Enhancing Ontology Matching using Logic-based Reasoning

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science (M.Sc.)

in

Computational Logic

by

Xichuan Wu

Registration Number 3736186

at the Faculty of Computer Science,
at the Dresden University of Technology (TU Dresden)

Supervisor: Dr. Yue Ma
External Supervisor: Susan Marie Thomas
Professor: Prof. Dr. -Ing. Franz Baader

(Signature of Author) (Signature of Supervisor) (Signature of Supervisor)

Dresden, July 27, 2013

Declaration Of Authorship

I, Xichuan Wu, declare that the thesis entitled "ENHANCING ONTOLOGY MATCHING USING LOGIC-BASED REASONING" and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;

Signed: . . . . . .

Date: . . . . . .

Acknowledgements

I would like to take this chance to express my gratitude to all those who have helped me in working out this thesis.

First and Foremost, thanks go to Prof. Dr. -Ing. Franz Baader, Dr. Yue Ma from Dresden University of Technology and Susan Marie Thomas from SAP Research Karlsruhe. This work would not be possible without their trust, patience and support. Special thanks to Géraud Amic, our French expert in financial accounting, for his remarkable dedication and patience in devising gold standard mappings between French and Spanish taxonomy, in answering each and every of our questions instantly. Special thanks also go to Prof. Dr. Philipp Cimiano and Dr. John McCrae from Semantic Computing Group in the University of Bielefeld, for their generous support in making COAL work. I am also grateful for the support and companionship graciously offered by all my mentors, friends and classmates in Dresden, Karlsruhe and the world over.

Some words that are not often spoken are dedicated here to my parents and my younger brother together with his family. Though my parents are among hundreds of millions of humble peasants, both of them are persons of distinguished character. My father, Zhengjun WU, is a caring son, loving husband, supportive father and truthful friend. He is the visionary and entrepreneurial leader of the family. My mother, Tongbi LEI, is a wise, diligent, frugal and loving lady. I love them more than can be put into words.

Abstract

Ontologies, as formal representations of conceptualizations of domain knowledge, have proven to be very useful in real world applications like the Semantic Web. Given the increasing number of ontologies which are independently developed, but which represent overlapping domain knowledge, there is an urgent need to integrate them. Ontology matching meets this need by (semi-)automatically finding correspondences (also called mappings) between entities in given pairs of ontologies. Most ontology matching methods are based on statistics. In contrast, this thesis is devoted to enhancing ontology matching by logic-based reasoning, both for mapping generation and for mapping refinement.

- Mapping generation. We first identify a number of underlying properties of various financial reporting ontologies and then formalize them into a Basic Accounting Ontology. Later we use this ontology to define concepts in the ontologies to be matched, so that reasoners can be used to infer mappings between them. This process is realized as a semantically enriched ontology matching mechanism (SEOM).

- Mapping refinement. We adapt a number of logical principles from existing work to identify and remove mapping suggestions that are logically unintended. Different approaches to resolving observed incoherence, inconsistency or violations of principles are analyzed. A number of heuristics are also considered to enhance the mapping refinement process. Combining a selected set of these together, we build a logic-based mapping refinement procedure (LOMR).

We show experimentally that 1) SEOM generates a set of logically consistent and accurate mapping suggestions; 2) LOMR is able to improve the quality of mapping suggestions by removing logically unintended mappings while keeping the logically sound ones.

List of Figures

List of Tables

List of Algorithms

Contents

Chapter 1

Introduction

1.1 Real World Problem: Comparing Companies

Financial reports inform interested parties about the current financial position of a company, and the results of operations for a reporting period. The volume of such reports has become so enormous that automated processing has become a necessity. To meet this need the eXtensible Business Reporting Language (XBRL) [1] was developed, and has been widely adopted by regulatory and governmental organizations such as the U.S. SEC[1], UK revenues and customs[2], the European Financial Reporting authority[3] and individual European Business Registries[4]. Such authorities use XBRL to define *XBRL taxonomies* for the financial and business data that they are legally authorized to collect from the organizations or companies under their jurisdiction. An XBRL taxonomy specifies the content (*concept*s in XBRL terms) and structure of financial reports created according to specific accounting regulations and conventions, which vary across country, jurisdiction, industry, time, etc. A taxonomy functions like an XML schema in that its concepts are used to tag data in financial reports so that it can be automatically processed by software. Much XBRL-based financial data is already available on the Web in multiple languages[5], ready for use by interested parties, such as regulators, potential investors, creditors, competitors, and the general public.

Authorities, financial analysts, and the public would like to compare financial

[1] See http://www.sec.gov/

[2] See http://www.hmrc.gov.uk/

[3] See http://www.eba.europa.eu/Supervisory-Reporting/FINER.aspx

[4] See http://www.ebr.org/

[5] See http://www.xbrl.org/knowledge_centre/projects/list

data from multiple jurisdictions, but language barriers and the diversity of XBRL taxonomies make this very difficult. In Europe, for example, each member state has jurisdiction specific rules for registering a company, publishing its bylaws, its annual financial statements, and other official documents. Accordingly, each national business register has defined its own sets of local taxonomies for the filing and publishing of XBRL instance documents, which contain the legal, economic and financial data of the registered companies. Although some progress has been made in Europe [2], the diversity of taxonomies makes it operationally difficult to compare company results across jurisdictions.

It is widely recognized that this XBRL taxonomy comparability problem has yet to be addressed [2, 3]. Frankel [3] identifies the basic problem as a lack of semantic clarity, a problem which, in general, accounts for the bulk of software integration costs. In response to the problem, the XBRL organization has recently formed the Comparability Task Force, which is currently collecting requirements around comparability[6]. This task force envisions a solution to the problem through the provision of mappings, in the form of XBRL assertions about relationships between comparable sets of elements in different taxonomies. Assertion creation is dependent, however, on taxonomy alignment. In this thesis the taxonomy alignment problem is approached by transforming it into an ontology matching problem. There are mainly two reasons for this approach. First, using ontological representation, some underlying semantics (call *background knowledge*) which are implied, but not explicit in XBRL taxonomies, can be made explicit. These semantics added to the ontologies of XBRL taxonomies can enhance the ontology matching process [4]. Thanks to the calculation hierarchies in XBRL taxonomies, a methodical way can be devised to formally explicate underlying semantics (see Section 2.3.1 in Chapter 2). In short, the transformation from XBRL taxonomies to ontologies makes it possible to exploit more information from given XBRL taxonomies. The second reason to transform the taxonomy alignment problem into an ontology matching problem is that reasoning with ontologies and mappings helps generate logically consistent mappings. The combination of background knowledge with reasoning is expected to generate mappings that are semantically intended and logically consistent.

A single XBRL taxonomy usually defines a number of financial reports. The *balance sheet* is one of the most common. It shows measures of the assets of the corporation, the debts owed, and the interests of the owners [5]. It consists of a

[6]XII Comparability Task Force: Comparability Business Requirements, http://www.xbrl.org/comparability-task-force (2012)

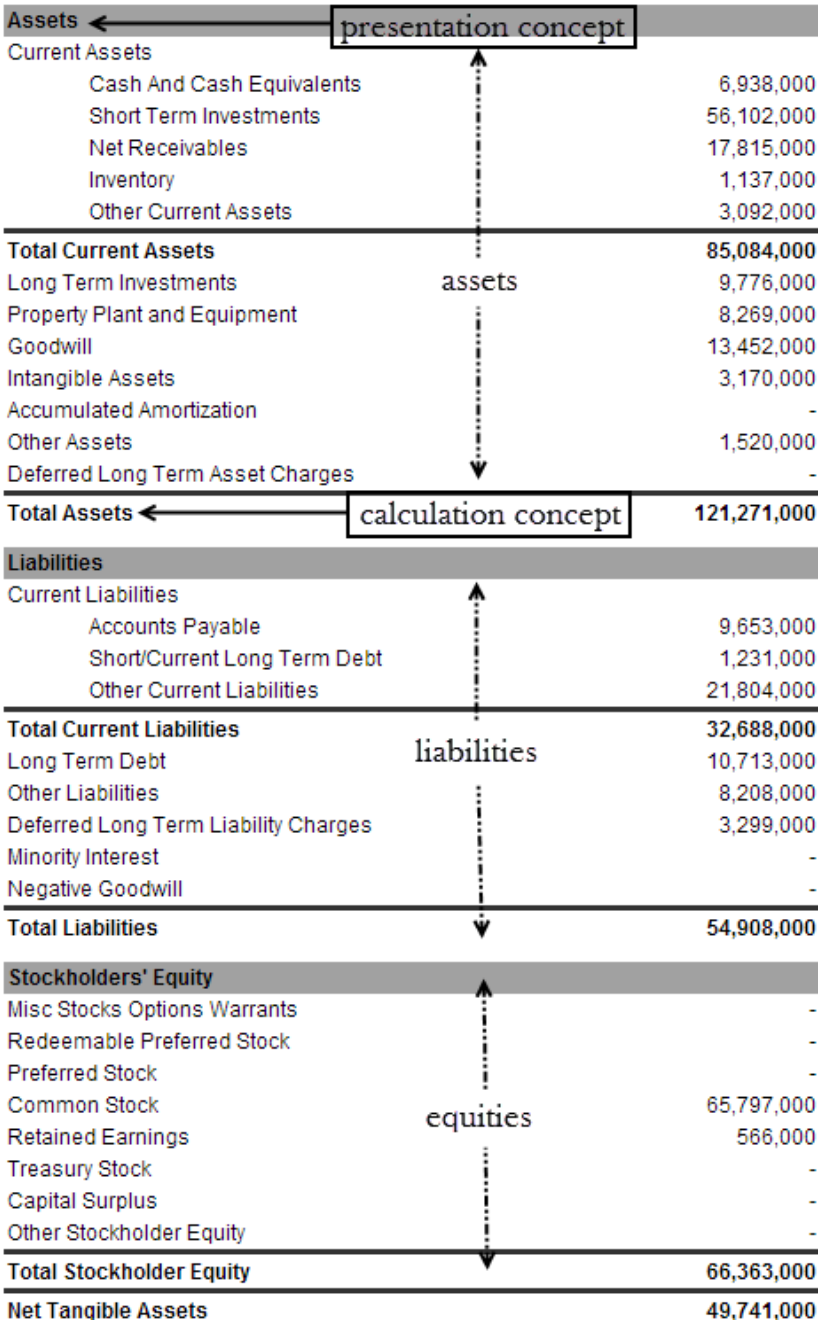


Figure 1.1: Example of real world balance sheet report

number of concepts presented in special hierarchies, with each concept usually followed by a number representing the amount of money. It usually contains three sections: assets, liabilities and stockholders equity, which further contain various sub-items. A balance sheet can usually be presented in two different hierarchies, i.e., presentation hierarchy and calculation hierarchy. Presentation hierarchy corresponds to the layout we usually see in a balance sheet, as shown in Figure 1.1[7]. There are abstract line items for presentation purpose. For example, "Assets", without referring to a specific amount of assets, usually is a header for all different kinds of assets, including total assets which corresponds the total amount of assets. Calculation hierarchy specifies the calculation relation among line items. For example, total assets equals the sum of current assets and non-current assets.

1.2 Ontology Matching

An ontology is a formal representation of a conceptualization of domain knowledge. With the development of the Semantic Web, an ontology is often encoded in the Web Ontology Language (OWL)[8], which is, in turn, based on description logics (DL) [8]. DL provides a formal ground for ontology, so that logical reasoning can be applied to infer implicit knowledge. Ontology matching (also called ontology alignment or ontology mapping) is the process of finding correspondences between entities in a pair of ontologies. In the following we introduce some basic definitions, on top of which *mapping* can be formally defined.

1.2.1 Description Logics

DL is a family of formal logics that are mostly used for knowledge representation (KR). In comparison with other earlier KR formalism (such as Semantic Network [6], or Frame systems [7]), DL provides the ability to unambiguously represent entities of interest and some fragments of it are tractable in terms of computational complexity. DL has many different variants. In order to understand how it is formally presented (syntax) and how it conveys meaning (semantics), we formally introduce the description language $\mathcal{SHOIN}(\mathbf{D})$, the DL underlying OWL-DL. Note that dialects of DL differ in the syntactic constructors used to build up com-

[7]This is an extract from Microsoft Corporation (MSFT) here http://finance.yahoo.com/q/bs?s=MSFT+Balance+Sheet&annual .

[8]See http://www.w3.org/TR/owl-features/ .

Table 1.1: Syntactic construction of concept descriptions

$\neg B$	(negation)
$B \sqcap C$	(intersection)
$B \sqcup C$	(union)
$\{o_1, \ldots, o_n\}$	(one of)
$\forall P.C$	(value restriction)
$\exists P.C$	(existential quantification)
$\exists_{\leq n} P$	(at least restriction)
$\exists_{\geq n} P$	(at most restriction)
$\exists R.D$	(data exists restriction)
$\forall R.D$	(data value restriction)
$\exists_{\leq n} R$	(data at least restriction)
$\exists_{\geq n} R$	(data at most restriction)

plex concepts and properties. An introduction to the members of the DL family is given by Baader and Nutt [8].

We start with the introduction of *vocabulary* in order to construct concept descriptions in DL.

Definition 1. (Vocabulary). *A vocabulary* $\boldsymbol{S}$ *is defined as a quadruple* $\boldsymbol{S} = \langle \boldsymbol{C}, \boldsymbol{P}, \boldsymbol{R}, \boldsymbol{I} \rangle$ *where* $\boldsymbol{C}$ *is a set of concept names,* $\boldsymbol{P}$ *a set of object properties,* $\boldsymbol{R}$ *a set of data properties and* $\boldsymbol{I}$ *a set of individual names.*

Having vocabulary $\mathbf{S}$, concept descriptions in $\mathcal{SHOIN}(\mathbf{D})$ are constructed using the following constructors which apply to concepts (also called classes), both atomic and complex (Table 1.1). In Table 1.1, B and C are concepts, D is a datatype, P is an object property and R is a data property, $n \in \mathbb{R}^+$ and $o_1, \ldots, o_n$ are individual names. In addition to the complex concept descriptions presented above, there are also top concept $\top$ and the bottom concept $\bot$. The set of terminological axioms, TBox $\mathcal{T}$, can be constructed as in Table 1.2, where B, C are concept descriptions, P, Q are two object properties and R, S are two data properties. The set of assertional axioms, ABox $\mathcal{A}$, contains axioms of the

Table 1.2: Types of terminological axioms

$B \sqsubseteq C, B \equiv C$	concept subsumption, equivalence
$P \sqsubseteq Q, P \equiv Q$	object property subsumption, equivalence
$R \sqsubseteq S, R \equiv S$	data property subsumption, equivalence

form in Table 1.3, where a, b two individual names and d is a concrete data value in

Table 1.3: Types of assertional axioms

$C(a)$	concept assertion
$P(a, b)$	object property assertion
$R(a, d)$	data property assertion
$a = b$	equality
$a \neq b$	inequality

D. An ontology $\mathcal{O}$ contains both a TBox and an ABox, $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$. In essence, an ontology can be seen as a set of ontological axioms, including assertional axioms and terminological axioms. For the use case in this thesis, ABox will be omitted because we are concerned with schema, not instances. Ontologies in the following discussion consist only of terminological axioms.

To assign meaning to descriptions of an ontology, we define *interpretation*s.

Definition 2. (Interpretation). *An interpretation* $\mathcal{I} := \langle \cdot^{\mathcal{I}}, \Delta^{\mathcal{I}}, \Delta^{\mathcal{I}}_D \rangle$*, where* $\cdot^{\mathcal{I}}$ *is an interpretation function,* $\Delta^{\mathcal{I}}$ *non-empty domain,* $\Delta^{\mathcal{I}}_D$ *a concrete domain and* ***D*** *a datatype theory.*

This function $\cdot^{\mathcal{I}}$ assigns to every atomic concept A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, to every object property P a binary relation $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, to every data property R a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}_D$, to every individual name in **I** to an element of $\Delta^{\mathcal{I}}$, to every datatype in **D** a subset of $\Delta^{\mathcal{I}}_D$, and to every data constant a value in $\Delta^{\mathcal{I}}_D$. Inductively, complex formulas can be interpreted as in Table 1.4. In the context of mapping generation and mapping refinement, we only consider relations between concepts, not between properties. From now on, anything other than concepts will be left out.

Table 1.4: Interpretation of different concept descriptions

$$
\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset \\
(\neg B)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \backslash B^{\mathcal{I}} \\
(B \sqcap C)^{\mathcal{I}} &= B^{\mathcal{I}} \cap C^{\mathcal{I}} \\
(B \sqcup C)^{\mathcal{I}} &= B^{\mathcal{I}} \cup C^{\mathcal{I}} \\
(\forall P.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} | \forall b.(a,b) \in P^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \\
(\exists P.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} | \exists b.(a,b) \in P^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\} \\
\{o_1, \ldots, o_n\}^{\mathcal{I}} &= \{o_1^{\mathcal{I}}, \ldots, o_n^{\mathcal{I}}\} \\
(\exists_{\leq n} P)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} | \#\{(a,b) \in P^{\mathcal{I}}\} \leq n\} \\
(\exists_{\geq n} P)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} | \#\{(a,b) \in P^{\mathcal{I}}\} \geq n\} \\
(\exists R.D)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} | \exists b.(a,b) \in R^{\mathcal{I}} \wedge b \in D^{\mathcal{I}}\} \\
(\forall R.D)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} | \forall b.(a,b) \in R^{\mathcal{I}} \rightarrow b \in D^{\mathcal{I}}\} \\
(\exists_{\leq n} R)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} | \#\{(a,b) \in R^{\mathcal{I}}\} \leq n\} \\
(\exists_{\geq n} R)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} | \#\{(a,b) \in R^{\mathcal{I}}\} \geq n\}
\end{aligned}
$$

Definition 3. *Given an interpretation $\mathcal{I} = \langle \cdot^{\mathcal{I}}, \Delta^{\mathcal{I}}, \Delta_D^{\mathcal{I}} \rangle$, $\mathcal{I}$ satisfies an axiom*

$$
\begin{aligned}
B \sqsubseteq C \quad & \textit{iff} \quad B^{\mathcal{I}} \subseteq C^{\mathcal{I}} \\
B \equiv C \quad & \textit{iff} \quad B^{\mathcal{I}} \subseteq C^{\mathcal{I}} \wedge C^{\mathcal{I}} \subseteq B^{\mathcal{I}}
\end{aligned}
$$

Definition 4. (Model). *An interpretation $\mathcal{I}$ is a model for an ontology $\mathcal{O}$, iff $\mathcal{I}$ satisfies each axiom and assertion in $\mathcal{O}$.*

Definition 5. (Entailment). *An ontology $\mathcal{O}$ entails an assertion or axiom α, iff each model for $\mathcal{O}$ is also a model for α. An ontology $\mathcal{O}$ entails a set of assertions or axioms A, iff each model for $\mathcal{O}$ is also a model for each $\alpha \in A$. It is denoted as $\mathcal{O} \models \alpha$ if $\mathcal{O}$ entails α, otherwise it is denoted as $\mathcal{O} \not\models \alpha$.*

Definition 6. (Concept Unsatisfiability). *A concept C is unsatisfiable iff each*

model $\mathcal{I}$ of $\mathcal{O}$ maps C to the empty set, i.e., there is no instance that belong to C by any given $\mathcal{I}$.

Definition 7. (Incoherence). *An ontology $\mathcal{O}$ is defined to be incoherent iff it contains unsatisfiable concept(s).*

Definition 8. (Inconsistency). *An ontology $\mathcal{O}$ is inconsistent iff there exist no model for $\mathcal{O}$, otherwise $\mathcal{O}$ is consistent.*

Having these definitions, we are ready to define mappings between ontologies.

1.2.2 Generating Mappings

Ontology matching is generally performed to integrate knowledge bases described by different ontologies. It is the process of generating a set of mappings (a.k.a correspondences or matches) between entities in the ontologies to be matched. There are a number of ways to formally represent a mapping. In the following, a pragmatic formal representation is adapted from the work of Euzenat [9, 10].

Definition 9. (Mapping) *A mapping μ is of the form*

$$\mu := \langle \rho(C_1, C_2), \epsilon \rangle,$$

where C_1 and C_2 are two concepts from two different ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ respectively. ρ is the relation between these two entities, such as equivalence and subsumption. ϵ is the confidence value *representing how "semantically close" these two entities are.*

In effect, mapping μ, considered in the sequel, is of the form, either $\langle C_1 \equiv C_2, \epsilon \rangle$, $\langle C_1 \sqsubseteq C_2, \epsilon \rangle$ or $\langle C_1 \sqsupseteq C_2, \epsilon \rangle$. Usually ϵ falls into the interval $[0, 1]$; and the higher it is, the more likely μ is a (semantically) correct mapping. In some parts of discussion that follows, when confidence values are irrelevant, a short hand notation for a mapping between concept C_1 and C_2 is $\rho(C_1, C_2)$, where $\rho \in \{\equiv, \sqsubseteq, \sqsupseteq\}$. We use $\mathcal{M}$ to denote a set of mappings. The union $\mathcal{O}_1 \cup_{\mathcal{M}} \mathcal{O}_2$ of $\mathcal{O}_1$ and $\mathcal{O}_2$ connected by $\mathcal{M}$ is defined as $\mathcal{O}_1 \cup_{\mathcal{M}} \mathcal{O}_2 = \mathcal{O}_1 \cup \mathcal{O}_2 \cup \{\tau(\mu) | \mu \in \mathcal{M}\}$ with τ being a translation function that converts a mapping into an axiom in the following way: $\tau(\langle \rho(C_1, C_2), \epsilon \rangle) = \rho(C_1, C_2)$. A corresponding mapping ontology $\mathcal{O}_m$ for $\mathcal{M}$ is therefore $\mathcal{O}_m = \{\tau(\mu) | \mu \in \mathcal{M}\}$.

Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, we assume that there is a set of *gold standard mappings*, denoted as $\mathcal{M}_{gs}$. These are created by domain experts to reflect their choice of correct correspondences. There are a number of requirements

$\mathcal{M}_{gs}$ should fulfill (which however cannot be precisely formalized). For example, $\mathcal{O}_1 \cup_{\mathcal{M}_{gs}} \mathcal{O}_2$ should be consistent and contain no unsatisfiable concepts. $\mathcal{M}_{gs}$ should also be minimal in the sense that adding any mappings would not provide more information, and removing any mappings would lead to absence of intended correspondences. Given mapping suggestions $\mathcal{M}$ for $\mathcal{O}_1$ and $\mathcal{O}_2$, we define *correct mapping*s as $\mathcal{M} \cap \mathcal{M}_{gs}$, *incorrect mapping*s as mappings that result in inconsistency, unsatisfiable concepts or violation of the semantics of $\mathcal{O}_1$ or $\mathcal{O}_2$. The remnant are *redundant mapping*s because they cause no logical problem but still do not belong to the gold standard.

In Euzenat's generic representation, correspondences can be between concepts, as well as properties. In this thesis, however, we only consider mappings between concepts. The reason is that financial ontologies converted from XBRL taxonomies contain only concepts.

There have been a number of sophisticated tools to perform ontology matching[9], the purpose of which is to (semi-) automatically find correct mappings. Hereafter, we call these ontology matching tools simply *matcher*s, unless explicitly noted otherwise.

Here are some notational convention used in the following discussion. Foremost, the signature Σ of an ontology or an axiom is the set of concepts occurring in the ontology or the axiom respectively.

Definition 10. (Signature Σ). *The signature $\Sigma(\mathcal{O})$ of ontology $\mathcal{O}$ is defined as follows,*

$$\Sigma(\mathcal{O}) := \{C | C \textit{ is a concept name occurring in } \mathcal{O}\}, \tag{1.1}$$

where the signature $\Sigma(\alpha)$ of axiom α can be defined analogously.

We use a number of symbols to denote different logical axioms. To be specific, μ is used to denote mappings, which connect concepts from different ontologies, and α or ν are used to denote logical relation between two concepts from the same ontology. The difference between α and ν will become clear later when specific use cases are considered.

Note that subscripts of logical symbols used in the work have special intention, where $_x$ indicates xEBR ontology[10], $_f$ indicates French ontology, $_s$ indicates Spanish ontology, all of which are derived from corresponding XBRL taxonomies; also $_m$ indicates mapping ontology. This notational pattern is adopted because these three ontologies are the main subjects of the discussion. On the other hand, this

[9] See http://www.ontologymatching.org/ for a comprehensive list.

[10] xEBR ontology is created based on xEBR taxonomy, which is described in Section 3.2.2.

notational pattern does not hinder the generic applicability of algorithms, if one supplies other ontologies and mapping ontology to replace $\mathcal{O}_f, \mathcal{O}_s$ and $\mathcal{O}_m$. To distinguish $\mathcal{O}_x$ from other ontologies, we often call $\mathcal{O}_f$ and $\mathcal{O}_s$ *local* ontologies.

