Contribution of ontology alignment to enterprise interoperability

Fuqi Song

N° d'ordre : 4880

THÈSE

PRÉSENTÉE A

L'UNIVERSITÉ BORDEAUX 1

ÉCOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE L'INGENIEUR

Par SONG Fuqi

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : PRODUCTIQUE

CONTRIBUTION À L'INTEROPERABILITE DES ENTREPRISES PAR ALIGNEMENT D'ONTOLOGIES

Soutenue le : 28 Octobre 2013

Devant la commission d'examen formée de :

M. BOURRIERES Jean-Paul	Professeur, Université Bordeaux 1	Président
M. BERIO Giuseppe	Professeur, Université de Bretagne Sud	Rapporteur
M. ARCHIMEDE Bernard	Professeur, Ecole Nationale d'Ingénieurs de Tarbes	Rapporteur
M. DULUC Franck	Ingénieur R&T, Airbus Operations SAS	Examinateur
M. CERBAH Farid	Ingénieur de Recherche, Dassault Aviation	Examinateur
M. ZACHAREWICZ Grégory	Maître de Conférences, Université Bordeaux 1	Co-directeur
M. CHEN David	Professeur, Université Bordeaux 1	Directeur

送给我的父母和姐姐

A mes parents et ma sœur

Acknowledgement

I would like to thank my advisors of thesis: Dr. Gregory Zacharewicz and Prof. David Chen with my sincere gratitudes and respects, they are the ones who have given me many guidance and valuable advices during Ph.D. study and the writing of the thesis. Without the much time that they have spent and the efforts that they have made to discuss the subject and research with me, the thesis will never come to an end.

I would like to thank my colleague and also friend, Zhang Xin, who is also a Ph.D. student at IMS. We share our experiences and support each other during Ph.D. study. Chen Xun and Ye Xia, whom I spent much time with, gave me many supports and encouragements during the writing of thesis, and many other dear friends who have helped and accompanied me.

Last but not least, my family in China, they give me their strongest support all the time, without their love, I will not be able to complete this thesis.（我最要感谢的是我在中国的家人，你们是我最坚强的后盾，没有你们长久以来的爱与关怀，我是不可能走到今天并完成博士论文的。）

Speech is not enough for me to express my feeling of gratitude, as the period of my Ph.D. study becoming an unforgettable story in my life, it is all of you who are the most important part of it. Bordeaux city and IMS laboratory are the places where I have all the memories, tough or happy, I will never forget, because I was there and passed three important years in my life there.

09/10/2013, IMS Bat. 31, Talence

Contribution of ontology alignment to enterprise interoperability

This thesis brings ontology alignment to contribute to federated enterprise interoperability focusing on data interoperability at the semantic level. In response to existing problems and challenges, aiming at improving the matching ability and precision, a pattern-based core word ontology alignment approach is proposed, as well as an analytic matcher aggregation approach, which allows combining the multiple matchers automatically and improving the combined results. A prototype system is implemented for validation and further application based on the proposed approaches. The experiments suggest that the proposed approaches have obtained promising results and reached expected goals. The proposed ontology alignment approach and implemented prototype system are applied to an ontology-driven architecture for querying data from multiple relational databases to develop enterprise interoperability.

Key words

Enterprise interoperability, ontology alignment, semantic interoperability, matcher aggregation, AHP

Contribution à l'interopérabilité des entreprises par alignement d'ontologies

Cette thèse propose l'utilisation de l'alignement d'ontologies pour contribuer à l'interopérabilité d'une fédération d'entreprises en se basant sur l'interopérabilité des données au niveau sémantique. Une approche d'alignement basée sur des modèles d'ontologie utilisant les mots noyaux est proposée en réponse aux problèmes et aux défis existants, visant ainsi à améliorer la capacité d'adaptation et la précision dans la mise en correspondance de concepts. De plus une étape d'agrégation des «matchers» analytique, qui permet de combiner automatiquement plusieurs adaptateurs et d'améliorer les résultats combinés, vient compléter l'approche. Un système prototype a été mis en œuvre à l'issue des travaux conceptuels pour la validation de l'approche proposée. Les expériences démontrent que l'approche proposée a obtenu des résultats prometteurs et a atteint les objectifs escomptés sur la définition de proximité des concepts. L'approche d'alignement d'ontologies proposée et le système de prototype mis en œuvre ont enfin été appliqués à une architecture dirigée par les ontologies et axée sur l'interrogation des données de plusieurs bases de données relationnelles pour appuyer l'interopérabilité des entreprises.

Mots clés

Interopérabilité des entreprises, alignement d'ontologies, interopérabilité sémantique, agrégation des matchers, AHP

Résumé Étendu en français de la Thèse de Fuqi SONG « Contribution à l'Interopérabilité des Entreprises par Alignement d'Ontologies »

Rappel de la situation des entreprises collaboratives et identification des problèmes

L'exigence croissante de collaboration entre les entreprises, demande une interopérabilité accrue des entreprises afin de faciliter leurs collaborations. Plusieurs dimensions et préoccupations sont impliquées dans le développement de l'interopérabilité de l'entreprise. En particulier, au niveau des points de vue techniques et conceptuels, l'hétérogénéité sémantique des concepts manipulés par les partenaires conclue à l'incompréhension des informations échangées. Cela consiste en un obstacle majeur qui entrave la réalisation de l'interopérabilité de l'entreprise.

Pour faire face à l'hétérogénéité sémantique, les approches basées sur les ontologies sont largement utilisés en raison de l'évolution rapide des technologies connexes du web sémantique et les avantages qu'elles apportent pour faciliter l'interopérabilité sémantique. Cette thèse vise à apporter un alignement d'ontologies pour contribuer à l'interopérabilité fédérée des entreprises en mettant l'accent sur l'interopérabilité des données au niveau sémantique. L'alignement d'ontologies cherche des correspondances sémantiques entre les différents systèmes d'information et joue un rôle important pour permettre l'interopérabilité sémantique. Cette thèse a contribué sur l'adoption de la technique d'alignement d'ontologies pour contribuer à une approche d'interopérabilité fédérée des entreprises, avec l'objectif de développer l'interopérabilité des données d'entreprise au niveau sémantique.

Basé sur les travaux existants dans ce domaine et en réponse aux problèmes et défis restants, des travaux de recherche ont été menés à travers cette thèse visant à améliorer les approches d'alignement d'ontologies, ceci afin de faciliter l'interopérabilité de l'entreprise. La recherche de cette thèse vise à répondre à deux questions cruciales de l'alignement d'ontologies: (i) améliorer la capacité à découvrir des correspondances sémantiques, et (ii) améliorer la méthode d'agrégation par comparateurs « Matchers » en combinant les résultats de plusieurs techniques.

En ce qui concerne la première question, la plupart des approches actuelles d'alignement d'ontologies cherchent des correspondances à partir

du niveau lexical et du niveau structurel plutôt que directement au niveau sémantique. Les difficultés sont causées par la diversité et l'ambiguïté du langage naturel, qui est utilisé pour représenter les entités de l'ontologie. Dans notre travail, une approche d'alignement d'ontologies se basant sur les 'mots de base' (Core Word) est proposée en appliquant les connaissances de traitement du langage naturel (Natural Language Processing, NLP) et d'extraction d'information (Information Extraction, IE). La motivation de cette proposition est que l'étiquette de l'entité dans l'ontologie est généralement nommée par mot composé, qui combine plusieurs mots significatifs simples. Cependant, le plus souvent un mot du mot composé représente le sens principal de l'entité entière. Ce genre de mots est appelé "mot de base". Nous soutenons que le constat de mots clés contribuera de manière significative à la découverte de correspondances sémantiques. Le processus de reconnaissance de mots de base est réalisé sur la base de règles prédéfinies et d'une partie du discours (part of speech, POS) de mots. Un algorithme de mesure spécifique est proposé sur la base du mot de base reconnu et des informations complémentaires pour calculer la similarité sémantique. Outre la méthode basée sur les mots base proposé pour l'alignement d'ontologies, pour gérer les diverses situations et améliorer la capacité d'adaptation, deux adaptateurs au niveau lexical et au niveau structurel sont appliqués par les algorithmes de réutilisation : la distance d'édition (edit distance), n-gram et l'inondation de similitude (similarity flooding).

Concernant une autre question importante dans l'alignement d'ontologies, l'agrégation de matcher a attiré l'attention de beaucoup de chercheurs, puisque classiquement plusieurs adaptateurs sont appliqués pour procéder à l'alignement d'ontologies. Bien que de nombreuses méthodes basée multi-stratégie aient été proposées, des méthodes plus automatiques et dynamiques sont attendues afin d'améliorer les résultats correspondants combinés. Dans cette thèse, une approche analytique appelée Analytic Hierarchy Process (AHP) est appliqué à déterminer les poids de chaque matcher. AHP a déjà été utilisé et validé dans un très large éventail de domaines et a d'applications matures pour le calcul de pondération dans certains domaines, tels que le E-learning et la prise de décision. Dans notre travail, il est tout d'abord proposé d'aligner les ontologies par l'agrégation de matcher. Le processus d'apprentissage est effectué avec l'aide de trois indicateurs de similarité qui proviennent du niveau tout-ontologie. Cette méthode vise à automatiser l'ensemble du processus d'agrégation et à améliorer les résultats combinés.

Basé sur les approches proposées, un prototype de système a été implémenté en Java pour l'évaluation et les utilisations futures. Ce système

se compose de trois éléments principaux: un « pré-processeur », un « matcher » et un « agrégateur ». Il contient environ 5000 lignes de codes. Les expériences ont été réalisées en comparaison avec l'ensemble de données de la base de référence OAEI, et avec d'autres approches. L'évaluation a montré que les approches proposées ont obtenu des résultats prometteurs et ont atteint les objectifs escomptés.

Pour appliquer les méthodes proposées d'alignement d'ontologies pour améliorer l'interopérabilité des données au niveau sémantique, une architecture axée sur l'ontologie pour l'interrogation des données de plusieurs bases de données relationnelles est proposé d'utiliser le système de prototype mis en œuvre. Cette architecture présente certaines extensibilités et peut être appliquée à plusieurs scénarios d'application en fonction des demandes spécifiques.

Contributions Principales

La thèse a permis d'améliorer l'alignement d'ontologies et les approches d'agrégation de « matcher » pour une meilleure performance et pour faciliter l'interopérabilité entre les différents systèmes d'information d'entreprise hétérogènes. En comparant les approches proposées au contexte EIS, la performance peut être considérée comme améliorée à deux niveaux : (i) une possibilité accrue de trouver des correspondances appropriées, et (ii) une meilleure précision des résultats de l'alignement. A cet égard, le travail effectué dans cette thèse a été publié dans plusieurs revues et conférences internationales.

Le chapitre 1 décrit la problématique, la portée de la recherche, l'investigation des approches basées sur l'ontologie et les propositions spécifiques de cette thèse à améliorer. Un état de l'art sur l'utilisation des technologies du web sémantique pour contribuer à l'interopérabilité des systèmes d'information d'entreprise sont décrits dans ce chapitre et les détails peuvent être trouvés dans Song et al. [93]. Les rôles que joue l'ontologie de l'interopérabilité de l'entreprise a été discuté dans Song et al. [121].

