# OntoPop or how to annotate documents and populate ontologies from texts

Florence Amardeilh

# OntoPop or how to annotate documents and populate ontologies from texts

Florence Amardeilh[1], [2]

[1] Lalicc, Université Paris 4 – Sorbonne,
18 rue serpente, 75006 Paris, France
florence.amardeilh@paris4.sorbonne.fr
[2] Mondeca, R&D Lab,
3 cité Nolez, 75018 Paris, France
florence.amardeilh@mondeca.com

**Abstract.** This paper presents an innovative solution for annotating documents and populating domain ontologies in a single process. In fact, this process can be viewed as a translation of a linguistic representation of a text into a more formal representation format, being RDF statements or OWL instances. Thus, we provide the OntoPop methodology for mapping linguistic extractions with concepts of ontology thanks to knowledge acquisition rules. This methodology has been implemented in a platform integrating a IE tool and a commercial ontology repository. The platform is currently under evaluation on various real world applications. Nevertheless, OntoPop already raises interesting issues such as semantic heterogeneity that are discussed at the end of this article.

## 1 Introduction

Document Annotation consists in (semi-)automatically adding metadata to documents, i.e. providing descriptive information about the content of a document such as its title, its author but mainly the controlled vocabularies as the descriptors of a thesaurus or the instances of a knowledge base on which the document has to be indexed. Ontology Population aims at (semi-)automatically inserting new instances of concepts, properties and relations to the knowledge base as defined by the domain ontology. Once Document Annotation and Ontology Population are performed, the final users of an application can exploit the resulting annotations and instances to query, to share, to access, to publish documents, metadata and knowledge [1].

Document Annotation and Ontology Population can be seen as similar tasks. Firstly, they both rely on the modelling of terminological and ontological resources (ontologies, thesaurus, taxonomies…) to normalize the semantic of the documentary annotations as well as the concepts of the domain. Secondly, as human language is a primary mode of knowledge transfer, they both make use of text-mining methods and tools such as Information Extraction to extract the descriptive structured information from documentary resources or Categorisation to classify a document into predefined cate-

gories or computed clusters. Thirdly, they both more and more rely on the Semantic Web standards and languages such as RDF for annotating and OWL for populating.

As shown in **Fig. 1**, some of the Text Mining solutions parse a textual resource, creating semantic tags to mark up the relevant content with regard to the domain of concern (cf. **Fig. 3** example). The semantic tags are then used either to semantically annotate the content with metadata [2] [3] [4] or to acquire knowledge, i.e. to semi-automatically construct and maintain domain terminologies [5] or to semi-automatically enrich knowledge bases with the named entities and semantic relations extracted [6] [7] [8] [9]. Therefore, the major issue is to conceive a mediation layer to map the semantic tags produced by the Text Mining tools into formal representations, being the content annotations (RDF) or the ontology instances (OWL). In other words, how can we transform a linguistic representation of a textual document into a semantic knowledge representation?

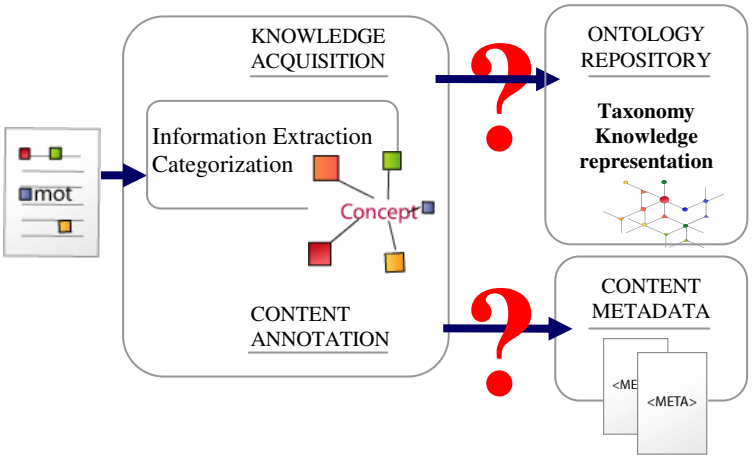


**Fig. 1.** Mastering the gap from IE to Semantic Representations issue

In that article, we introduce a new methodology, named OntoPop, which is addressing that particular issue. It allows the creation of a gateway between Information Extraction (IE) tools and knowledge representation (KR) tools. One of the specificity of the methodology is to keep the tools independent from each other for a greater flexibility and reusability of the overall application. The OntoPop methodology has been implemented in a platform that is currently under evaluation. The platform integrates a IE tool with a KR tool in order to perform document annotation and ontology population during a single process over a corpus of representative textual resources for the domain of application.