1.2.3 Mapping Refinement

Mapping refinement, also known as mapping revision [11], is a recent and debated area of study. This verification process aims to preserve those correct mappings, which are logically consistent with both ontologies and reflect correct correspondence between concepts, while removing unintended mappings from mapping suggestions. Here mapping suggestions can be obtained manually or automatically. Manual mappings are not free from errors due to a number of reasons, for example, different conceptualizations of the same domain, or incomplete knowledge of ontology engineers, etc. In fact, Unified Medical Language System® Metathesaurus (UMLS-Meta)[11], which stems mainly from manual effort, was reported to contain a considerable number of unintended mappings [12]. Automated mapping suggestions for now contains even more incorrect mappings because most matchers depend mainly on analysis of text, in forms of concept names, labels, descriptions, etc, and structural information, e.g., number of descendants. These analyses based on statistics are incapable of exploiting semantics of ontologies. Mapping refinement is the process of exploiting logical principles to remove unintended mapping suggestions, which is a necessary step to ensure the quality of output mapping suggestions.

1.3 Problem Statement

Many systems and tools have been proposed to solve the problem of ontology matching in general. As far as financial ontologies are concerned, there are only a few matchers due to a number of reasons. First, there have not been any well-accepted financial ontologies formalizing national or international schemas. Second, financial reporting schemas vary across countries, jurisdictions and industries. All accounting actions, like identifying and measuring, are guided by principles, rules or practices which have evolved over time, and which are specific to the society or economic field in which they are embedded. This diversity of financial reporting schemas makes it very difficult to establish correspondences

[11] http://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/

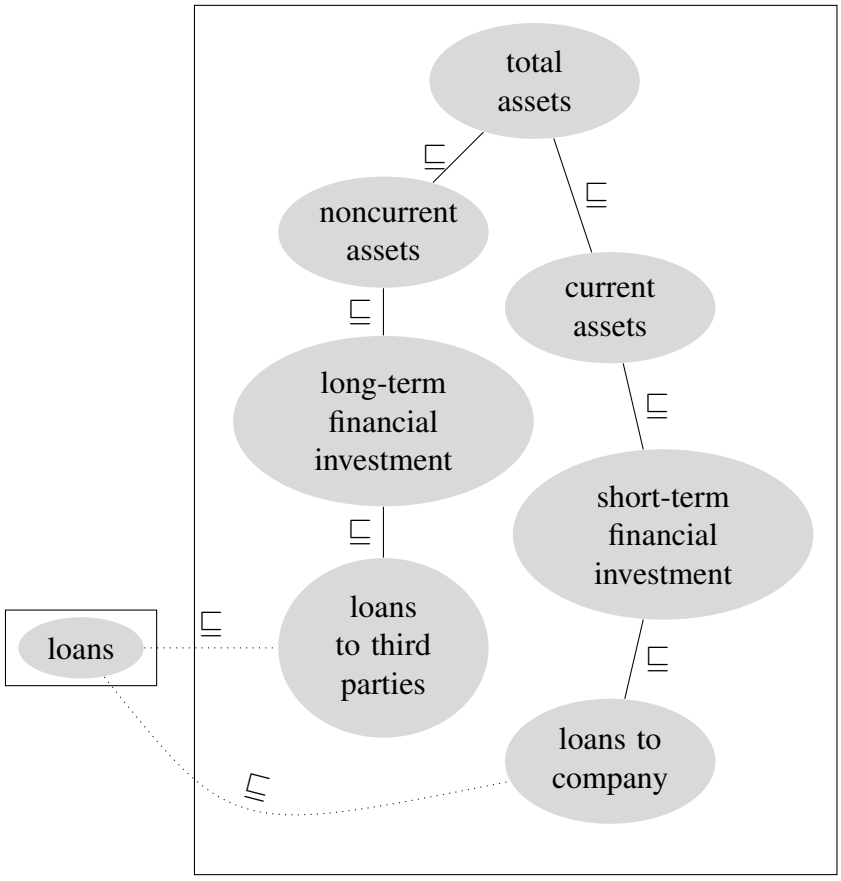


Figure 1.2: Automated mappings for financial concepts from different schemas

between them. Third, matchers have to consider the multilingual characteristics of financial reporting schemas. It is very common for financial reporting schemas to have labels in multiple languages, which is especially true in Europe. Then when one wants to align schemas that do not have labels in shared languages, the problem becomes cross-lingual ontology matching. The few matchers that try to match some emerging financial reporting schemas rely heavily on either the quality of translation [13], or a set of terminological features for machine learning techniques [14]. The mappings those matchers generate contain many incorrect mappings, i.e., ones that cause inconsistency, as illustrated by the following example.

In Figure 1.2, *loans* in one source schema is mapped by certain string-based matcher as a narrower concept than two different concepts in a target schema,

loans to third party and *loans to company*. When one examines the structure of the target schema, it turns out that *loans to third party* belongs to *long-term financial investment*, which in turn belongs to *noncurrent assets*, whereas *loans to company* belongs to *short-term financial investment*, which in turns belongs to *current assets*. As a partition of *total assets*, *noncurrent assets* and *current assets* are meant to have no overlap. Therefore, *loans* in source schema cannot be a narrower concept to both *loans to third parties* and *loans to company*. These two mappings of course have led to unintended consequences. To resolve this problem, there are at least two approaches. First, we could make explicit the distinction between *loans*, *loans to third-party* and *loans to company* by formal constructions. By exploiting background knowledge or underlying semantics, this approach would avoid the generation of these two mappings from the beginning. The second approach is to remove at least one of these two mappings. There should be principles one can follow to resolve such inconsistencies. Principles can be of statistics, or of logic reasoning. Statistical principles can assign confidence value to each mapping and then select the mapping with lowest value to remove. But those statistics are often computed based on text analysis, which would provide only an approximation of correctness of each mapping. Logical principles, however, exploit the underlying semantics explicitly presented in ontologies to soundly determine which mappings are logically intended and which are not.

In this thesis, we attempt to enhance ontology matching following the ideas of the two approaches above. We first investigate how to exploit background knowledge or underlying semantics in order to obtain mapping suggestions that are semantics-aware and logically consistent. Then we present a number of logic-based principles, some of which are adapted from existing work, with also a few proposed principles based on observation of the current use case. These principles are discussed in detail and then implemented and integrated together in order to remove incorrect mappings from automated mapping suggestions.

1.4 Research Questions

Most state-of-the-art matchers use only lexical and structural information to construct mapping suggestions [4]. The quality of the mappings produced by matchers is evaluated mainly by Precision/Recall metrics with respect to gold standard, i.e., reference mappings produced by domain experts. The goal of this thesis is to enhance an existing matcher by logic-based reasoning. In particular, the following questions are addressed.

R1 What kind of background knowledge will help generate better mappings?

R2 What mechanism will enable the exploitation of background knowledge?

R3 Why is the refinement process for automatically generated mappings necessary?

R4 What are the logic-based principles for mapping refinement?

R5 How do the logic-based principles perform?

R1 is concerned with determining what kind of background knowledge to use in order to improve mapping results. It is suggested [4] that relevant background knowledge, in addition to the ontologies to be matched, can help the ontology matching process by increasing recall; on the other hand, considering background knowledge may also introduce redundant or even incorrect mappings, which decreases precision. How best to achieve higher recall while maintaining precision is use case specific. In the case of matching financial reporting schemas, it is observed that existing financial ontologies fail to explicate some very important background knowledge. This thesis addresses this problem by identifying and then formalizing a range of fundamental concepts and properties such that logical reasoning service can be used to achieve better matching results. Thus, a mechanism is proposed to exploit the background knowledge, which is in response to **R2**.

R3 is concerned with the necessity and importance of the work we have done for **R4** and **R5**. We now give theoretical arguments and empirical evidence to motivate the work on mapping refinement, which is in response to **R3**. On the theoretical side, arguments are given by Meilicke [11] for the need to refine automatically generated mapping suggestions. In short, it is argued that 1) a correct mapping will always be a coherent mapping; 2) the presence of incoherent mappings in the context of reasoning with a merged ontology will always lead to unintended consequences. As empirical evidence, Jiménez-Ruiz et al. [12] identified a number of logically unintended mappings from a comprehensive medical thesaurus which is mostly the result of manual effort. Also for the use case considered in this thesis, it is of great interest for the end users of ontology matching systems that manual effort can be reduced in selecting those correct mappings out of all the mapping suggestions. The end users of any matchers would right-

fully expect mapping suggestions of high quality (in particular when there is a large amount of mapping suggestions), which requires, among other things, that the mappings are logically consistent with ontologies. Based on these arguments and evidence, the main part of this thesis work is devoted to enhance ontology matching by logic-based mapping refinement.

R4 is concerned with logic-based principles that would help to remove incorrect mappings. Logic-based mapping refinement is to provide such support in an automatic and logically sound manner. Several logical principles are adapted or proposed to delete those mapping suggestions that cause inconsistency. To answer **R5**, all the logical principles are implemented and tested.

1.5 Outline & Contribution

To approach these research questions, this thesis is structured in several parts. In Chapter 2, in response to **R1**, background knowledge of financial reporting domain is analyzed and then later harnessed by a proposed mechanism, which leads to a semantically enriched ontology matching method. In Section 1.4, we answered **R3** by providing both theoretical arguments and empirical evidence. In Chapter 3, a number of logic-based principles are discussed in detail. These principles are either adapted from existing work or proposed for the first time in this work, which serves as an answer to **R4**. To answer **R2** and **R5**, Chapter 4 presents implementations of the principles after introducing datasets used in this thesis. Also presented are experimental results of both semantic enriched ontology matching system and logic-based mapping refinement system. Chapter 5 summarizes all the work done for this thesis and gives pointers to possible future work.

Chapter 2

Semantically Enriched Ontology Matching

There are a couple of heuristics and a methodical approach proposed in this work. This is to address the research question **R2**, concerning the use of background knowledge. The structure of this chapter is organized as follows. Section 2.1 briefly surveys existing techniques in ontology matching and hightlights the challenge to address. Section 2.2 presents a new ontology for the financial reporting domain. This ontology consists of underlying semantics of the domain and is used to define other financial concepts. In Section 2.3, an alignment process based on semantic enrichment is presented. Section 2.4 concludes the approach presented and highlights the contribution of this work.

Note that the main idea presented in this chapter is also in a paper to be published. Credits therefore go to the coauthors of that paper, namely Susan Marie Thomas from SAP Research Karlsruhe, Dr. Yue Ma from Dresden University of Technology and Dr. Sean O'Riain from Digital Enterprise Research Institute (DERI), National University of Ireland. My main contributions are the implementation and evaluation of the proposed idea.

2.1 Related Work

The proposed approach addresses a problem of ontology matching by means of formalization. In this section the approach is first compared to existing work on ontology matching, then to work on formalization in the accounting and financial reporting domain.

Table 2.1: Analytical comparison of the recent ontology matching systems.

System	Input	Output	GUI	Operation	Terminological	Structural	Semantic
LogMap	OWL	n:m alignments	Yes, web interface	-	labels, lexical variations	extended class hierarchy	logical repairing
COAL	OWL	n:m alignments	Yes, web interface	-	Levenshtein edit distance, bag-of-words cosine similarity, substring distance	direct or elementary children	-
SEOM	OWL	n:m alignments	-	question answering	-	calculation hierarchy	underlying properties, logical inference

2.1.1 Ontology Matching

There have been many different tools and systems for ontology matching, as recently surveyed by Shvaiko and Euzenat [4, 9], Noy [15] and Godugula [16]. The survey [4], as a follow-up work of [9], identified several aspects to describe and compare a number of existing matchers. Table 2.1 adopts their descriptions and applies them to LogMap[1] [17], COAL [14] and our logic-based approach (SEOM). The left part of the table is a general outlook of different systems and the right part classifies them in terms of which kind of data they exploit: strings (*terminological*), structure (*structural*), data instances (*extensional*) or *semantic*s. SEOM takes as input OWL ontologies and generates $n : m$ alignments [4]. As a prototype, it does not have a GUI (graphic user interface). Though it is designed as an ontology matcher, it can also be used as a question answering tool. Moreover, it does not exploit ontologies' textual descriptions, but their calculation hierarchies. First properties underlying the calculation hierarchies are identified and formalized. Sets of related properties constitute concept definitions, which is then to be reasoned using standard logical reasoning service to get mapping proposals. A few systems have additional functionality that tests the generated correspondences to decide whether to discard or retain them. These tests, which, in essence, refine the correspondence set generated by the matchers, may be rule-based, as in LILY [18], or logic-based as in LogMap.

In contrast to current state-of-the-art matchers, which primarily generate correspondences, and sometimes refine them via logic, our approach generates the correspondences entirely via logic, deducing them from the merge of $\mathcal{O}_1$, $\mathcal{O}_2$ and the Basic Accounting Ontology (BAO). Assuming the concepts in $\mathcal{O}_1$ and $\mathcal{O}_2$ have been correctly defined in terms of the BAO, the confidence score of every correspondence is one.

Among a number of challenges identified by Shvaiko and Euzenat [4], ontology matching requires a common context or background knowledge for the ontologies to improve mapping results. The logic-based approach is such a research effort in exploiting the underlying knowledge of financial domain by explicitly formalizing commonly shared properties. At its early stage, it still needs human intervention in creating and extending the BAO, defining financial concepts using those from the BAO. We envision the future semi-automation of these processes so as for the logic-based approach to be scalable.

[1]LogMap offers two functionality: mapping generation and mapping debugging. We call LogMap Repair the mapping debugging part. The LogMap web interface: http://csu6325.cs.ox.ac.uk/

According to Noy [15], matchers can be categorized into two main kinds of approaches: indirect and direct matching. Indirect matching uses a shared ontology as a common ground for the ontologies to be matched. Ideally, these ontologies are extensions of the shared ontology, so that matching profits considerably from the fact that they share common vocabulary with the shared ontology. Our approach is closer to this first category, with the BAO serving as the shared ontology, which, in our case, is used to enrich XBRL taxonomies. Aleksovski et al. [19] have shown that, in general, a shared ontology is helpful for matching ontologies whose semantics is shallow. However, the approach there is quite different from ours in the way it builds connections between the shared ontology and the ontologies to be matched; matching is based on lexical comparison rather than precise logical definitions.

The survey of Godugula [16] agrees with the position of Noy in classifying matching methods into indirect and direct. It is specifically devoted to the direct approach, which it considers to be the "harder" problem because it attempts to directly match two often very distinct ontologies, which have been independently developed. Most commonly these algorithms are based upon textual analysis, often supplemented by structural analysis. An example of textual analysis is the lexico-syntactic analysis of concept labels and descriptions, possibly also exploiting external lexical sources like WordNet [8]. Structural analysis, on the other hand, makes use of features like hierarchical structures, for example, the number of direct descendants of a concept. Shvaiko and Euzenat [9] conclude that the best solutions to the alignment problem are achieved by combining multiple, different matching algorithms. Their conclusion is supported by experiments with a direct alignment system COAL (Cross-lingual Ontology ALignment)[2], which combines lexico-syntactic analysis with structural analysis, and also exploits machine learning techniques which benefit from multilingual ontologies. COAL performed best when the different algorithms were combined. It applies machine learning techniques to a series of string-based and structural features of taxonomies, and, for each concept, it ranks all its possible corresponding concepts in descending order by confidence value. However, even though COAL was developed with the aim to match multilingual financial ontologies, its recall, and especially, its precision were not nearly as good as our logic-based approach (see Table 4.7 for the experimental results.). Another recently developed generic ontology mapping tool of the direct type, LogMap, which uses textual matching algorithms, yielded very poor results when applied to our use case. One reason is that LogMap is designed

[2]See COAL web interface: http://monnet01.sindice.net:8080/coal

to match large bio-medical ontologies with much semantics explicated, while the ontologies in our use case are both shallow in semantics and small in size. Our initial problem falls into this harder category but, as mentioned, we solved it by creating a shared ontology, which then makes our approach closer to the indirect one.

Some work has been done in Multilingual Ontologies for Networked Knowledge (MONNET) [20] project to match XBRL taxonomies from different countries. Spohr et al. [14] used machine learning techniques in order to account for the fact that taxonomies are usually multilingual and that cross-lingual transformation is needed while matching two taxonomies that are in distinct languages. To be precise, in the cross-lingual matching scenarios, native labels of source ontologies have to be automatically translated into labels in target language. This target language can be one of languages used in the ontologies; it may as well be a third language. For example, if we have French ontology with only French labels and Spanish ontology with only Spanish labels, one possibility is to translate all French labels into Spanish, another possibility is to translate both French and Spanish labels into English. They also developed a tool to generate matches given two ontologies, which is used in this thesis to get candidate mappings, to which logical refinement applies. The tool is called Cross-lingual Ontology Alignment (COAL). To use it in our use case, we need first to train it to obtain suitable matching function and relevant parameters.

A recent work [13] on cross-lingual ontology matching focuses on the effect of different label translations on matching results. It's shown that a set of features, like semantics, task intent, feedback etc, can be configured in the proposed matching system in order to get better results.

2.1.2 Accounting Ontologies

There has been considerable effort in modeling or formalizing accounting concepts. Recent work can be divided into two categories. First, there has been a lot of work which converts XBRL reports into Semantic Web Representations (see [21], [22] and [23]). While these efforts are useful for linking XBRL data to other data on the web of linked data as discussed by O'Riain [24], there is little further semantics addition during the conversion process. This is also true of the XBRL-related ontologies listed in a recent survey of financial ontologies [25].

In contrast, the second category of work focuses on a direct ontological specification of basic accounting concepts and processes. Krahel [26] proposes the formalization of accounting standards as a means to discover and resolve incon-

sistencies and ambiguities in the standards. Gailly and Poels [27] redesigned the Resource Event Agent (REA) model, popular in the accounting literature, and formalized it in OWL. Chou and Chi [28] proposed the EPA model (Event, Principle and Account) as a way to model the correct accounting classification of business transactions. And Gerber and Gerber [29] built a small OWL ontology as an experiment in formalization.

Our research has a similar departure point as the second category. But, unlike existing work, which attempts to model the accounting process, we aim at a detailed characterization of the concepts in XBRL taxonomies in order to perform cross-taxonomy alignment. In spite of cultural and linguistic diversity, there are many concepts common to the XBRL taxonomies used in different countries. In general, these common concepts are finer grained than the XBRL concepts, so that each XBRL concept can be described by means of multiple Accounting Ontology concepts, which it often shares with other XBRL concepts, even in the same taxonomy. Thus, our approach extends the semantics of each XBRL taxonomy. It makes explicit the fine-grained shared semantics which is only implicit in an XBRL taxonomy, often visible in labels or textual descriptions, but not available for machine processing. Moreover, our approach encodes this fine-grained semantics in such a way that logical reasoners can be used to infer mappings between taxonomies represented as ontologies. Although, the authors of [30] propose the use of ontologies to extend the semantics of XBRL, they do not propose to do so in a methodical way for the purpose of enabling alignment.

2.2 Basic Accounting Ontology

To align different XBRL taxonomies, we create a Basic Accounting Ontology by identifying and formalizing a set of concepts and properties that underlie various taxonomies.

Each concept in a balance sheet is specified by jurisdiction-specific accounting regulations and conventions. They are often calculation-oriented in the sense that accountants focus on adding up several items equal to another item. The fact that different jurisdictions, e.g., countries or governmental agencies, would have very different views and approaches to accounting makes it very difficult to compare/integrate accounting concepts. On the other hand, there is commonality shared by all accounting schemas. For example, to distinguish assets of different types, a concept “current assets” is commonly used to represent cash or assets which will be converted into cash within one year or less; concept “non-current

(a) Underlying financial concepts identified and formalized

(b) Object properties used in the enrichment step

Figure 2.1: Protégé screen shot of the BAO.

assets" would represent assets that last more than one year. So for many concepts in balance sheet, there is a common property, "currentness", with value of either Current or Noncurrent. There are also many other properties like this that characterize concepts. And these properties are shared by financial reporting schemas across national borders. In the light of this observation, we defined financial concepts using these commonly shared properties. By doing this, we make implicit knowledge of each financial concept explicit, which is our principled approach to semantic enrichment. We identify and formalize these underlying semantics as concepts and properties. All these became part of a new financial ontologies, the Basic Accounting Ontology.

We created a Basic Accounting Ontology that defines all these properties and the relations among these commonly shared concepts. It consists of 189 basic accounting concepts, 36 object properties and 5 data properties. Each basic concept is designed to represent certain underlying properties shared by most taxonomies. For example, $\texttt{Depreciability} \equiv (\texttt{Depreciable} \sqcup \texttt{Nondepreciable})$ because one financial concept can be either depreciable or nondepreciable, as shown in Fig. 2.1.

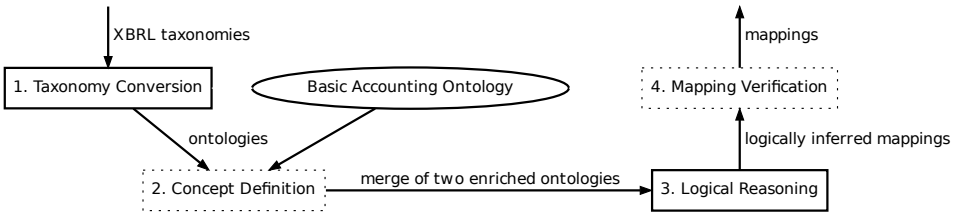


Figure 2.2: Three phases of the proposed alignment process

2.3 Semantically Enriched Matching Process

We propose to align XBRL taxonomies using a four-phased process: 1) conversion of XBRL taxonomies to ontologies, 2) enrichment of these ontologies, 3) generation of matches, 4) match verification. Phase 1 automatically converts each XBRL taxonomy into an OWL ontology. In Phase 2 these two OWL ontologies are manually enriched by describing the concepts in each ontology using a BAO which contains fundamental accounting concepts. These enriched ontologies are the input to Phase 3, which automatically computes cross-taxonomy equality and subsumption relationships by means of logical reasoning services (see [31]). Phase 4 presents the computed relationships to an expert for confirmation or correction.

Fig. 2.2 illustrates the four phases of the alignment process. Boxes with bold lines are phases that have been automated, whereas boxes with dotted lines are phases that need human intervention. As indicated in the figure, the linchpin of the process is the BAO, which we created to add more fine-grained semantics to the very coarse-grained semantics of XBRL taxonomies. This section outlines the three-phased process responsible for ontology matching.

In this work, the balance sheets of the French Taxonomie Comptes Annuels (TCA)[3] and Spanish Taxonomía del Nuevo Plan General de Contabilidad 2007 (PGC07)[4] taxonomies are the actual inputs of this process. Fig. 2.3, which has been used to explain Phase 1, is representative of the OWL class hierarchies created from the Spanish PGC07 taxonomy.

[3]See http://www2.xbrl.org/fr/frontend.aspx?clk=SLK&val=226 .

[4]See http://www.icac.meh.es/Taxonomia/pgc2007/Taxonomia.aspx .

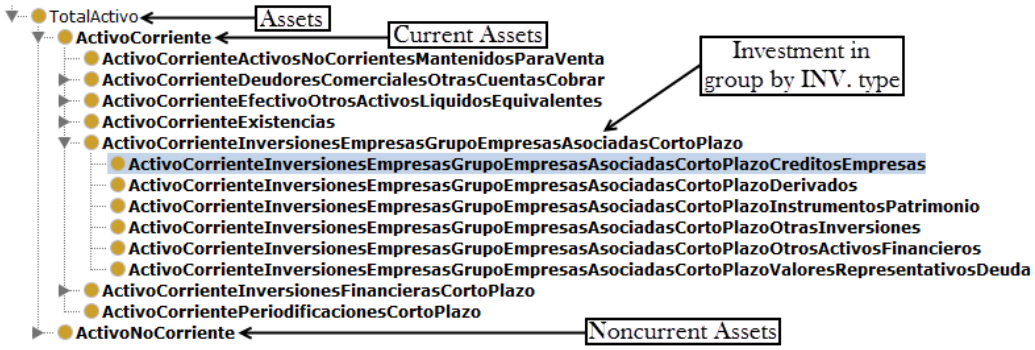


Figure 2.3: Protégé screen shot of calculation hierarchy of Spanish balance sheet assets; parent calculated from children. 'INV.' is short for investment.