Le chapitre 2 élabore une nouvelle méthode de mesure basée sur la sémantique de similitude des mots de base « core-word » pour l'alignement d'ontologies. Cette méthode consiste à mesurer la similarité basée sur le mot de base reconnu, qui représente le sens principal dans un mot composé ou une phrase courte. Un algorithme spécifique est proposé pour calculer la valeur de similarité sémantique [122]. Ainsi deux comparateurs de niveau lexical et de niveau structurel sont conçus en réutilisant les algorithmes existants pour améliorer la capacité d'appariement. Plus tard, le chapitre 4 valide par les résultats d'évaluation obtenus que l'approche proposée

possède une bonne capacité d'adaptation et atteint les objectifs définis. Ce chapitre conclue par le fait que la méthode de mesure de similarité sémantique proposée peut aussi être appliquée à des domaines autres que l'alignement d'ontologies, comme, la recherche sémantique, le web sémantique, l'extraction d'information, etc.

Le chapitre 3 développe une nouvelle méthode d'agrégation de matcher analytique qui permet de combiner les multiples adaptateurs proposés [123, 124]. Cette méthode est développée sur la base de AHP et de trois indicateurs de similarité visant à automatiser le processus d'agrégation et à améliorer les résultats combinés. Les résultats des expériences dans le chapitre 4 montrent que cette nouvelle méthode d'agrégation proposée améliore considérablement les résultats combinés. En outre, la méthode est facile à appliquer en raison du processus automatique et peut également être appliquée à d'autres domaines pour faire face aux problèmes de pondération qui implique de multiples facteurs et variables complexes.

Le chapitre 4 met en œuvre les approches proposées en Java et teste avec les jeux de données de référence de l'OAEI. Les résultats expérimentaux conduisent à la conclusion que les approches d'alignement d'ontologies proposées et la méthode d'agrégation apportent des améliorations au regard de l'existant et donnent des résultats prometteurs. En comparaison avec les 13 autres approches qui ont participé à la *biblio* de benchmarking de OAEI 2011, le HF1 de notre approche est de 0,84 et arrive classé au deuxième rang. Elle reste inférieure à la meilleure approche (YAM++, 0,86) de 0,03 mais est supérieure à la troisième CSA [53] de 0,01. Le principal avantage de notre approche est le compromis entre la simplicité d'application et une bonne capacité d'adaptation, en particulier la bonne capacité d'adaptation aux les ontologies réelles, ainsi que les potentiels d'être applicable à d'autres domaines. Les améliorations possibles de nos approches ont été identifiées et sont discutées dans la section de conclusion du chapitre 4. Ces perspectives, détaillées ci-après, pourraient permettre si elles étaient mises en œuvre d'obtenir un résultat encore meilleur.

Le chapitre 5 présente une architecture axée sur l'ontologie pour l'interrogation de données provenant de plusieurs bases de données relationnelles [40]. Ceci afin de mettre œuvre les approches proposées dans un système prototype dont l'objectif est de développer l'interopérabilité de l'entreprise. Cette architecture est axée sur le traitement de l'interopérabilité des données au niveau sémantique. Elle peut être appliquée à plusieurs domaines pour soutenir l'interopérabilité des données d'entreprise en conséquence.

Perspectives

Le travail présenté dans cette thèse est également préoccupé par plusieurs autres pistes de recherche et des questions qui peuvent être considérées comme des questions ouvertes pour la recherche future comme suit :

- Concernant les règles pour la reconnaissance des mots de base, afin de s'adapter à un domaine de connaissances spécifique et à des cas particuliers, une recherche future pourrait consister à étendre ces règles, telles que par la distinction de règles plus générales et de règles spéciales. Actuellement, les règles définies dans ce travail sont limitées au cas général et peuvent-être moins efficaces dans les cas particuliers ;
- Le comparateur PCW et le comparateur lexical effectuent des tâches d'alignement qui reposent principalement sur les étiquettes des entités dans l'approche, si les commentaires et les annotations supplémentaires de l'entité étaient également pris en compte en tant que sources sémantiques pour faciliter l'alignement, les résultats pourraient être améliorés ;
- Dans la méthode d'agrégation de matcher, trois indicateurs de similarité sont utilisées pour automatiser l'attribution d'échelles de pondération dans l'application AHP. Afin d'appliquer cette méthode de pondération proposée dans d'autres domaines, la façon de calculer les indicateurs de similarité peut être adaptée en conséquence, comme en se basant sur certains paramètres susceptibles de refléter l'importance des alternatives ;
- Le système prototype actuellement mis en place utilise une interface en ligne de commande qui prend les paramètres en entrée et génère des alignements au format XML en sortie. Bien que nous pensons que c'est suffisant pour les utilisateurs d'utiliser les résultats en fonction de leurs besoins spécifiques, L'IHM peut être améliorée pour rendre le système plus facile à utiliser en développant une interface graphique et un affichage graphique des résultats de l'alignement;
- Concernant la solution de construire une couche sémantique de l'information (SIL) pour développer l'interopérabilité des données, les règles d'extraction de l'ontologie à partir de bases de données relationnelles sont principalement définies au niveau du schéma. De nouvelles de règles relatives au niveau de l'instance pour extraire les enregistrements dans les bases de données relationnelles vers des instances d'ontologies pourraient être proposées afin d'enrichir la sémantique des données ;
- Compte tenu des besoins d'interopérabilité sémantique accrus au niveau conceptuel dans les entreprises, un travail d'application et d'extension des résultats conceptuels de la thèse est en cours [125]. Il propose

notamment d'appliquer l'alignement d'ontologies pour le développement de l'interopérabilité entre le Model-Driven Architecture (MDA) et des modèles de simulation, il est effectué sur la base de l'alignement d'ontologies proposée dans les approches de cette thèse. La méthodologie générale a été proposée et élaborée dans [125]. L'application de l'alignement d'ontologies pour échanger des informations fournit une connexion sémantique qui rend le lien flexible entre les modèles et la simulation. Au stade actuel, le travail qui a été fait met l'accent sur la définition d'une méthodologie générale et d'un cadre. Le travail restant concerne principalement l'élaboration de la méthode et des mesures d'application opérationnelles. Ce travail comprend : (i) les règles et le formalisme de l'échange d'informations, et (ii) le moyen d'échanger des informations entre les deux parties à l'aide de l'alignement d'ontologies.

Table of content

List of figures

List of tables

Acronyms

AHP	Analytic Hierarchy Process
API	Application Programming Interface
CIM	Computation Independent Model
ED	Edit Distance
EIS	Enterprise Information System
FEI	Framework for Enterprise Interoperability
HF1	Harmonic Measurement 1
HP	Harmonic Precision
HR	Harmonic Recall
LBM	Lexical-based Matcher
MDA	Model-Driven Architecture
NER	Named Entity Recognition
NG	N-Gram
OAEI	Ontology Alignment Evaluation Initiative
OWL	Ontology Web Language
PCG	Pairwise Connectivity Graph
PCW	Pattern-based Core Word similarity measurement
PIM	Platform Independent Model
POS	Part-Of-Speech
PSM	Platform Specific Model
RDB	Relational DataBase
RDF	Resource Description Framework
SBM	Structural-based Matcher
SES	System Entity Structure
SF	Similarity Flooding
SFA	Similarity Flooding Algorithm
SIL	Semantic Information Layer
SMA	Semantic MAtching
SPARQL	SPARQL Protocol and RDF Query Language
SPG	Similarity Propagation Graph
SQL	Structured Query Language
TS	Test Set
UML	Unified Modeling Language
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

General Introduction

As the demanding requirements for collaboration among enterprises increasing, enterprise interoperability is pursued continuously in order to enable the collaboration. Several dimensions and concerns are involved in developing enterprise interoperability. From the conceptual and technical points of view, semantic heterogeneity is becoming a major barrier that obstructs achievement of enterprise interoperability.

To address semantic interoperability, ontology-based approaches are widely applied due to the rapid development of semantic web related technologies and the benefits that they brought to facilitate semantic interoperability. This thesis aims at bringing ontology alignment to contribute to federated enterprise interoperability focusing on data interoperability at semantic level. Ontology alignment seeks semantic correspondences between different information systems and plays a significant role in enabling semantic interoperability.

Based on the existing works in this domain and in response to remaining problems and challenges, some research works have been carried out through this thesis aiming to improve the ontology alignment approaches, in order to facilitate enterprise interoperability. The research of this thesis focuses on addressing two crucial issues in ontology alignment: (i) to improve the ability to discover semantic correspondences, and (ii) to improve the matcher aggregation method for combining matching results from multiple matching techniques.

Regarding the first issue, most of current ontology alignment approaches seek correspondences from lexical level and structural level rather than from semantic level directly. The difficulties are caused by the diversity and ambiguity of natural language, which is used to represent the entities in ontology. In our work, a core word - based ontology alignment approach is proposed by applying the knowledge from the fields of Natural Language Processing (NLP) and Information Extraction (IE). The motivation of this proposal is that the label of entity in ontology is usually named with compound word, which combines several single meaningful words. However, usually only a few words of them represent the main meaning of the whole entity. This kind of words is called “core word”. We argue that the finding of core words will contribute significantly to discover semantic correspondences. The recognition process of core words is performed based on pre-defined rules and part of speech (POS) of words. A specific measurement algorithm is proposed based on the recognized core

word and complementary information to compute the semantic similarity. Besides the proposed core word-based method for ontology alignment, to handle diverse situations and enhance matching ability, two matchers at lexical level and structural level are applied by reusing algorithms: edit distance, n-gram and similarity flooding.

Concerning another significant issue in ontology alignment, matcher aggregation has drawn much researcher's attention, since normally multiple matchers will be applied to perform ontology alignment. Although many multi-strategy based methods have been proposed, more automatic and dynamic methods are expected in order to improve the combined matching results and facilitate the aggregation process. In this thesis, an existing analytic approach called Analytic Hierarchy Process (AHP) is applied to learn the weight of each matcher. AHP has been previously used and validated in a very wide range of domains and has some mature applications for weighting computation in certain fields, such as E-learning. It is first proposed to the domain of ontology alignment for matcher aggregation in our work. The learning process is carried out with aids of three similarity indicators that derived from whole-ontology level. This method aims to automate the whole aggregation process and to improve the combined matching results.

Based on the proposed matching and aggregation approaches, a prototype system is implemented in Java for evaluation and further application. This system consists of three major components: pre-processor, matcher and aggregator. It contains approximately 5,000 lines of codes. The experiments are carried out based on reference OAEI benchmarking datasets and comparisons with other approaches. The evaluation suggested that the proposed approaches have obtained promising results and reached expected goals.

To apply the proposed ontology alignment approaches for improving data interoperability at semantic level, an ontology-driven architecture for querying data from multiple relational databases is proposed using the implemented prototype system. This architecture showed certain extendibility, it can be applied to several application scenarios to solve effectively interoperability issues according to specific demands.

1> Ontology Alignment for Enterprise Interoperability

1.1 Introduction

The needs for collaboration among enterprises demand enabling interoperability between their systems. In order to develop interoperability, multiple aspects are involved. From information system point of view, semantic interoperability and technical interoperability are the major ones. Ontology is brought to address these issues due to the advert and rapid development of semantic web, as well as the benefits that ontology brought to address semantic heterogeneity. Although many works have been carried out by using ontologies to tackle semantic interoperability problems, but the results obtained did not reach to a satisfactory level. The work of this thesis

is to adopt ontology as a key component and to seek improvements for the performance of ontology alignment approaches, in order to better contribute to enterprise interoperability from these two aspects (information system and semantic interoperability).

