The technique takes as input a list of all the goals and facts of the system provided by the domain expert. The goals are the requirements that the KBS has to achieve, for instance, "Calculate the inheritance of a person" and the facts are general statements like "A person might have descendants". In the activity "Selection of Goals and Facts", the developer, in consensus with the domain expert, selects the most representative goals and facts to be used as input of next activity. In the activity "Representation of Predicates in FOL", the developer translates the goals and facts in natural language to predicates in FOL.

The activity "Specification of Axioms in FOL" takes as input the predicates specified in the previous activity and specifies in FOL the rules needed to achieve the goals of the system. This activity is iterative, that is, a goal predicate may require the achievement of other subgoals. For example, to satisfy the goal "Determine the ascendants of a person", other subgoals should be achieved, such as "Determine the genitor of a person". All the process is iteratively executed until all the goals have been decomposed and expressed as simple facts. Finally, the activity "Specification/Extension of the Application Ontology" uses axioms generated on the previous activity and extracts from them the necessary elements to compose the application ontology. The created application ontology can be extended by performing a semantic search in a repository of application ontologies. In the next subsections GAODT activities are explained in further detail.

2.2 Selection of Goals and Facts

This activity takes as input a list of all the goals and facts of the system, provided by the domain expert. From this list, the developer and the specialist sets

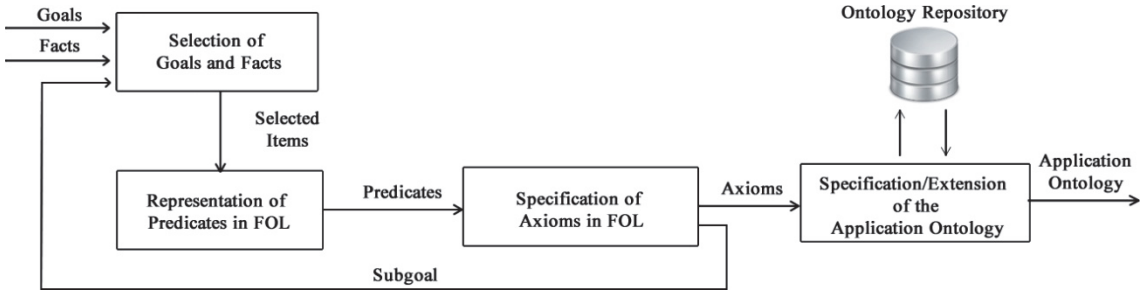

Figure 1: An overview of the GAODT technique.

which of them will be given as input to the next activity. As an example, Table 1 presents a list of goals and facts for an application ontology for a knowledge-based decision support system on Intestate Succession.

Initially it must be defined the general goal and the main specific goals of the system. For instance, for the general goal 1: "Calculate the inheritance of a person", the main specific goals are 2: "Identify the heirs of a person" and 3: "Determine the inheritance of each heir". To satisfy these subgoals, other goals could be defined in subsequent iterations, in a process performed recursively for all the goals in the list.

2.3 Representation of Predicates in FOL

This activity consists in translating the items selected in the previous activity, expressed in natural language to predicates in FOL. It consists of seven sub-activities: "Identification of entities", "Redefinition of entities", "Identification of relationships", "Redefinition of relationships", "Definition of the arity", "Definition of predicates" and "Redefinition of the entities of the predicate" (Figure 2).

In the sub-activity "Identification of entities", all explicit or implicit subjects and objects in a sentence are identified from the items selected in the previous activity and exemplified in Table 1. The result of this sub-activity is shown in Table 2.

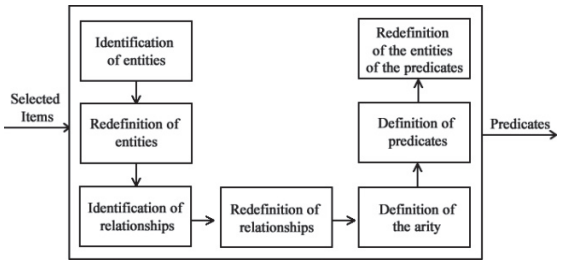

Figure 2: "Representation of Predicates in FOL" activity.

Table 1: List of goals and facts of the ontology.