In the next section this paper, we will present the OntoPop methodology. The platform implementing the OntoPop methodology is described in detail in section 3. Experimentations and relative issues are the aim of section 4. Section 5 will provide conclusion and directions for future work.

## 2 The OntoPop Methodology

The OntoPop methodology was elaborated after implementing and testing a first version of a platform for document annotation and ontology population [10]. From the obtained results, we output the following requirements:

- **Ease of implementation**. Mapping IE tools with KR tools requires different kind of experts (from the domain, linguistic, knowledge representation). Thus, the chosen solution must be easily understood by them three and must be an iterative process.
- **Mapping the structure of the ontology and the structure of the linguistic extractions, modelled in separate ways**. Annotating a document and/or populating an ontology must not impose new constraints on the way the terminological and ontological resources are modelled neither on the format produced by the IE tools.
- **Capacity to evolve**. The platform must be able to take into account the evolutions of both the ontological resources and the IE tools.
- **Completion**. The platform must be able to map all information given by the IE tools.
- **Standardisation**. The platform must not be dependant of the IE tool used and it must produce Semantic Web compliant formats, such as RDF and OWL.
- **Consistency**. The instances created in the knowledge base and the semantic annotations produced must be consistent with the ontology model.

Furthermore, the OntoPop methodology must solve questions related to mapping different knowledge representation formats, such as:

- *How to transform the linguistic representation format of a textual document into the ontology model representation format?* On the one hand, the IE tools produce a set of semantic tags organized as a conceptual tree, mostly represented as a XML document. Those IE tools tend to extract a maximum of information without a constant worry about the normalisation of the extracted data because the result is usually exploited by search engines and the formalism is thus less crucial. On the other hand, the ontology and its knowledge base are used to store and exploit the information in a rigorous and constraining way. Formalisms such as OWL and RDF(S) are used to define the semantic representation of the knowledge along with the required quality for normalising the instances.
- *How to master the gap between the ontology modelling and the construction of the linguistic resources for a specific domain?* As the customisation of the IE tool and of the KR tool for a specific domain is achieved independently from one another, the domain coverage of each tool might be slightly different. Indeed, the semantic tags produced by the IE tool are not necessarily aligned with every concept of the ontology. The IE tool generally reuses existing linguistic resources where a subpart is possibly not relevant for the domain or the ontology models concepts needed for other purposes of the final application, etc.

- How do we know if the meaning of a semantic tag produced by the IE tool corresponds to the meaning of a concept of the ontology? A single semantic tag can be used to map several concepts of the ontology. On the contrary, different semantic tags can be used to map the same concept. How do we actually map a semantic tag with a concept of the ontology the same way that Gruber asked: « what information about terms is most critical for supporting sharability? The names? Textual definitions? Type, arity and argument restrictions? Arbitrary axioms? » [11]?

Based on those requirements emanating from the first platform's implementation, we sketched the OntoPop methodology. The methodology is willing to guide the users through integrating an IE tool and a KR tool together in order to design domain-oriented applications for knowledge management. The OntoPop methodology defines a progressive and iterative framework until a common agreement is reached upon the quality of the tools' integration between all the users implementing the application's solution. Those users are:

- the client, i.e. the domain expert, who specifies the application needs and borderlines and who validates the whole solution provided by the other users;
- the linguist, i.e. the expert in charge of the linguistic developments needed to adapt the IE tool for the domain;
- the ontograph, i.e. the expert in charge of modelling the domain ontology;
- and the integrator, i.e. the expert in charge of the mapping and the implementation of the solution from a technical point of view.

The OntoPop methodology is composed of five stages:

- **The Study Stage**. Discussion between the linguist, the ontograph, the integrator and the client upon the data to manage (the corpus to analyse, the knowledge to model and the metadata to produce): they evaluate the work load for adapting each tool to the domain, they estimate the capacity of extraction of the IE tool on a new domain and the coverage that can be obtained on representative corpus, they define the targeted coverage for the new application and thus the concepts to be modelled also considering transitioning the existing data to the new model.
- **The Structuring Stage**. Structuring the IE resulting semantic tags into a conceptual tree and modelling the domain ontology: the integrator identifies as soon as possible the overlapping and the missing information in the conceptual trees or in the ontology model to adjust them according to the client's needs; the integrator, the ontograph and the linguist produce a synchronised development planning; they exchange specifications documents such as the structure of the conceptual trees produced by the IE tool and the ontology model for the concerned domain to facilitate the integrator's task.
- **The Mapping Stage**. Mapping each element defined in the domain ontology with the semantic tags contained in the conceptual trees in order to create a set of Knowledge Acquisition Rules, as discussed in section 3.1.