This chapter aims at presenting some basic concepts and definitions relating to enterprise interoperability and ontology alignment as well as the problems tackled and objectives of this thesis. Section 1.2 recalls the Framework for Enterprise Interoperability (FEI) and defines the research scope of this thesis under the FEI. Semantic interoperability is addressed in Section 1.3, including: (i) semantic heterogeneity issues; (ii) relevant approaches for solving semantic interoperability issues, much investigation is focused on ontology-based methods; (iii) the roles that ontology played in contributing to enterprise interoperability. Section 1.4 situates the relation of ontology alignment and federated approach regarding FEI. In Section 1.5, main contributions of the thesis are outlined. Firstly, the main problems and challenges in the domain of ontology alignment are analyzed and the proposals for improving them are presented. Secondly, an ontology-driven architecture by using ontology alignment as major component is introduced briefly. Section 1.6 describes the organization of thesis and Section 1.7 draws some conclusions.

1.2 Enterprise Interoperability and Research Scope

As the economic and industry globalizing and increasing, the collaboration among multiple partners is demanded more than ever. However, there exist many barriers to obstruct the collaboration because the heterogeneities of culture, data and computing techniques. In order to enable collaboration, a key point is to develop interoperability among enterprises. Interoperability is the ability that two systems could interact and communicate with each other.

Definition 1.1 (Enterprise interoperability)
The ability for two systems to understand one another and to use functionality of one another [1].

In enterprises, solutions need to be implemented from multiple aspects to develop interoperability. In Chen et al. [2], Framework for Enterprise Interoperability (FEI) with three dimensions has been defined to represent and structure various issues and concerns (see Figure 1.1). FEI has been published as an international standard (ISO 11354[1]). First the

[1] http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=50417

framework identifies the categories of barriers that are obstacles for achieving interoperability among enterprises: conceptual, technological and organizational. Then four concern points are defined: data, service, process and business, these points should be taken into account when building architectures and solutions. Generic approaches to address interoperability problems are categorized into integrated, unified and federated.

1.2.1 Interoperability Barriers

Conceptual barriers

They are concerned with the syntactic and semantic differences of information to be exchanged. These problems concern the modeling at the high level of abstraction (such as for example the enterprise models of a company) as well as the level of the programming (for example XML models).

Technological barriers

These barriers refer to the incompatibility of information technologies (architecture and platforms, infrastructure, etc.). These problems concern the standards to present, store, exchange, process and communicate the data through the use of computers.

Organizational barriers

They relate to the definition of responsibility (who is responsible for what?) and authority (who is authorized to do what?) as well as the incompatibility of organization structures (matrix vs. hierarchical ones, for example).

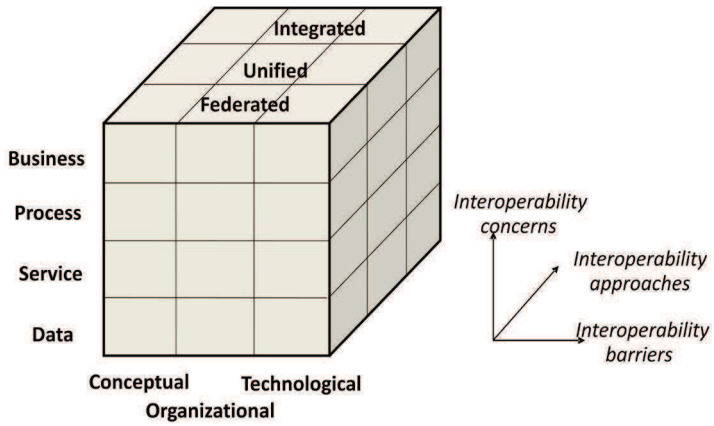


Figure 1.1 Framework for Enterprise Interoperability (FEI)

1.2.2 Interoperability Concerns

The interoperability of data

It refers to make different data models and query languages working together. The interoperability of data deals with finding and sharing information from heterogeneous data sources, and which can moreover reside on different machines under different operating systems and data base management systems.

The interoperability of services

It is concerned with identifying, composing and making various applications function together (designed and implemented independently). The term 'service' is not limited to the computer based applications; but also functions of companies and networked enterprises.

The interoperability of processes

The aim is to make various business processes work together: a process defines the sequence of the services (functions) according to some specific needs of a company. In a networked enterprise, it is also necessary to study how to connect internal processes of two companies to create a common process.

The interoperability of business

It refers to working in a harmonized way at the level of organization and company in spite of, for example, the different modes of decision-making, methods of work, legislations, culture of the company or commercial approaches so that business can be developed between companies.

1.2.3 Interoperability Approaches

Integrated approach

There exists a common format for all models. This format must be as detailed as models. The common format is not necessarily a standard but must be agreed by all parties to elaborate models and build systems.

Unified approach

There exists a common format but only at a meta-level. This meta-model is not an executable entity as it is in the integrated approach but provides a means for semantic equivalence to allow mapping between models.

Federated approach

There is no common format. To establish interoperability, parties must accommodate on the fly. Using a federated approach implies that no partner imposes its models, languages and methods of work. This means that they must share an ontology to map their concepts at the semantic level.

1.2.4 Research Scope

The research work in this thesis is based on Framework for Enterprise Interoperability (FEI). Figure 1.1 suggests that conceptual (semantic) barrier exists through all the enterprise, including the concerns of data, service, process and business. Among all these concerns, the conceptual barrier to data interoperability is most essential and crucial. Because some semantic barriers occurred in concerns of service, process and business is caused by data interoperability barriers. Thus to remove conceptual barrier at data level is significant to contribute to enterprise interoperability at all levels. Concerning the approaches to remove the barriers, federated approach is the most flexible and extensible one, an approach developed from this category is expected. Based on this motivation, the thesis is dedicated to developing data interoperability and removing semantic (conceptual) barriers by adopting federated approach.

Data Interoperability: Information system, as critical part of modern enterprises, has been adopted widely in every professional domain for decades, and the number of new information systems is increasing rapidly. Huge amount of data is stored in these systems via diverse formats including databases, text files and multimedia files. Most of data is isolatedly stored in specific information systems and often is difficult to share even in the same systems, as quite often the right information could not be found when needed. The essential reasons are that semantic heterogeneity obstructs the exchange and the information is not linked in an interoperable way.

Semantic barriers: In enterprise interoperability, semantic heterogeneity is one of the main difficulties that obstruct to exchange information and to use information exchanged. Semantic heterogeneity lays in each aspect of enterprise, namely, data, service, process and business. Different interpretations of same concept and knowledge in different organizations lead to interoperation difficulties.

Federated approach: In modern enterprises, enterprises deal with many changes in a complex environment, such as, new partner joining, market changing. To enhance the competitiveness, flexible and extendable systems are expected to exchange information rapidly and effectively

among enterprises. Federated approach is one promising solution to contribute to such kind of system. Thus this thesis aims to propose a federated approach based on ontology, which is regarded, in this domain, as the main solution to represent semantic and to share concepts.

In the following sections, firstly, semantic heterogeneity, which is the root that causes semantic barriers, is analyzed. Secondly, the relevant works to develop semantic data interoperability are surveyed and classified into two categories: MDR-based and ontology-based. We focus on studying ontology-based approaches. Thirdly, the roles that ontology played for contributing to data interoperability are classified and discussed. And then different modes to use ontology for data interoperability are investigated, in which ontology alignment is taken as federated approach. Concerning ontology alignment, existing challenges and problems are analyzed and some proposals for improving are presented.

1.3 Semantic Data Interoperability

According to He et al. [3], semantic data interoperability refers to *the capability that two software modules or systems can exchange the data with precise meaning, and the receiving party can accurately translate or convert the information carried by the data, including the knowledge, i.e. information and knowledge that can be understood, and ultimately produce an effective collaborative results*. Semantic interoperability stays at the conceptual level, and it concerns both organizational and technical aspects.

To address how to facilitate semantic interoperability in enterprises, at first, the semantic heterogeneity that obstructs semantic interoperability is described in §1.3.1. Secondly, the existing approaches for solving semantic interoperability problems are investigated in §1.3.2. The roles that ontology played are discussed in §1.3.3.

1.3.1 Semantic Heterogeneity

Buccella et al. [4] classified heterogeneity into four categories: (i) structural heterogeneity involves different data models; (ii) syntactical heterogeneity presents different languages and data representations; (iii) systemic heterogeneity involves hardware and operating systems; and (iv) semantics heterogeneity involves different concepts and their interpretations.

In general, the semantic heterogeneity includes three types of concept heterogeneity as follows [5]:

1) The semantically equivalent concepts;
2) The semantically unrelated concepts;

3) The semantically related concepts.

In the first case, different models use different terms to refer the same concept. For example, one system uses *teacher* whereas the other one uses *professor* to address teaching staff. In the second case, the same term is used by different systems to denote completely different concepts, such as *dear* may refer to *expensive* or be used to address a person. In the third case, two concepts have part of semantic relations, and they are not completely equivalent. For instance, *kid* refers to children aged between 3 to 10 years old in ontology *O*, while *child* is used to refer the children aged between 7 to 14 years old in ontology *O'*. They both refer to children, but the meaning is not exactly the same. The semantic of them has part of intersection. The other situations for the third case include different generalization and specification, different definable terms or abstraction, and different conceptualization. To evaluate the semantic relations between them, usually a decimal (0.0 to 1.0) similarity value is used to measure the degree of the semantic relation. 0 means no semantic related relation between two entities, while 1.0 means semantic equivalent concepts.

Regarding this issue in enterprise interoperability, semantic heterogeneity causes conceptual barriers and technical barriers, which obstruct systems interoperations. Taking order management for example, in order to present the serial id of product, there can be many varieties, such as *id*, *sid*, *product id, productId*. When different systems communicate, these different interpretations constitute barriers at both conceptual level and technical level. Conceptual level concerns the understanding of peoples and technical level concerns the specific processing of information systems. The key issue to solve the two problems is to construct a mechanism that is able to find semantic correspondences between different concepts.

1.3.2 Relevant Works

In order to develop semantic interoperability, specific approaches are involved to seek semantic relations via certain interfaces. Figure 1.2 from Yahia et al. [6] illustrated the problems and showed the semantic interoperability properties. Semantic relations (P2) are connected through technical interface (P1) by investigating semantic information, including concept, non mandatory information, semantic bloc and minimal mandatory information (P3).

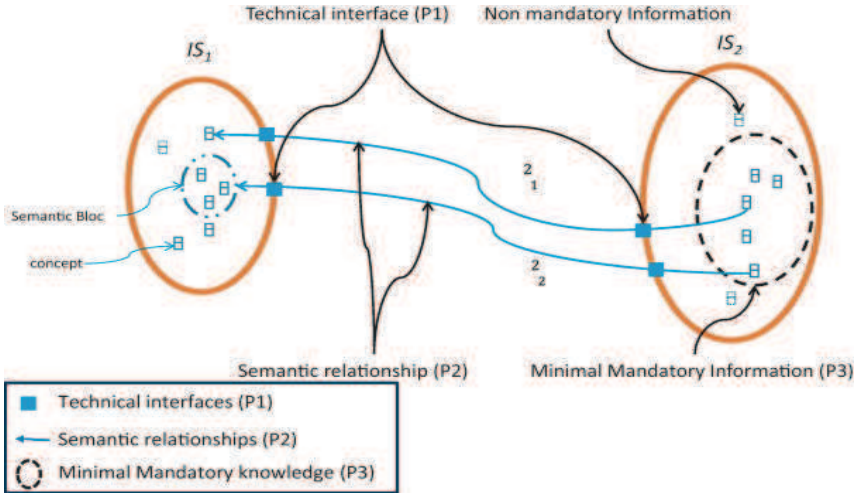


Figure 1.2 Semantic interoperability properties (Yahia et al., [6])

To find the semantic relation, different kinds of approaches are used in accordance with specific goals. According to the research of Vernadat [7], there are two general types of approaches to address semantic interoperability: MetaData Registry (MDR) - based and ontology-based approaches. These two kinds of approaches are investigated in the follows.