2.3.1 Taxonomy Conversion

In *Phase 1* of the alignment process each taxonomy is first converted into RDF using the MONNET *xblr2rdf* converter, which preserves the calculation hierarchies (cf. Section 1.1). In the conversion process each XBRL concept in a calculation hierarchy becomes an OWL class with the same uniform resource identifier (URI) as the XBRL concept. Given this one-to-one relationship, these classes are often referred to as XBRL concepts in the following explanations. In the next step of Phase 1, the calculation hierarchies are converted into OWL subclass relationships using SPARQL[5]. This conversion is done strictly as a way to facilitate the rapid addition of semantics to concepts in Phase 2. Later these subclass relationships are removed in order to model cases where some financial concepts do not comply the property restrictions of their parents. Figure 2.3 is a Protégé screen shot showing an extract of the output from Phase 1, namely, part of the class hierarchy corresponding to the calculation hierarchy for assets in the Spanish balance sheet. One of its numeric relationships, as indicated by the screen shot, is `Assets` = `Current Assets` + `Noncurrent Assets`. Phase 1 was applied to the balance sheets of the TCA and PGC07 taxonomies to convert them into ontologies. To differentiate concepts from different ontologies, the standard short form for a concept URI is used, i.e., with *namespace prefix*[6] followed by *local name*. Prefix `ca:` indicates French concepts, prefix `pgc07:` Spanish; concepts from the BAO have no prefix.

[5]See http://www.w3.org/TR/rdf-sparql-query/
[6]See http://www.w3.org/TR/REC-xml-names/

2.3.2 Concept Definition

The purpose of the class hierarchies created in Phase 1 is purely to speed up *Phase 2* which is a manual process in which each XBRL concept is described using concepts from the BAO, in order to add more fine-grained semantics to it. As mentioned, each class with its direct subclasses represents a computational roll-up (sum) in the XBRL world. This roll-up works by virtue of the fact that the concepts being rolled up share certain properties. For instance, the subclasses of Assets (`TotalActivo`) in Fig. 2.3 all share the property of being classified as assets. Rather than editing each concept individually to add this property, it is given just once to Assets, and is then inherited by all its subclasses, thus speeding up the process of enrichment. In rare cases where this inheritance is incorrect, it is fixed later when the inheritance hierarchy is removed, and affected concepts are edited as necessary. This procedure of adding shared properties is repeated for each class (roll-up). Moreover, care is taken that each sibling in a roll-up is differentiated from the others by means of properties. Siblings must be mutually disjoint (non-overlapping), otherwise they could not be added up to create a total. For example, Current Assets and Noncurrent Assets must be disjoint, otherwise the total Assets would be incorrect, having double counted the overlap between the two addends.

An example of the result of Phase 2 can be seen in Table 2.2, which shows the semantics added to the Spanish concept highlighted in Fig. 2.3. The additional semantics take the form of property restrictions. Most of the restrictions for the Spanish concept under discussion are inherited from its superclasses. From Assets it inherits the property restriction `(∃hasClassification.Asset)`. It also inherits
`(∃hasGrossOrNet.Net⊔∃hasDepreciability.Nondepreciable)`, stating that it could be either net assets or nondepreciable assets. In fact, it belongs to net assets. Similarly, `(∃hasCurrentness.Current)` is inherited from current assets, and
`(∃hasClassification.FinancialInvestmentAsset)` as well as
`(∃investmentIn.GroupCompanyOrAssociate)` are inherited from the superclass investment in group. On the other hand,
`(∃hasFinancialInstrument.Loan)` was added directly, and specifies the type of financial investment asset. In this case, the label for the highlighted concept, which can be translated as “current assets; short-term investments in group companies and associates; loans”, exhibits the components of meaning, each of which is formalized as property restriction as in Table 2.2.

Table 2.2: Properties for the highlighted Spanish financial concept

pgc07:ActivoCorrienteInversionesEmpresasGrupo-EmpresasAsociadasCortoPlazoCreditosEmpresas
$\sqsubseteq$
(∃hasGrossOrNet.Net ⊔ ∃hasDepreciability.Nondepreciable)
(∃hasClassification.Asset)
(∃hasCurrentness.Current)
(∃hasClassification.FinancialInvestmentAsset)
(∃investmentIn.GroupCompanyOrAssociate)
(∃hasFinancialInstrument.Loan)

Phase 2 was applied to the asset concepts in the French and Spanish ontologies created in Phase 1. This resulted in the description, or enrichment, of 94 French asset concepts, and 74 Spanish, with each concept having 7 property restrictions on average.

2.3.3 Logical Reasoning

In *Phase 3* of the alignment process the concept descriptions created in Phase 2, which are called (*primitive classes*), are converted into definitions, that is, (*defined classes*). In this conversion process, we 1) remove the subclass relationships converted from the calculation hierarchy in order to avoid any undesired inheritance, and 2) systematically add disjointness among siblings in order to detect inconsistencies. Finally, we merge the two ontologies, and then use HermiT [33] to infer the mappings between them. The reasoner is used to classify the merge of both enriched ontologies so that concepts from different ontologies with logically equivalent definition would be inferred to be equivalent. With equivalence come also subsumption mappings.

Phase 3 was applied to the French and Spanish ontologies enriched in Phase 2. An example is that `ca:ActifCirculantNet` $\sqsubseteq$ `pgc07:TotalActivo` is inferred from the concept definitions shown in Table 2.3. This inference is due to the fact that the subclass has all the restrictions of the superclass, plus one more, as can be seen from inspection of the table.

The effectiveness of this semantically enriched mapping generation approach

Table 2.3: Example of concept definitions resulting in an inferred mapping, i.e., ca:ActifCirculantNet $\sqsubseteq$ pgc07:TotalActivo

ca:ActifCirculantNet	pgc07:TotalActivo
$\sqsubseteq$((∃hasDepreciability.Nondepreciable $\sqcup$ (∃hasGrossOrNet.Net))	$\sqsubseteq$((∃hasDepreciability.Nondepreciable $\sqcup$ (∃hasGrossOrNet.Net))
$\sqsubseteq$(∃hasClassification.Asset)	$\sqsubseteq$(∃hasClassification.Asset)
$\sqsubseteq$((∃hasCurrentness. (Current $\sqcup$ CurrentByException))	
$\Downarrow$transforming into concept definition	$\Downarrow$transforming into concept definition
ca:ActifCirculantNet $\equiv$	pgc07:TotalActivo $\equiv$
((∃hasDepreciability.Nondepreciable $\sqcup$ (∃hasGrossOrNet.Net)) $\sqcap$ (∃hasClassification.Asset) $\sqcap$ (∃hasCurrentness. (Current $\sqcup$ CurrentByException))	((∃hasDepreciability.Nondepreciable $\sqcup$ (∃hasGrossOrNet.Net)) $\sqcap$ (∃hasClassification.Asset)
rdfs:label “current assets, net”	rdfs:label “total assets”

is implemented and evaluated in Chapter 4.

2.4 Summary

In this chapter, a mapping generation mechanism (SEOM) that exploits background knowledge of financial reporting domain by semantically enriching two financial ontologies is presented. Identified as one of future challenges of ontology matching by Shvaiko and Euzenat [4], exploiting background knowledge is attempted in this work by building a basic ontology, the Basic Accounting Ontology (BAO), consisting of basic concepts and properties that formalize the underlying semantics across different financial reporting schemas. Having background knowledge formalized in this fashion, all other financial concepts can be defined by basic concepts and properties. Both defined using the same background knowledge ontology, two concepts with logically equivalent definitions constitute a good match. The presented approach also differs from most other ontology matching systems in that it is solely logic-based. Given two financial ontologies defined on a common ground, logical reasoners are used to reason over the merge of these two ontologies, so that all equivalent matches are revealed. With equivalence relations and class hierarchies of ontologies come also subclass matches.

The BAO is designed to be extensible and is expected to formalize more semantics that is common to financial reporting schemas from various countries. With a more advanced BAO, this logic-based approach can be deployed to match a wide range of financial reporting schemas.

Chapter 3

Logic-based Mapping Refinement

This chapter describes a logic-based mapping refinement mechanism, which is the main part of this thesis, and an answer to the research question of what are the logic-based principles for mapping refinement (**R4**). There are already a number of logic-based principles and techniques for mapping refinement [11]. This thesis extends some of the existing principles and also proposes one novel principle, which is based on observations of the use case considered herein. To be specific, the conservativity principle is extended to handle the subclass relation. The novel principle, the grouping principle, is proposed based on observations of different granularity of financial ontologies. The consistency principle and the locality principle are also built into our mapping refinement mechanism. The principles are used to detect inconsistency, incoherence or violations. Then we can find explanations for them and then identify those mapping suggestions that cause the inconsistency, incoherence or violations. We also consider a number of heuristics to optimize the refinement process. In addition to a conceptual discussion of these heuristics, we select some as applicable components for the final refinement procedure.

3.1 Related Work

Mapping refinement has been studied by many researchers. On the one hand, mapping refinement is sometimes built into matchers. Among other matchers, LILY [18] uses a few patterns to detect unlikely mappings. Notably, it considers two patterns as follows.

- mappings which cause subclass circles. A mapping $\mathcal{M}$ is inconsistent if it

causes a subclass circle in the involved ontologies, because subclass circles result in the equivalence of all concepts involved in the circle.

- mappings which introduce new equivalence axioms. Two mappings $\mu_1 = \langle A \equiv A', \epsilon_1 \rangle$ and $\mu_2 = \langle B \equiv B', \epsilon_2 \rangle$ are inconsistent if $\mathcal{O}_1 \models (A \equiv B) \wedge \mathcal{O}_2 \not\models (A' \equiv B')$. The same holds if we replace equivalence by disjointness.

These patterns are of an *ad hoc* nature and are just specific cases in which mappings result in new consequences that cannot be derived without the mappings. That is to say, they are, in essence, special cases of the conservativity principle as presented in Section 3.2.1.

On the other hand, there is research devoted entirely to mapping refinement. Meilicke et al. [11] introduced a novel formalization of the problem and proposed several principles to remove unintended mappings. Their approach was evaluated on a commonly used dataset and achieved good results. Qi et al. [34] devised a conflict-based operator, and also proposed relevance-based reasoning techniques for mapping refinement. The authors reported some 'preliminary but interesting evaluation results' to show the usefulness of their approach. Jiménez-Ruiz et al. [12] proposed three logic principles based on observations of large medical ontologies and UMLS-Meta, i.e., the conservativity, consistency and locality principle. Our work follows that in [12], so we give a brief description of the definitions and techniques presented there.

Conservativity

The conservativity principle, first proposed by Jiménez-Ruiz et al. [12], is based on the assumption that ontologies are self-contained, in the sense that they have a proper coverage of a given knowledge domain, and contain no logical inconsistencies within themselves. There it was informally defined as follows. In the process of manually or automatically matching independently developed ontologies, mapping suggestions may introduce new semantic relations to ontologies, where these new semantic relations cannot be entailed by the ontologies separately. In such cases, those mapping suggestions that cause the new relations are questionable, and at least one of them is to be removed in order to avoid those new unintended relations. For example, on the left side of Figure 3.1 there are two mapping suggestions, $\mu_1 = \langle C_{f1} \equiv C_{s1}, \epsilon_1 \rangle$ and $\mu_2 = \langle C_{f2} \equiv C_{s1}, \epsilon_2 \rangle$, proposed for $\mathcal{O}_f$ and $\mathcal{O}_s$. Given μ_1 and μ_2, it follows logically that $C_{f1} \equiv C_{f2}$. If $\mathcal{O}_f \not\models (C_{f1} \equiv C_{f2})$, at least one of the mappings $\{\mu_1, \mu_2\}$ is incorrect because the introduced relation

$C_{f1} \equiv C_{f2}$ cannot be entailed by $\mathcal{O}_f$ alone and thus it can be seen as violation of the semantics of $\mathcal{O}_f$.

This is known as *conservative extension* [35, 36], and it is a computationally hard problem. A simplification of it is presented in [12] to only consider a special kind of axiom, i.e., *equivalent classes*. It only detects conflict cases like $C_{f1} \equiv C_{s1}$ and $C_{f2} \equiv C_{s1}$, where C_{f1} and C_{f2} are in one ontology and C_{s1} in another (illustrated on the left side of Figure 3.1). This simplification reduces the complexity of the reasoning problem considerably and as argued by the authors is very important because very large bio-medical ontologies and thesaurus are computationally demanding. Our definition of the conservativity principle is adopted from the work of Jiménez-Ruiz et al. [12], with an extension to its original simplified implementation, as presented in Section 3.2.1.

Consistency

The consistency principle says that mapping suggestions should be consistent with the ontologies. This is to say $\mathcal{O}_u = \mathcal{O}_f \cup_{\mathcal{M}_{fs}} \mathcal{O}_s$ is consistent and every concept in its signature is satisfiable. We will present an implementation of the conservativity principle in Section 3.2.3, which follows the same idea as that of Jiménez-Ruiz et al. [12].

Locality

The locality principle says that two concepts are likely to be a correct mapping if there are mappings among their respective neighbors. On the other hand, if there is no mapping between neighbors of two concepts, it is necessary to check the validity of the mapping (if any) between these two concepts. So the neighborhood relationship of concepts in a graph indicates semantic similarity. If the locality principle does not hold, the following situations can be identified: (1) mappings among them may be incomplete and new mappings should be discovered, (2) the definitions of concepts in these ontologies may be different or incompatible, or (3) the existing mappings may be erroneous. This principle can be used to compute confidence value for a given mapping, which is discussed in Section 3.2.1.

3.2 Logic Principles

In the following sections, we present a number of logic principles for mapping refinement. Some of principles are adapted from existing work, and there are a

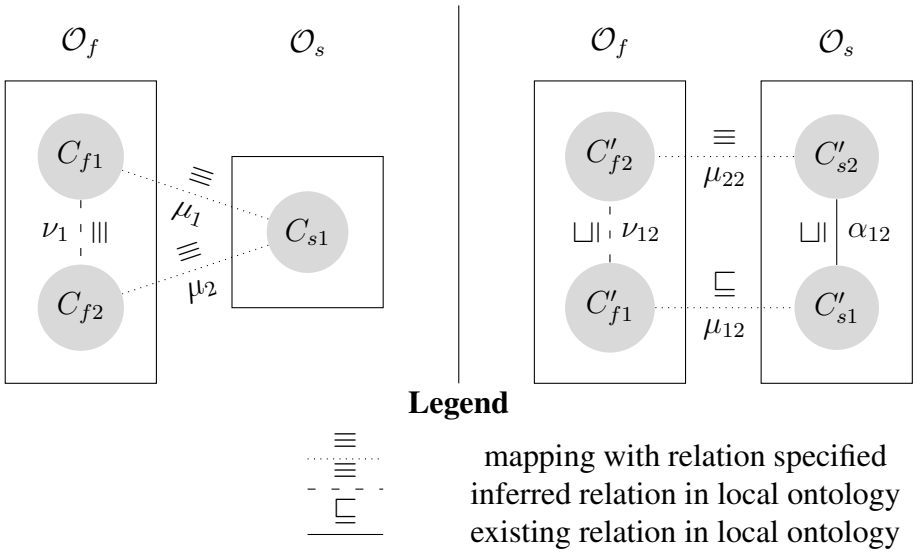


Figure 3.1: The extension of conservativity principle presented in this thesis

few new principles based on observations of the use case ontologies. We will first formalize these principles and then propose methods to realize them.

3.2.1 Extended Conservativity

Here an extension of Jiménez-Ruiz et al.'s implementation of the conservativity principle is presented. The extension is two fold. First, subsumptions (subclass and super-class) are also considered. Second, a generic pattern of violation of the principle is presented. One example of such violation is shown on the right side of Figure 3.1.

An important step to mapping refinement using the extended conservativity principle is to get those violating axioms, i.e., axioms that could be entailed only after introducing mapping suggestions. The set $\mathcal{S}_{vio}(\mathcal{O}_l)$ of violating axioms with respect to a local ontology $\mathcal{O}_l$ is formally defined as follows,

$$\mathcal{S}_{vio}(\mathcal{O}_l) := \{\nu | \Sigma(\nu) \subseteq \Sigma(\mathcal{O}_l), \mathcal{O}_u \models \nu, \mathcal{O}_l \not\models \nu\}. \tag{3.1}$$

where ν can be either subsumption or equivalence of concepts in $\mathcal{O}_l$.

To find those mappings that cause these violations, logical reasoning can be exploited. Consider the example in Figure 3.1. Given two mappings $\mu_{11} = \langle C'_{f1} \sqsubseteq C'_{s1}, \epsilon_{11} \rangle$ and $\mu_{22} = \langle C'_{f2} \equiv C'_{s2}, \epsilon_{22} \rangle$, together with one existing axiom

$\alpha_{12} = (C'_{s1} \sqsubseteq C'_{s2})$ from $\mathcal{O}_s$, a violating axiom $\nu_{12} = (C'_{f1} \sqsubseteq C'_{f2}) \in \mathcal{S}_{vio}(\mathcal{O}_f)$ can be inferred, that is,

$$\{\mu_{11}, \alpha_{12}, \mu_{22}\} \models \nu_{12}. \tag{3.2}$$

In this example $\{\mu_{11}, \alpha_{12}, \mu_{22}\}$ is the minimal set that entails ν_{12}, is called the *explanation* of ν_{12}.

Definition 11. (Explanation). *An explanation Ω_ν of axiom ν is a minimal set of axioms that entail ν, i.e.,*

$$\Omega_\nu \models \nu, \text{ s.t. } \Omega_\nu \subseteq \mathcal{O}_u \text{ and } \forall \Omega' \subset \Omega_\nu, \Omega' \not\models \nu. \tag{3.3}$$

An explanation indicates the causing mappings of a violating axiom. Note that we define an explanation as a minimal set of axioms, though in other work explanations are not necessarily minimal. As for the running example, the causing mapping could be either μ_{11} or μ_{22}. If either one of them is deleted, ν_{12} no longer follows and the violation is resolved (Note that we assume $\{\mu_{12}, \alpha_{12}, \mu_{22}\}$ is the only explanation for ν_{12}.). In fact, either one of them constitutes a *diagnosis* of ν_{12}, as formally defined as below.

Definition 12. (Diagnosis). *A diagnosis Δ_ν of ν is a minimal set of axioms from $\mathcal{O}_m$, without which ν does not follow, i.e.,*

$$\Delta_\nu \subseteq \mathcal{O}_m, \text{ s.t. } \mathcal{O}_f \cup \mathcal{O}_s \cup \mathcal{O}_m \models \nu, \mathcal{O}_f \cup \mathcal{O}_s \cup \mathcal{O}_m \backslash \Delta_\nu \not\models \nu, \tag{3.4}$$

where $\forall \Delta' \subset \Delta_\nu, \mathcal{O}_f \cup \mathcal{O}_s \cup \mathcal{O}_m \backslash \Delta' \models \nu$.

Note that in the context of mapping refinement, diagnosis is defined to contain only mappings.

The difference between explanation and diagnosis can be illustrated using the example above. In Figure 3.1, ν_{12} is a violating axiom, $\{\mu_{11}, \alpha_{12}, \mu_{22}\}$ is an explanation for it, whereas $\{\mu_{11}\}$ and $\{\mu_{22}\}$ are two different diagnoses for it. Following Reiter's theory [37], there are two ways to identify and remove mappings which cause violating axioms: computing explanations and computing diagnosis, which are detailed as follows.

Compute Explanation

One way to resolve violations is to compute one explanation for each violating axiom and remove one mapping from the explanation, followed by a test to check

Algorithm 1 Compute one explanation for an unsatisfiable concept

Require: a local ontology $\mathcal{O}$, and a concept C that is unsatisfiable in $\mathcal{O}$
Ensure: explanation Ω_C, i.e., a minimal set of axioms in $\mathcal{O}$ that cause C to be unsatisfiable

```
1: procedure GETEXPLANATION(O, C)
2:     Ω_C ← O
3:     for α ∈ O do
4:         Ω_C ← Ω_C \ {α}
5:         if Ω_C ⊭ (C ⊑ ⊥) then
6:             Ω_C ← Ω_C ∪ {α}
7:         end if
8:     end for
9:     return Ω_C
10: end procedure
```

whether this violating axiom is successfully resolved. If not, repeat the process by computing a new explanation for this violating axiom. This is possible because there can be more than one explanation for a violating axiom. In Figure 3.2, for example, if there is another mapping $\mu_{12} = \langle C'_{f1} \sqsubseteq C'_{s2}, \epsilon_{12} \rangle$, the set $\{\mu_{12}, \mu_{22}\}$ is also an explanation for ν_{12}, which can be formally denoted as

$$\{\mu_{12}, \mu_{22}\} \models \nu_{12}.$$

This explanation still remains if the mapping μ_{11} from the explanation $\{\mu_{11}, \alpha_{12}, \mu_{22}\}$ is removed.

Algorithm 1 presents the process of computing one explanation Ω_C for an unsatisfiable concept C in the context of an ontology $\mathcal{O}$. To begin with, all the logical axioms are taken as an initial explanation Ω_C. Then one axiom α is deleted from Ω_C and the updated Ω_C is checked whether it still entails $C \sqsubseteq \bot$, i.e., C is unsatisfiable. If this Ω_C can now make C satisfiable, it means α is an indispensable part of Ω_C, thus it is added back to Ω_C. Each one of the axioms is checked following this procedure. In the end, Ω_C contains the set of axioms that constitute one explanation for C. This is a rather intuitive procedure and is used later in resolving violations of the conservativity principle, the consistency principle, etc.

A complete procedure for the extended conservativity principle is presented in Algorithm 2. It takes as input two involved ontologies $\mathcal{O}_f, \mathcal{O}_s$ and the mapping ontology $\mathcal{O}_m$, applies the extended conservativity principle to remove all mappings which cause violating axioms, resulting in a refined mapping ontology $\mathcal{O}'_m$.

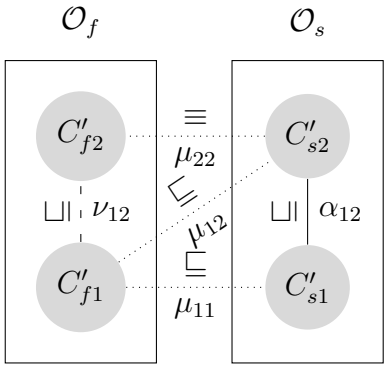


Figure 3.2: Example of multiple explanations for one violating axiom

The algorithm is explained step by step.

- ▷ MERGEONTOLOGIES takes $\mathcal{O}_f$, $\mathcal{O}_s$ and $\mathcal{O}_m$ as input and merges these three ontologies into $\mathcal{O}_u$, that is, $\mathcal{O}_u = \mathcal{O}_f \cup \mathcal{O}_s \cup \mathcal{O}_m$. In OWL API [38], there is *OWLOntologyMerger* class specifically designed for this task.

- ▷ GETALLINFERREDAXIOMS takes an ontology as an input and outputs all inferred axioms of interest of this ontology. All inferred axioms are generated by using logic reasoner. There are different kinds of inferred axioms that can be generated by a given reasoner, for example, subclass, equivalent classes, equivalent object properties, class assertions, etc. Because in the current use case it is only interesting to check whether candidate mappings have introduced new logical relations (subclass and equivalence relationships) among local concepts, only two kinds of inferred axioms, subclass and equivalent classes, are generated. This is configured by supplying *InferredOntologyGenerator* with a list consisting of *InferredEquivalentClassAxiomGenerator* and *InferredSubClassAxiomGenerator* in OWL API.

- ▷ GETVIOLATINGAXIOMS takes the merged ontology $\mathcal{O}_u$ and one of two local ontologies $\mathcal{O}_l$ as input and computes the set $\mathcal{S}_{vio}$ of axioms that are not in $\mathcal{O}_l$, but inferred in $\mathcal{O}_u$. As presented in Algorithm 3, $\mathcal{O}_l$ serves as a filter because it is only interesting for us to see whether there are violating axioms w.r.t this local ontology. While traversing all logical axioms in $\mathcal{O}_u$, if one of them, α, concerns only concepts in $\mathcal{O}_l$ but cannot be entailed by $\mathcal{O}_l$, α is then added to the set $\mathcal{S}_{vio}$ of violating axioms.