– **The Validation/Quality Stage**. Validation of the produced document annotations and knowledge base instances: the integrator tests the mapping implemented and the client validates the overall solution for the new application. If improvements are needed, users reiterate the methodology phases from the Structuring stage.

– **The Delivery Stage**. Delivery of the application to the client and maintenance.

## 3 The OntoPop Platform

We developed a second version of the document annotation and ontology population platform based on the OntoPop methodology. Although the methodology can be applied to various IE and KR tools, the platform is made of two specific tools. It allows us to demonstrate the methodology's validity by integrating existing commercial tools. On the one hand, Mondeca's Intelligent Topic Manager (ITM™) is used for representing and managing the domain ontology, the thesaurus and the knowledge base [14]. On the other hand, Temis' Insight Discoverer Extractor (IDE™) is used for extracting information from semi and unstructured texts [15]. The aim of the platform is to provide a generic software solution, 100% customisable and producing the most comprehensive mapping between the ontology model and the results of the IE tool. As shown in **Fig. 2**, the OntoPop platform is composed of:

– a set of Knowledge Acquisition Rules,

– a Knowledge Acquisition Rules Compiler,

– a module for processing the ontology population and document annotation tasks.

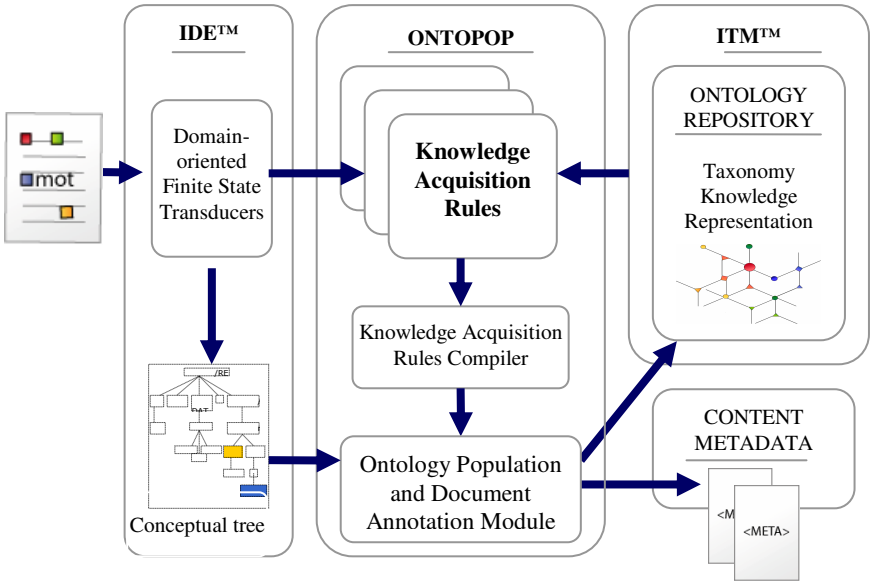


**Fig. 2.** The OntoPop's platform

### 3.1 Defining the Knowledge Acquisition Rules

The purpose of the OntoPop methodology is to facilitate the mapping between the two knowledge representation formats and to maximise its coverage. During the Mapping stage, the integrator compares the semantic tags produced by the IE tool with the concepts modelled in the domain ontology to design mapping rules, or **knowledge acquisition rules**. A knowledge acquisition rule describes the way a concept of the ontology will be instantiated or used for document annotation. It points out the semantic tag triggering the instantiation or the annotation process on that concept. The whole set of rules have to be defined before launching the document annotation or the ontology population tasks. Therefore, they constitute the core of the OntoPop methodology and platform.

We will illustrate the creation of those knowledge acquisition rules through an example taken from one of our projects. The domain deals with media publishing, and especially processing news about "famous people" to acquire their biographical data (aliases, date of birth, place of birth, zodiac sign, etc.), their relationships (engaged, married, divorced…) and their connections with artefacts such as movies, plays, songs, etc. Our client needs to semi-automatically acquire that information to enrich the knowledge base and to index the document with the named entities extracted from the news such as the "famous people" names and the locations.

Firstly, the integrator runs the IE tool on a subset of free-text documents. Each document produces a conceptual tree as shown in **Fig. 3**. A conceptual tree represents not the entire content of the document but only some textual units (usually sentences) that have been found relevant wrt the domain by the IE tool. The conceptual tree is made of hierarchically organised nodes following the structure of the extracted information in the original text. Actually, each node represents a semantic tag associated to the extracted information recalled in parenthesis. For example, the semantic tag "/ActorNamed" is associated to the original text "Francis Ford Coppola".