1.3.2.1 MetaData Registry (MDR) - Based Approaches

According to ISO/IEC 11179[2], *a Metadata Registry (MDR) is a database of metadata that supports the functionality of registration*. The core function of metadata schema registries is to collect, store and provide reference descriptions of metadata schemata. The examples to apply MDR include enterprise LDAP for users and IT resources metadata, UDDI repositories for web service registries and thesauri. Shukair et al. [8] classified the approaches to adopt MDR as follows:

- **ISO/IEC 11179 13 -** based approaches follow ISO MDR standards and implement the registration of elements from multiple systems, the examples include CORES [9] and DESIRE [10];
- **Dublin Core (DC) -** based approaches reuse and extend DC set. *DC is one of the most influential, domain-independent initiatives in the area of digital resource metadata description* [8]. Some work uses DC as metadata standard, such as, developing e-Government [11] and digital libraries [12];

[2] http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=50340

- **Resource Description Framework (RDF) -** based approaches use RDF[3] as data model to exchange information, for instance, Maali et al. [13] proposed DCAT, which is a RDF-based vocabulary to describe government catalogues and datasets.

1.3.2.2 Ontology - Based Approaches

As the advert and the rapid development of semantic web technologies, many ontology-based tools for semantic integration have been developed. In addition, several working groups of W3C are dedicated to standardizing various techniques that are needed in semantic web, such as RDF, OWL, RDB2RDF[4], SPARQL[5]. This supplies solid technical and theoretical ground for applying semantic web to enterprises. Although semantic web was developed from the domain of Internet and is being applied mostly in the domain of Internet, the semantic web technologies can be adverted to enterprise interoperability and make certain contributions.

Semantic web links the data in a machine-readable way. Ontologies are used to represent the concepts and their relations semantically for a specific knowledge domain. Ontology is the key component to construct semantic web. The ways to apply ontology are diverse, from simple single one to combined ones. In this section, first three basic modes to apply ontology for data interoperability are introduced. Secondly, the relevant works about using ontology for semantic interoperability are overviewed.

Data integration was proposed since decades to address data exchange issues, it has drawn many researchers' attention and obtained promising results [14-16]. Latterly, ontology is widely applied to data interoperability, since semantic heterogeneity became a more significant issue over the structural aspects. Wache et al. [17] surveyed the ontology - based approaches for information integration and summarized three modes of using ontology: single ontology, multiple ontologies and hybrid ontologies approach as shown in Figure 1.3.

For the single ontology approach, a commonly shared ontology is proposed. This ontology contains the common vocabularies for all the data sources in order to share concepts. In this case, besides proposing a new specific ontology, usually, domain ontology and reference ontology are used as this shared vocabulary. Domain ontology concerns a particular area of

[3] http://www.w3.org/RDF/
[4] https://metacpan.org/release/RDF-RDB2RDF
[5] http://www.w3.org/TR/rdf-sparql-query/

knowledge, for instance, ONTO-PDM [18] is an ontology specializing in product design.

Regarding the multi-ontology based approach, there is a local ontology for each data source, and then the links between local ontologies are built in order to make them interoperable. While for hybrid ontologies approach, shared vocabularies are kept to minimize interlinks between local ontologies. The way to construct the links between ontologies is ontology alignment, which seeks the correspondences between entities in ontology.

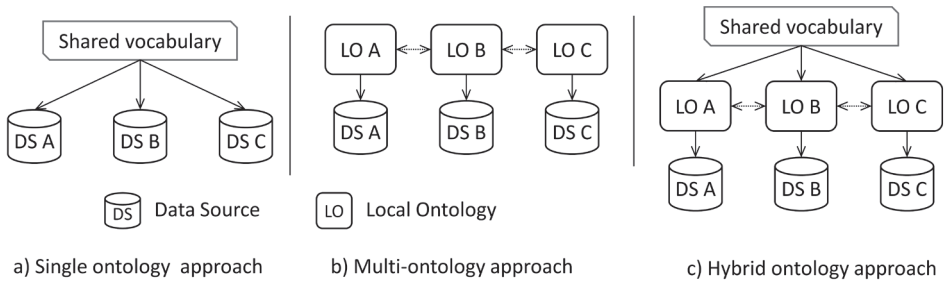


Figure 1.3 Three modes of using ontology for data interoperability

Each mode has its limitations and strengths, single ontology mode is easy to implement but is not adaptive when adding or removing a data source, whereas multi-ontology mode implements costly but has more flexibility with data source changes. The highlight of hybrid ontology mode is that it combines the advantages of both modes and it makes easier to compare different source ontologies due to the shared vocabulary, but it is relatively difficult to maintain.

Some works using ontology alignment to address semantic interoperability issues have been carried out. Martínez-Costa et al. [19] proposed an approach for the semantic interoperability between ISO EN 13606 and OpenEHR archetypes in the domain of health and hospital information systems. Fallahi et al. [20] proposed a hybrid ontology structure to facilitate semantic interoperability for GIS and environment modeling. Quant et al. [21] also applied a hybrid ontology approach to enable interoperability among relational databases (RDB). A local domain ontology is proposed for each RDB, and then a shared global ontology is designed to share common vocabularies between local ontologies.

Lu et al. [22] proposed to use ontology alignment for interoperating between Supply Chain Operations Reference (SCOR) model and product design in the environment of networked enterprise information systems. They proposed a SCOR-ontology based on SCOR model. The product ontology that they used is ONTO-PDM [18]. Ontology alignment is used to discover the semantic relations between the two ontologies.

1.3.3 Roles of Ontology

This thesis focuses on ontology-based approach to address semantic interoperability, because the benefits to adopt ontology are multifold [23, 24], and ontology plays multiple roles in facilitating the semantic interoperability. The main roles are summarized as follows: (i) knowledge representation; (ii) concepts annotation and enrichment, and (iii) serving as mediation media for knowledge sharing.

Knowledge representation

Ontology is used for conceptualizing the concepts in a certain domain, for example, ONTO-PDM [18] presented a product model in manufacturing environment, and Kucuk et al. [25] presented a domain ontology for electrical Power Quality (PQ) called PQONT. Ontology ONTO-PDM aims at facilitating the interoperability among software applications during the physical product lifecycle. PQONT tries to build a set of shared vocabularies in the domain of PQ for different systems. In Naudet et al. [26], an ontology of interoperability (OoI) was described to formalize the concepts and their relations in the domain of enterprise interoperability from a system theory point of view.

Concepts annotation and enrichment

Ontology is adopted to annotate and enrich certain concepts or knowledge. The object to be enriched can be a model or a document or even a concept. The enrichment can be performed in two ways: (i) based on the original object, enrichment is made by creating a new enriched larger object, which is different to the original one. The added information becomes part of the object and is essential. Liao et al. [27] defined a method to add semantic annotations to concepts; (ii) enrichment provides additional information and semantics to the object to be enriched, it is complementary. Zouggar et al. [28] used ontology to develop semantics for enterprise models, in order to understand among different modeling languages. In Lim et al. [29], the authors proposed a methodology to build semantically annotated multi-faceted ontology for product family modeling, and Fernandes et al. [30] used semantic methods to support engineering design innovation.

Serving as mediation media for knowledge sharing

Ontology is used as a representation form to share concepts among heterogeneous systems. Unlike the traditional data integration approaches, ontology-based methods possess the features of higher extendibility and

lower coupling among the existing systems. Rezgui et al. [31] proposed a knowledge-centered approach with ontology to integrate different information portals of European Commission (EC) and British institutions. The work aimed to provide an integrated and semantic-based user interface for accessing documents, which are published by different partners. In their work, ontology was taken as a library of shared concepts with global ontology mode. This ontology is used to eliminate semantic ambiguity among different contexts. Colombo et al. [32] aimed to implement an integrated "intelligent" environment for CAD by using ontology. Since ontology is a kind of semantic representation, the model and data can be denoted like human beings, in order to be understood by computers and eventually to achieve the goal of "intelligence". MAFRA [33] was an ontology-based methodology for integrating different systems. The authors proposed a comprehensive framework to illustrate how to integrate distributed ontologies. The method can be adapted for integrating various knowledge and information sources.

1.4 Federated Approach with Ontology Alignment

This section tries to situate the relation between ontology alignment and the approaches to develop enterprise interoperability, in order to clarity how ontology alignment can be used as a federated approach. Firstly, different ontology mapping modes are investigated in §1.4.1, including ontology integration, ontology merging and ontology alignment. Secondly, the position between different ontology mapping modes and approaches to develop enterprise interoperability are situated and analyzed in §1.4.2.

1.4.1 Ontology Mapping Modes

The key issue to apply ontology for semantic interoperability is ontology mapping. There are different modes of mapping for various purposes: ontology integration, ontology merging, and ontology alignment. Choi et al. [34] gave a comprehensive comparison of them. Ontology alignment seeks correspondences and sets up links between ontologies, while ontology integration and merging will create a new integrated ontology with the information from source ontologies. The illustrations are shown in Figure 1.4 a) and b) respectively.

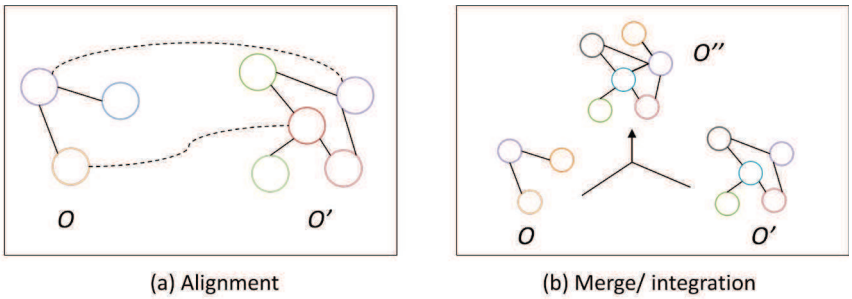


Figure 1.4 a) Ontology alignment and b) Ontology integration and merging

Table 1.1 listed the input and output of different ontology mapping modes. The second column presents the input of ontology mapping and third column shows the output of the mapping results. The input and output reflect the conditions and purposes of each mapping mode. Ontology merging and integration are similar, since for both of them, the output is one single ontology. Regarding the inputs, ontology from similar domains will adopt ontology merging to generate one single coherent ontology that specializes in one specific domain and usage. Ontology integration is used for integrating ontologies that are from less related domains. The generated ontology contains the knowledge that is from both domains. Ontology alignment takes two ontologies, which are supposed to be partially similar and share some common relevant concepts, as input. The output is a set of correspondences and the linked ontologies.

Ontology alignment provides a loose integration between ontologies, because original ontologies keep unchanged and are linked with correspondences. Ontology alignment is a way to implement the federated approach for contributing to enterprise interoperability. The work of this thesis focuses on using ontology alignment to improve current approaches for semantic data interoperability.