Algorithm 2 Extended conservativity via computing explanation

Require: two local ontologies $\mathcal{O}_f$ and $\mathcal{O}_s$, corresponding mapping ontology $\mathcal{O}_m$
Ensure: refined mapping ontology $\mathcal{O}'_m$

procedure EXCONSERVEXP($\mathcal{O}_f, \mathcal{O}_s, \mathcal{O}_m$)
  $\mathcal{O}_u \leftarrow$ MERGEONTOLOGIES($\mathcal{O}_f, \mathcal{O}_s, \mathcal{O}_m$)
  for $\mathcal{O}_l \in \{\mathcal{O}_f, \mathcal{O}_s\}$ **do**
    $\mathcal{S}_{vio} \leftarrow$ GETVIOLATINGAXIOMS($\mathcal{O}_u, \mathcal{O}_l$)
    repeat
      for $\nu \in \mathcal{S}_{vio}$ **do**
        if $\mathcal{O}_u \models \nu$ **then**
          $S_{err} \leftarrow \emptyset$
          $C_{unsat} \leftarrow$ SATISFIABILITYCONVERTER(ν)
          repeat
            $\Omega_{C_{unsat}} \leftarrow$ GETEXPLANATION($\mathcal{O}_u, C_{unsat}$)
            $\mathcal{S}_{sus} \leftarrow \Omega_{C_{unsat}} \cap \mathcal{O}_m$
            $\mu_{err} \leftarrow$ GETERRMAPPING($\mathcal{S}_{sus}$)
            $S_{err} \leftarrow S_{err} \cup \mu_{err}$
            $\mathcal{O}_m \leftarrow \mathcal{O}_m \backslash \{\mu_{err}\}$
            $\mathcal{O}_u \leftarrow \mathcal{O}_u \backslash \{\mu_{err}\}$
          until $\mathcal{O}_u \not\models (C_{unsat} \sqsubseteq \bot)$
          for $\mu \in S_{err}$ **do**
            $\mathcal{O}_u \leftarrow \mathcal{O}_u \cup \mu$
            if $\mathcal{O}_u \models (C_{unsat} \sqsubseteq \bot)$ **then**
              $\mathcal{O}_u \leftarrow \mathcal{O}_u \backslash \{\mu\}$
            else
              $\mathcal{O}_m \leftarrow \mathcal{O}_m \cup \mu$
            end if
          end for
        end if
      end for
      $\mathcal{S}_{vio} \leftarrow$ GETVIOLATINGAXIOMS($\mathcal{O}_u, \mathcal{O}_l$)
    until $\mathcal{S}_{vio} = \emptyset$
  end for
  $\mathcal{O}'_m \leftarrow \mathcal{O}_m$
  return $\mathcal{O}'_m$
end procedure

Algorithm 3 Compute violating axioms introduced by mapping suggestions

Require: merged ontology $\mathcal{O}_u$ and local ontologies $\mathcal{O}_l$
Ensure: set of violating axioms $\mathcal{S}_{vio}$

```
1: procedure GETVIOLATINGAXIOMS(O_u, O_l)
2:     S_vio ← ∅
3:     S_u ← GETALLINFERREDAXIOMS(O_u)
4:     for α ∈ S_u do
5:         if Σ(α) ⊆ Σ(O_l) and O_l ⊭ α then
6:             S_vio ← S_vio ∪ α
7:         end if
8:     end for
9:     return S_vio
10: end procedure
```

▷ SATISFIABILITYCONVERTER takes a logical axiom α as input and generates a corresponding concept C_{unsat} as output such that α is entailed if and only if C_{unsat} is unsatisfiable. Debugging unsatisfiable concepts has been studied extensively and has been a part of most OWL reasoners [39, 40]. This can be done by the following observation.

Given logical axiom $\nu \in \mathcal{S}_{vio}$ of the form $A \sqsubseteq B$, let $C_{unsat} = A \sqcap \neg B$, then we have

$$\mathcal{O}_u \models \nu \Leftrightarrow \textit{ } C_{unsat} \textit{ is unsatisfiable w.r.t } \mathcal{O}_u. \tag{3.5}$$

Axioms of the form $A \equiv B$ are first converted to subsumptions $\{A \sqsubseteq B, B \sqsubseteq A\}$. Their corresponding unsatisfiable concepts can be computed following the same method. This conversion only works for DL languages which allow for negation and is a standard practice to generate explanations for arbitrary axioms.

▷ GETEXPLANATION takes the merged ontology $\mathcal{O}_u$ and the unsatisfiable concept C_{unsat} as input and computes one explanation Ω_C for C_{unsat}. We have proposed a procedure to get an explanation of an unsatisfiable concept in Algorithm 1, which is a rather primitive procedure by using reasoners in a black-box way. For efficiency reasons, we use Pellet in our mapping refinement system to compute one explanation at a time.

▷ GETERRMAPPING takes as input the set of suspect mappings $\mathcal{S}_{sus}$ and selects one mapping μ_{err} to delete. In some cases, the candidate mappings

have their corresponding confidence values given by matchers. In such cases, the mapping with the lowest confidence value is picked out as μ_{err}. In some other cases, mapping candidates are without confidence values. In such cases, heuristics are applied to compute a temporary confidence value for each mapping in order to pick out μ_{err}. In the following, there are a couple of heuristics discussed for this purpose.

Heuristics for computing confidence value. From $\mathcal{S}_{sus}$ one mapping μ_{err} should be identified as the one mapping that is most likely to be erroneous. In order to distinguish this mapping, *confidence value* ϵ, a measurement of degree of correctness, for each suspect mapping is computed. One idea to compute ϵ proposed by Jiménez-Ruiz et al. in [12] is based on the locality principle. Intuitively the locality principle suggests that if there is an established mapping between two ontologies, the neighbors of the two concepts are likely to be matched as well. The ontology modularization framework [41] is used to compute the set of neighbors for concepts, and confidence value can also be computed. For instance, given a mapping μ from C_f in $\mathcal{O}_f$ to C_s in $\mathcal{O}_s$, modules $\mathsf{M}^f_{C_f}$ and $\mathsf{M}^s_{C_s}$ can be computed as the sets of neighbors for C_f and C_s respectively. $\epsilon(\mu)$ can be computed as follows.

$$\epsilon(\mu) = \frac{|\text{Mapped concepts in } \Sigma(\mathsf{M}^f_{C_f})| + |\text{Mapped concepts in } \Sigma(\mathsf{M}^s_{C_s})|}{|\Sigma(\mathsf{M}^f_{C_f})| + |\Sigma(\mathsf{M}^s_{C_s})|}. \tag{3.6}$$

Now for each $\mu \in \mathcal{S}_{sus}$, there is confidence value $\epsilon(\mu)$. The mapping to delete μ_{err} is defined as

$$\mu_{err} \in \mathcal{S}_{sus}, s.t. \ \forall \mu \in \mathcal{S}_{err} \backslash \{\mu_{err}\}, \epsilon(\mu) \geq \epsilon(\mu_{err}).$$

Now that μ_{err} is successfully identified, it should be removed from $\mathcal{O}_m$, resulting in an updated mapping ontology. Since $\mathcal{O}_m$ also contributes to $\mathcal{O}_u$, $\mathcal{O}_u$ is updated by removing μ_{err}.

S_{err} is the set of mappings that are from different explanations for the violating axiom ν. It is possible that the order in which explanations are computed would delete more mappings than necessary. For example, in Figure 3.2, there are two different explanations for violating axiom ν_{12}, i.e., $\Omega_1 = \{\mu_{11}, \alpha_{12}, \mu_{22}\}$ and $\Omega_2 = \{\mu_{12}, \mu_{22}\}$. Suppose that the confidence values of these three mappings are in such an order, $\epsilon_{11} > \epsilon_{22} > \epsilon_{12}$. In the process of resolving ν_{12}, if Ω_2 is the first explanation found, the algorithm would remove μ_{12} because it has a lower

confidence value than μ_{22}. After this removal, the violation remains. Another explanation Ω_1 will then be found, the algorithm would remove μ_{22} due to the same reason. At this point, $S_{err} = \{\mu_{12}, \mu_{22}\}$. It is, however, not the minimal set of mappings to delete from $\mathcal{O}_m$. Instead, deleting μ_{22} alone would successfully resolve the violation. Considering such cases, it is necessary to re-check each mapping in S_{err} in order to delete the minimal set of mappings that cause violations.

After all steps above (line 11 – 16), the violating axiom ν may have been correctly resolved. The procedure is however incomplete because only one explanation of ν is investigated. There can be more than one explanation for ν. To make sure that ν is successfully resolved, it is necessary to recheck whether $\mathcal{O}_u \models \nu$. If $\mathcal{O}_u \models \nu$ holds, steps (line 11 – 16) are repeated until ν is no longer entailed by $\mathcal{O}_u$. This design ensures that each ν is successfully resolved and that this refinement process is complete in discarding all mappings that violate the extended conservativity principle. After successfully resolving one violating axiom, the procedure proceeds to check all other violating axioms. Note that correctly resolving one violating axiom may cause other violating axioms to be resolved at the same time. Therefore, we check each violating axiom before resolving it (line 7). This process ends when there are no violating axioms present. It is important to note that the mapping ontology $\mathcal{O}_m$ is constantly changing and any time an entailment check is invoked, the latest $\mathcal{O}_m$ is used. This also demands incremental reasoning capability of logical reasoners, i.e., logical reasoners can take account of new updates to an ontology without reloading the whole ontology. As surveyed by Dentler et al. [42], there are only a limited number of existing logical reasoners that support incremental reasoning, among which Pellet [39] supports both incremental consistency checking and incremental classification. Therefore Pellet is used in the implementation.

In Algorithm 2 (line 18 – 25), there is such a rechecking procedure after each violation is successfully resolved. Simply S_{err} is traversed and each mapping μ in it is first added to the latest $\mathcal{O}_u$. After this addition, if C_{unsat} becomes again unsatisfiable, that is to say, the violation again arises, μ is removed from $\mathcal{O}_u$; otherwise, μ does not belong to the minimal set of mappings to be deleted and is therefore added back to $\mathcal{O}_m$. This process is to ensure that only a minimal set of mappings are removed.

After the whole set $\mathcal{S}_{vio}$ of violating axioms have been resolved, the updated merged ontology $\mathcal{O}_u$ is used again to check whether there are still some violating axioms present or not. If there are, the algorithm goes again through the process of computing new $\mathcal{S}_{vio}$ and resolving each one of them. If, however, there are no

more violating axioms, the violation checking for one local ontology ends. The process of recomputing $\mathcal{S}_{vio}$ is necessary due to the *timeout* mechanism in the current implementation. For some violating axioms, there are a large number of explanations present. We look into a given number of its explanations before continuing with the next violating axiom. Thus, to ensure there are no more violating axioms, we have to recompute $\mathcal{S}_{vio}$ to make sure that it is empty. This timeout mechanism proves to be helpful because the whole refinement process is dynamic in the sense that resolving one violating axiom may help resolve some other violating axioms. This refinement process is applied to both local ontologies (line 2). Finally the algorithm outputs the latest mapping ontology as $\mathcal{O}'_m$. $\mathcal{O}'_m$ satisfies two conditions: 1) $\mathcal{O}'_m \cup \mathcal{O}_f \cup \mathcal{O}_s$ is consistent and contains no unsatisfiable concepts; 2) $\mathcal{O}'_m$ does not introduce violating axioms to either $\mathcal{O}_f$ or $\mathcal{O}_s$.

Due to the interconnectivity of local ontologies and the mapping ontology, deleting one mapping changes the merge of the three ontologies. This requires updating ontologies constantly, which undermines the efficiency of this algorithm. Moreover, this algorithm depends heavily on the logical reasoning service to compute explanations, which is a major obstacle toward scalability.

Compute Diagnosis

The other way to removing mappings that cause violations is to compute a diagnosis of the violating axiom, and then remove this diagnosis from the mapping suggestions. It is clear that one violating axiom ν could not be entailed by the merge of two local ontologies, i.e.,

$$\mathcal{O}_f \cup \mathcal{O}_s \not\models \nu.$$

Now on the basis of $\mathcal{O}_{fs} = \mathcal{O}_f \cup \mathcal{O}_s$, mappings from $\mathcal{O}_m$ are added to $\mathcal{O}_{fs}$ one at a time following a certain order. Each time a mapping $\mu \in \mathcal{O}_m$ is added to $\mathcal{O}_{fs}$, it is then checked whether the latest $\mathcal{O}_{fs}$ entails ν. If yes, according to the definition of diagnosis, μ belongs to the diagnosis Δ_ν and is added to Δ_ν. Otherwise, another mapping μ' is added to $\mathcal{O}_{fs}$, followed by another entailment check. This process is repeated until all the mappings from $\mathcal{O}_m$ are checked.

Algorithm 4 presents such a process. In fact, it is a much general process in the sense that 1) any logical axiom can be converted to its corresponding unsatisfiable concept; 2) the initial set S_{tmp} of axioms can be set as a baseline, which can be the empty set or any set of axioms that make C satisfiable. A baseline ontology $\mathcal{O}_b$ is to initialize S_{tmp}. In the case above, violating axiom ν can be converted to

C such that ν is entailed if and only if C is unsatisfiable. The baseline ontology is $\mathcal{O}_{fs}$, since $\mathcal{O}_{fs} \not\models \nu$.

Algorithm 4 Compute diagnosis for an unsatisfiable concept

Require: ontology $\mathcal{O}$, a concept C that is unsatisfiable in $\mathcal{O}$ and ontology $\mathcal{O}_b$ serving as a baseline where C is satisfiable

Ensure: diagnosis Δ_C, i.e., minimal set of axioms in $\mathcal{O}$ that cause C to be unsatisfiable

```
1:  procedure GETDIAGNOSIS(O, C, O_b)
2:      S_tmp ← O_b
3:      Δ_C ← ∅
4:      for α ∈ O do
5:          S_tmp ← S_tmp ∪ α
6:          if S_tmp ⊨ (C ⊑ ⊥) then
7:              Δ_C ← Δ_C ∪ α
8:              S_tmp ← S_tmp \ {α}
9:          end if
10:     end for
11:     return Δ_C
12: end procedure
```

As pointed out by Meilicke et al. [43], the order in which mappings are added and checked in the process is very important. Take again the example in Figure 3.1. Suppose ν_{12} is a violating axiom in $\mathcal{O}_f$. To compute its diagnosis $\Delta_{\nu_{12}}$, mapping μ_{11} is added to $\mathcal{O}_{fs}$ as the first mapping. At this point, without μ_{22}, ν_{12} cannot be entailed by $\mathcal{O}_{fs}$. The process proceeds by adding μ_{22} to $\mathcal{O}_{fs}$, causing $\mathcal{O}_{fs} \models \nu_{12}$. Then it would delete μ_{22} from $\mathcal{O}_m$. It might be the case that μ_{22} is actually more likely to be correct than μ_{11}. In this case, the procedure deletes more likely mappings while keeping ones that are less likely. Except for relying on domain experts to find most likely mappings, one can make use of confidence values generated by matchers or the presented heuristic. Now if each mapping has its own confidence value, it is intended that a mapping with higher confidence value should be preferred to ones with lower values. Following this idea, the set of mappings $\mathcal{O}_m$ can first be sorted in descending order with respect to their confidence values. Then mappings with higher confidence values are added to $\mathcal{O}_{fs}$ before the ones with lower values. Back to the example, if it is known that the confidence value ϵ_{22} of mapping μ_{22} is higher than ϵ_{11}, the confidence value of μ_{11}, μ_{22} would be added to $\mathcal{O}_{fs}$. When later violation arises, μ_{11} is removed from

$\mathcal{O}_m$. Thus, good mappings, i.e., mappings with higher confidence value here, are kept while bad mappings are deleted.

A complete procedure of the extended conservativity principle following the diagnosis approach is presented in Algorithm 5. As with Algorithm 2, the merged ontology $\mathcal{O}_u$ of all three ontologies is computed. In addition, the merge $\mathcal{O}_{fs}$ of local ontologies $\mathcal{O}_f$ and $\mathcal{O}_s$ is also computed and later serves as the starting point of the diagnosis process. As discussed, the set of mappings in $\mathcal{O}_m$ is sorted in descending order in terms of confidence values of the mappings. This is exactly what the function SORTDESCENDING does, though in the algorithm as a short hand notation it outputs a 'sorted' ontology $\mathcal{O}_m$ (line 4).

Algorithm 5 Extended conservativity via computing diagnosis

Require: two local ontologies $\mathcal{O}_f$ and $\mathcal{O}_s$, corresponding mapping ontology $\mathcal{O}_m$
Ensure: refined mapping ontology $\mathcal{O}'_m$

```
1:  procedure EXCONSERVDIAG(O_f, O_s, O_m)
2:      O_u ← MERGEONTOLOGIES(O_f, O_s, O_m)
3:      O_fs ← MERGEONTOLOGIES(O_f, O_s)
4:      O_m ← SORTDESCENDING(O_m)
5:      for O_l ∈ {O_f, O_s} do
6:          S_vio ← GETVIOLATINGAXIOMS(O_u, O_l)
7:          for ν ∈ S_vio do
8:              C_unsat ← SATISFIABILITYCONVERTER(ν)
9:              Δ_ν ← GETDIAGNOSIS(O_u, C_unsat, O_fs)
10:             if Δ_ν ≠ ∅ then
11:                 O_m ← REMOVEAXIOMS(O_m, Δ_ν)
12:                 O_u ← REMOVEAXIOMS(O_u, Δ_ν)
13:             end if
14:         end for
15:     end for
16:     O'_m ← O_m
17:     return O'_m
18: end procedure
```

The most important difference from Algorithm 2 lies in the way of handling a given violating axiom ν. The diagnosis Δ_ν of ν is initiated as an empty set. Note that at the outset, according to the definition of violating axiom, $\mathcal{O}_{fs} \not\models \nu$ holds at the beginning. Then each mapping μ from the mapping ontology $\mathcal{O}_m$ is added to $\mathcal{O}_{fs}$. Note that as the mappings are in a specific order the one mapping

with highest confidence value will be the first one added to $\mathcal{O}_{fs}$. The updated ontology $\mathcal{O}_{fs}$ is then checked whether $\mathcal{O}_{fs} \models \nu$ holds. In fact, after only one mapping is added, the entailment would not change. Because a mapping only establish a connection between two concepts from different ontologies, it would need at least two mappings to cause violations, in which case two concepts from the same ontology are indirectly connected with the given mappings. In the process of adding new mappings to $\mathcal{O}_{fs}$, if ever $\mathcal{O}_{fs} \models \nu$, the mapping added last μ^* is responsible for this violation. Compared to other mappings before μ^*, it has the lowest confidence value and is therefore added to the diagnosis Δ_ν. Further, it has to be removed from the latest $\mathcal{O}_{fs}$ so that the set of axioms in $\mathcal{O}_{fs}$ contains only mappings which do not violate the conservativity principle. After traversing all mappings in $\mathcal{O}_m$, the diagnosis Δ_ν now contains all those mappings that could lead to the violating axiom ν. If there is more than one mapping in Δ_ν, these mappings are independent from each other in causing the violating axiom ν. Therefore, all those mappings should be deleted from the mapping ontology $\mathcal{O}_m$ (line 11). Because those mappings are also part of the merged ontology $\mathcal{O}_u$, they also need to be removed from $\mathcal{O}_u$ (line 12).

After one violating axiom is successfully resolved, the algorithm proceeds to resolve other violating axioms. It is important to note that to check every violating axiom a new $\mathcal{O}_{fs}$ is needed (line 9). There are two ways to obtain a new $\mathcal{O}_{fs}$. First, one can delete all mappings added to it in the previous round. This demands some recording mechanism to keep track of all those mappings added to $\mathcal{O}_{fs}$ previously. One advantage of this approach is that the reasoner hosting $\mathcal{O}_{fs}$ does not need to be re-initiated. The second approach is to compute again the merge of local ontologies $\mathcal{O}_f$ and $\mathcal{O}_s$. This approach however demands that the reasoner reload the newly merged ontology $\mathcal{O}_{fs}$ again. Because initiating logical reasoner is more expensive than removing axioms from ontology, the first approach is used in the implementation. After the whole set $\mathcal{S}_{vio}$ of violating axioms have been resolved, this violation checking process proceed with another local ontology since there are two local ontologies to be matched. Finally, the algorithm outputs a refined mapping ontology $\mathcal{O}_m$ as $\mathcal{O}'_m$.

The complexity of this algorithm is analyzed as follows. For a given local ontology, suppose the size of violating axioms $\mathcal{S}_{vio}$ is m, i.e., $|\mathcal{S}_{vio}| = m$. Suppose there are n candidate mappings, i.e., $|\mathcal{O}_m| = n$, the algorithm calls entailment check $(m \times n)$ times. For each violating axiom ν, Algorithm 2 (the explanation approach) has to compute all its explanations and then check each one of them. In contrast, Algorithm 5 (the diagnosis approach) computes its diagnosis only once. Considering these observations, the diagnosis approach is expected to perform

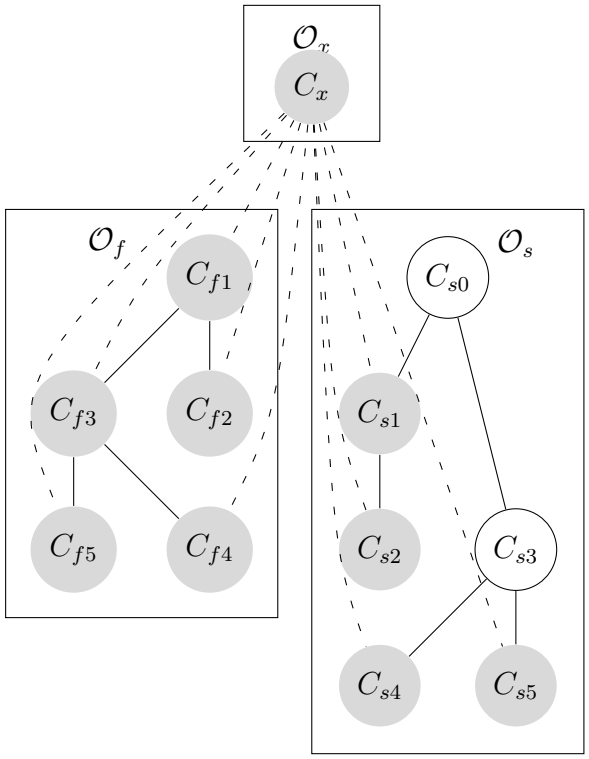


Figure 3.3: Grouping principle

better than the explanation approach in terms of runtime.

3.2.2 Grouping Principle

Grouping principle is an *ad hoc* principle based on observations of the use case considered. In the use case, there is an xEBR taxonomy, which is built to facilitate interoperability among various national taxonomies, called *local* taxonomies. While a given local taxonomy contains several thousands of concepts, the xEBR taxonomy contains only a couple of hundred of concepts, and therefore can be seen as an abstraction of financial concepts used in different countries. It is meant to be a generic reflection of concepts and hierarchies in the financial reporting domain and many details (e.g., low level concepts, or country-specific concepts, etc) are omitted. Thus it is less granular than local taxonomies. The XBRL Europe Business Registers Working Group (xEBR WG)[1], who had created this xEBR tax-

[1] See http://www.xbrleurope.org/working-groups/xebr-wg .

onomy, also created manual mappings from several local taxonomies to it. Given these xEBR WG mappings, it is observed that one xEBR concept can be mapped to more than one local concepts with different relations in certain local taxonomy.

From a different point of view, it can be argued that all local concepts that are mapped to the same xEBR concept share certain common property. And that each of these concepts is mapped to xEBR concept with different relations makes it interesting to investigate possible *internal structure* among these local concepts. In Figure 3.3, the dashed lines represent mapping suggestions from $\mathcal{O}_x$ to $\mathcal{O}_f$ and $\mathcal{O}_s$. On the left side, concepts C_{fi}, where $i \in [1, 5]$, are all mapped to C_x, possibly with different relations and different confidence values. If all these mappings are correct in semantic and logical sense, this set of concepts must share certain properties and have semantic relations among themselves. Like in the above example, concept C_{f1} is a super concept all the rest concepts. C_x could be a high level concept, semantically equivalent to C_{f1} and broader than C_{f2}, C_{f3}, C_{f4} and C_{f5}. This assumption is also consistent with the fact that all of them are mapped to C_x (possibly) with different relations and confidence values. The mapping suggestions, therefore, are more likely to be correct. In contrast, on the right side of the figure, in $\mathcal{O}_s$, concepts C_{sj}, where $j \in \{1, 2, 4, 5\}$, are mapped to C_x, but C_{s0} and C_{s3} are not. It is suggested by the mappings that C_{s1} and C_{s2} are semantically related to C_{s4} and C_{s5}. As can be seen in the figure, there are no apparent relations among them. This observation undermines the correctness of (some of) the mapping suggestions. In the following, we formalize this observation as the grouping principle.

Formal Definition

A set of local concepts that are mapped to a common concept in another ontology is called a *group*.

Definition 13. (Anchor concept and Group). *Given a local ontology $\mathcal{O}_l$ and a mapping ontology $\mathcal{O}_m$, group G_{C_x} is defined as*

$$G_{C_x} = \{C_l | \rho(C_x, C_l) \in \mathcal{O}_m, C_l \in \Sigma(\mathcal{O}_l)\}, \tag{3.7}$$

where $C_x \in \Sigma(\mathcal{O}_m) \backslash \Sigma(\mathcal{O}_l)$ and C_x is called an anchor *concept.*

Given this definition, the grouping principle can be defined as follows.