Secondly, the integrator compares all the semantic tags produced in the conceptual trees with every element (class, attribute or relation) of the ontology. Concerning the "Famous People" domain, only 10 classes (3 named entities and 7 events), 11 relations and 15 attributes from the complete domain ontology can be mapped to semantic tags. For example, the class "Personality", subclass of "Person", has attributes "date of birth", "alias", "zodiac sign"... whilst the class "Article" has attributes "publication date", "source", "author"… The class "Article" is also related to the class "Personality" through the "personality indexation" property.

```
/REFERENCE-ACTOR (Francis Ford Coppola)
   /ActorNamed (Francis Ford Coppola)
      /Personality (Francis Ford Coppola)
/REFERENCE-ACTOR (Spike Jonze)
   /ActorNamed (Spike Jonze)
      /PotentialNameOfPerson (Spike Jonze)
         /FirstName(Spike)
            /MASCULIN_SEX(Spike)
         /ProperName(Jonze)
/BIRTH-DATE (Coppola was born on the 7th of April 1939 in Detroit)
   /Person (Coppola)
      /ProperName (Coppola)
   /Birth (was born)
      /DATE (on the 7th of April 1939)
   /Location (Detroit)
      /America (Detroit)
         /UnitedStates (Detroit)
/COUPLE (his cousin, Nicolas, is going to divorce Patricia)
   /ActorNamed (his cousin, Nicolas)
      /Family (his cousin, Nicolas)
      /FirstName (Nicolas)
   /ImminentEvent (is going to)
   /Divorce (divorce)
   /ActorNamed (Patricia)
      /FirstName (Patricia)
```

**Fig. 3.** Extract of the conceptual tree produced by the linguistic analysis of the article "La tribu Coppola" published in Elle Magazine (the tree is translated from French)

During the comparison task, the integrator centralises all the necessary information as in **Table 1** to define the knowledge acquisition rules. Here are the mapping cases:

- A single semantic tag maps only one element; cf. the semantic tag /FilmWork.
- Several semantic tags map the same element; cf. the class "Personality".
- A semantic tag maps several elements of the same type; cf. the tag /COUPLE.
- A semantic tag maps several elements of different types; cf. the tag /ActorNamed.
- A semantic tag has no element to map; cf. the semantic tag /ImminentEvent.
- A element has no corresponding semantic tag; cf. the data property "Zodiac sign".

When a semantic tag maps several concepts, the integrator must use the context of that semantic tag in the tree, i.e. the other semantic tags being its ancestors, descendents or siblings, to resolve any ambiguities. For instance, if the semantic tag "/COUPLE" has a child node "/Divorce", as it is the case in the Coppola example, an event of the class "Divorce" will be instantiated. Otherwise, if the child is the node "/Marriage", it will be an event of the class "Wedding". The notion of context is extremely important, even crucial, as the mapping is rarely a simple projection between a semantic tag and an element of the ontology. Hence, that context must be easily grasped by the knowledge acquisition rules in the hierarchical structure of the conceptual tree.

**Table 1.** Examples for the comparison task

| Name in the ontology | Type in the ontology | Semantic tag | Context in the conceptual tree |
|---|---|---|---|
| Movie | Class | /FilmWork | |
| Personality | Class | /Personality | |
| | | /PotentialNameOfPerson | |
| Wedding | Class | /COUPLE | ∃ Child = /Mariage |
| Divorce | Class | | ∃ Child = /Divorce |
| Personality | Class | /ActorNamed | ∃ Child = /Personality |
| Spouse | Relation | | ∃ Child = /Personality **and** ∃ Parent = /COUPLE |
| Personality indexation | Relation | | |
| Birth location | Attribute | /Location | ∃ Parent = /BIRTH-DATE |
| Wedding location | Attribute | | ∃ Parent = /COUPLE **and** ∃ Brother = /Wedding |
| Location indexation | Attribute | | |
| Birth date | Attribute | /DATE | ∃ Oncle = /Person **and** ∃ Father = /Birth **and** ∃ Ancestor = /BIRTH-DATE |
| | | /ImminentEvent | |
| Zodiac sign | Attribute | | |

First, the knowledge acquisition rules must draw a parallel between a concept of the ontology to be mapped and a semantic tag triggering that process. Second, they must take into account every semantic tag conditioning the context. In order to formalise the knowledge acquisition rules, we studied the work on Contextual Exploration such as defined by Desclès to identify some discursive categories in unstructured texts [12]. The Contextual Exploration method parses a free text and tags it accordingly to the presence of linguistic markers (linguistic indicators and contextual linguistic clues). In [13] Crispino defined a language, LangText, to express contextual exploration rules. In OntoPop, we explore already tagged documents to find not linguistic but semantic indicators and their contextual clues to create new instances in the ontology. Because of the analogy between the two methods, we decided to adapt LangText in order to define a language for the OntoPop's knowledge acquisition rules, cf. **Fig. 4**.