Table 1.1 Input and output of ontology mapping

Type	Input 2..* ontologies from	Output
Ontology merging	Similar domains	One single coherent ontology
Ontology alignment	Partially Similar domains	Two or more ontologies that are linked
Ontology integration	Different domains	One single ontology

1.4.2 Ontology Alignment and Enterprise Interoperability

Regarding the three-dimensional FEI framework (see Figure 1.1), the relations between the approaches/barriers and ontology mapping are identified as Figure 1.5 shows. Figure 1.5 illustrated two dimensions:

interoperability approaches and interoperability barriers, while interoperability concerns are omitted. This figure explains using what kind of approaches to remove what kind of barriers. As stated by the roles of ontology for contributing to enterprise interoperability, ontology can aid to remove the conceptual barriers and technical barriers. Concerning the different types of ontology mapping modes, ontology integration and merging are applied to address unified approach, while ontology alignment is in the category of federated approach. Ontology alignment does not require community (in terms of model, data and time) between two systems. It provides flexible and loose connection to develop semantic interoperability. This thesis adopts ontology alignment as a component to develop federated approach to address data interoperability issues at semantic level.

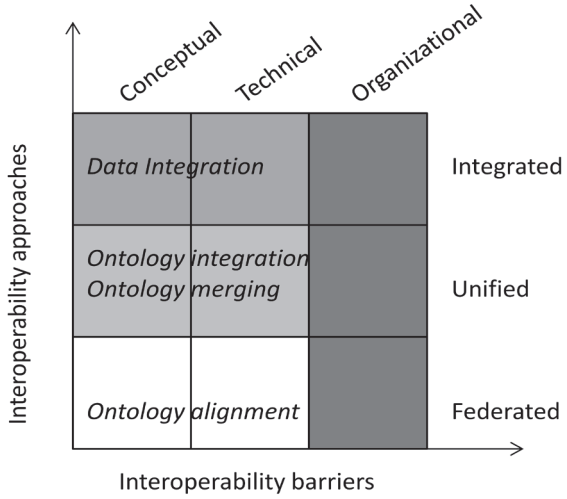


Figure 1.5 Ontology mapping and FEI

1.5 Problems and Contributions of the Thesis

The research in this thesis is carried out in the domain of enterprise information systems and under FEI (sees Figure 1.1). It focuses on addressing data interoperability issues among multiple enterprise information systems. Ontology is brought to this issue as a major component. Ontology alignment, as a federated approach and a way to make data interoperable and sharable, is taken as the major research topic in this thesis. The hypothesis of the research is that the ontologies from enterprises already exist. The creation of ontology is not the concern and beyond of the research scope. The main challenges and our proposals to improve existing issues in current research of ontology alignment is presented in §1.5.1. A concrete architecture for developing semantic data interoperability is

introduced in§1.5.2. In §1.5.3, main contributions of the thesis are presented.

1.5.1 Major Challenges of Ontology Alignment and the Proposals

Shvaiko and Euzenat [35] stated the ten challenges for ontology matching:

- Large-scale ontology evaluation;
- **Performance of ontology matching techniques;**
- Discovering missing background knowledge;
- Uncertainty in ontology matching;
- **Matcher selection and self-configuration;**
- User involvement;
- Explanation of matching results;
- Social and collaborative ontology matching;
- Alignment management;
- Reasoning with alignments.

The first five statements are concerned directly to the matching problem, while the last five items are related to the other parts of the matching systems, such as user involvement and alignment management. In our research work, the focus is made on *performance of ontology matching techniques* and *matcher selection and self-configuration*, which are two key issues of the challenges, to propose some approaches for improving ontology alignment.

Performance of ontology matching techniques

Ontology alignment involves many aspects to discover the correspondences because of the complexity of the ontology itself. The matching tasks can be performed using different approaches based on one specific aspect of ontology, for instance, graph-based approaches or lexical-based approaches.

Most of current matching techniques try to measure the similarity from lexical and structural level of source ontologies in order to discover the correspondences. It is argued that the semantic matching is more important than the two above aspects, because the final goal of ontology alignment is to find semantic correspondences in enterprise interoperation, especially for the ontologies that are retrieved from enterprise systems, they contain many labels and comments that presented in natural languages. Therefore, the focus of this work is on measuring the semantic similarity of entities.

In this thesis, we propose an approach to perform ontology alignment by studying the semantics of entities. This approach focuses on finding correspondences from semantic level. It measures the semantic

similarity of two entities based on core word, which represents the main meaning of a compound word or short phrase, namely, the entities of ontologies. The core words are recognized by pre-defined patterns. In order to adapt diverse situations, especially when no core words could be identified, two other matchers are also proposed at lexical level and structural level. These two matchers reuse and adapt the existing similarity measurement algorithms. The three proposed matchers could complement each other from different levels, so that to find maximally the correspondences.

Matcher selection and self-configuration

Ontology alignment seeks correspondences at different levels of source ontologies. In this process, multiple matching techniques (matcher) are usually involved. Therefore how to choose and aggregate these matchers becomes an issue in ontology alignment. The purpose of this step is to aid generating the best final correspondences automatically and dynamically via selection and aggregation of matchers.

Some existing works apply different strategies in order to adapt the selection of matchers. It is argued that strategy-based matcher selection is not flexible enough to adapt the various situations and manual work and adjustments need to be involved. An automatic method is expected to improve this issue, meanwhile to improve the precision of final aggregated results.

In this thesis, an analytic approach based on Analytic Hierarchy Process (AHP) is proposed to aggregate the matchers. The motivation is to assign the weight by evaluating and balancing the importance of each matcher considering the various factors. This approach assigns the weight of each matcher automatically and dynamically according to three indicators, which reflects the similarity of two ontologies from one specific aspect at whole-ontology level.

The proposed alignment approaches are described in Chapter 2 and analytic aggregation approach is presented in Chapter 3. Implementation and testing of the proposals are carried out and presented in Chapter 4.

1.5.2 Ontology-Driven Architecture to Build Semantic Information Layer

Ontology alignment, as a method to find semantic correspondences, needs to be combined with the other processes and components so that to develop data interoperability. With this idea, an ontology-driven architecture is proposed for building semantic information layer (SIL) using the proposed ontology alignment approach as one of the key components. SIL refers to an

information layer with semantic representation, interface to interact with upper level applications and access to lower data source. Three main steps are involved to develop SIL: ontology extraction, ontology enrichment and ontology alignment. The objective is to query data from multiple Relational Database (RDB) systems that are used in enterprises, so that to enable enterprise data interoperability at semantic level. The architecture focuses on RDB as data sources, since RDB is the main data storage type that is being widely used in enterprises.

This architecture is elaborated in Chapter 5, including the main components of SIL and main steps to build it. Relevant methods and techniques are fully described. An illustrative example is given to show the construction of this architecture by applying the proposed ontology matching and combination approaches. Potential applications of this architecture to contribute to enterprise interoperability are discussed.

1.5.3 Main Contributions

The following points are considered as the main contributions of this thesis:

- Present an improved ontology alignment approach to facilitate federated enterprise interoperability among different enterprise information systems focusing on data interoperability at semantic level;
- Propose a novel core word-based similarity measurement method for ontology alignment;
- Develop a new analytic matcher aggregation approach to allow combining automatically the proposed matchers and improving the combined the results;
- Propose an ontology-driven architecture for query data from multiple relational databases by applying the proposed ontology alignment approaches.

1.6 Organization of Thesis

The thesis consists of five chapters. It is organized based on Software Development Life Cycle (SDLC) and Scientific Method (SM) as shown in Figure 1.6. From the perspective of SDLC, Chapter 1 corresponds to the step of “requirement specification”. Chapter 2 and Chapter 3 correspond to the phase of “design”. “Implementation” and “Testing” are detailed in Chapter 4. Chapter 5 applies the proposed approaches. From the point of view of scientific method, “Observation/Research” is carried out in Chapter 1. Chapter 2 and Chapter 3 correspond to “Hypothesis” and “Prediction”. “Experimentation” and “Conclusion” are completed in Chapter 4, Chapter 5 (and “General conclusion”) respectively.

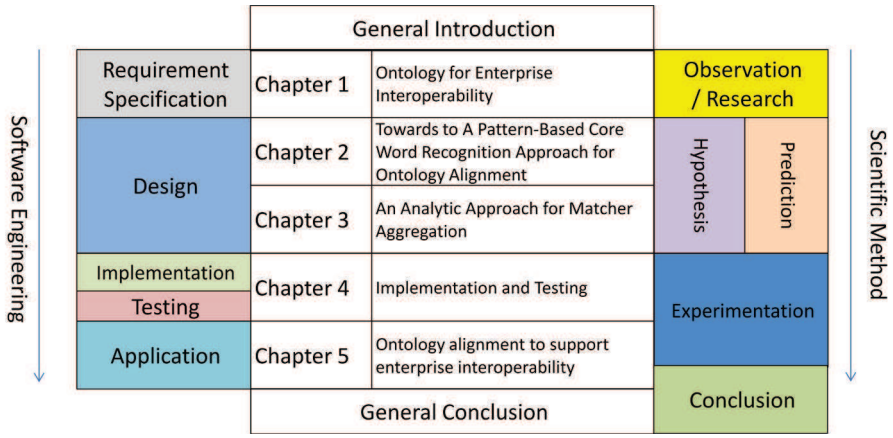


Figure 1.6 Organization of thesis

- **General Introduction** gives a brief introduction of the thesis, mainly including research problem, background and main proposals;
- **Chapter 1** states the research background, problems and challenges. The proposal and main contributions are also outlined;
- **Chapter 2** proposes a core word - based approach with pattern recognition for ontology alignment. Two other matchers from lexical level and structural level are also presented;
- **Chapter 3** proposes an analytic matcher aggregation approach for combining the matching results obtained by the proposed core word-based matcher and the other two matchers;
- **Chapter 4** implements the proposed alignment and aggregation approaches. Experiments are carried out for evaluating and validating the proposed approaches;
- **Chapter 5** applies the proposed ontology alignment approaches to improve enterprise interoperability by proposing an ontology-driven architecture for querying data from multiple relation databases;
- **General conclusion** draws some main conclusions obtained by the research work of the thesis and points out some future perspectives.

An overview (see Figure 1.7) is presented to illustrate the problem statement and the main relations among key parts of the thesis. In summary, ontology contributes to enterprise interoperability, particularly for removing conceptual barriers and technical barriers from semantic interoperability point of view. Ontology alignment, as a way of ontology mapping, enables data interoperability among different systems at semantic level (Chapter 1). Based on current works and in response to existing challenges of ontology alignment, some improvements are made to propose a core word-based ontology alignment approach (Chapter 2) and an analytic matcher aggregation method (Chapter 3). A software prototype is implemented to validate the proposed approaches and for further

application. The testing is done with benchmarking tests (Chapter 4). One application is proposed to facilitate enterprise interoperability by applying the proposed approaches and using the implemented systems (Chapter 5).

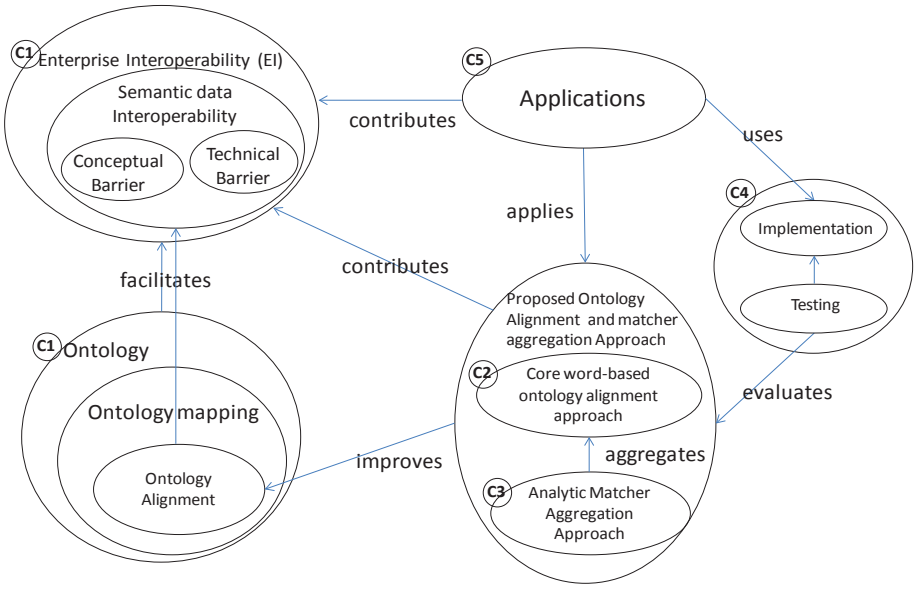


Figure 1.7 Main issues and their relations of the research

1.7 Conclusion

In a more and more globalised world economy environment, developing interoperability is becoming a key success factor for many enterprises. Although many works have been done, enterprise interoperability is still not developed at a satisfactory level due to many reasons. Among various solutions approaches, federated approach is considered as most promising one in a constantly changing and dynamic enterprise collaboration environment.