1	Calculate the inheritance of a person
2	Identify the heirs of a person
3	Determine the inheritance of each heir
4	Identify the spouse of a person
5	Define the inheritance portion of descendants in the universal regime
6	Identify the ascendants of a person
7	Identify the descendants of a person
8	Determine the moiety of the spouse in the universal regime
9	Determine the moiety of the spouse in the limited regime
10	Determine the quantity of descendants of a person
11	Determine the quantity of ascendants of a person
12	Determine the portion of the descendants in the universal regime
13	Determine the portion of the descendants in the limited regime
14	Divide the moiety by the quantity of descendants
15	A person might have descendants
16	A person might have spouse
17	A person might have ascendants
18	A person might have marital properties
19	A person might have pre-marital properties
20	A person might have universal regime
21	A person might have partial regime
22	A marital property has value
23	A pre-marital property has value
24	Assign inheritance portion to spouse and descendants in the universal regime
25	Assign inheritance portion to spouse and descendants in the limited regime
26	Sum all the properties of a person
27	Verify the existence of ascendants
28	Verify the existence of descendants
29	Verify the existence of universal regime
30	Verify the existence of limited regime

The sub-activity "Redefinition of entities" takes into account the entities identified in Table 2, and for each one verifies if that is an entity or a relationship. The entity "Heirs" is actually a

relationship between two “People”, i.e., “A person is heir of another person”. So, “Heirs” is redefined as a “Person”, considering the entities that integrate the relationship. However, the word “Heirs” is not discarded, it will be useful in the sub-activity “Redefinition of relationships”. Table 3 shows the result of this sub-activity.

Table 2: Inputs and outputs of the sub activity “Identification of entities’’.

Selected items	Entities
Calculate the inheritance of a person	Inheritance, Person
Identify the heirs of a person	Heirs, Person
Determine the inheritance of each heir	Heirs, Inheritance

Table 3: Inputs and outputs of the sub-activity “Redefinition of entities”.

Entities	Redefinition of entities
Inheritance, Person	Inheritance, Person
Heirs, Person	Person, Person
Heirs, Inheritance	Person, Inheritance

The sub-activity “Identification of relationships” uses the items selected in the activity “Selection of Goals and Facts” to identify verbs in the phrases, which represent the relationships to be extracted. For instance, in the selected item “Calculate the Inheritance of a person”, the verb “Calculate” is identified as a relationship. Table 4 shows the relationships identified.

Table 4: Inputs and outputs of the sub-activity “Identification of relationships”.

Selected items	Relationships
Calculate the inheritance of a person	Calculate
Identify the heirs of a person	Identify
Determine the inheritance of each heir	Determine

The sub-activity “Redefinition of relationships” takes into account the relationships identified in the previous activity (exemplified in Table 4) and verifies if these relationships are transitive verbs, as they need a supplement to make sense. For example, the relationship “identify” needs a supplement to give it sense, using their respective entities identified in Table 3 or the words that were considered entities in the first sub-activity, for example, the word “Heirs”. Table 5 shows the result of this sub-activity applied to the examples in Table 3 and Table 4.

The sub-activity “Definition of the arity” defines the number of entities involved in the relationships previously identified. This quantity is determined according to by the number of entities identified on each selected item. Table 6 shows the arity identified for the items 1, 2 and 3 in Table.

Table 5: Inputs and outputs of the sub-activity “Redefinition of relationships”.

Relationships	Entities	Redefinition of relationships
Calculate	Inheritance	calculateInheritance
Identify	Heirs	identifyHeirs
Determine	Inheritance	determineInheritance

Table 6: Inputs and outputs of the sub-activity “Definition of the arity”.

Relationships	Entities	Arity
calculateInheritance	Inheritance, Person	2
identifyHeirs	Person, Person	2
determineInheritance	Person, Inheritance	2

In the sub-activity “Definition of predicates” the entities and relationships identified and illustrated in Tables 3 and 5 are represented in FOL. Table 7 presents the predicates resulting from the realization of this sub-activity.

Table 7: An example of translation of the selected items into predicates in FOL.

Selected items	Predicates
Calculate the inheritance of a person	calculateInheritance (Person, Inheritance)
Identify the heirs of a person	identifyHeirs (Person, Person)
Determine the inheritance of each heir	determineInheritance (Person, Inheritance)

The sub-activity “Redefinition of the entities of the predicate” aims at renaming the arguments of the predicates, defined in the sub-activity “Definition of predicates”, when the arguments have the same name. For example, for the predicate “identifyHeirs(Person, Person)”, the entities are considered variables since they represent distinct persons. So it is redefined to “identifyHeirs (PersonX, PersonY)” and this change is also propagated to all other predicates in Table 7. Table 8 presents the result of this sub-activity and the final product of this activity.