**RuleName**: name given by the integrator
**ConceptType**: nature of the ontology concept (class, attribute, relation)
**ConceptURI**: URI of the concept uniquely identified in the ontology
**IndicatorNode**: semantic tag triggering the annotation and population processes
**ContextualClues**: conditions on the (non-)existence of some semantic tags in the context of the IndicatorNode
**Output** (optional): default value is the original textual value of the node, otherwise an expression to construct the value from another node(s) in the tree
**Position** (optional): indicates if the positions of the extracted information in the original document are required or not (Boolean: true/false, by default: 'faux')
**Trust** (optional): indicates a trust level (high/medium/low, by default: 'high')
**endRule**

**Fig. 4.** Structure of the knowledge acquisition rules for the OntoPop methodology

As an example, the semantic tag "/DATE", previously seen in **Fig. 3**, is used to instantiate the attribute "Birth Date" on the class "Personality". The corresponding knowledge acquisition rule is formalised as so:

```
RuleName: BirthDateR1
ConceptType: Attribute
ConceptURI: http://www.mondeca.com/onto#Birth_Date
IndicatorNode: DATE
ContextualClues: {Exist: [TreeSearchSpace: parent]
                         [ClueNode: Birth]
                 }
                 {Exist: [TreeSearchSpace: ancestor]
                         [ClueNode: BIRTH-DATE]
                         {Exist: [TreeSearchSpace: child]
                                 [ClueNode: Person]}
                 }
Output: text
Position: false
Trust: high
endRule
```

**Fig. 5.** Example of a knowledge acquisition rule on the attribute "Birth Date"

### 3.2 Performing the Annotation and Population tasks

Since the language adapted from LangText is human-oriented to ease the maintenance task, the platform needs an interpreter to translate the knowledge acquisition rules into a computable machine-oriented language. We chose the XPath[1] language as it can parse any tree (conceptual tree, XML document…) to reach directly any node and select any of its descendants, ancestors or siblings. For instance, the knowledge acquisition rule in **Fig. 5** that instantiates the attribute "Birth Date" has the following XPath rule: "Birth/DATE[ancestor::BIRTH-DATE/Person]".
That XPath rule means: "*find a semantic tag 'DATE' whose parent is the node 'Birth' and whose ancestor is the node 'BIRTH-DATE', itself having a child node 'Person'*".

1 Site web du W3C: http://www.w3.org/TR/xpath

While translating all the knowledge acquisition rules in XPath rules, the interpreter will automatically construct two XSLT[2] stylesheets: one associated with the ontology population process to generate OWL or XTM instances and the other associated with the document annotation process to generate RDF metadata.

Once the knowledge acquisition rules have been successfully defined for the domain of application and automatically computed into XSLT stylesheets, the Ontology Population and Document Annotation module can be run on a testing set of documents during the Validation stage of the OntoPop methodology. The module is made of three components that are 100% customisable through independent configuration files. For example, the integrator defines if the final application has to annotate the content of a document with metadata or to enrich the knowledge base with new instances or both. The three components correspond to the different phases when processing a document for ontology population and/or document annotation:

- Transforming the conceptual trees resulting from the linguistic analysis to the required format, using the corresponding stylesheet: XTM or OWL for the knowledge base enrichment and/or RDF for the document annotation.
- Controlling the XTM or OWL instances with regards to the ontology (restrictions, cardinalities, data type…), the thesaurus (controlled vocabularies) and the knowledge base (already existing instances to avoid duplicates), creating the valid new instances and storing the invalid new instances for user validation.
- Controlling the RDF statements with regards to the ontology (restrictions, cardinalities…), the thesaurus (controlled vocabularies) and the knowledge base (controlled named entities), creating the valid statements as new metadata of the document and storing the invalid statements for user validation.

Optionally, in the case of a semi-automatic application, the final user validates the results of the ontology population process and/or the document annotation process in a single user interface that displays both valid and invalid instances and/or metadata.

Besides, as the IE tool partly relies on relevant domain-oriented lexical entries to improve its results, a maintenance process has been set up with the ontology repository. Indeed, new instances stored in the knowledge base are regularly exported as an XML document. The instances can be either those automatically populated using the OntoPop platform or those manually created by the final user through the editing user interfaces of the ontology repository. The XML document is made of the list of all the instances names organized by classes, no properties are exported. The XML document is then computed by the IE tool which completes its named entities and other lexical entries with the names of the new instances. The linguistic resources of the IE tool are then compiled to be taken into account during the next information extraction task. That maintenance process enhances to overall performance of the OntoPop platform since the linguistic resources are kept coherent with the stored knowledge.