The Grouping Principle: For every group G_C, where $C \in \Sigma(\mathcal{O}_m) \backslash \Sigma(\mathcal{O}_l)$, if $|G_C| \geq 2$, there exists semantic relation among the concepts in G_C. Formally, let

$C_i, C_j \in G_C$ and $C_i, C_j \in \Sigma(\mathcal{O}_l)$, where $i \neq j$; let $\rho(C_i, C_j)$ denote the semantic relation between them, where $\rho \in \{\equiv, \sqsubseteq, \sqsupseteq\}$. There must exist $\rho \in \{\equiv, \sqsubseteq, \sqsupseteq\}$, such that

$$\mathcal{O}_l \models \rho(C_i, C_j).$$

In considering only $\{\equiv, \sqsubseteq, \sqsupseteq\}$ as the possible relations between any pair of local concept, there must be at least one concept in the group that is the super concept of all the rest. In the following, an algorithm is presented based on this idea.

Algorithm 6 Grouping principle

Require: local ontologies $\mathcal{O}_l$, corresponding mapping ontology $\mathcal{O}_m$
Ensure: whether $\mathcal{O}_m \cup \mathcal{O}_l$ follow the grouping principle

procedure GROUPPRINCIPLE($\mathcal{O}_l, \mathcal{O}_m$)
  $groupCounter \leftarrow 0$
  $satisfyingGroupCounter \leftarrow 0$
  $\mathcal{S}_{anchor} \leftarrow \Sigma(\mathcal{O}_m) \backslash \Sigma(\mathcal{O}_l)$
  for $C_a \in \mathcal{S}_{anchor}$ **do**
    $G_{C_a} \leftarrow$ GETCONCEPTSGROUP($\mathcal{O}_m, C_a$)
    if $|G_{C_a}| \geq 2$ **then**
      $isSatisfyingGroup \leftarrow$ FALSE
      $groupCounter$++
      for $C_{sup} \in G_{C_a}$ **do**
        $isSuperClass \leftarrow$ TRUE
        for $C_{sub} \in G_{C_a} \backslash \{C_{sup}\}$ **do**
          $isConsis \leftarrow$ is $\mathcal{O}_l \models (C_{sub} \sqsubseteq C_{sup})$?
          $isSuperClass \leftarrow isSuperClass$ && $isConsis$
        end for
        $isSatisfyingGroup \leftarrow isSatisfyingGroup \,||\, isSuperClass$
      end for
      if $isSatisfyingGroup$ **then**
        $satisfyingGroup$++
      end if
    end if
  end for
  PRINT($satisfyingGroupCounter/groupCounter$)
  return $isSatisfyingGroup$
end procedure

The grouping principle is different from the locality principle in their assumptions. The locality principle is based on the assumptions that neighbors of correctly matched concept are also likely to be matched; while the grouping principle assumes that a set of concepts that are mapped to a common concept should have some common properties and thus there is some interesting internal structure among them.

Proposed Method

Algorithm 6 presents a primitive procedure to found out whether the merge of local ontology $\mathcal{O}_l$ and its mapping ontology $\mathcal{O}_m$ with a core ontology follows the grouping principle. It takes as input a local ontology $\mathcal{O}_l$ and a relevant mapping ontology $\mathcal{O}_m$, i.e., $\Sigma(\mathcal{O}_l) \cap \Sigma(\mathcal{O}_m) \neq \emptyset$. There are also a couple of counters to give more information about the mappings. $groupCounter$ counts the number of sets of local concepts that have more than one concept, while $satisfyingGroupCounter$ counts the number of sets of local concepts that follow the grouping principle. Later the set $\mathcal{S}_{anchor}$ of potential anchor concepts is computed as those concepts occurring in $\mathcal{O}_m$ but not in $\mathcal{O}_l$. For each of these potential anchor concepts C_a, the function GETCONCEPTSGROUP computes the set G_{C_a} of local concepts that are connected to C_a via the mapping ontology $\mathcal{O}_m$. The group G_{C_a} is a set of concepts from $\mathcal{O}_l$, i.e., $G_{C_a} \subseteq \Sigma(\mathcal{O}_l)$. If G_{C_a} contains more than one local concept, $groupCounter$ increases by 1. The next step is to check whether this group G_{C_a} follows the grouping principle. Another boolean indicator $isSatisfyingGroup$ is also initiated to be FALSE in order to keep track of the satisfiability property of the group G_{C_a}.

Again there are different patterns in which the grouping principle can be checked. In this work, one pattern is identified, i.e., whether in a group there is a concept that is the super concept of all the rest concepts. Now given a group of concept G_{C_a}, each one of its member concept is checked whether it is this super concept. The idea is very intuitive: a concept $C_{sup} \in G_{C_a}$ is assumed to be this super concept of all; then for each of the rest concepts $C_{sub} \in G_{C_a} \backslash \{C_{sup}\}$, the subclass relation $C_{sub} \sqsubseteq C_{sup}$ is checked whether it can be entailed by the local ontology $\mathcal{O}_l$, i.e.,

$$\mathcal{O}_l \models (C_{sub} \sqsubseteq C_{sup}),$$

the result of which is stored in a boolean variable $isConsis$. Moreover, for each C_{sup} there is a boolean indicator $isSuperClass$, initiated to be TRUE. The result of each entailment check is accumulated in $isSuperClass$ via the boolean

operation "AND", denoted here in Java syntax as &&. Thus, if C_{sup} is in fact the super concept of all the rest, $isSuperClass$ would remain TRUE after all the rest concepts have been tried and checked. Later $isSuperClass$ is also added to $isSatisfyingGroup$ via the boolean "OR" operation, denoted in Java syntax as $||$. Thus, if one concept is known to be the super concept of the rest, this group G_{C_a} follows the grouping principle. The number of groups that follow the grouping principle is also counted by $satisfyingGroupCounter$.

After all groups have been checked, the percentage of groups following the principle can also be computed as

$$percentage = \frac{satisfyingGroupCounter}{groupCounter}.$$

The percentage indicates to what extent the mapping ontology $\mathcal{O}_m$ complies with the grouping principle. The following is some detailed descriptions of the introduced functions occurring in Algorithm 6.

Algorithm 7 Get groups of concepts that are mapped to the same foreign concept

Require: mapping ontologies $\mathcal{O}_m$ and anchor concept C_a
Ensure: set of concepts that comprise G_{C_a}

```
1: procedure GETCONCEPTSGROUP(O_m, C_a)
2:     G_{C_a} ← ∅
3:     for μ ∈ O_m do
4:         if C_a ∈ Σ(μ) then
5:             G_{C_a} ← G_{C_a} ∪ (Σ(μ)\{C_a})
6:         end if
7:     end for
8:     return G_{C_a}
9: end procedure
```

The procedure GETCONCEPTSGROUP takes mapping ontology $\mathcal{O}_m$ and a potential anchor concept C_a as input. Formally a concept C_a is an anchor concept iff 1) $C_a \in \Sigma(\mathcal{O}_m)\backslash\Sigma(\mathcal{O}_l)$, 2) there are at least two mappings in $\mathcal{O}_m$ that are related to C_a. Detail of this process are presented in Algorithm 7. The group G_{C_a} is initiated as an empty set, i.e., $G_{C_a} = \emptyset$. Then for each mapping μ from the set $\mathcal{O}_m$ of all logical axioms in $\mathcal{O}_m$, it is then checked that whether the mapping μ involves the concept C_a, i.e., whether C_a is contained in the signature $\Sigma(\mu)$ of μ. If $C_a \in \Sigma(\mu)$, the rest of $\Sigma(\mu)$, i.e., $(\Sigma(\mu)\backslash\{C_a\})$, is added to G_{C_a}. After all

the mappings in $\mathcal{O}_m$ have been traversed, G_{C_a} contains all the concepts from $\mathcal{O}_l$ that are connected to C_a via $\mathcal{O}_m$. G_{C_a} is also the output of the function GETCONCEPTSGROUP.

Algorithm 6 is a straightforward and non-optimized procedure. It is based on the assumption that different granularity of local ontologies would lead to the scenarios that one concept in one ontology is mapped to a non-trivial set of concepts in another ontology. Here non-trivial sets refer to those sets that have more than one element. Due to the ad hoc nature of this proposed principle, it should first be verified on xEBR WG mappings before being applied to automated mappings.

3.2.3 Consistency

The consistency principle ensures that mappings do not make any concept unsatisfiable. For example, in local ontology $\mathcal{O}_1$ it holds that $B \sqsubseteq A$ and in $\mathcal{O}_2$ it is known that C is disjoint with D. There are two mappings, i.e., $A \equiv C$ and $B \sqsubseteq D$. Now it can be inferred that B is a subclass of both C and D. Because C and D are disjoint, B becomes unsatisfiable. In such cases, at least one mapping should be removed to resolve the incoherence. Let $\mathcal{O}_f$ and $\mathcal{O}_s$ be two local ontologies, and $\mathcal{O}_m$ be a mapping ontology between them.

Algorithm 8 Coherence check

Require: two local ontologies $\mathcal{O}_f$ and $\mathcal{O}_s$, corresponding mapping ontology $\mathcal{O}_m$
Ensure: $\mathcal{O}_f \cup \mathcal{O}_s \cup \mathcal{O}_m$ is coherent

```
1:  procedure COHERENCECHECK(O_f, O_s, O_m)
2:      O_u ← MERGEONTOLOGIES(O_f, O_s, O_m)
3:      O_fs ← MERGEONTOLOGIES(O_f, O_s)
4:      if ISCONSISTENT(O_u) then
5:          for C ∈ Σ(O_u) do
6:              if O_fs ⊭ (C ⊑ ⊥) && O_u ⊨ (C ⊑ ⊥) then
7:                  return FALSE
8:              end if
9:          end for
10:     else
11:         return FALSE
12:     end if
13:     return TRUE
14: end procedure
```

Algorithm 8 is a non-optimized procedure testing whether the merged ontology is coherent or not, i.e., whether it contains unsatisfiable concept. It takes as input two ontologies $\mathcal{O}_f$ and $\mathcal{O}_s$ and a corresponding mapping ontology $\mathcal{O}_m$. First, all three ontologies are merged into $\mathcal{O}_u$. Then $\mathcal{O}_f$ and $\mathcal{O}_s$ are merged into $\mathcal{O}_{fs}$. Next the function ISCONSISTENT determines whether $\mathcal{O}_u$ is consistent, i.e., whether there exists a model for it. This is very basic functionality of logical reasoners. If $\mathcal{O}_u$ is consistent, then for each concept C in the signature of $\mathcal{O}_u$, if it is satisfiable with respect to $\mathcal{O}_{fs}$ but becomes unsatisfiable with respect to $\mathcal{O}_u$, $\mathcal{O}_u$ is shown to contain unsatisfiable concept, therefore, is incoherent. Note it is also possible to resolve those unsatisfiable concepts found in $\mathcal{O}_u$ by computing their explanations or diagnoses. The computation of explanations or diagnoses would be the same as presented in Section 3.2.1.

As illustrated in the example above, consistency checking relies on disjointness relations. So it is desired that local ontologies contain disjointness relations. As defined in Chapter 2, SEOM can enrich the financial ontologies partly by adding disjointness axioms, so that the consistency principle is expected to help the mapping refinement process in this case and will be examined in Chapter 4 experimentally.

3.3 Extension & Discussion

There are several possible optimization techniques for both extended conservativity algorithms.

3.3.1 Compute a minimal set of violating axioms

The number of violating axioms can be very large, therefore reducing it to a minimal set can be useful. For example, the two financial ontologies considered in this work contain some 300 concepts altogether. And a set of mappings generated by COAL contains some 500 mappings. But the set of violating axioms amounts to several thousand. Thus it is interesting to check whether some violating axioms can be entailed by others, in which case these violating axioms would be resolved automatically, if their causing axioms were resolved.

Definition 14. (Minimal violating axiom set). *A minimal set of violating axioms,*

denoted as $\mathcal{S}^*_{vio}$, satisfies

$$\mathcal{S}^*_{vio} \subseteq \mathcal{S}_{vio},\ \textit{s.t.}\ \bigcup_{\nu \in \mathcal{S}^*_{vio}} \Delta_\nu = \bigcup_{\nu \in \mathcal{S}_{vio}} \Delta_\nu, \forall \mathcal{S}'_{vio} \subset \mathcal{S}^*_{vio}, \bigcup_{\nu \in \mathcal{S}'_{vio}} \Delta_\nu \neq \bigcup_{\nu \in \mathcal{S}_{vio}} \Delta_\nu. \tag{3.8}$$

Intuitively, the diagnosis of all violating axioms in $\mathcal{S}^*_{vio}$ should be the same as the diagnosis of all violating axioms in $\mathcal{S}_{vio}$. Though the minimal set of violating axioms $\mathcal{S}^*_{vio}$ is defined using diagnosis of violating axioms, the idea can also be applied when explanation is computed in Algorithm 2.

3.3.2 Resolve multiple violating axioms at a time

Instead of resolving one violating axiom at a time, a set of violating axioms can be considered together. This set can be of any fixed size, which can be determined heuristically. Ideally, this would speed up the process of resolving violating axioms. To be specific, a set S_{vio} of violating axioms, where $S_{vio} \subseteq \mathcal{S}_{vio}$, can be transformed into one axiom ν^* such that S_{vio} is entailed if and only if ν^* is entailed. For example, let $S_{vio} = \{\nu_a, \nu_b\}$, where $\nu_a = (A \sqsubseteq B)$ and $\nu_b = (C \sqsubseteq D)$. It is known that $A \sqsubseteq B \Leftrightarrow \top \sqsubseteq (\neg A \sqcup B)$. Following this proposition, ν^* is the conjunction of the new axioms, i.e.,

$$\nu^* = \top \sqsubseteq (\neg A \sqcup B) \sqcap (\neg C \sqcup D). \tag{3.9}$$

For an axiom of the form $G \equiv H$, it can first be transformed into a set of subsumptions like $\{G \sqsubseteq H, H \sqsubseteq G\}$. For $\nu \in S_{vio}$, we further have

$$\mathcal{O} \not\models \nu \Rightarrow \mathcal{O} \not\models S_{vio}, \tag{3.10}$$

where $\nu \in S_{vio}$. Any mappings that belong to the diagnosis Δ_ν of the violating axiom ν also belong to the diagnosis Δ_{ν^*}. Therefore, resolving multiple violating axioms by the techniques presented here is expected to be more effective in deleting unintended mappings. The size of S_{vio} is determined later by running the algorithms with different sizes. The result is reported in the evaluation section.

3.3.3 Replace entailment check with satisfiability check

As introduced in Section 3.2.1, it is possible to convert any logical axiom α to a concept C_{unsat} such that α is entailed if and only if C_{unsat} is unsatisfiable. This can also be applied to compound violating axiom constructed following the technique presented in the above section.

3.3.4 Alternative ordering metrics for mappings

So far, only confidence values of mappings are considered to sort the mappings in a specific order so that given two mappings with different confidence values the one with higher value is preferred. Meilicke et al. [44] proposed to calculate the *potential impact* of a mapping candidate. Though they devised the method in the context of distributed description logics (DDL), the method can also be adapted to the current work.

Definition 15. (Potential impact of a mapping). *The potential impact of a mapping $\mu = \langle id, C_f, C_s, \epsilon, \rho \rangle$ from $\mathcal{O}_f$ to $\mathcal{O}_s$ is defined as*

$$imp(\mathcal{O}_f, \mathcal{O}_s, \mu) \mapsto \begin{cases} sub(\mathcal{O}_f, C_f) \cdot (super(\mathcal{O}_s, C_s) + dis(\mathcal{O}_s, C_s)) & \text{if } \rho = \sqsubseteq \\ super(\mathcal{O}_f, C_f) \cdot (sub(\mathcal{O}_s, C_s) + dis(\mathcal{O}_s, C_s)) & \text{if } \rho = \sqsupseteq \\ imp(\mathcal{O}_f, \mathcal{O}_s, \mu_{\sqsubseteq}) + imp(\mathcal{O}_f, \mathcal{O}_s, \mu_{\sqsupseteq}) & \text{if } \rho = \equiv \end{cases} \tag{3.11}$$

where $sub(\mathcal{O}_f, C_f)$ returns the number of all subclass of concept C_f in $\mathcal{O}_f$, $super(\mathcal{O}_s, C_s)$ returns the number of all super-class of concept C_s in $\mathcal{O}_s$, and $dis(\mathcal{O}_s, C_s)$ returns the number of all classes that are disjoint with C_s in $\mathcal{O}_s$; $\mu_{\sqsubseteq}$ refers to a mapping μ where its relation is $\sqsubseteq$, analogously for $\mu_{\sqsupseteq}$.

3.4 Summary

In this chapter we extend existing mapping refinement techniques by using a reasoner as an oracle. This includes several principles that constitute a logic-based mapping refinement (LOMR) mechanism. One principle, the grouping principle, is proposed based on observations of the ontologies in the current use case. An extended discussion about the conservativity principle is presented to consider subsumption mappings. The consistency and locality principle are also adopted from existing work and implemented in the current implementation. A number of optimization techniques are also presented. By integrating the conservativity, consistency and locality principles, together with a set of selected optimization techniques, a complete mapping refinement mechanism, LOMR, is developed in order to refine automated mapping candidates. The above optimization techniques are applicable to both Algorithm 2 and Algorithm 5. These techniques include resolving multiple violating axioms at a time and replacing an entailment check with a satisfiability check.

Chapter 4

Experiment & Evaluation

In this chapter we describe the experiments conducted and the results obtained. It has the following parts: Section 4.1 describes the datasets prepared and used in the experiments; Section 4.2 introduces the metrics used in the evaluation; Section 4.3 describes the implementation and the experiments; Section 4.4 presents the evaluation results of both SEOM and LOMR; Section 4.5 summarizes the chapter.

4.1 Datasets

In the experiments we consider two datasets: financial dataset and conference dataset. The financial dataset contains financial ontologies we created based on XBRL taxonomies, a set of gold standard mappings between a pair of financial ontologies and mapping suggestions from a couple of matchers. Conference dataset contains a number of ontologies from the CONFERENCE track [45] of OAEI 2010, reference mappings between them and mapping suggestions generated by some matchers. In the following sections, we will describe the two datasets in detail.

4.1.1 Financial dataset

We created a series of financial ontologies based on French, Spanish and xEBR taxonomies[1]. Note that we only focus on concepts occurring in the balance sheet, which are determined by different role references in different taxonomies (shown

[1]More details about the French and Spanish taxonomies are in Section 2.3. For xEBR taxonomy, confer Section 3.2.2.

Table 4.1: Role references of taxonomies used in the experiments

Taxonomy	Role reference
xEBR	`Assets`
	`EquityLiabilities`
TCA	`ca:BilanActifDeveloppe`
	`ca:BilanPassifAvantRepartitionDeveloppe`
PGC07	`pgc07:BalanceSituacion`

in Table 4.1). Financial ontologies constructed, and later used in the experiments, are presented in Table 4.2. $\mathcal{O}_x$, $\mathcal{O}_f$ and $\mathcal{O}_s$ are balance sheet ontologies for xEBR, French and Spanish taxonomies respectively. $\mathcal{O}_b$ refers to the BAO that we constructed using shared properties underlying various taxonomies. $\mathcal{O}_{fa}$ and $\mathcal{O}_{sa}$ are two ontologies, which contain only asset concepts, while $\mathcal{O}_{fae}$ and $\mathcal{O}_{sae}$ are the enriched ontologies we created in the process of SEOM. As a reminder, these two enriched ontologies only contain concepts relating to assets. $\mathcal{O}_{fm}$ and $\mathcal{O}_{sm}$ are minimal ontologies containing only concepts relating to current assets.

Gold Standard Mappings. We consulted one French accounting expert to manually create mappings between $\mathcal{O}_f$ and $\mathcal{O}_s$. To help the expert, we applied heuristics to manual mappings from the xEBR ontology to each of the two ontologies in order to get an initial set of mappings. Then we presented these mappings via a tailored web application[2] to the expert so that he could modify or add to this initial set of mappings, until the gold standard mappings were properly identified. As the domain expert pointed out, there are lots of difficulties and subtleties in matching financial concepts from different countries. In effect, the expert gave many equations comparing concepts, some using equality ($=$), others using less than ($<$) or greater than ($>$). We have to be very careful in interpreting these formulas in order to truthfully reflect relations among different financial concepts. The set of gold standard mappings, denoted as $\mathcal{M}_{gs}$, is presented in Table 4.3. More details about interpreting $\mathcal{M}_{gs}$ and relevant discussions can be found in Appendix A.

Mapping suggestions for the financial ontologies. We use SEOM and COAL to generate mapping suggestions for the financial ontologies. SEOM generates 93 mappings for the enriched ontologies $\mathcal{O}_{fae}$ and $\mathcal{O}_{sae}$. COAL is able to gener-

[2] See a screen shot of the web application in Figure A.1 in Appendix A.

Table 4.2: Financial ontologies used in the experiments

Ontology	# Concepts	# Objectprop.	# Dataprop.	DL
$\mathcal{O}_x$	66	0	0	$\mathcal{AL}$
$\mathcal{O}_f$	179	0	0	$\mathcal{AL}$
$\mathcal{O}_s$	138	0	0	$\mathcal{AL}$
$\mathcal{O}_{fa}$	138	0	0	$\mathcal{AL}$
$\mathcal{O}_{sa}$	74	0	0	$\mathcal{AL}$
$\mathcal{O}_b$	189	36	5	$\mathcal{ALCHQ}(\mathbf{D})$
$\mathcal{O}_{fae}$	328	36	5	$\mathcal{ALCHQ}(\mathbf{D})$
$\mathcal{O}_{sae}$	264	36	5	$\mathcal{ALCHQ}(\mathbf{D})$
$\mathcal{O}_{fm}$	19	0	0	$\mathcal{AL}$
$\mathcal{O}_{sm}$	42	0	0	$\mathcal{AL}$

Table 4.3: Statistics of gold standard mappings. The 32 simple subsumptions consist of 13 *narrowMatch* mappings and 19 *broadMatch* mappings.

	simple		complex		Total
	Asset	Other	Asset	Other	
Subsumption	32(**13+19**)	25	7	4	68
Equivalence	**14**	13	13	13	53
Total	**46**	38	20	17	
	84		37		121

Table 4.4: Different sets of mappings for top-n tests

# COAL mapping suggestions	top 1	top 5	top 10
(a) $(\mathcal{O}_{fm}, \mathcal{O}_{sm})$	30	123	217
(b) $(\mathcal{O}_{f}, \mathcal{O}_{s})$	537	2685	5370

ate different numbers of mapping suggestions. For example, if we consider the top ten mapping suggestions for a single concept, the total number of mapping suggestions is ten times more than when we only consider the top one mapping suggestion. Different top-n COAL mappings are presented in Table 4.4.

4.1.2 Conference dataset

The CONFERENCE track consists of a collection of 15 ontologies all describing the domain of conference organization. There are reference mappings only among 7 out of the 15 ontologies, which will be the subject of our experiments for the following reasons: 1) they have disjointness relations, which is crucial for the consistency principle to be effective; 2) there are reference mappings for these ontologies; 3) we are able to compare LOMR against a recent mapping refinement system ALCOMO, which was also evaluated on this dataset. Hereafter we refer to these 7 ontologies as conference ontologies. Conference ontologies together with mapping suggestions among them are called the conference dataset. More details about these 7 ontologies are given in Table 4.5. For conference ontologies in OAEI 2010 campaign, there are in total 14 matchers[3] that submitted their mapping suggestions, which are shown in Table B.1 in Appendix B.

4.2 Metrics

In the following sections, evaluation metrics used in this thesis are presented. Apart from the well-known precision, recall and F measure, a new metric is introduced to measure the benefit gained from the mapping refinement process.

[3] A detailed description of the matchers can be found in Meilicke's PhD thesis [11].

Table 4.5: A number of ontologies from the CONFERENCE track and the reference mappings among them

# Reference mappings	ekaw	Conference	sigkdd	iasted	ConfOf	cmt	edas	# Concepts	DL
ekaw	×	25	15	19	20	11	23	77	$\mathcal{SHIN}$
Conference		×	15	14	15	15	17	60	$\mathcal{ALCHIF}(\mathcal{D})$
sigkdd			×	15	7	12	15	49	$\mathcal{ALEI}(\mathcal{D})$
iasted				×	9	4	19	140	$\mathcal{ALCIN}(\mathcal{D})$
ConfOf					×	16	19	38	$\mathcal{SIN}(\mathcal{D})$
cmt						×	13	36	$\mathcal{ALCIN}(\mathcal{D})$
edas							×	104	$\mathcal{ALCOIN}(\mathcal{D})$

4.2.1 Precision, Recall and F measure

Precision, Recall and F measure are well-known metrics, originating from Information Retrieval (IR), which require a gold standard. In this thesis, gold standard mappings $\mathcal{M}_{gs}$ are given by a domain expert, as explained in Section 4.1.1. A set of mapping suggestions $\mathcal{M}$ can be evaluated following the *confusion matrix*, where $truepositive$ refers to the number of mappings that are in both $\mathcal{M}_{gs}$ and $\mathcal{M}$; $falsenegative$ refers to the number of mappings that are in $\mathcal{M}_{gs}$, but not in $\mathcal{M}$; $falsepositive$ refers to the number of mappings that are in $\mathcal{M}$, but not in $\mathcal{M}_{gs}$. Thus, precision (p), recall (r) and F measure (f) are defined as follows.