This chapter stated the research problem in the domain of enterprise semantic interoperability and defined precisely research scope under the framework of the FEI. Relevant background and survey have been investigated. Semantic problems encountered in data interoperability are seen as a fundamental issue that also affects other interoperability concerns. In particular, existing approaches mainly focused on developing integrated and unified approaches, few solutions were proposed through federated one. Consequently this thesis aims at bring ontology-based approaches to address and contribute to semantic data interoperability, more specifically, using ontology alignment as a federated enterprise interoperability solution. To this aim, some proposals have been made for improving ontology alignment based on current relevant works, in order to improve ontology alignment performance, and thus contribute to enterprise interoperability.

2> Towards A Pattern-Based Core Word Recognition Approach for Ontology Alignment

2.1 Introduction

Ontology alignment, as a crucial research issue for data integration and semantic interoperability, seeks the correspondences between two ontologies. To find correspondences, the matching tasks need to be performed from different levels of source ontologies, namely, lexical, structural and semantic. The matching algorithms in each category focus on certain aspects and try to find correspondences from this level. Usually multiple matchers are chosen and applied to improve the accuracy and matching ability. Two key points that need to be considered when designing

ontology alignment approaches are: (i) how to design the matchers, and (ii) how to select and combine the multiple matchers.

In this thesis, three matchers are designed concerning three levels of ontology alignment. The main matcher proposed is a core word-based matcher, which is based on natural language principles and tries to learn the semantic relations from the labels of entity in ontology. It measures the semantic similarity based on the core word, which represents the primary meaning of a compound word or short text. The core words are recognized based on the pre-defined patterns and part of speech (POS) of words. A specific algorithm to compute the value of similarity is proposed based on the recognized core words and complementary information.

Besides the core word -based matcher, the other two matchers are proposed at lexical level and structural level to enhance the matching ability and adapt to diverse situations. At lexical level, the algorithms used are edit distance [36] and n-gram [37] model. At structural level, similarity flooding algorithm (SFA) [38] is applied.

To aggregate the multiple matchers dynamically and automatically, an analytic approach is proposed to learn the weight of each matcher based on AHP and indicators. The aggregation part about combining different matchers is presented in Chapter 3.

This chapter describes the proposed ontology matching approaches. Section 2.2 defines relevant definitions and states the matching problems. Section 2.3 overviews some relevant works to multi-matcher based approaches, as well as the structure of the proposed approach and each matcher. The proposed pattern-based core word matching approach is described in Section 2.4. Non-semantic based matchers, which are based on edit distance, n-gram and similarity flooding algorithm, are described in Section 2.5. Section 2.6 draws some conclusions.

2.2 Ontology Alignment: Concepts and Problem Statement

2.2.1 Relevant Definitions

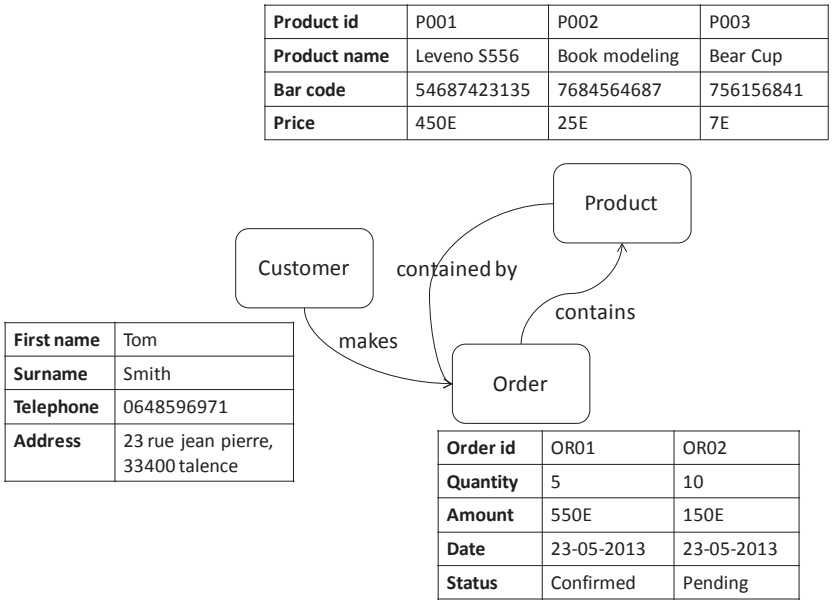
In this section, the definitions of basic concepts and explanations relating to the problem of ontology alignment are presented. The definitions help to clarify the problems and the research scope. The concepts include ontology (Definition 2.1), entity (Definition 2.2), correspondence (Definition 2.3), similarity (Definition 2.4) and ontology alignment (Definition 2.5).

Definition 2.1 (Ontology)
An ontology is defined as a formal, explicit specification of a shared conceptualization [39]. In this definition, *formal* indicates that ontology is machine-readable, which can be processed by computers. *Explicit* refers to that all the concepts and relations in an ontology are defined explicitly and directly. An ontology is described formally as a 6-uple:

$$\{C, P, H^c, H^p, A, I\},$$

including a set of concepts C and a set of properties P. The hierarchy relationship between concepts and sub-concepts is denoted by H^c, in the same way, H^p denotes the hierarchy relations between properties and sub-properties. A is a set of axioms, while I is the set of instances of concepts and properties.

An example of ontology is illustrated in Figure 2.1. It is not represented in any formal ontology languages. In the figure, we can see the concepts: *customer, product* and *order*, as well as the relations among them: *makes, contains* and *contained by*. They refer to *customer makes order*, *order contains product* and *product is contained in the order.* The table beside each concept elaborates the attributes of concepts and their instances. For instance, *product* has four attributes: *product id, product name, bar code* and *price.* The following three columns are the instances of *product*. This figure demonstrates an overall idea about how ontology looks like and what it can represent.



Product id	P001	P002	P003
Product name	Leveno S556	Book modeling	Bear Cup
Bar code	54687423135	7684564687	756156841
Price	450E	25E	7E

First name	Tom
Surname	Smith
Telephone	0648596971
Address	23 rue jean pierre, 33400 talence

Order id	OR01	OR02
Quantity	5	10
Amount	550E	150E
Date	23-05-2013	23-05-2013
Status	Confirmed	Pending

Figure 2.1 An example of ontology about order management

There are many ways to represent ontology, Song et al. [40] described the stack of ontology languages, among which, Ontology Web

Language (OWL) [6] is a standard semantic web ontology language developed by W3C. OWL is derived from the DAML and built upon the RDF. It has strong support for expression of web ontology and powerful engine to represent semantics. Besides OWL Full, there are two specific subsets: OWL DL (Description Logic) and OWL Lite. OWL DL was developed for supporting existing description logic and supplying a subset of language that includes computational properties for reasoning systems. OWL Lite was designed for simple implementation to provide users with a functional subset at the beginning of using OWL. In this thesis, the ontology representation and the matching carried out are based on OWL.

Definition 2.2 (Entity)
Entity of ontology is the basic element for composing ontology.

From the formal representation of ontology (see Definition 2.1), we can see that the main entities in ontology (OWL) are classes, properties, individuals, axioms and hierarchies. The first three types of entities are the main objects of study in ontology alignment. They are elaborated in following part and some examples of them in RDF are given corresponding to the example in Figure 2.1.

```
<xmlns:example="http://www.ims-bordeaux.fr/grai/example#">
```

Classes: A class defines a group of individuals that belong together because they share some properties[7]. In the example of Figure 2.1, *product, customer* and *order* are classes.

```
<Class rdf:about="&example;Customer"/>
<Class rdf:about="&example;Order"/>
<Class rdf:about="&example;Product"/>
```

Construct *rdfs:subClassOf* is used to represent the hierarchy between classes. For instance, *VIP* is used to refer to some very important customers who have priorities. Therefore *VIP* is a sub class of *customer*.

```
<Class rdf:about="&example;VIP">
    <rdfs:subClassOf rdf:resource="&example;Customer"/>
</Class>
```

Property: The relations between classes or between class and data value are denoted by properties. There are two types of properties: object property and datatype property. Object property is used to denote the relation between classes, such as, *makes, contains* and *contained by*.

[6] http://www.w3.org/TR/owl-features/
[7] http://www.w3.org/TR/2004/REC-owl-features-20040210/#Class

Datatype property is used to link classes and data values, for example, *product* has *product id*. *Product id* is a string value, such as, *P001*.

```
<ObjectProperty rdf:about="&example;contains">
   <rdf:type rdf:resource="&owl;FunctionalProperty"/>
   <rdfs:domain rdf:resource="&example;Order"/>
   <rdfs:range rdf:resource="&example;Product"/>
</ObjectProperty>
<DatatypeProperty rdf:about="&example;product_id">
   <rdf:type rdf:resource="&owl;FunctionalProperty"/>
   <rdfs:domain rdf:resource="&example;Product"/>
   <rdfs:range rdf:resource="&xsd;string"/>
</DatatypeProperty>
```

Individual: Individual is the instance of a class and the property used to relate individuals. For instance, *Jack* is an individual of *Customer*, and *Jack makes* an order *OR01* that is an instance of *Order*.

```
<NamedIndividual rdf:about="&example;Jack">
    <rdf:type rdf:resource="&example;Customer"/>
    <example:makes rdf:resource="&example;OR01"/>
</NamedIndividual>
```

Definition 2.3 (Correspondence)
Given two ontologies *o* and *o'* with associated entity languages O_L and $O_{L'}$, a set of alignment relations Θ and a confidence structure over Ξ, a correspondence *c* is a 5-uple:

$$\{id, e, e', r, n\},$$

such that *id* is a unique identifier of the given correspondence, $e \in Q_L(o)$, $e' \in Q_{L'}(o')$, $r \in \Theta$ and $n \in \Xi$ [41]. An example format to present correspondence in implementation is as follows:

```
<entity1 rdf:resource="http://ekaw#Document" />
<entity2 rdf:resource="http://openconf#Text" />
<measure rdf:datatype="http://www.w3.org/2001/ XMLSchema#float">0.9</measure>
<relation>=</relation>
```

The types of relation *r* are diverse, which can be hierarchal, equivalent, etc. Equivalent relation is the focus in ontology alignment, while the other types of relations usually are for some specific needs. For example of hierarchal relations, they can be sub-class, super-class or is-part-of. The relations can also be some very specific relation for a particular purpose, for instance, *happened_at* is sought to link the accidents and locations between two accidents report. In the work of this thesis, the

focus is made on discovering equivalent relations with similarity-based approaches. The other kinds of relations are not considered.