[2] Site web du W3C: http://www.w3.org/TR/xslt

## 4 Experimentations and Major Issues

The OntoPop methodology and platform is currently tested in several real projects dealing with media publishing, legal edition, competitive intelligence… As summarised in **Table 2**, those projects are in different stages of the OntoPop methodology. The most achieved project is the Media Publishing on the "Famous People" domain whereas other projects are still in the Mapping stage such as the ones on Scientific or Industrial Competitive intelligence.

**Table 2.** Experimentations over various domains of application

| Domain of application | OntoPop Project Phase | Population and/or annotation | Number of ontology concepts concerned | Number of different semantic tags | Number of created rules | Number of processed documents |
|---|---|---|---|---|---|---|
| Media Publishing on "Famous People" | Delivery | Both | 10 classes, 11 object prop, 15 data prop | 64 tags | 62 rules | Hundreds of news articles per day |
| Legal Edition | Validation | Population | 10 classes, 4 object prop, 36 data prop | 66 tags | 77 rules | + 100000 law cases |
| Media Publishing on "News Event" | Mapping | Both | 18 classes, 26 object prop, 7 data prop | 27 tags | 34 rules | Thousands of news articles per day |
| Industrial Competitive Intelligence | Mapping | Both | 29 classes, 39 object prop, 76 data prop | 68 tags | 70 rules | To be estimated |
| Scientific intelligence | Mapping | Population | 5 classes, 15 data prop | 18 tags | 24 rules | 50 millions of abstracts |

They all both semi-automatically populates the ontology and annotates the parsed documents except the Legal Edition and Scientific Competitive Intelligence projects just interested in knowledge acquisition. Not all the classes, attributes and relations are mapped by the knowledge acquisition rules but a subpart of the ontology concepts. It usually depends of the requisite domain coverage for achieving the application's goals. Moreover, from the above table, we can see there is roughly as much knowledge acquisition rules as available semantic tags for each domain corpus. Nevertheless, the integrator still has to manually maintain the entire set of knowledge acquisition rules and the more there are, the heavier that burden is.

At that time, we can not provide any insight analysis of the performance of the OntoPop platform as we are currently performing that evaluation. By the time this paper will be published we would be able to provide an evaluation framework along with relevant measures and figures for each project. In the platform's first version, we used the classic recall and precision measures but we are not very satisfied with them. They

can not really apprehend the complexity of the ontology population and the document annotation tasks and we need to define better adapted measures that would include for example the number of rules for each mapped concept.

Nevertheless, while implementing the OntoPop methodology and applying it to our different projects, important issues came up that we want to discuss here in more details. While defining knowledge acquisition rules, a general mapping question arose: how is it possible to transform a non-formal representation of a text into a more formal semantic representation of knowledge? This question leads us to the different issues presented below, accompanied by proposed solutions, based on examples extracted from "La tribu Coppola" conceptual tree mentioned in section 3.1.

**Input Format.** The extracted information can not be stored in the data structure modelled in the domain ontology.

- *Example*: Textual dates such as "the 7th of April 1939", "yesterday", "the month before"… are not formal descriptions to be exploited in a calendar system like a date value composed of three fields "dd/mm/yyyy", such as "07/04/1939".
- *Solutions*: Either the IE tool is able to compute the linguistic date into a semantic tag that respects the data type format or the data structure in the domain ontology has to be changed from a date field to a simple text field inducing a capacity loss in exploiting that information, especially in information retrieval systems.

**Information accuracy (I).** The semantic modelled in the ontology is more accurate that the one produced by the semantic tags.

- *Example:* The class "Family" has three properties "hasFather", "hasMother" and "hasChild". In the following conceptual tree, the semantic tag "/ActorParent" is the only one to describe the parent information. Thus it is impossible to know if the person tagged as the parent is in fact the father or the mother of this "Family".

```
/QualificationPerson (Francis Coppola with his daughter Sofia)
   /ActorParent (Francis Coppola)
   /Parenthood (with his daughter Sofia)
      /Child (Sofia)
         /FirstName (Sofia)
```

- *Solution*: An inference layer added to the OntoPop platform would solve some of these problems. In the "Family" case, if the existing instance "Francis Coppola" of class "Person" has an attribute "sex" valued as "male", the inference layer would deduce that if semantically tagged as a parent, then he should be the father.