Table 8: Inputs and outputs of the sub-activity “Redefinition of the entities of the predicate”.

Predicates	Predicates redefined
calculateInheritance (Person, Inheritance)	calculateInheritance (PersonX, Inheritance)
identifyHeirs (Person, Person)	identifyHeirs (PersonX, PersonY)
determineInheritance (Person, Inheritance)	determineInheritance (PersonY, Inheritance)

2.4 Specification of Axioms in FOL

The purpose of this activity is to specify the rules that lead to the achievement of the goals of the system which are represented as predicates in FOL. The process is iterative, because there is an iteration with the activity "Selection of Goals and Facts". For each goal contained in a rule a search is performed in the list of goals and facts to retrieve the subgoals that satisfy it.

This activity consists of four sub-activities: "Definition of the condition and conclusion", "Definition of Boolean operators", "Definition of quantifiers" and "Definition of implication or equivalence". Figure 3 shows the sub-activities of this activity.

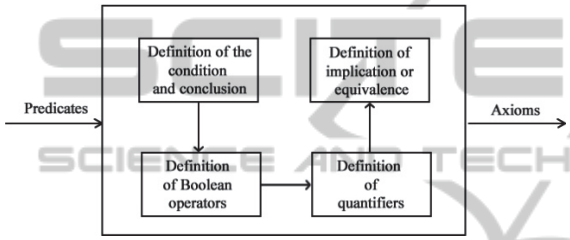

Figure 3: "Specification of axioms in FOL" activity.

The sub-activity "Definition of the condition and conclusion" determines the condition and the conclusion of each rule. The conclusion is the main goal that has to be achieved and the condition can be considered as a set of assumptions or subgoals that lead to the achievement of the main goal. This sub-activity receives as input the predicates identified in Table 8. Table 9 shows the result of this sub-activity.

Table 9: Output of the sub-activity "Definition of the condition and conclusion of the axiom".

Condition and predicates of the axiom	Conclusion
identifyHeir (PersonX, PersonY)	calculateInheritance (PersonX, Inheritance)
determineInheritance (PersonY, Inheritance)	

The sub-activity "Definition of Boolean operators" specifies the Boolean operators which integrate the predicates of the axiom condition. The operators used are the conjunction represented by the symbol (^) and the disjunction represented by the symbol (∨).

Predicates in the condition are joined by an "and" operator when all of them are needed to achieve the conclusion; by an "or" operator when they are alternative predicates to achieve the conclusion. For example, to achieve the goal "Calculate the inheritance of a person" (calculateInheritance(PersonX, Inheritance)), it is necessary to satisfy all the goals "Identify the heirs of a person" (identifyHeirs(PersonX, PersonY)) and "Determine the inheritance of each heir" (determineInheritance(PersonY,Inheritance)). Therefore, a conjunction is used to integrate these two predicates.

The sub-activity "Definition of quantifiers" defines the appropriate quantifiers associated to entities present in the axiom. Quantifiers can be universal (∀) or existential (∃). The first one is used to indicate that a predicate is true for all the elements of a given set while the last one is used to indicate that a predicate is true for at least one element in a given set. For instance, the variable "PersonX" refers to "at least one person who died" so the existential quantifier is associated to this entity. The variables "PersonY" and "Inheritance" follow the same principle, being set to the existential quantifier.

The sub-activity "Definition of implication or equivalence" takes as input a set of predicates like those in the example of Table 9 and to determines whether the axiom to be created is an implication or an equivalence.

The implication is used when the satisfaction of the condition leads to the conclusion. The equivalence occurs when there is a symmetry between the condition and conclusion. For instance, an implication is used to form the following axiom: "∃ PersonX, PersonY, Inheritance | identifyHeirs (PersonX,PersonY) ^ determineInheritance (PersonY,Inheritance) ⇒ calculateInheritance (PersonX,Inheritance)".