Table 4.6: Confusion matrix for mapping evaluation

	$\mathcal{M}_{gs}$	
$\mathcal{M}$	$truepositive$	$falsepositive$
	$falsenegative$	

$$p := \frac{truepositive}{truepositive + falsepositive} \tag{4.1a}$$

$$r := \frac{truepositive}{truepositive + falsenegative} \tag{4.1b}$$

$$f := \frac{2 \times p \times r}{p + r} \tag{4.1c}$$

4.2.2 Human Effort Saved

Another metric reflects the amount of human effort saved by mapping refinement systems from the perspective of an end user who checks all the mapping suggestions in order to find correct ones. Specifically, the metric, *human effort saved*, denoted as hes, measures the percentage of incorrect mappings removed with respect to the total number of incorrect mappings before the refinement process. In Table 4.6, let $\mathcal{M}$ be the mapping suggestions before the refinement process and $\mathcal{M}'$ be the refined mappings. And $falsepositive'$ denotes the number of mappings that are in $\mathcal{M}'$, but not in $\mathcal{M}_{gs}$, which is exactly the number of incorrect mappings

after the refinement process. The human effort saved can then be calculated as

$$hes := \frac{falsepositive - falsepositive'}{falsepositive}. \tag{4.2}$$

This is a rather intuitive metric to reflect the usefulness of the refinement process. Considering the case that $falsepositive$ is non-zero, if $hes = 1$, it means all incorrect mappings are removed by the refinement process, i.e., $falsepositive' = 0$; if $hes = 0$, it means none of the incorrect mappings has been removed by the refinement process and no human effort is saved. The hes metric is analogous to a metric, *relative effort reduction* [46], which has been found to be useful in the domain of knowledge revision.

4.3 Implementation

We have implemented a semi-automatic mapping generation mechanism (SEOM) and a fully automated mapping refinement mechanism (LOMR). Next, we introduce the tools and libraries used, followed by descriptions of both mechanisms.

A number of logic reasoners are used in this thesis. Next, short descriptions of these reasoners are given, focusing on those aspects that are important to this work. HermiT reasoner is an OWL 2 DL reasoner based on hyper-tableau calculus. Given an ontology, it can perform consistency checking, classification and entailment checking. Pellet reasoner is an OWL 2 reasoner providing many well-advanced logical reasoning services, among which incremental reasoning contributes considerably to our algorithms.

Among the logical principles presented in Chapter 3, the grouping principle, proposed based on observations of different granularity between shared and national financial ontologies, is first to be verified before being used to refine mapping suggestions. A combination of the conservativity, consistency and locality principles is implemented as a complete logic-based mapping refinement (LOMR) procedure. Given two ontologies and a set of mapping suggestions, LOMR first checks whether there is incoherence (unsatisfiable concept). Then it resolves incoherence by removing mapping suggestions until no incoherence exists. The next step is to check if there is any violation of the conservativity principle. Again to resolve these violations, LOMR removes mapping suggestions following the algorithms and heuristics presented in Section 3.2 of Chapter 3. The locality principle is used to compute confidence values, in case mapping suggestions do not

have corresponding confidence values. After this whole mapping refinement process, the output is a set of refined mappings that are consistent with the two local ontologies, and cause no violations of the conservativity principle is returned. To evaluate this refined set of mappings, they are compared against reference mappings.

4.4 Evaluation

To show the effectiveness of SEOM, we compare SEOM against COAL on the basis of $\mathcal{O}_{fe}$ and $\mathcal{O}_{se}$. As for our mapping refinement mechanism, we first conducted experiments to determine a set of optimal heuristics to be included in LOMR. After obtaining an optimal LOMR, we show its usefulness by refining both SEOM and COAL mappings for the financial ontologies. Then we compare LOMR against another mapping refinement system on the conference dataset.

4.4.1 SEOM

Concept definition based on the BAO can be used as a mapping generation mechanism. With each concept in the local taxonomies properly defined, a pair of concepts from two different taxonomies can be matched by using reasoners, if they have similar definitions. There are in total 46 mappings in $\mathcal{M}_{gs}$ (subsumption together with equivalence) that relate exclusively to asset concepts, as shown in Table 4.3. In order to distinguish *narrowMatch*es from *broadMatch*es, we require that mappings are directed and always go from a French concept to a Spanish one. For the purpose of reasoning, we converted the *exactMatch*es into equivalence relations, the *narrowMatch*es and *broadMatch*es into subclass and superclass relations respectively.

Table 4.7 compares SEOM with COAL. For each concept in the French balance sheet, we use COAL to get the top *exactMatch* concept from the Spanish balance sheet. We do the same for *narrowMatch* and *broadMatch*. Then we eliminate all the mappings that do not involve asset concepts. This results in 352 mappings, in contrast to 93 mappings generated by SEOM. As can be seen in Table 4.7, the logic-based approach produces much better results in terms of precision and recall for all 3 kinds of mappings. For example, out of 14 *exactMatch*es from the gold standard mappings, the logic-based approach finds 12, whereas COAL only finds 3. Note that we also used LogMap to match the datasets and obtained no mappings relating exclusively to asset concepts. The results therefore are omitted in

Table 4.7: Comparison of alignment from COAL and SEOM

Recall	*exactMatch*	*narrowMatch*	*broadMatch*	Overall Recall	Precision
COAL	3/14	5/13	3/19	$23.9\%(11/46)$	$3.1\%(11/352)$
SEOM	12/14	9/13	13/19	$73.9\%(34/46)$	$36.6\%(34/93)$

Table 4.7. SEOM achieves 73.9% recall and 36.6% precision, as compared with 23.9% recall and 3.1% precision by COAL.

Out of the 93 mapping suggestions from SEOM, there are 59 redundant mappings. Recall is not 100% mainly due to two sources of difficulty. One is mappings involving 'Other' concepts, which are catchalls for anything that does not fall into another sibling category. The other source of difficulty is divergent categorization, e.g., the Spanish taxonomy includes prepayments to suppliers in the inventory category, but the French does not. It is also worth noting that the logic-based approach detected inconsistencies which we resolved by deleting what we considered to be incorrect mappings. The incorrectness of these mappings was later confirmed by the domain expert. The limitation of SEOM lies in that 1) the BAO is to be extended, 2) human intervention is needed in both extending the BAO and defining other financial concepts using it.

4.4.2 Mapping Refinement System

In this section we present results of LOMR. Before that we first investigate different configurations of LOMR.

Grouping Principle

First, we need to verify this principle before building it into the whole logical refinement system. For this verification, we consider $\mathcal{O}_x$, $\mathcal{O}_f$,$\mathcal{O}_s$ and the xEBR WG mappings among them. Given $\mathcal{O}_f$ and $\mathcal{O}_x$ and the manual mappings $\mathcal{M}_{fx}$, we first find those pivot concepts from $\mathcal{O}_x$, i.e., each concept which is mapped to more than one concept in $\mathcal{O}_f$. Then for each pivot concept C_p, we can find a group of concepts from $\mathcal{O}_f$ that are connected to C_p in $\mathcal{M}_{fx}$. Afterwards, we check whether there is a super-concept of the rest in each group.

The experimental results can be seen in Table 4.8. For $\mathcal{O}_f$, 11 out of 18 groups follow the grouping principle. This indicates that the grouping principle reflects the internal relations among the French concepts in a group. For $\mathcal{O}_s$, however, there are 5 groups found, but there is only 1 group of concepts following the

Table 4.8: Experimental results of the grouping principle

Taxonomy	# Groups of local concepts	# Groups following the grouping principle
TCAG	18	11
PGC07	5	1

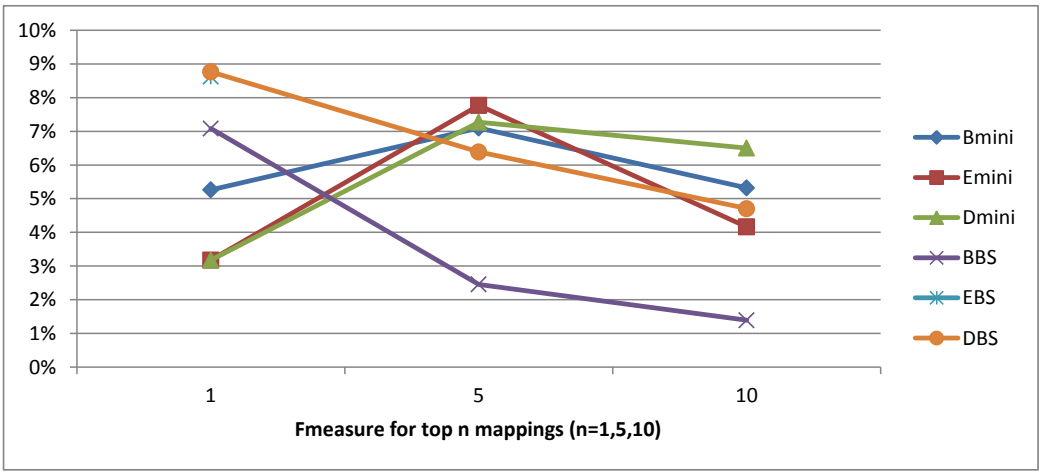


Figure 4.1: Effect of the refinement process

grouping principle. The grouping principle therefore reflects the internal relation of concepts in groups for $\mathcal{O}_f$, and is able to propose new mappings. Though there are granularity differences between xEBR taxonomy and local taxonomies, this principle cannot tell whether a certain mapping is good or not. Therefore, this principle is not included in the final mapping refinement mechanism. On the other hand, the principle reveals that there are possible good mappings between two concepts consisting of similar sets of sub-concepts (See Appendix C for an example).

Extended Conservativity

Here we describe experiments concerning the extended conservativity principle. In Figure 4.1, the F measure f of top-n mappings, before and after the refinement process, are presented. The horizontal axis refers to different top-n mappings generated by COAL, as shown in Table 4.4. Both the explanation approach and the diagnosis approach are applied to minimal ontology pair (mini) and balance sheet

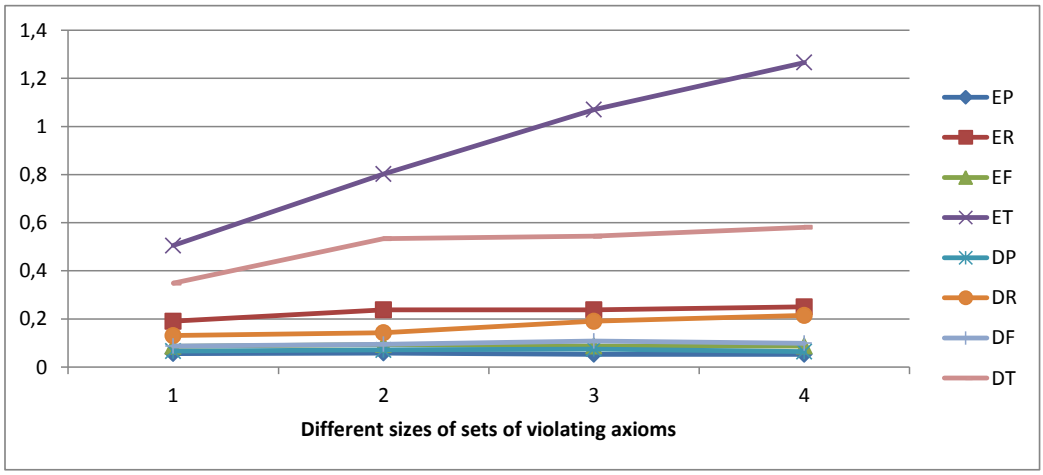


Figure 4.2: Comparative analysis of the explanation and diagnosis approaches

ontology pair (BS). The curves "Bmini" and "BBS" refer to baseline F measures, i.e., F measures of the mapping suggestions before the refinement processes. Prefix "E" of curves refers to the explanation approach, while prefix "D" to the diagnosis approach.

First, it can be observed that F measures are, in general, very low, due to the quality of mapping candidates automatically generated by COAL. For $\mathcal{O}_{fm}$ and $\mathcal{O}_{sm}$, the curve "Bmini" gives the baseline f before the refinement process. The diagnosis approach achieved better f than baseline f two cases out of three, while the explanation approach gave better results in only one case. For the balance sheet ontologies $\mathcal{O}_f$ and $\mathcal{O}_s$, both the explanation and diagnosis approaches give better results than the baseline. Note that the explanation approach failed to give results within a given timeout in two cases. Timeout is set as 60 minutes for all experiments in this thesis. On the basis of these results, it can be concluded that both approaches, especially the diagnosis approach, give better F measure f than the baseline in the case of financial ontologies. This is to say, after the refinement process, the refined mappings are of better quality compared to the original mappings.

For both approaches, there is only a moderate change in f. One reason is that in the refinement process a small portion of correct mappings are also removed. An increase of p can be observed in general, while at the same time r decreases, thus resulting in no big change in f.

Explanation vs. Diagnosis. Two approaches have been proposed to resolve inco-

herence and violations introduced by mappings. A comparative analysis of these two approaches is in Figure 4.2. In the figure, the horizontal axis indicate different sizes of sets of violating axioms that are to be resolved at the same time. Curves starting with "E" refer to the explanation approach; while "D" indicates the diagnosis approach. Suffix "P" refers to p, "R" to r, "F" to f and "T" refers to runtime (minutes) of the refinement procedure. Thus, "EP" refers to the precision of refined mappings through the explanation approach; others can be interpreted analogously.

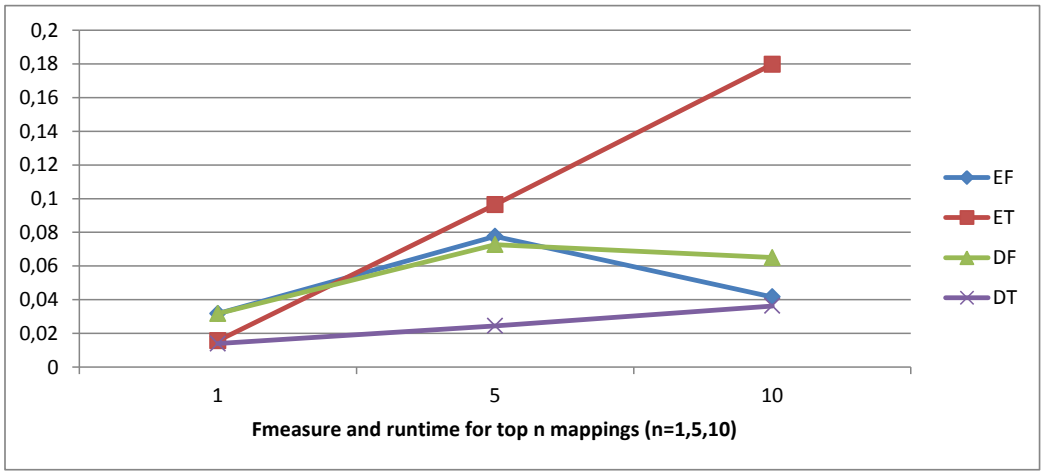
The following observations can be drawn from the experimental results. In general the explanation approach took more time than the diagnosis approach. Indeed, the explanation approach took much more time than the diagnosis approach when more mapping candidates are to be refined, which is shown by difference in the slopes of curves "ET" and "DT". Further, the diagnosis approach achieves better results in terms of f. Thus the diagnosis approach is built into the LOMR system.

Resolving multiple violating axioms at a time. In Section 3.3, it is suggested that the procedure would speed up by resolving multiple violating axioms at a time because mappings that cause violations would be more quickly discovered. The experimental results concerning this hypothesis is shown in Figure 4.2. It shows that it takes more time to resolve a larger size of violating axioms. It is also interesting to note that r has a slight increase, while f decreases moderately. This is true for both explanation and diagnosis approaches. Given these results, the heuristic of resolving multiple violating axioms at a time does not provide a gain that outweighs the loss it causes. Therefore, this heuristic is not included in LOMR, and all other results in this thesis are concerned with resolving only one violating axiom at a time.

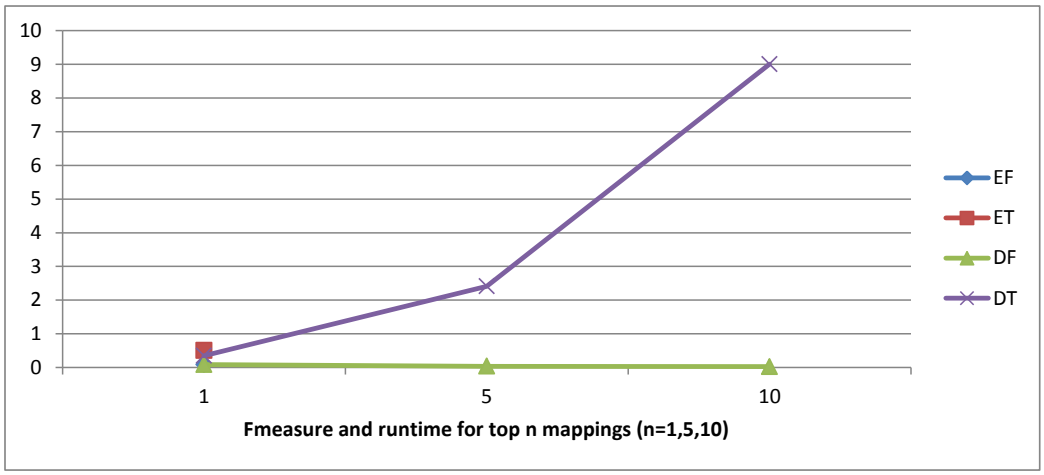
LOMR

With the help of the experimental results presented above, an optimal mapping refinement system (LOMR) constituting the extended conservativity, consistency and locality principles, together the diagnosis approach, resolving single violating axiom at a time, is implemented and evaluated in the following sections.

Scalability. A scalability test is also conducted to reveal the capacity of the presented algorithms in dealing with different numbers of mapping suggestions. In Figure 4.3, the naming of curves follows that of Figure 4.2, with the exception that only f and runtime are presented here. The horizontal axis refers to different top-n mappings to be refined, as detailed in Table 4.4.



(a) Mini test



(b) Balance sheet ontologies

Figure 4.3: Scalability test on different sets of mappings

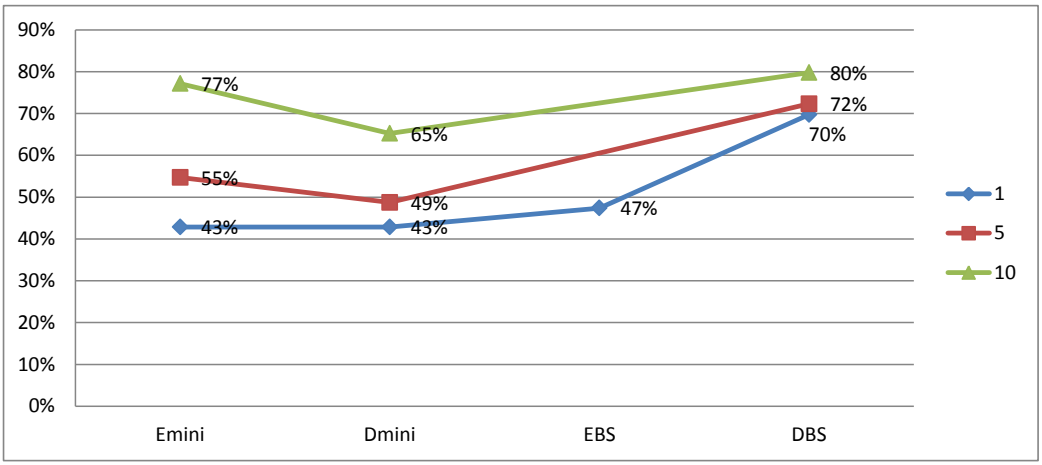


Figure 4.4: Human effort saved on different mapping sets

It can be seen from Subfigure 4.3a that the explanation approach does not scale as well as the diagnosis approach. This is confirmed by Subfigure 4.3b, where with even more mappings, the explanation approach failed to give any result within a given timeout. The diagnosis approach can finish refining some 5,000 mappings in less than 10 minutes. On the other hand, it can also be observed that with more mappings to refine, the time the diagnosis approach takes increases rapidly.

Human Effort Saved. As introduced in Section 4.2.2, hes reflects the usefulness of a refinement mechanism from the viewpoint of end users. Figure 4.4 presents the experimental results on two different approaches on two different datasets with different top-n mappings. The vertical axis is the percentage of hes. It can be seen that given the datasets, LOMR reduces human effort between 43% and 80%.

Refinement with Enriched Ontologies. In this test, enriched ontologies, i.e., $\mathcal{O}_{fae}$ and $\mathcal{O}_{sae}$, are used for the mapping refinement process. The enriched ontologies contain disjointness relations, which are vital for the consistency principle to be effective. Therefore, it can be expected that with $\mathcal{O}_{fae}$ and $\mathcal{O}_{sae}$, LOMR is better enabled to detect incoherence caused by the mapping suggestions. Table 4.9 presents the refinement results using enriched ontologies and those simple ontologies with only class hierarchies, i.e., $\mathcal{O}_{fa}$ and $\mathcal{O}_{sa}$. Since enriched ontologies are particularly relevant for the consistency principle, the refinement procedure with consistency check (rows with "C", as opposed to without "C") is compared against one without consistency check (rows without "C"). Again we conducted the test with different top-n mapping suggestions. The statistics presented in Table 4.9 take the following pattern. We collected the F measures of the refined mappings,

Table 4.9: Refining different sets of mappings with enriched ontologies

(f(%), t(min))	1	5	10
$(\mathcal{O}_{fa}, \mathcal{O}_{sa})$	(3.2, 0)	(5.1, 0)	(4.6, 0)
$(\mathcal{O}_{fa}, \mathcal{O}_{sa})$,C	(3.2, 0)	(5.1, 0)	(4.6, 0.1)
$(\mathcal{O}_{fae}, \mathcal{O}_{sae})$	(6.3, 0.3)	(8.6, 2.3)	(10.1, 4.0)
$(\mathcal{O}_{fae}, \mathcal{O}_{sae})$,C	(6.2, 0.4)	(8.6, 1.6)	(12.4, 3.7)

and the time t the procedure takes for each test, and then put them into a pair. Take for example the pair (6.3,0.3) in the second column and fourth row. It means that with $\mathcal{O}_{fae}$ and $\mathcal{O}_{sae}$ LOMR takes 0.3 minute to refine 537 mapping suggestions from COAL and the F measure of the refined mappings is 6.3%.

It can be seen that enriched ontologies help increase the f of the refined mappings. The consistency check also helps improve the quality of the refined mappings. On the other hand, LOMR takes much more time while considering enriched ontologies. The reason is that in the enriched ontologies there are concepts from the BAO so that more violations of logic principles are detected, and have to be resolved.

Refining logic-based mappings. It is also interesting to refine the mappings generated by SEOM. As mentioned in Chapter 2, all mappings generated by SEOM are consistent with the ontologies involved. The question would then be whether the mappings would violate the conservativity principle or not. LOMR is used to refine the mappings from SEOM. The results support the previous hypothesis that all the good mappings are preserved and no violation is found. This experimental result shows that mappings proposed by SEOM, as expected, do not violate any of LOMR's principles.

LOMR vs. ALCOMO

LOMR is compared against other existing mapping refinement systems, in order that a fair conclusion can be drawn for LOMR.

ALCOMO is a recent logic-based mapping refinement system designed and implemented by Meilicke [11]. A series of experiments have been conducted to compare LOMR against ALCOMO. We conducted experiments on seven pairs of conference ontologies, where the results of three representative pairs are pre-

sented in Figure 4.5. Specifically, Subfigure 4.5a concerns two ontologies `cmt` and `ekaw`, whereas Subfigure 4.5b concerns ontologies `cmt` and `Conference`, and Subfigure 4.5c concerns ontologies `Conference` and `ekaw`. The vertical axes indicate f of refined mappings from different matchers. Each curve in each figure corresponds to one matcher. For relevant descriptions of each ontology matcher, readers are refered to the PhD thesis of Meilicke [11]. We compare f of refined mappings from LOMR and ALCOMO against the f of unrefined mappings (Baseline).

The following observations can be made. In most cases, LOMR gives comparable, sometimes even better results, than ALCOMO does. For example, LOMR always gave better results when refining mapping suggestions produced by LILY for the three representative pairs of conference ontologies. Specifically, when refining mapping suggestions from LILY for ontologies `cmt` and `ekaw`, LOMR achieved higher F measure than ALCOMO by 7.7%. More detailed comparison of LOMR and ALCOMO is presented in Appendix D.