**Information accuracy (II).** The semantic tags from the linguistic analysis are more accurate than the semantic modelled in the domain ontology.

- *Example*: The "Famous People" domain ontology has two classes regarding couple events: "Wedding" and "Divorce". But in the following example, the semantic tag "/Break" under "/COUPLE" indicates another type of event in a couple's life: Is it a break in a non-married relationship? Is it announcing a probable divorce? It is not possible to instantiate it as there is no corresponding class in the ontology.

```
/COUPLE (Spike Jonze and Sofia Coppola broke up in 2001)
   /ActorNamed (Spike Jonze)
      /Personality (Spike Jonze)
   /ActorNamed (Sofia Coppola)
      /Personality (Sofia Coppola)
   /Break (broke up)
   /DATE (2001)
```

- *Solution:* It should be interesting to (semi-)automatically add new concepts to the ontology based on certain sort of semantic tags. For instance, a new class "Break" could be created with the property "hasBrokeUp" on range "Person".

**Information Proximity.** The semantic tags mapping the properties of an instance of a class should be situated underneath the semantic tag representing the instance itself.

- *Example:* The class "Personality" has an attribute "kinship". That property is mapped to the semantic tag "/FamilyLink" which is a sibling of the semantic tag "/ActorNamed" instantiating the "Personality" class. In the following conceptual tree, the semantic tag "/FamilyLink" has two "/ActorNamed" siblings. The property will be instantiated for both instances of "Personality" even if one is wrong.

```
/QualificationPerson (Anton Coppola, Francis' uncle, …)
     /ActorNamed (Anton Coppola)
        /Personality (Anton Coppola)
     /ActorNamed (Francis)
        /FirstName (Francis)
     /FamilyLink (uncle)
```

- *Solution:* To avoid such errors, a knowledge acquisition rule must not be defined if there is a potential risk of ambiguity arising from the hierarchical structure of the parsed conceptual trees.

Other issues raised by the OntoPop platform can be summarised in the following three points:

– **Consistency** between the semantic tags and the instances or the annotations generated, i.e. the semantic must be respected between a concept of the ontology and its corresponding semantic tags in the conceptual tree. For example, the semantic tag "Personality" must be used to create new instances of the class "Personality" and not instances of the class 'Movie Character", although some of them are considered by extension as some sort of personalities such as "Rocky" or "Zorro".

– **Conflict** between acquisition rules, i.e. a rule may instantiate another concept than the one it has been defined for. For instance, a class and its subclasses may have very similar acquisition rules only differentiated by the context of their triggering semantic tags. Actually they usually have the same triggering semantic tag and that's why solving the context is so important in those complex cases. For example, the "Event" class has two subclasses, i.e. "Wedding" and "Divorce". The semantic tag "/COUPLE" is the same triggering tag for the three classes. But if its child node is "/Marriage", an instance of the class "Wedding" will be created; if the child node is "/Divorce", an instance of the class "Divorce" will be created; otherwise, the upper class "Event" will be instantiated.

– **Maintenance** of the knowledge acquisition rules, i.e. when modifying either the domain ontology or the conceptual tree structure, the rules have to be updated to reflect the changes. Although processing a document is fully automated, the rules' maintenance is still a manual operation. Depending of the domain's coverage and complexity, it can rapidly become an overwhelming task for the integrator.

## 5 Conclusion and Future Work

The OntoPop methodology and its platform are an innovative solution for performing ontology population and document annotation from linguistic extractions. Thanks to the definition of knowledge acquisition rules, it becomes possible to integrate an IE tool with a KR tool for any specific domain analysis. Similar approaches in the Semantic Web Framework can be found, especially platforms such as KIM [6], Artequakt [7] and OntoSophie [8]. But they are rather using machine learning techniques or populating high-level generic ontologies such as PROTON[3] or not annotating the document at the same time. The other major difference between those platforms and OntoPop is the fact that OntoPop preserves the independence between the IE tool and the KR tool. The mediation layer provides flexibility and adaptability capacities to specific domains as in real world applications.

Nonetheless, there are still issues to solve and we need to improve our solution to facilitate the definition of the knowledge acquisition rules during the Mapping stage. Not only this evolution of the platform must respect the requirements mentioned in section 2 but it must also tackle the issues raised in section 4. To our opinion, a possible evolution would be to express the linguistic representation of a document (the actual conceptual tree) in a more formal knowledge representation. In fact, a conceptual tree can always be represented in XML making it possible to extract a XML Schema from this XML conceptual tree.