As illustrated in Figure 1, the GAODT activities are executed iteratively. Therefore, in order to construct new axioms, each one of the predicates in the condition of the current axiom submitted to the "Selection of Goals and Facts" activity where once more time the items in Table, satisfying this condition will be selected.

For instance, the predicate goal "identifyHeirs (PersonX,PersonY)" which is part of the condition in the axiom example of the previous paragraph is submitted to the "Selection of Goals and Facts" and the domain specialist informs that the items "Verify the existence of descendants", "Identify the descendants of a person", "Verify the existence of spouse", "Identify the spouse of a person" satisfy that goal. Then, new goals specified in natural language are given as input to the activity "Representation of Predicates in FOL" (Table 10)

and a new cycle of the GAODT technique begins.

Table 10: Goals represented as predicates.

Goals in Natural Language	Predicates
Verify the existence of descendants	verifyExistence Descendants (PersonX, PersonY)
Identify the descendants of a person	identifyDescendants (PersonX, PersonY)
Verify the existence of spouse	verifyExistenceSpouse (PersonX, PersonY)
Identify the spouse of a person	identifySpouse (PersonX, PersonY)

These predicates are given as input to the activity "Specification of Axioms in FOL" to generate the new axiom presented in Table 11.

Table 11: Axiom developed in the activity "Specification of Axioms in FOL".

CONDITION
verifyExistenceDescendants(PersonX, PersonY) ^ identifyDescendants(PersonX, PersonY) ^ verifyExistenceSpouse(PersonX, PersonY) ^ identifySpouse(PersonX, PersonY)
CONCLUSION
identifyHeirs(PersonX, PersonY)

The predicate "determineInheritance(PersonY, Inheritance)" that also makes part of the axiom condition pass through the same process of specification and representation in FOL to which the predicate "identifyHeirs(PersonX,PersonY)" was submitted, generating the new axiom in Table 12.

Table 12: Axiom developed in activity "Specification of Axioms in FOL".

CONDITION
verifyExistenceDescendants(PersonX, PersonY) ^ verifyExistenceSpouse(PersonX, PersonY) ^ verifyExistenceUniversalRegime (PersonX,UniversalRegime) ^ assignInheritancePortionDescendants SpouseUniversalRegime (PersonX Inheritance, PersonY, UniversalRegime)
CONCLUSION
determineInheritance(PersonX,PersonY)

The process is then recursively executed for each one of the subgoals, until all the goals given as input to the technique (as the ones illustrated in Table 1) have been satisfied. The product of this activity is a set of axioms specified in predicates in FOL. A sub-set of the axioms generated from the activity "Specification of Axioms in FOL" is shown in Table 13.

2.5 Specification/Extension of the Application Ontology

The constituent elements of the axioms specified in the previous activity are extracted for the construction of the application ontology. This activity consists of six subactivities: "Translation of axioms", "Definition of classes", "Definition of non-taxonomic relationships", "Definition of taxonomic relationships", "Definition of properties" and "Retrieval of application ontologies" (Figure 4).

Table 13: A sub-set of axioms generated from the activity "Specification of Axioms in FOL.

Condition	Conclusion
identifyHeirs(PersonX, PersonY) ^ determineInheritance(PersonX, Inheritance)	calculateInheritance (PersonX, Inheritance)
verifyExistenceDescendant(PersonX, PersonY) ^ verifyExistenceSpouse(PersonX, PersonY) ^ verifyExistenceUniversalRegime(PersonX, UniversalRegime) ^ assignInheritancePortionDescendantsSpouseUniversalRegime (PersonX, Inheritance, PersonY, UniversalRegime)	determineInheritance (PersonX, PersonY)
mightHaveDescendants(PersonX, PersonY)	verifyExistenceDescendants (PersonX, PersonY)
mightHaveSpouse(PersonX, PersonY)	verifyExistenceSpouse (PersonX, PersonY)
verifyExistenceDescendants(PersonX, PersonY) ^ identifyDescendants(PersonX, PersonY) ^ verifyExistenceSpouse(PersonX, PersonY) ^ identifySpouse(PersonX, PersonY) ∨ verifyExistanceAscendants(PersonX, PersonY) ^ identifyAscendants(PersonX, PersonY)	identifyHeirs (PersonX, PersonY)
determinePortionDescendantsUniversalRegime(PersonX, Portion) ^ determineMoietySpouseUniversalRegime(PersonX, PersonY)	assignInheritancePortionDescendants SpouseUniversalRegime (PersonX, Inheritance, PersonY, UniversalRegime)