The computation methods of confidence *n* vary depending on the approach and the type of relation. Similarity-based approaches try to discover the correspondences by measuring the similarity between entities from different levels. In this thesis, the value of confidence *n* is the similarity between entities.

Definition 2.4 (Similarity)
A similarity $\sigma : o \times o \rightarrow R$ is a function from a pair of entities to a real number expressing the similarity between two objects [41] such that:

$\forall x, y \in o, \sigma(x, y) \geq 0$ (positiveness)
$\forall x \in o, \forall y, z \in o, \sigma(x, x) \geq \sigma(y, z)$ (maximality)
$\forall x, y \in o, \sigma(x, y) = \sigma(y, x)$ (symmetry)

The approaches for calculating similarity vary depending on the purposes and algorithms. In ontology alignment, the measurements of entity similarity are performed from several levels: lexical, structural and semantic. Each algorithm measures the similarity from certain aspects. In this thesis, three matchers (see Section 2.4 & 2.5) with different algorithms are proposed to measure the similarity.

Definition 2.5 (Ontology Alignment)
Given two ontologies *o* and *o'*, an alignment *A* is made up of a set of correspondences between pairs of entities belonging to *QL(o)* and *QL'(o')* respectively.

Figure 2.2 shows an illustration of the main parts in ontology alignment. Entities *e* and *e'* are from ontology *O* and *O'* respectively. Alignment *A* is made up of the discovered correspondences *c* between ontology *O* and *O'*.

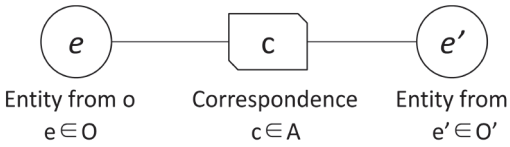


Figure 2.2 Relation between entity and correspondence

2.2.2 Matching Problem Statement

The purpose of ontology alignment is to find the correspondences. Matching process is the main step applied to find correspondences between entities of ontologies. As shown in Figure 2.3, the matching process takes a

pair of source ontologies O and O' as input. Parameters (e.g. threshold) and external resources (e.g. lexical database) are optional input depending on the matching approaches. Matching process uses specific algorithms and generates the alignment A.

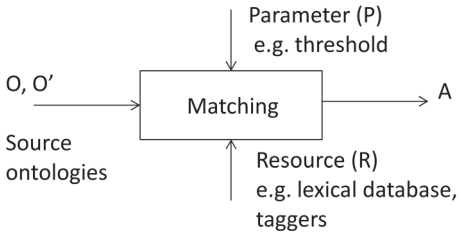


Figure 2.3 Ontology matching process

Concerning the features of ontology, the matching process can be preceded at different levels of source ontology. Three levels are summarized and presented in Figure 2.4. Each level emphasizes on one aspect, almost all the existing techniques can be categorized into a specific level.

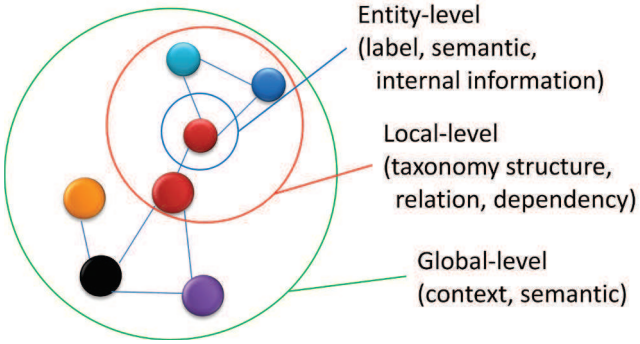


Figure 2.4 Three levels for ontology alignment

- At entity level, the class itself is treated as the object of study; the label, comment and internal information of it are investigated. The mostly used techniques are string metric [42] and string similarity, as well as domain, property and data type comparison [43].
- At local level, the objects and the relations linked to the studied entity are taken into account, such as, graph-based methods [44], and taxonomic-based methods. Directly linked entities and the studied entity compose into a local group, this group of objects is taken as object of study.
- At global level, the whole ontology is regarded as a context and environment. The relation and affect between objects, as well as the studied object are investigated. Machine learning and artificial neural network [45] are some methods applied at this level.

2.3 Related Work and the Proposals

2.3.1 Related Work

In the research domain of ontology alignment, some basic matching techniques have been proposed and developed. The authors, including Euzenat and Shvaiko [41], Granitzer et al. [46] and Alasoud et al. [47], gave comprehensive introduction and comparison of different basic matching techniques and their applications. In this section, the focus is to investigate the multiple matchers-based approaches in order to see how the matchers are designed and combined. Table 2.1 listed the surveyed approaches.

Li et al. [48] proposed an approach called Risk Minimization based Ontology Mapping (RiMOM). RiMOM applied different strategies to match ontologies according to two similarity factors. The used strategies are classified into two categories: linguistic-based and structure-based. For linguistic-based strategies, RiMOM used edit-distance and vector distance, while for structure-based strategy, it adapted similarity flooding algorithm to match. It proposed two similarity factors: label similarity *F_LS* and structure similarity *F_SS*, which are computed from conceptual and structural levels of source ontologies. The weight of each matcher is computed based on these two factors. A strategy is applied to choose which matcher to use according to an experimental threshold value.

Pirro and Talia [49] proposed and implemented a system called User Friendly Ontology mapping environment (UFOme). It combined multiple matchers based on strategies and focused on taking systematic requirements into account, namely, functional and user point of views. We investigated the part about matchers in UFOme. Four matchers are used: lucene matcher, string matcher, WordNet matcher and structural matcher. Two affinity coefficients are proposed: lexical affinity coefficient *La* and structural affinity coefficient *Sa*, which are concluded from source ontologies. The weight is calculated based on the two affinity coefficients with a heuristic function.

Mao et al. [50] proposed a concept "harmony" *h* to estimate the importance and reliability of similarities and used it as the weights. Tu and Yu [51] first calculated the credibility of each matcher and then used this credibility as weight of each matcher. Huang [45] applied an artificial neural network approach to learn the weights from training data.

Table 2.1 Investigation of multiple matchers-based ontology alignment approaches

Matching techniques	Factors	Weight aggregation
RiMOM [48] -Edit distance -Vector distance -Similarity flooding	Label similarity factor: $F\_LS = \frac{\#ide\_conc\_l + \#ide\_prop\_l}{\max(\lvert C_1 \rvert + \lvert P_1 \rvert, \lvert C_2 \rvert + \lvert P_2 \rvert)}$ Structural similarity factor: $F\_SS = \frac{\#nonl\_conc + \#nonl\_prop}{\max(\#NC_1 + \#NP_1, \#NC_2 + \#NP_2)}$	$w_{name} = \frac{F\_LS}{\max(F\_LS, F\_SS)}$ $w_{vec} = \frac{F\_SS}{\max(F\_SS, F\_SS)}$ $sim = \frac{w_{name}\sigma(sn) + w_{vec}\sigma(sv)}{w_{name} + w_{vec}}$
UFOme [49] -Lucene matcher -String matcher -WordNet matcher -Structural matcher	Lexical affinity coefficient : $L_a(O_s, O_t) = \frac{\#common\_entities\_l}{\min(\lvert S \rvert, \lvert T \rvert)}$ Structural affinity coefficient : $S_a(O_s, O_t) = \frac{\#common\_entities\_s}{\min(\lvert S \rvert, \lvert T \rvert)}$	$w_l = \frac{e^{\eta L_a} - e^{-\eta L_a}}{e^{\eta L_a} + e^{-\eta L_a}}$ $w_s = \frac{e^{\psi S_a} - e^{-\psi S_a}}{e^{\psi S_a} + e^{-\psi S_a}}$
PRIOR+ [50] -Name similarity -Edit distance -Profile similarity -Structural similarity	-NA	$h = \frac{\#s\_\max}{\min(\#e_1, \#e_2)}$, h: harmony $sim = \frac{\sum_k h_k \times Sim_k(e_{1i}, e_{2j})}{n}$
CMC [51] -Credibility prediction -Multiple matchers	-NA	Mean square error: $MSE = E_F[(sim - sim_{ac})^2]$ Credibility: $c = e^{-C \times MSE}$ $similarity = \sum c \cdot sim / \sum c$
SFS [45] -Similarity in concepts names(s1),properties(s2), and relationships(s3) -Artificial neural network	-NA	$\sum w_i = 1$, w_i is initialized randomly, adjusted via learning $sim = \sum_{i=1}^{3}(w_i s_i)$
YAM++ [52] Terminological matcher Extensional matcher Similarity Flooding	NA	Multiple strategies -Double-Hungarian -Dynamic weighted aggregation
CSA [53] edit distance, WordNet TF-IDF	-NA	Cluster-based aggregation

Akbari and Fathian [54] also proposed a combined approach with lexical and structural matchers. The weights are set manually according to experiments, the weight for lexical matcher is α =0.4 and for structural matcher is β =0.6. Xu et al. [55] proposed a metric called "*differentor*" to integrate different similarity measurement results obtained by different matching techniques. The weights are computed based on this metric. The

weights are at entity level, which means that each pair of matched entities has a different weight. The matching algorithms proposed to use are lexical, structural, extensional and relation similarity.

YAM++ [52] adopted several matchers from different levels. For terminological matchers, they include name metrics, label metrics and context metrics. Extensional matcher tries to find more correspondences based on the basic terminological matcher. At structural level, similarity flooding is reused by them. A dynamic multiple strategies weighted aggregation approach is used to finalize the results. CSA [53] proposed a cluster-based similarity aggregation approach, including five basic similarity measurement techniques. The similarity aggregation is based on analysis of each similarity matrix of each basic measure, so that to find which one is actually effective for the alignment.

How to design?

For each approach listed in Table 2.1, it tries to adopt several matching techniques, which complement each other. In this thesis, three matchers from three levels (see Figure 2.4) of source ontologies are proposed. The three levels are lexical, structural and semantic. The three levels are different from the existing categories of matcher selection. It is argued that this classification can cover the basic matching issues and approaches. Each of them can discover certain mappings from a different perspective. From the three levels, the matchers can discover relatively complete correspondences with higher accuracy. Especially, the thesis focuses on seeking correspondences from semantic level by studying the semantics of entities in ontology.

How to combine?

The matching usually adopts several techniques and algorithms. An issue is how to combine the different matching results to generate the final correspondence in order to improve the results. The above surveyed methods are weighted-mean based, the final goal and key point is to obtain the weights. From the investigation of these multiple matchers-combined approaches, there are mainly two kinds of methods to decide the weights:

- Based on certain factors, which are obtained from source ontologies, such as RiMOM and UFOme, the factors are not utilized directly as weights, further computation is required.
- Based on certain variables that are defined via a specific method, and then these variables are used directly as weights, such as PRIOR+, CMC and SFS.

The methods to decide the weights vary depending on the matchers, but the assignment of weights should be able reflect correctly and precisely the importance of each variable. Additionally, an approach with automatic and dynamic aggregation process is expected to facilitate the work with high flexibility. In our work, an analytic approach based on Analytic Hierarchy Process (AHP) is proposed to learn the weight of each matcher automatically and dynamically with three indicators, which are learned from the source ontologies. This approach allows generating the weights, which could balance the important factors according to the specific source ontologies. It will assign higher weights for important matchers after evaluating the importance of each matcher with the aids of these indicators. This part will be elaborated in Chapter 3.