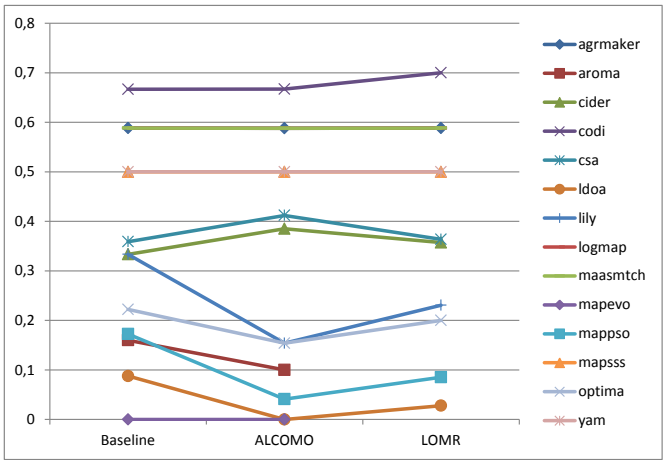
Additionally, with a few exceptions, the quality of mappings, measured by f, has been improved slightly after the mapping refinement process. This shows that LOMR is also applicable to datasets other than the financial dataset only.

4.5 Summary

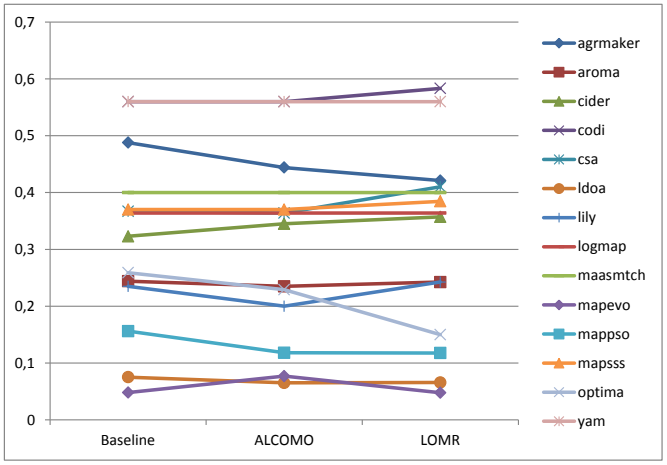
This chapter mainly presents experiments and evaluation results concerning the mapping generation mechanism by semantic enrichment presented in Chapter 2 and the logic-based mapping refinement mechanism presented in Chapter 3. All these address the research questions **R2** and **R5**.

Section 4.1 presents all relevant information of the datasets used in this thesis: financial dataset and conference dataset. While the conference dataset stems from benchmarking efforts, we created the financial dataset by constructing financial ontologies based on XBRL taxonomies, building the gold standard mappings between an ontology pair and generating mappings suggestions using SEOM and COAL. Section 4.2 presents a number of different metrics for measuring the quality and the gain of refined mappings. Among existing metrics like Precision p, Recall r and F measure f, a new metric, human effort saved hes, is introduced. In Section 4.3 some general information about the implementation is presented, including the libraries, reasoning tools and descriptions of both mechanisms.

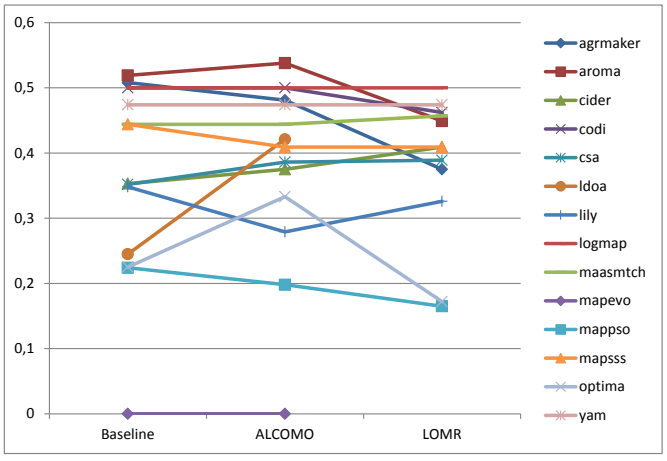
Section 4.4.1 shows the advantage of the presented approach against another mapping generation tool, in terms of precision and recall of generated mappings.



(a) Ontologies `cmt` and `ekaw`



(b) Ontologies `cmt` and `Conference`



(c) Ontologies `Conference` and `ekaw`

Figure 4.5: Refining mappings submitted for CONFERENCE track to OAEI 2010

SEOM gives better results than a system based on heuristics and machine learning techniques. Section 4.4.2 starts with verification results concerning the grouping principle. The principle is shown to be inadequate to detect incorrect mappings, therefore is excluded from LOMR. In the same section, results are presented for LOMR, consisting of the extended conservativity (using the diagnosis approach), consistency and locality principles and heuristics like resolving only one violating axiom at a time. LOMR is shown to improve the quality of mapping suggestions by increases of f. hes is also computed in different settings, where there is up to 80% human effort saved when considering the financial dataset. Moreover, it is shown that our system LOMR gives comparative, sometimes even better, results than ALCOMO does. With a few exceptions, the quality of mappings, measured by f, has been improved slightly after the mapping refinement process. Also there have been, on average, around 20% human effort reduction.

With all the experimental results summarized above, the questions of what mechanism will enable the exploitation of background knowledge (**R2**) and how the logic-based principles perform (**R5**) are addressed with a mapping generation mechanism based on semantic enrichment (SEOM) and a logic-based mapping refinement mechanism (LOMR) respectively. Both mechanisms give comparable, sometimes better, results than existing ones.

Chapter 5

Conclusion

In this thesis, we have investigated the potential contribution of logic-based reasoning to ontology matching. Specifically, we start out with five research questions, concerning two different aspects of the problem: exploiting background knowledge and refining mappings. The first two research questions concern exploiting background knowledge. To answer the question of what kind of background knowledge will help generate better mappings (**R1**), we identify and formalize a set of underlying concepts and properties shared by financial reporting schemas of different origins. These commonly shared concepts and properties constitute a Basic Accounting Ontology, which serves as the foundational semantics to define different financial concepts. In response to the question of what mechanism will enable the exploitation of background knowledge (**R2**), we define other financial concepts on the basis of this shared ontology, and use reasoners to determine mappings between them. The remaining three research questions concern mapping refinement. In response to the question of why the refinement process is necessary for automatically generated mappings (**R3**), we motivate the necessity and importance of mapping refinement by presenting theoretical arguments and empirical evidence. It is argued that while correct mappings are always coherent, incorrect mappings always lead to unintended consequences, which could be detected by logical reasoning. Moreover, it is shown that there are many incorrect mappings in both manual mappings and automatically generated mappings, which might be removed by mapping refinement. To answer the question of what the logic-based principles for mapping refinement (**R4**) are, we present a number of logic principles and several heuristics for mapping refinement. We answer the question of how the logic-based principles perform (**R5**) by combining a set of selected principles and heuristics into a logic-based mapping refinement

system, and evaluating it.. The following sections highlight the contributions of this work and give a number of pointers for possible future work.

5.1 Contribution

The contribution of this thesis work is two fold, i.e., a logic-based mapping generation mechanism (SEOM) and a logic-based mapping refinement mechanism (LOMR).

In the work on the logic-based mapping generation mechanism, the following results have been achieved.

- The Basic Accounting Ontology. The BAO is designed to lay a common ground for various financial reporting schemas by identifying and then formalizing underlying semantics as basic concepts and properties in OWL syntax. By devising this shared ontology, other financial ontologies can be aligned on this common ground.

- Logic-based mapping generation procedure. A three-phase procedure to assist the alignment of financial taxonomies is presented. The procedure starts with taxonomy conversion into ontologies. The next step is to define all the financial concepts using basic concepts and properties from the BAO. Then reasoners are used to infer mappings from the merge of the enriched ontologies, so that financial concepts with the same definition (definition based on the BAO) are given as good matches. All the matches generated in this way are logically consistent with the ontologies.

This mechanism is tested on French and Spanish financial reporting schemas and yields better results than existing ontology matching systems based on heuristics and machine learning techniques.

In the work on the logic-based mapping refinement mechanism, the following has been achieved.

- Extension of the conservativity principle. The conservativity principle was first proposed and implemented by Jiménez-Ruiz et al. [12] in a simplified fashion. Their simplification is very restrictive in terms of mapping relations and violation patterns. In this thesis an extended procedure is devised and implemented, where mappings in the form of both equivalence and subsumption are considered, and a generic pattern covers all possible kinds of violations of the conservativity principle.

- Comparative analysis of explanation and diagnosis approaches. To resolve violation of principles or incoherence caused by unsatisfiable concepts, there are two approaches: computing explanations and computing diagnoses. Two corresponding procedures are devised and evaluated in order to determine which approach is superior. As shown in Chapter 4, the diagnosis approach is better than the explanation approach in general, especially in terms of scalability. Thus, the diagnosis approach is built into the final mapping refinement procedure.

- Complete mapping refinement procedure. In addition to the extended conservativity principle, the consistency principle and the locality principle are also built into the mapping refinement procedure. Moreover, a number of heuristics are proposed and tested. Notably the idea of resolving multiple violating axioms at a time is shown to offer no real advantage. Combining all these principles and some heuristics a complete procedure is built and is shown to produce results comparable with the state of the art. The mapping refinement procedure is also shown to be useful from the end user's point of view, in that it reduces incorrect mappings.

The mechanism is also shown to improve the quality of mapping suggestions in terms of F-measure when compared with the gold standard mappings.

5.2 Future Work

A number of interesting questions have yet to be addressed, which are rightfully good pointers for future work.

SEOM for now depends largely on human intervention. Especially, the BAO covers only a limited number of underlying basic concepts and properties. It is expected to be extended with more concepts and properties formalizing the underlying semantics of the knowledge domain. For this approach to be scalable and efficient, automation of the following steps are of great importance. First, there can be a semi-automated approach to extending the BAO. The expected tools can make use of natural language processing (NLP) and machine learning (ML) techniques in order to identify and formalize new components of underlying semantics in the financial reporting domain. Second, the process of defining financial concepts using the BAO can also be semi-automated by establishing the correspondence of the BAO with other financial ontologies. Third, as mentioned above, the logic-based approach generates redundant mappings, which can be minimized. In

other words, it is interesting to compute a minimal set of mappings. Another interesting research challenge is to let matchers generate complex mappings. As we have seen in the gold standard mappings, a complex mapping is a logical axiom that relates more than two concepts, e.g., $\langle A \equiv (B \sqcup C), \epsilon \rangle$. All of the systems compared in Shvaiko and Euzenat [4] are limited to computing simple mappings. SEOM currently shares this limitation, but has the potential to generate complex mappings, because of its semantics-oriented nature.

The performance of the mapping refinement mechanism LOMR is influenced by ontology matching systems by means of mapping suggestions and corresponding confidence values. Mapping generation tools, to a large extent, determine the recall of mapping suggestions. Therefore, the mapping refinement mechanism improves the quality of mapping suggestions mainly by removing incorrect mappings, while preserving the correct ones. It is interesting to investigate how best to avoid undesired deletion, i.e., deletion of correct mappings, which is unavoidable, if confidence values are misleading. Another possible improvement of this work concerns the types of reasoning techniques that are used in our algorithms. We have used reasoners as a black-box, querying consistency and entailments without looking into the internal process of reasoning. As Stuckenschmidt [47] has concluded that black-box approaches suffer from their computational complexity, moving to a white-box approach, i.e., tracking and analyzing the internal reasoning process, could offer significant speed-up of the algorithms.

Appendix A

Gold standard mappings $\mathcal{M}_{gs}$

To obtain gold standard mappings for French and Spanish ontologies, a domain expert is consulted to conduct manual matching of concepts in these two ontologies. In order to support this manual process, an initial set of mappings are computed by applying heuristics to manual mappings from xEBR ontology to each of the two ontologies. The idea is that the domain expert could start modifying or adding to this inital set of mappings until gold standard mappings are properly identified. This supporting process can be justified in two aspects. First, the manual mappings from xEBR ontology to local ontologies are product of joint effort and accepted as standard, thus good in quality. Second, the heuristics applied there is based on the rather intuitive transitivity of subclass relations. Next, one subsection is devoted to applying the heuristics to xEBR WG mappings in order to get a initial set of mappings. The remaining subsections explain the making and interpretation of the gold standard mappings.

A.1 Heuristic Mappings

Because we are ultimately interested in matching concepts from local ontologies, heuristically inferred mappings (*heuristic mappings* hereafter) among local ontologies can serve as baseline to 1) verify our proposed principles for logic-based reasoning, 2) evaluate mappings automatically generated by various matchers. For Italian and German ontologies, heuristic mappings have already been created using some simple techniques [14]. In our case, we need to construct the inferred mappings between French and Spanish ontologies.

Note xEBR WG mappings are in Simple Knowledge Organization System

(SKOS) format, which is "a common data model for sharing and linking knowledge organization systems via the Web". This set includes relations like *exactMatch*, *broadMatch*, *narrowMatch* and *closeMatch*, which are called SKOS relations as a whole in the following discussion. SKOS relations have well-defined semantics and have been used in many different fields to represent correspondences between entities. For example, in the financial domain when xEBR WG create mappings from a international financial reporting schema to national schemas, their choice of mapping representation is evidently influenced by SKOS relations. A heuristic conversion from SKOS relations to description logic representation is presented in Table A.2 and used for data preparation later in this thesis.

Table A.1: Heuristics to infer mappings between $\mathcal{O}_f$ and $\mathcal{O}_s$ on the basis of xEBR WG mappings.

	$C_f \sim C_x$	$C_x \sim C_s$	$C_f \sim C_s$
1	=	=	=
2	=	$\sqsubseteq$	$\sqsubseteq$
3	=	$\sqsupseteq$	$\sqsupseteq$
4	$\sqsubseteq$	=	$\sqsubseteq$
5	$\sqsubseteq$	$\sqsubseteq$	$\sqsubseteq$
6	$\sqsubseteq$	$\sqsupseteq$	*null*
7	$\sqsupseteq$	=	$\sqsupseteq$
8	$\sqsupseteq$	$\sqsubseteq$	*overlap*
9	$\sqsupseteq$	$\sqsupseteq$	$\sqsupseteq$

In Table A.1, C_f is a concept in French ontology, C_x XBRL Europe Business Registers (xEBR) ontology, C_s Spanish ontology; $\sim$ indicates the relation between two concepts; = stands for *exactMatch*, $\sqsubseteq$ *narrowMatch*, $\sqsupseteq$ *broadMatch* and *overlap* the relation that is weaker than all previous relations and *null* means no obvious relation between two concepts. Each row is interpreted as follows: if both relations in column 2 and 3 hold, the relation in the last column follows. For instance, in row 2, if C_f *exactMatch* C_x and C_x *narrowMatch* C_s hold, then it follows C_f *narrowMatch* C_s. A special case is in row 6 where both C_f and C_s have

Table A.2: Logical interpretation of the gold standard mappings.

	DL Syntax	Description
C *exactMatch* D	$C \equiv D$	C is equivalent to D
C *narrowMatch* D	$C \sqsubseteq D$	C is narrower than D
C *broadMatch* D	$C \sqsupseteq D$	C is broader than D

narrowMatch relation with C_x, in which case we cannot determine the relation between C_f and C_s and hence we simply denote it as *null*.

We consider only those concepts in the calculation hierarchy of *balance sheet*s. The ontologies we constructed here contain only those concepts. The reasons for such treatment are 1) Balance sheet is one most common report that uses most basic financial concepts, like Asset, Liability and Stockholders' Equity, etc; 2) Concepts of monetary type in balance sheet bear a rather independent and complete hierarchy. Comparing the financial ontologies constructed above and the conference ontologies, it is clear that most financial ontologies are bigger in size. We have thus obtained heuristic mappings between $\mathcal{O}_f$ and $\mathcal{O}_s$ in Table A.3.

Table A.3: Heuristic mappings between $\mathcal{O}_x$, $\mathcal{O}_f$ and $\mathcal{O}_s$. The numbers in parentheses are the numbers of *closeMatch* while the reminder are the numbers of *exactMatch*.

# Heuristic mappings	$\mathcal{O}_x$	$\mathcal{O}_f$	$\mathcal{O}_s$
$\mathcal{O}_x$	$\times$	24(61)	23(18)
$\mathcal{O}_f$		$\times$	10(62)
$\mathcal{O}_s$			$\times$

As presented in Section 4.1.1, while working with the domain expert to create $\mathcal{M}_{gs}$, we face some difficulty in matching financial reporting schemas from different countries and in interpreting $\mathcal{M}_{gs}$. In the following, we present our work on both issues. Note that each concept in the calculation hierarchy is uniquely located, therefore numeric ID (starting from 1) is assigned to each of them. This simplifies discussion considerably.

A.2 Mismatches

Ontology mismatches are commonplace while integrating ontologies of different sources because of the distributed nature of ontology development. According to Visser et al. [48], ontology mismatches are of two kinds: *conceptualisation mismatches* and *explication mismatches*. Conceptualisation mismatches arise when there are more than one conceptualisations for a given domain. Distinct conceptualisations differ in the set of concepts identified or relations among these concepts. Therefore, conceptualisation mismatches can be further distinguished into two sub-categories: class mismatches and relation mismatches. Class mismatches are concerned with classes of different levels of abstraction or having different descendants. Relation mismatches are associated with the relations identified. They can be the same set of concepts structured differently, or assigning an attribute to different concepts, or having different ranges for the same attribute. Explication mismatches are concerned with the way conceptualisations are specified, and can also be further distinguished into 6 sub-categories, which are clearly illustrated with a diagram and examples by Smart and Engelbrecht [49]. There are also other classifications of ontology mismatches, as shown in the work of Hameed et al. [50].

We observe a number of mismatches between French and Spanish balance sheet ontologies. In the following, we perform an analysis of these mismatches. A most telling case is about how "treasury shares" are classified differently in those two ontologies. Both French and Spanish ontologies identify this concept and present it as `ca:ActionsPropresNet` and `pgc07:PatrimonioNetoFondosPropiosAccionesParticipacionesPatrimonioPropias` respectively. Apart from this difference in terminology, this French concept is classified as assets while the Spanish concept as liabilities. Treasury shares are stock that has been bought back by the issuing corporation. In French financial reporting schemas, they bear a positive sign in assets as it is truly something valuable owned by the corporation. In Spanish financial reporting schemas, however, they bear a negative sign in liabilities because these are the part of liabilities corporation actually do not need to pay. Thus, this mismatch belongs first to conceptualisation category, then secondarily to *term mismatch* in the explication category. Based on semantics of these two concepts, this match is included in the gold standard mappings as domain expert suggested. To reconcile this difference in conceptualisation, we specifically identify and define `TreasuryShare` as being either a part of assets or a part deducted from equity, and a subclass of `VariablyClassified` (see in Table A.4).

Table A.4: Definition of `TreasuryShare` in the BAO

`TreasuryShare`
$\equiv$ (∃`hasClassification.(Asset` $\sqcup$ `DeductionFromEquity))`
$\sqcap$ (∃`hasFinancialInstrument.OwnStock)`
$\sqsubseteq$ `VariablyClassified`

Table A.5: Concepts involved in a mapping from the gold standard mappings

Concept ID	Concept URI	English Label
$F114$	`ca:ParticipationsNet`	Investments, Net
$S19$	`pgc07:ActivoNoCorrienteInversiones-EmpresasGrupoEmpresasAsociadasLargoPlazoInstrumentos-Patrimonio`	long-term investments in group companies and associates, equity instruments
$S26$	`pgc07:ActivoNoCorrienteInversiones-FinancierasLargoPlazo-InstrumentosPatrimonio`	equity instruments

There are still a number of cases illustrating mismatches between $\mathcal{O}_f$ and $\mathcal{O}_s$. We need to keep in mind this fact when we analyze gold standard mappings and later when we evaluate candidate mappings.

A.3 Interpreting $\mathcal{M}_{gs}$

For example in Table A.5, $F114$ is less than the sum of $S19$ and $S26$. In this case, it is not appropriate to simply add $F114 \sqsubseteq S19$ and $F114 \sqsubseteq S26$. Instead, in order to truthfully represent the knowledge, this relation should be transformed into $F114 \sqsubseteq S19 \sqcup S26$. For another example in Table A.6, we have one equation, $F168 = S106 + S117 - S119$. It means the value of $F168$ in French equals the sum of $S106$ and $S117$, with $S119$ left out. $S119$ is a part of $S117$. There is no property in OWL syntax that corresponds to subtraction. While looking at the Spanish calculation hierarchy, it is clear that we can enumerate all other sub-

Table A.6: Another example of interpreting arithmetic equations in gold standard mappings. Subtraction is interpreted semantically as exclusion.

Concept ID	Concept URI	English Label
$F168$	`ca:Dettes`	debts
$S106$	`pgc07:PasivoNoCorriente-DeudasLargoPlazo`	noncurrent liabilities; long-term debts
$S117$	`pgc07:PasivoCorriente`	current liabilities
$S119$	`pgc07:PasivoCorrienteProvisionesCortoPlazo`	current liabilities; short-term provisions

components of $S117$ than $S119$, so as to get what $(S117 - S119)$ refers to.

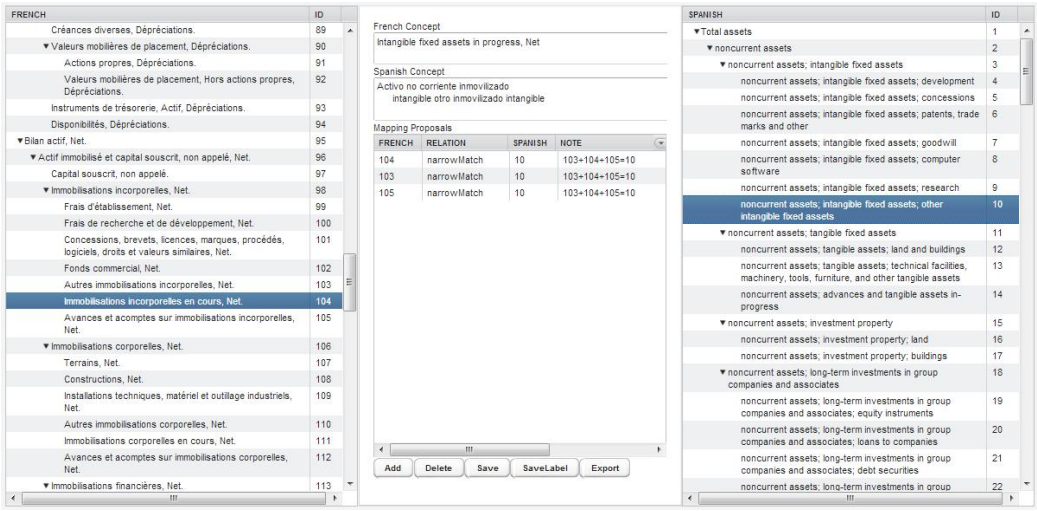


Figure A.1: A screen shot of ConceptMatcher as the tool to create gold standard mappings.

Appendix B

Automated mapping suggestions for conference dataset

Table B.1 presents the number of mapping suggestions from different matchers for different pairs of conference ontologies.

Table B.1: Automated mapping suggestions for ontologies from conference dataset

	agrmaker	aroma	cider	codi	csa	ldoa	lily	logmap	maasmtch	mapevo	mappso	mapsss	optima	yam
# mapping suggestions														
cmt-ekaw	6	14	19	10	28	103	19	6	6	15	47	9	34	5
cmt-Conference	26	26	16	10	34	119	19	7	5	27	62	12	39	10
Conference-ekaw	34	29	26	15	66	114	21	15	11	19	82	20	64	13
cmt-confOf	17	12	13	9	25	98	13	6	5	12	44	12	23	7
confOf-ekaw	23	23	20	15	36	79	18	13	9	9	50	13	37	14
cmt-edas	8	12	20	9	27	112	24	8	9	14	66	12	38	12
edas-ekaw	25	23	30	16	50	137	24	15	9	21	85	11	56	14

Appendix C

Mapping proposals from the grouping principle

As can be seen in Table C.1, we have one concept from xEBR taxonomy, `xebr:AmountsPayableWithinOneYearTotal`[1], mapped to 4 different Spanish concepts, which are all sub-concepts of another Spanish concept `pgc07:PasivoCorriente`. It seems reasonable that there should be a mapping from this core concept to `pgc07:PasivoCorriente`. But this mapping is not among the manual mappings. There are a couple of cases like this one, it is interesting to know whether we can make mapping suggestions like

`xebr:AmountsPayableWithinOneYearTotal`

narrowMatch

`pgc07:PasivoCorriente`.

We consulted Spanish expert on this matter. The expert points out that `pgc07:PasivoCorriente` contains additionally other concepts that are not mapped to `xebr:AmountsPayableWithinOneYearTotal` and suggests that there should be the mapping above.