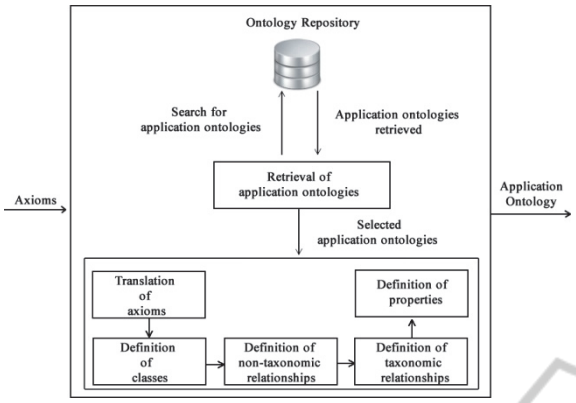

Figure 4: "Specification/extension of the application ontology" activity.

The sub-activity "Translation of axioms" converts the axioms defined in the previous activity expressed in FOL into rules expressed in an ontology rule based language, like RuleML (Harold, 2001). The experiences conducted to evaluate GAODT use RuleML because of its expressiveness.

To perform this translation, the following heuristics are applied. First, regular expressions (Jeffrey, 2001) are used to extract the premises and the conclusions of the axioms. For example, for the rule "∃ PersonX, PersonY, Inheritance | identifyHeirs(PersonX,PersonY) ^ determine Inheritance(PersonY,Inheritance) ⇒ calculate Inheritance(PersonX,Inheritance)", the following regular expression was used "^(\w+\(.*\)) ^ (\w+\(.*\)) ⇒ (\w+\(.*\))". Then, the premises and conclusion are specified in POSL (Boley, 2004) and finally automatically translated to RuleML axioms (Figure 5).

The sub-activity "Definition of classes" extracts the variables of the axioms of the "Specification of Axioms in FOL" activity illustrated in Table 13. For example, the predicate "identifyHeirs (PersonX,PersonY)" has the variables "PersonX" and "PersonY" both referring to the class "Person".

The sub-activity "Definition of non-taxonomic relationships" extracts non-taxonomic relationships of the ontology from the predicates in the list of axioms outputted from the activity "Specification of Axioms in FOL" (Table 13). A non-taxonomic relationship is defined for each predicate having the same name and arity. For example, the predicates "sumProperties(PersonX,Properties)", "sumProperties(PersonX,MaritalProperty)" and "sumProperties (PersonX,Pre-maritalProperty)" have the same name and arity; therefore, the non-taxonomic relationship "sumProperties/2" is defined.

```
<Assert>
 <Rulebase mapClosure='universal'>
<Implies>

<And>
<Atom>
 <Rel> identifyHeirs </Rel>
 <Var>PersonX</Var>
 <Var>PersonY</Var>
</Atom>
<Atom>
 <Rel>determineInheritance</Rel>
 <Var>PersonX</Var>
    <Var>PersonY</Var>
</Atom>
</And>
<Atom>
 <Rel>calculateInheritance</Rel>
 <Var>PersonX</Var>
 <Var>PersonY</Var>
</Atom>

</Implies>
 </Rulebase>
</Assert>
```

Figure 5: Example of an axiom represented in RuleML.

The sub-activity "Definition of taxonomic relationships" extracts a set of taxonomic relationships based on the hierarquical relation between the variables of the predicates outputted from the previous sub-activity. For example, there is a hierarchy between the classes, "Property", "MaritalProperty" and "Pre-maritalProperty", extracted from the predicates "sumProperties (PersonX,Properties)", "sumProperties(PersonX, MaritalProperty)" and "sumProperties(PersonX,Pre-maritalProperty)".

The sub-activity "Definition of properties" extracts from the axioms the predicates describing attributes of the classes. For example, "hasValue(MaritalProperty,Value)" and "hasValue (Pre-maritalProperty,Value)" describe that the classes "MaritalProperty" and "Pre-maritalProperty" has the property "hasValue".