In the last decades, researchers have been working on database schema integration, then on XML Schemas integration and more recently on OWL ontology integration [16] [17]. They are looking for methods to align different representations of the same format in order to merge them or to find a translation between them. They are deducing and creating mappings between those representations formats. Consequently, we are investigating the different approaches to: first, use the XML Schemas representing the conceptual trees, over various domains and produced by different IE tools, to model a generic ontology of linguistic extractions; second, find a way to align the ontology of linguistic extractions directly with a domain ontology. The purpose is to help the integrator to generate not all but at least some high-level knowledge acquisition rules, to maintain those rules when detecting a change in the ontologies and to detect conflicts when possible.

[3] PROTON website: http://proton.semanticweb.org/

## References


1. Laublet P., Reynaud C., Charlet J.: Sur Quelques Aspects du Web Sémantique. Assises du GDR I3, Cépadues, Nancy (2002) 59-78
2. Kahan J., Koivunen M., Prud'Hommeaux E. and al.: Annotea: An Open RDF Infrastructure for Shared Web Annotations. In Proceedings of the WWW10 International Conference, Hong Kong (2001) 623-632
3. Handschuh S., Staab S., Ciravegna F.: S-CREAM – Semi-automatic CREAtion of Metadata. In Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW02), Spain (2002) 358-372
4. Vargas-Vera M., Motta E., Domingue J.: MnM: Ontology Driven Tool for Semantic Markup. In Proceedings of the Workshop Semantic Authoring, Annotation & Knowledge Markup (SAAKM 2002), Lyon (2002) 43-47
5. Bourigault D., Aussenac-Gilles N., Charlet J.: Construction de Ressources Terminologiques ou Ontologiques à partir de Textes: un Cadre Unificateur pour Trois Etudes de Cas. Revue d'Intelligence Artificielle (RIA), Vol. 18, No. 1, Paris (2004) 87–110
6. Popov B., Kiryakov A., Ognyanoff D., and al.: KIM - A Semantic Platform for Information Extaction and Retrieval. In Journal of Natural Language Engineering, Vol. 10, Issue 3-4. Cambridge Press University (2004) 375-392
7. Alani H., Kim S., Millard D. E., and al.: Web based Knowledge Extraction and Consolidation for Automatic Ontology Instantiation. In Proceedings of Knowledge Capture (K-CAP03), Workshop on Knowledge Markup and Semantic Annotation, Florida (2003)
8. Celjuska D., Vargas-Vera M.: Ontosophie: A Semi-Automatic System for Ontology Population from Text. In Proceedings International Conference on Natural Language Processing ICON 04, India (2004)
9. Valarakos A., Paliouras G., Karkaletsis V., Vouros G.: Enhancing the Ontological Knowledge through Ontology Population and Enrichment. In Proceedings of the International Conference on Knowledge Engineering and Knowledge Management (EKAW04). Lecture Notes in Artificial Intelligence, Vol. 3257. Springer-Verlag (2004) 144-156
10. Amardeilh F., Laublet P., Minel J-L.: Document Annotation and Ontology Population from Linguistic Extractions. In Proceedings of Knowledge Capture (KCAP05), Banff (2005)
11. Gruber T.: The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Base. (eds) J.A. Allen, R. Fikes & E. Sandewall, Principles of Knowledge Representation and Reasoning. Proceedings of the 2nd International Conference, Vol. 601-602 (1991)
12. Desclés J.-P., Cartier E., Jackiewicz A., Minel J.-L.: Textual Processing and Contextual Exploration Method. In Proceedings of CONTEXT 97, Rio de Janeiro (1997) 189-197
13. Crispino G: Une Plate-forme Informatique de l'Exploration Contextuelle : Modélisation, Architecture et Réalisation (ContextO). Application au Filtrage Sémantique de Textes. PhD Thesis, Paris 4-Sorbonne University (2003)
14. Amardeilh F., Francart T: A Semantic Web Portal with HLT Capabilities, In Veille Stratégique Scientifique et Technologique (VSST04), Vol. 2, Toulouse (2004) 481-492
15. Grivel L., Guillemin-Lanne S., Lautier C. and al.: La Construction de Composants de Connaissance pour l'Extraction et le Filtrage de l'Information sur les Réseaux. In 3ème Congrès du Chapitre Français of International Society for Knowledge Organization, Paris (2001)
16. Noy F. N.: Semantic Integration: A Survey of Ontology-Based Approaches. In SIGMOD Record, Special Issue on Semantic Integration, Vol. 33, No. 4 (2004) 65-70
17. Shvaiko P., Euzenat J.: A Survey of Schema-based Matching Approaches. In Journal of Semantics, Vol. 3730. Springer-Verlag, Berlin Heidelberg (2005) 146-171