[1] Concepts with prefix `xebr:` belong to xEBR taxonomy.

Table C.1: Mapping proposals from the grouping principle.

XEBR Ontology	Spanish Ontology	
xebr:AmountsPayable-WithinOneYearTotal	pgc07:PasivoCorrienteDeudasEmpresasGrupoEmpresasAsociadas	pgc07:PasivoCorriente
	pgc07:PasivoCorrienteAcreedoresComercialesOtrasCuentasPagar	
	pgc07:PasivoCorrienteDeudasCaracteristicasEspecialesCortoPlazo	
	pgc07:PasivoCorrienteDeudasCortoPlazo	
xebr:AmountsPayable-AfterMoreThanOneYearTotal	pgc07:PasivoNoCorrienteDeudaCaracteristicasEspecialesLargoPlazo	pgc07:PasivoNoCorriente
	pgc07:PasivoNoCorrienteDeudasEmpresasGrupoEmpresasAsociadasLargoPlazo	
	pgc07:PasivoNoCorrienteDeudasLargoPlazo	

Appendix D

Refining OAEI 2010 mappings

The following are comparative analyses of LOMR and ALCOMO in terms of hes and f on a number of ontologies from conference dataset.

HES

Now there is a comparative analysis of LOMR and ALCOMO in terms of hes on the same set of conference ontologies. In Table D.1, given different pairs of ontologies in the left most column, LOMR and ALCOMO are used to refine mapping suggestions from a list of matchers in the first row. Blank cells indicate that at least one of the refinement systems fails to produce valid values. Comparing refined mappings agaist the respective original mapping suggestions, we get hes_L (of LOMR) and hes_A (of ALCOMO) for each ontology pair. We compute $hes_L - hes_A$ in order to see whether LOMR performs better than ALCOMO. Take for example the number 0.105 in the second column and fourth row of the table. It means that given the mapping suggestions generated by AgreementMaker [51] (abbreviated as "agrmaker") for the ontology pair `Conference` and `ekaw`, LOMR achieves hes of 0.263 ($hes_L = 0.263$) and ALCOMO of 0.158 ($hes_A = 0.158$), where $hes_L - hes_A = 0.105$.

F measure

The differences in f of refined mappings from LOMR and ALCOMO are presented in the upper part of Table D.1. f_L refers to F-measure of refined mappings using LOMR, while f_A refers to that of ALCOMO. We compute $f_L - f_A$ in order to see whether LOMR achieves better f than ALCOMO. Take for example the

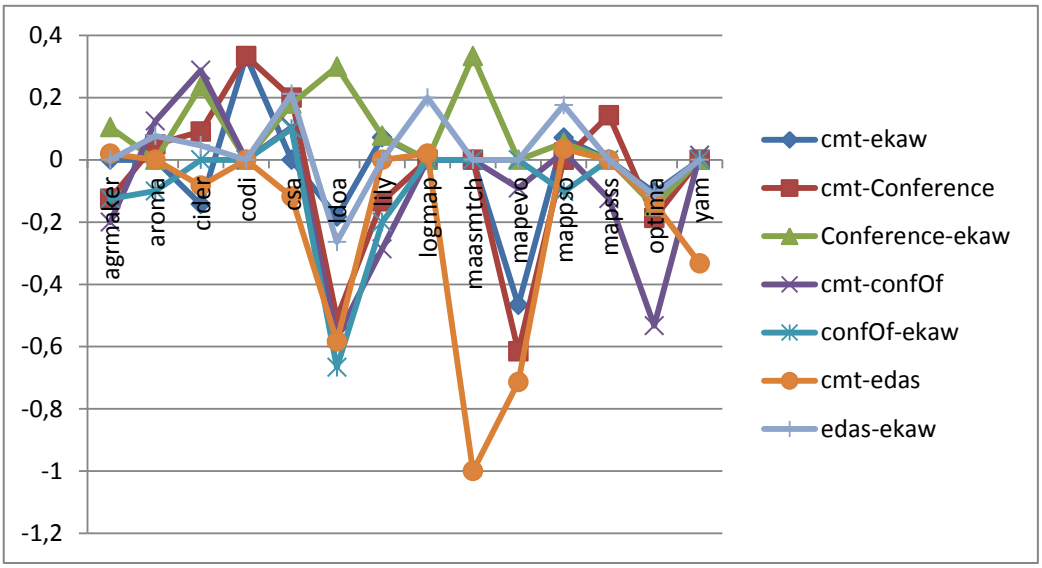


Figure D.1: Illustration of difference of hes of LOMR and ALCOMO over a number of ontology pairs from conference dataset

column "lily". It can be seen that for the mapping suggestions generated by LILY LOMR achieves better f in most cases. Note that cells are left blank because there is no f available. It happens when all mapping suggestions are removed, resulting in $r = 0$ or $p = 0$. According to Equation 4.1, no f can be obtained.

Table D.1 can be illustrated as in Figure D.1. It can be seen that in general ALCOMO achieves higher hes than LOMR. There are also a number of cases where LOMR achieves better hes. One such case is the ontology pair `Conference` and `ekaw`, as illustrated in Figure D.2.

Table D.1: Comparative analysis of LOMR and ALCOMO on ontologies from conference dataset

	agrmaker	aroma	cider	codi	csa	ldoa	lily	logmap	maasmtch	mapevo	mappso	mapsss	optima	yam
$hes_L - hes_A$														
cmt-ekaw	0	0	-0.143	0.333	0	-0.183	0.072	0	0	-0.466	0.071	0	-0.104	0
cmt-Conference	-0.125	0.05	0.091	0.333	0.2	-0.509	-0.133	0	0	-0.615	0	0.143	-0.187	0
Conference-ekaw	0.105	0	0.236	0	0.18	0.299	0.077	0	0.333	0	0.055	0	-0.149	0
cmt-confOf	-0.2	0.125	0.288	0	0.105	-0.571	-0.286	0	0	-0.091	0.025	-0.125	-0.533	0.015
confOf-ekaw	-0.125	-0.1	0	0	0.1	-0.667	-0.2	0	0	0	-0.103	0	-0.116	0
cmt-edas	0.02	0	-0.083	0	-0.118	-0.583	0	0.02	-1	-0.714	0.035	0	-0.138	-0.333
edas-ekaw	0	0.077	0.046	0	0.212	-0.264	0	0.2	0	0	0.176	0	-0.114	0
$f_L - f_A$														
cmt-ekaw	0		-0.028	0.033	-0.048	0.028	0.077	0	0		0.044	0	0.046	0
cmt-Conference	-0.023	0.007	0.012	0.023	0.046	0	0.042	0	0	-0.029	0	0.015	-0.029	0
Conference-ekaw	-0.106	-0.089	0.034	-0,038	0.003		0.047	0	0.013		-0.033	0	-0.161	0
cmt-confOf	-0.027	0.08	0.03	0	0.021	-0.041	-0.03	0	0	-0.003	0.002	-0.01	0.008	0
confOf-ekaw	-0.085	-0.089	-0.038	-0.076	-0.031	-0.18	-0.019	-0.104	0		-0.041	0	-0.074	-0.041
cmt-edas	0	0	-0.017	0	-0.03		0.095	0	-0.035		**0.176**	-0.054	0.108	-0.03
edas-ekaw	-0.033	0.05	0.007	-0.074	0.133	-0.034	0.036	-0.026	0		0.029	0	0.041	0

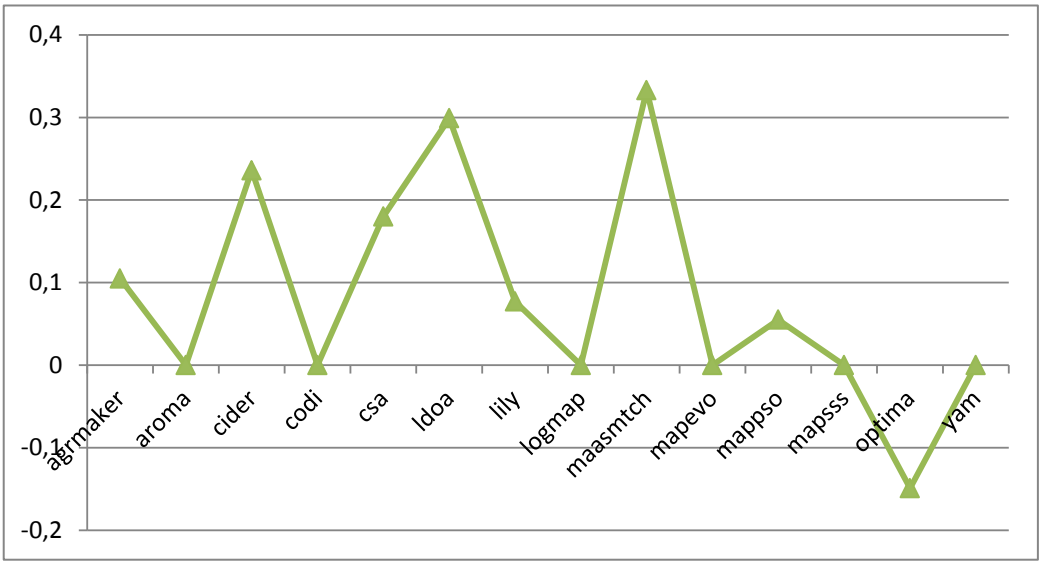


Figure D.2: Illustration of difference of hes of LOMR and ALCOMO over the ontology pair `Conference` and `ekaw`

Appendix E

Case study: deleting a correct mapping

Mapping refinement systems sometimes delete correct mappings, those that are also in gold standard mappings. Even though it is not the desired behavior of a mapping refinement system from the view point of end users, these undesired deletions are the consequence of many other reasons than the refinement procedure. It's observed that mapping generators provide misleading information. The following is a case study illustrating such situation.

In Figure E.1, there is a violating axiom $\nu_{bd} = (C_{fb} \sqsubseteq C_{fd})$, with one explanation $\pi_{bd} = \{\mu_{bc}, \alpha_{ca}, \mu_{ad}\}$, where $\mu_{bc} = \langle C_{fb} \sqsubseteq C_{sc}, 0.4\rangle$, $\alpha_{ca} = (C_{sc} \sqsubseteq C_{sa})$ and $\mu_{ad} = \langle C_{sa} \equiv C_{fd}, 0.6\rangle$. Note these two mappings along with their respective confidence values are generated using COAL. To resolve this violation, at least one of the two mappings μ_{bc} and μ_{ad} has to be removed. Now given that μ_{ad} has a higher confidence value, μ_{bc} is deleted. But μ_{bc} is in fact in the gold standard, though the mapping generator assigns a low confidence value to it. On the other hand, according to their semantics, C_{fd} is not equivalent to C_{sa}.

This example is actually taken from the French and Spanish ontologies used in the evaluation. C_{fb} refers to the French concept `ca:EnCoursDeProductionDeBienNet`, whose English label is "in-progress goods, net" and C_{sc} refers to the Spanish concept `pgc07:ActivoCorrienteExistenciasProductosCurso`, whose English label is "partly-finished goods". Clearly, there should be a mapping between them. On the other hand, C_{fd} refers to the French concept `ca:StocksDeMarchandisesNet`, meaning "merchandise inventories, net" and C_{sa} refers to the Spanish concept

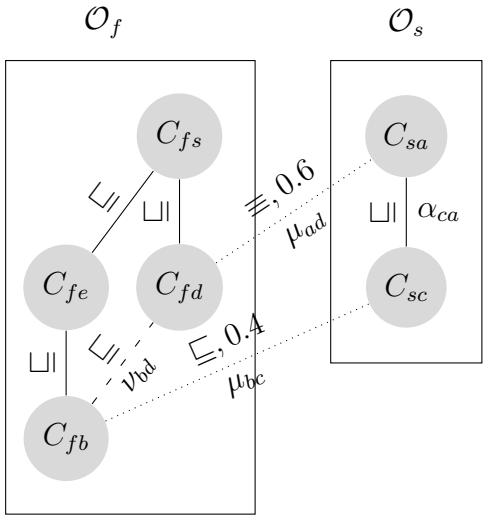


Figure E.1: Case study of deletion of a correct mapping

Table E.1: The corresponding concepts in $\mathcal{O}_f$ and $\mathcal{O}_s$ in the case study

C_{fb}	`ca:EnCoursDeProductionDeBienNet`
C_{sc}	`pgc07:ActivoCorrienteExistenciasProductosCurso`
C_{fd}	`ca:StocksDeMarchandisesNet`
C_{sa}	`pgc07:ActivoCorrienteExistencias`

`pgc07:ActivoCorrienteExistencias`, "inventories" in English. These two concepts are as their labels reveal not equivalent. μ_{ad} has a higher confidence value probably because the labels of C_{sa} and C_{fd} are more similar via some string-based techniques. Now it is also clear why ν_{bd} is a violating axiom because C_{fd} is about merchandise inventories while C_{fb} is about in-process goods.

This case study here tries to shed some light in cases where correct mappings are also deleted. The reason could be misleading information of ontology matcher, or inadequacy of gold standard, etc.

List of Symbols

$\mathcal{O}$	An ontology
α	A logical axiom
Σ	The signature of an ontology, an axiom, a set of axioms, etc
$\mathcal{M}$	Mapping suggestions
$\mathcal{O}_m$	Mapping ontology
$\mathcal{M}_{gs}$	Gold standard mappings
μ	A mapping suggestion
ρ	Logical relation between two concepts, being one of $\{\equiv, \sqsubseteq, \sqsupseteq\}$
ϵ	A confidence value of a mapping suggestion
τ	A translation function that converts a mapping into an axiom
Ω	The explanation of an unsatisfiable concept
Δ	The diagnosis of an unsatisfiable concept
ν	A violating axiom, i.e., an axiom that is originally not entailed in one ontology, but introduced by adding mapping suggestions
$\mathcal{S}_{vio}$	The complete set of violating axioms w.r.t an ontology
S_{vio}	A subset of violating axioms
G	A group of concepts that are mapped to a single foreign concept

M	A module of a concept in an ontology
p	Precision
r	Recall
f	F-measure
hes	Human effort saved
t	Time (minute)

Glossaries

ALCOMO	Applying Logical Constraints on Matching Ontologies
API	application programming interface
BAO	Basic Accounting Ontology
COAL	Cross-lingual Ontology Alignment
DL	description logics
DDL	distributed description logics
HES	Human Effort Saved
IFRS	International Financial Reporting Standard
IR	Information Retrieval
IRI	internationalized resource identifier
KR	knowledge representation
LOMR	Logic-based Mapping Refinement
ML	machine learning
MONNET	Multilingual Ontologies for Networked Knowledge
NLP	natural language processing
OAEI	Ontology Alignment Evaluation Initiative
OWL	Web Ontology Language
PGC07	Taxonomía del Nuevo Plan General de Contabilidad 2007
RDF	Resource Description Framework

SEC	Securities and Exchange Commission
SEOM	Semantically Enriched Ontology Matching
SKOS	Simple Knowledge Organization System
TCA	Taxonomie Comptes Annuels
XBRL	eXtensible Business Reporting Language
xEBR	XBRL Europe Business Registers
xEBR WG	XBRL Europe Business Registers Working Group
XML	Extensible Markup Language
UMLS-Meta	Unified Medical Language System® Metathesaurus
URI	uniform resource identifier

Bibliography

[1] C. Hoffman and L. Watson, *XBRL For Dummies*. Wiley Publishing, Inc., 2009.

[2] T. Verdin, G. Maguet, and S. Thomas, "Promoting xbrl for cross-border data exchange by business registers in europe," *Interactive Business Reporting*, vol. 2, pp. 18–21, 2012.

[3] D. S. Frankel, "Xbrl and semantic interoperability," *Model Driven Architecture Journal*, 6 2009.

[4] P. Shvaiko and J. Euzenat, "Ontology matching: State of the art and future challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 158–176, 2013.

[5] H. Bierman, *An introduction to accounting and managerial finance: a merger of equals*. World Scientific Publishing Co. Pte. Ltd., 2009.

[6] M. R. Quillian, "Word concepts: A theory and simulation of some basic semantic capabilities," *Behavioral Science*, vol. 12, no. 5, pp. 410–430.

[7] M. Minsky, "Frame theory," *Thinking: Reasings in Cognitive Science*, pp. 355–376, 1977.

[8] F. Baader and W. Nutt, "Basic description logics," in *Description Logic Handbook* (F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, eds.), pp. 43–95, Cambridge University Press, 2003.

[9] J. Euzenat, "Semantic precision and recall for ontology alignment evaluation," in *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, (San Francisco, CA, USA), pp. 348–353, Morgan Kaufmann Publishers Inc., 2007.

[10] J. Euzenat and P. Shvaiko, *Ontology matching*. Springer, 2007.

[11] C. Meilicke, *Alignment incoherence in ontology matching*. PhD thesis, University of Mannheim, Chair of Artificial Intelligence, 2011.

[12] E. Jiménez-Ruiz, B. Cuenca Grau, I. Horrocks, and R. Berlanga, "Logic-based assessment of the compatibility of UMLS ontology sources," *Journal of Biomedical Semantics*, vol. 2, 2011.

[13] B. Fu, R. Brennan, and D. O'Sullivan, "A configurable translation-based cross-lingual ontology mapping system to adjust mapping outcomes," *J. Web Sem.*, vol. 15, pp. 15–36, 2012.

[14] D. Spohr, L. Hollink, and P. Cimiano, "A machine learning approach to multilingual and cross-lingual ontology matching," *The Semantic WebISWC 2011*, p. 665680, 2011.

[15] N. F. Noy, "Semantic integration: A survey of ontology-based approaches," *SIGMOD Record*, vol. 33, p. 2004, 2004.

[16] S. Godugula, "Survey of ontology mapping techniques," June 2008.

[17] E. Jiménez-Ruiz, B. Cuenca Grau, Y. Zhou, and I. Horrocks, "Large-scale interactive ontology matching: Algorithms and implementation," in *Proc. of ECAI*, 2012.

[18] P. Wang and B. Xu, "Lily: Ontology alignment results for oaei 2008," in *OM* (P. Shvaiko, J. Euzenat, F. Giunchiglia, and H. Stuckenschmidt, eds.), vol. 431 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2008.

[19] Z. Aleksovski, W. ten Kate, and F. van Harmelen, "Exploiting the structure of background knowledge used in ontology matching," in *Ontology Matching*, 2006.

[20] T. Declerck, H.-U. Krieger, S. M. Thomas, P. Buitelaar, S. O'Riain, T. Wunner, G. Maguet, J. McCrae, D. Spohr, and E. Montiel-Ponsoda, "Ontology-based multilingual access to financial reports for sharing business knowledge across europe," in *Internal Financial Control Assessment Applying Multilingual Ontology Framework* (J. Roóz and J. Ivanyos, eds.), (1142 Budapest, Erzsébet Királyné útja 125. Hungary), Kiadja a Memolux Kft., Készült a HVG Press Kft. nyomdájában, 9 2010.

[21] T. Declerck and H.-U. Krieger, "Translating xbrl into description logic. an approach using protege, sesame & owl," in *BIS*, pp. 455–467, 2006.

[22] R. Garca and R. Gil, "Publishing xbrl as linked open data," 2009.

[23] J. Bao, G. Rong, X. Li, and L. Ding, "Representing financial reports on the semantic web: - a faithful translation from xbrl to owl," in *RuleML*, pp. 144–152, 2010.

[24] S. O'Riain, E. Curry, and A. Harth, "Xbrl and open data for global financial ecosystems: A linked data approach," *International Journal Of Accounting Information Systems*, pp. 141–162, 2011.

[25] S. O'Riain, *Semantic Paths in Business Filings Analysis*. PhD thesis, National University of Ireland, Galway, 2012.

[26] J. P. Krahel, *On the Formalization of Accounting Standards*. PhD thesis, State University of New Jersey, 2012.

[27] F. Gailly and G. Poels, "Towards ontology-driven information systems: redesign and formalization of the rea ontology," in *Proceedings of the 10th international conference on Business information systems*, BIS'07, pp. 245–259, Springer-Verlag, 2007.

[28] C.-C. Chou and Y.-L. Chi, "Developing ontology-based epa for representing accounting principles in a reusable knowledge component," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2316 – 2323, 2010.

[29] M. C. Gerber and A. J. Gerber, "Towards the development of consistent and unambiguous financial accounting standards using ontology technologies," in *proceedings of the International Conference on accounting 2011*, 2011.

[30] B. Li and L. Min, "An ontology-augmented xbrl extended model for financial information analysis," in *ICIS*, vol. 3, pp. 99–130, 11 2009.

[31] J. Bock, P. Haase, Q. Ji, and R. Volz, "Benchmarking owl reasoners," *ARea2008 - Workshop on Advancing Reasoning on the Web: Scalability and Commonsense*, June 2008.

[32] P. Allen, "Case study: Taxonomy packages - a simple specification to solve a universal problem," *Interactive Business Reporting*, vol. 2, p. 32, 2012.

[33] I. Horrocks, B. Motik, and Z. Wang, "The HermiT OWL reasoner," in *OWL Reasoner Evaluation Workshop (ORE 2012)*, 2012.

[34] G. Qi, Q. Ji, and P. Haase, "A conflict-based operator for mapping revision," in *Description Logics* (B. C. Grau, I. Horrocks, B. Motik, and U. Sattler, eds.), vol. 477 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2009.

[35] C. Lutz, D. Walther, and F. Wolter, "Conservative extensions in expressive description logics," in *IJCAI*, pp. 453–458, 2007.

[36] C. Lutz and F. Wolter, "Conservative extensions in the lightweight description logic el," in *CADE*, pp. 84–99, 2007.

[37] R. Reiter, "A theory of diagnosis from first principles," *Artif. Intell.*, vol. 32, no. 1, pp. 57–95, 1987.

[38] M. Horridge and S. Bechhofer, "The owl api: A java api for owl ontologies," *Semantic Web*, vol. 2, no. 1, pp. 11–21, 2011.

[39] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical owl-dl reasoner," *J. Web Sem.*, vol. 5, no. 2, pp. 51–53, 2007.

[40] D. Tsarkov and I. Horrocks, "FaCT++ description logic reasoner: System description," in *Proceedings of the International Joint Confererence on Automated Reasoning (IJCAR˜2006)*, vol. 4130 of *Lecture Notes in Artificial Intelligence*, pp. 292–297, Springer, 2006.

[41] B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler, "Just the right amount: extracting modules from ontologies," in *WWW*, pp. 717–726, 2007.

[42] K. Dentler, R. Cornet, A. ten Teije, and N. de Keizer, "Comparison of reasoners for large ontologies in the owl 2 el profile," *Semantic Web*, vol. 2, no. 2, pp. 71–87, 2011.

[43] C. Meilicke, H. Stuckenschmidt, and A. Tamilin, "Repairing ontology mappings," in *AAAI*, pp. 1408–1413, AAAI Press, 2007.

[44] C. Meilicke, H. Stuckenschmidt, and A. Tamilin, "Reasoning support for mapping revision," *J. Log. Comput.*, vol. 19, no. 5, pp. 807–829, 2009.

[45] V. Svátek and P. Berka, "Ontofarm: Towards an experimental collection of parallel ontologies," in *In: Poster Session at ISWC*, 2005.

[46] N. Nikitina, "Semi-automatic revision of formalized knowledge," in *ECAI 2010: 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16–20, Proceedings* (H. Coelho, R. Studer, and M. Wooldridge, eds.), vol. 215 of *Frontiers in Artificial Intelligence and Applications*, (Amsterdam), pp. 1097–1098, ECAI 2010: 19th European Conference on Artificial Intelligence, IOS Press, August 2010.

[47] H. Stuckenschmidt, "Debugging owl ontologies - a reality check," in *EON* (R. Garcia-Castro, A. Gómez-Pérez, C. J. Petrie, E. D. Valle, U. Küster, M. Zaremba, and M. O. Shafiq, eds.), vol. 359 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2008.

[48] P. R. S. Visser, D. M. Jones, B. T. J. M. Capon, and M. J. R. Shave, "An analysis of ontological mismatches: Heterogeneity versus interoperability," in *AAAI 1997 Spring Symposium on Ontological Engineering*, (Stanford, USA), 1997.

[49] P. R. Smart and P. C. Engelbrecht, "An analysis of the origins of ontology mismatches on the semantic web," in *16th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2008)*, September 2008.

[50] A. Hameed, D. H. Sleeman, and A. D. Preece, "Detecting mismatches among experts' ontologies acquired through knowledge elicitation," *Knowl.-Based Syst.*, vol. 15, no. 5-6, pp. 265–273, 2002.

[51] I. F. Cruz, F. P. Antonelli, and C. Stroe, "Agreementmaker: efficient matching for large real-world schemas and ontologies," *Proc. VLDB Endow.*, vol. 2, pp. 1586–1589, Aug. 2009.