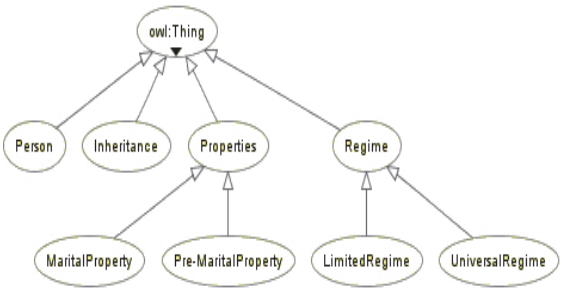

Figure 6: Taxonomic relationships of the Intestate Succession application ontology.

Table 14: A sub-set of the non-taxonomic relationships of the Intestate Succession application ontology.

Domain	Relationship	Range
Person	calculateInheritance	Inheritance
Person	identifyAscendants	Person
Person	identifySpouse	Person
Person	identifyDescendants	Person
Person	determineInheritance	Inheritance
Person	identifyHeirs	Person
Person	mightHaveDescendants	Person
Person	mightHaveMarital	MaritalProperty
Person	mightHave Pre-Marital	Pre-Marital Property
Person	mightHave UniversalRegime	Universal Regime
Person	mightHave LimitedRegime	Limited Regime

Table 15: A sub-set of the properties of the Intestate Succession application ontology.

Domain	Properties
Person	determineQuantityAscendants
Person	determineQuantityDescendants
Person	sumMaritalProperties
Person	determineMoietySpouse UniversalRegime
Person	determinePortionDescendants UniversalRegime
Person	determinePortionDescendants LimitedRegime
Person	determineMoietySpouse LimitedRegime
Pre-MaritalProperty	hasValue
MaritalProperty	hasValue

```
<Assert>
    <Rulebase mapClosure='universal'>
    <Implies>
    <And>
<Atom>
 <Rel>hasValue</Rel>
     <Var>MaritalProperty</Var>
     <Var>Value</Var>
</Atom>
<Atom>
 <Rel> sumMaritalProperties</Rel>
 <Var>Person</Var>
 <Var> MaritalProperty </Var>
</Atom>
 </And>
</Implies>
 </Rulebase>
</Assert>
```

Figure 7: Example of an axiom represented in RuleML.

If there is a need to extend the application ontology developed, the sub-activity "Retrieval of application ontologies" performs a semantic search for reusable ontologies in a repository. Several similarity measures (Lee et al., 2008); (Claudia et al., 2008) can be used to rank the ontologies retrieved.

Figure 6, Figure 7, Table 14 and Table 15 show respectively the taxonomy, an example of an axiom, a sub-set of the non-taxonomic relationships and properties of the application ontology developed for the domain of Intestate Succession, which make up the final product of GAODT.

3 RELATED WORK

Various techniques and methodologies for building ontologies have been proposed like the 101 technique (Noy and McGuinness, 2001), DERONTO (Caliari, 2007), Uschold and King (Uschold and King, 1995) (Fernandez et al., 2004), Gruninger and Fox (Gruninger and Fox, 1995); (Fernandez et al., 2004) and Methontology (Perez, 2004); (Fernandez et al., 2004). Table 16 shows the results of a comparative analysis among them and GAODT technique according to the following criteria: the type of developed ontology, the order through which the ontology elements are discovered, the type of life cycle (classic or iterative), the reuse of existing ontologies and the use linguistic knowledge to find classes and relationships.

Ontologies can be of four types: high-level, domain, task and application ontologies (Guarino, 1998). The chosen technique should be appropriate for the construction of a particular type of ontology. Among the state of the art of the techniques analyzed in this work only GAODT and DERONTO support the development of application ontologies.

An application ontology is composed of six elements: classes, taxonomy, relationships, properties, axioms and instances (Girardi, 2010). The order in which these elements are discovered influences the development of the subactivities of the technique or methodology. For example, GAODT starts by discovering axioms considering that they represent the requirements of the knowledge-based system. Other techniques begin with the identification of relevant domain terms corresponding to ontology classes.

An ontology development process should preferably be incremental allowing the engineer to add new elements to the ontology at each process interaction. DERONTO, 101, Methontology and GAODT are supported by an incremental development process.

Good techniques and methodologies should

Table 16: Comparative analysis of techniques for the construction of ontologies.

Comparison criteria	DERONTO	Uschold and King	101	Methontology	Gruninger and Fox	GAODT
The type of developed ontology	Domain and application ontologies	Domain ontologies	Domain ontologies	Domain ontologies	Domain ontologies	Application ontologies
The order through which the ontology elements are discovered	Classes Taxonomy Properties Relationships Axioms	Classes Taxonomy Relationships	Classes Taxonomy Properties Relationships Axioms Instances	Classes Taxonomy Relationships Properties Axioms Instances	Classes Relationships Properties Instances Axioms Taxonomy	Axioms Classes Relationships Taxonomy Properties
The type of life cycle	Iterative	Classic	Iterative	Iterative	Classic	Iterative
The reuse of existing ontologies	No	Yes	Yes	Yes	No	Yes
Uses linguistic knowledge	No	No	Yes	No	No	Yes

consider the reuse of existing ontologies like domain and task ontologies and even application ontologies. Most of the analyzed techniques approach reusability with the exception of DERONTO and Gruninger and Fox.

Only GAODT and the 101 technique define rules to find classes from nouns and relationships from verbs. This is an advantage because it allows easy identification of the elements of the ontology based on a linguistic strategy. Another advantage of the use of linguistic knowledge is the possible automation of the extraction of these elements using NLP techniques (Cunningham et al., 2012).

4 CONCLUSIONS AND FUTURE WORK

This article described GAODT, a technique for building application ontologies through a goal-oriented development cycle. The technique also provides the developer, a well-defined way to translate the knowledge expressed in natural language to FOL predicates. This feature is not covered by any other technique or methodology of the state of the art presented in this paper.

GAODT has been evaluated through the development of a study case for the construction of an application ontology to be used on an Intestate Succession Knowledge-based System.

Building reusable ontologies is a costly process. Among the four types of ontologies defined by Guarino (Guarino, 1998), application ontologies are the less reusable once they are developed for specific software applications. However, they are generally easier and faster to develop. Building application ontologies and then generalizing its elements to domain and task ontologies is a good alternative approach for developing these reusable artifacts. In this context, GAODT consists of a first step in this direction by defining a systematized way for building application ontologies.

Future improvements of GAODT include the development of a software tool to support it and a technique to perform the semantic search for ontologies to be reused. GAODT will also be integrated into a knowledge based process for the development of multi-agent systems (Leite, 2009) already constructed by the authors´ research group. The main objective is to use GAODT to construct the knowledge bases of deliberative agents of knowledge-based systems developed with this process.

ACKNOWLEDGEMENTS

This work is supported by CNPq, CAPES and FAPEMA, institutions of Brazil and Maranhão Governments for scientific and technologic development.

REFERENCES

Article nº 1.829., 2002. *Brazilian Civil Code*. Federal Legislation.

Boley, H., 2004. POSL: An Integrated Positional-Slotted Language for Semantic Web Knowledge. *W3C*.

Caliari, F., 2007. DERONTO: Method for Building Ontologies from Entity-Relationship Diagrams. *Master thesis - Federal Technological University of Paraná*. (In Portuguese)

Claudia, A., Fanizzi, N., Esposito, F., 2005. A semantic similarity measure for expressive description logics. In

Proceedings of Italian Conference on Computational Logic, Roma.

Costa, A., 2009. MADAE-Pro A knowledge-based process for Domain and Application Engineering. *Master thesis - Federal University of Maranhão*. In Portuguese.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., 2002. 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.

Fernández, M., Pérez, A., 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, p. 480 – 484.

Gonçalves, C., 2009. *Brazilian Civil Law: Inheritance law*. São Paulo, Saraiva. (In Portuguese).

Gruninger, 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*.

Jeffrey, F., 2006. Mastering Regular Expressions. *O'Reilly Media*, 3rd Edition.

Lee, W., Shah, N., Sundlass, K.., Musen, M., 2008. Comparison of ontology-based semantic-similarity measures. *AMIA Symposium*, p. 384–388.

Noy, N., McGuinness, D., 2001. *Ontology Development 101: A Guide to Creating Your First Ontology*.

Pérez, A., et al., 2004. *Ontological Engineering: with examples from the areas of knowledge management, e-commerce and the semantic web*. London. Springer-Verlag.

Russel, S., Norvig, P., 2004. *Artificial Intelligence*. Rio de Janeiro: Ed. Campus. (In Portuguese).

Uschold, M., King, M., 1995. Towards a Methodology for Building Ontologies. In *IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing*. Montreal, Canada.