2.3.2 Overview of the Proposal

Figure 2.5 overviewed the structure of the proposal in this thesis, including the main components and the processes. A pair of ontologies is taken as input and final correspondences are obtained as output. *Pre-processor* processes the source ontologies, including analyzing each entity and tokenizing the labels. The processed ontologies are sent to matchers. One or more matcher(s) will be chosen according to a pre-defined selection process and the features of the entities to be matched. *Matchers* perform the matching task and generate intermediate correspondences. *Aggregator* combines the results obtained by the matchers and generates the final correspondences.

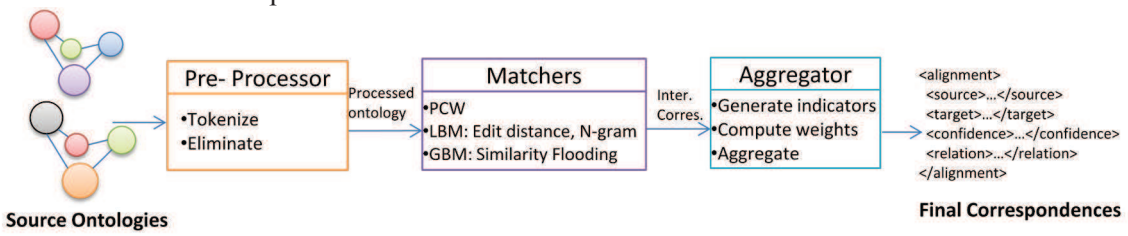


Figure 2.5 General structure of main components of proposed approach

The proposed matchers are classified as two types: semantic-based and non-semantic based. Semantic-based matcher seeks the correspondences from semantic level of entities to be matched, while non-semantic matcher tries to find the correspondences from lexical and structural aspects of ontologies to be matched. The matchers are elaborated in Section 2.4 and Section 2.5 respectively.

- **Pattern-based Core Word (PCW)** is a semantic-based matcher. It tries to find correspondences by investigating semantics in the labels of

entities. In the measurement, PCW adopts core word as fundamental component to measure the semantic similarity.

- **Lexical-Based Matcher *(*LBM)** adopts Jaro-winkler [28] edit distance (ED) and n-gram (NG) model [37]. The selection of which algorithm to use depends on the constraint of label's length. LBM seeks the lexical similarity matching from entity level.
- **Structure-Based Matcher (SBM)** applies similarity flooding algorithm [38] to learn the similarity between entities at a global level.

The *matchers* are elaborated in the rest of this chapter. An analytic approach based AHP is proposed for *aggregator* and is elaborated in Chapter 3. *Pre-processor* is presented in implementation part of Chapter 4.

2.4 Pattern-based Core Word (PCW) Similarity Measurement

2.4.1 Background and Overview

"NLP (Natural Language Processing) strives to enable computers to make sense of human language". NLP has been proposed and studied more than half a century. It seeks ways to make computers understand human natural languages. The input resources could be speech, text and multimedia. In the domains of Artificial Intelligence (AI) and Human-Computer Interaction (HCI), NLP is a major research topic. In NLP, there are many research issues involved. Concerning identifying core word in ontology, a few topics are involved: Information Extraction (IE) and Named Entity Recognition (NER).

IE refers to extracting structured information from information sources automatically. The extraction process respects to certain pre-defined rules. NER is a subtask of IE. NER tries to find and recognize the atomic elements in text. For instance, *the book title* will be recognized as *the (article) book (noun) title (noun)*. The recognition rules are various, in this example, it is recognized by part-of-speech (POS) of words.

In the domain of enterprises information system, most ontologies are created in natural languages. The labels used for naming entities are alike natural language. Normally, they consist of several single meaningful words. These compound words or short phrases focus on expressing one core meaning, unlike the normal complete sentence, which may intend to express several meanings. Therefore, to find out which word(s) represent(s) the main meaning is helpful to understand the semantics. This kind of words is called core word. A core word-based approach for measuring semantic similarity is proposed based on this motivation.

Definition 2.1 (Core word)
Core word is a word or a set of words that is extracted from a compound word or short text and able to represent the main meaning of the compound word, namely, only with these words, the meaning of compound word can be understood without ambiguity.

The hypothesis to apply the method is that the labels of entities in ontology should be alike natural language. For the situation with randomly generated strings and less meaningful compound words, the method is less applicable. The thesis focuses on single core word patterns.

Some related works in this area are listed in Table 2.2. Muslea [56] investigated the different extraction patterns in information extraction. The authors of [57-59] applied patterns to extract information from free text and documents. In Ceausu [57] and Sari et al. [59], the patterns are focused on specific information, such as, the date and location, which are important in accident report. Maynard et al. [58] used patterns to extract and create ontology from free text.

Table 2.2 Investigation of IE and NER approaches

Author(s)	Type	Pattern recognition	Extraction source	Application
Muslea, 1999 [56]	survey	-	-	-
Ceausu et al., 2007 [57]	framework	POS-based	Accident report	Text categorization Accident report
Maynard et al., 2009 [58]	tool	Hearst pattern; Lexical-syntactic pattern; Contextual pattern ;	Free text	Ontology extraction ; ontology creation ;
Sari et al., 2010 [59]	method	Date and time; Location; Accident effect ;	Free text, document; Structured documents	Creating extraction pattern
Ritze et al., 2008 [60]	method	Class by Attribute type pattern	Ontology	Detecting complex correspondence
Svab-zamazal et al., 2011 [61]	theory	NER	OWL ontology	Ontology matching

Ritze et al. [60] and Svab-Zamazal et al. [61] adopted patterns to perform ontology matching for discovering complex correspondences, which are relevant to the research domain of this thesis. They defined a set of patterns from one or several related entities in ontology and use the patterns to find correspondences. The patterns are learned from the mostly used forms when creating ontology. In this thesis, the patterns are recognized based on POS and linguistics. The difference is that the purpose

of obtaining patterns is not to use them for finding the matching directly, rather a way to find the core word. And then, the core word is used for discovering semantic correspondences.

A process (see Figure 2.6) is proposed for measuring the similarity confidence between two compound words. A pair of compound words or short phrases is as input. First the stop words and superfluous information are eliminated from the label, and then the label is tokenized into several single words. With pre-defined patterns, the short text is recognized into each category. In this process, POS tagger and grammar parser are applied. At last, the recognized pattern and core word will be used to measure the similarity. Details are elaborated in following sections.

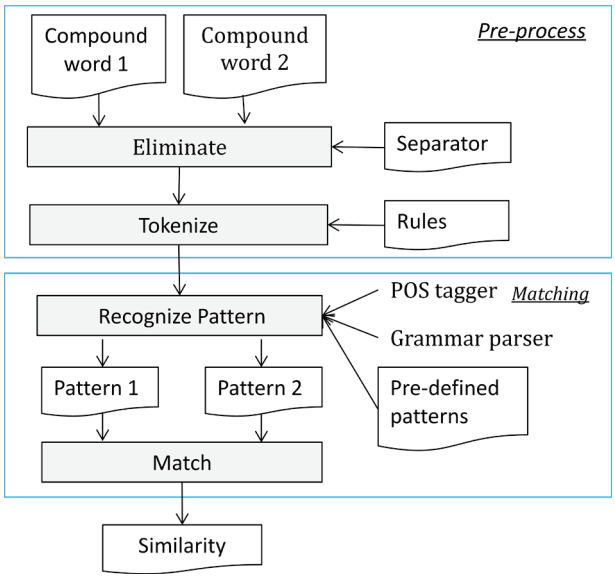


Figure 2.6 Process of core word recognition

2.4.2 Pattern Recognition and Core Word Identification

The creation of labels of entities in ontology commonly follows some specific rules. Usually verb-based labels are used for labeling object property (relation), such as, *hasName* and *applyTo*. Noun-based labels are used for labeling class and data property. For instance, *conferenceMember* and *blackBook* are examples of class, while *title* is an example of data property. From this perspective, certain patterns could be recognized from labels of ontology. In our approach, the recognition process is based on part-of-speech (POS).

The types of POS used in the approach are listed in Table 2.3. To tag the POS of words, postagger [62] from Stanford University is used. Mainly nouns, verbs, adjectives and part of prepositions are tagged. The

words with the other POS are ignored, such as, articles and conjunctions, because they do not contribute much in representing the main meaning. For nouns, there are four types: singular noun (NN), singular proper noun (NNP), plural noun (NNS) and plural proper noun (NNPS). For verbs, there are different tenses and participles as listed in the table. Some of prepositions (IN) are tagged. For adjectives, there are base form (JJ), comparative form (JJR) and superlative form (JJS). Sometimes present and past participate are used as adjectives, such as, *edited book*. For adjectives and this kind of verbs, they are called as "modifier" in general.

Table 2.3 Part-of-speech (POS) tagging

POS	Prefix	POS tagging type	Remark
Noun	NN-	NN, NNP, NNPS, NNS	Noun and proper noun, singular and plural
Verb	VB-	VB, VBP, VBZ, VBD	Verb base form, singular present, past tense
		VBG	Verb, Present participle
		VBN	Verb, past participle
Preposition	IN	IN	Preposition, of, by,
Adjective	JJ-	JJ, JJR, JJS	Adjective, comparative form, superlative form
Other	O-		Except the above POS

In order to obtain the patterns that are mostly used, some real-life ontologies and experimental ontologies are studied. The most commonly used patterns are concluded in Table 2.4. The first column shows the composition modes of word, and then the patterns in second column. A star symbol (*) indicates that the tagged word is identified as core word. Besides the core word, complementary information is also noted, such as, multiple nouns and the passive tense. The representation of this information is denoted as *(core word, <type, complement info. 1, type, complement info. 2, ...>)*. For instance, *(title, <book, MULTI_NOUN >)* denotes that the core word is *title* and the complementary information is *book* with type of *multiple nouns (MULTI_NOUN)*.

NNG is used to represent a group of nouns, including one or many nouns. *NNs* represents the complementary information, it is composed of several nouns in a sequential order. There are two special cases with preposition "*of*" and "*by*". "*Of*" changes the position of core word in multi-nouns mode. For example, for both labels "*titleOfBook*" and "*bookTitle*", the core word is title, but due to "*of*", the position of core word is changed. Normally, for multi-nouns mode, the recognition rule is that the last noun is taken as core word. But for the multi-nouns with "of", the rule

is changed to that the core word is the noun just before "of". "*By*" is used to identify whether a verb is past form or modifier. For example, in "*editedBook*", "*edited*" is a modifier. In "*editedByAuthor*", "*edited*" is a past participate form of verb "*edit*" with passive voice. The details and examples of each pattern are presented in Table 2.5.

Table 2.4 Core word recognition patterns

		Composition mode	Pattern	Com. info.	Remark
Noun -based	Nouns group (NNG)	Single noun	NN*	-	The noun
		Multi-nouns	NN(+)-NN*	NNs	The last noun
		Multi-nouns with 'of'	NN*-of-NNG	NNs	Noun just before 'of'
	Modifier-noun (MM-NNG)	Adjective-noun(s)	JJ-NNG*	JJ, NNs	The noun
		Past participate-noun(s)	VBN-NNG*	VBN, NNs	The noun
		Present participle -noun(s)	VBG-NNG*	VBG, NNs	The noun
Verb -based	Verb	Single verb	VB*	NNs	Verb
	Verb-object	Verb-noun	VB*-NNG	NNs	Verb
		Verb-prep-noun	VB*-PP(-NNG)	NNs	Verb
		Passive form	VBN-by(-NNG)	NNs	Verb

* core word + one to many

Table 2.5 Examples of patterns and core word recognition

Type	Composition mode	Example	Core word	Compl. info.
Nouns group	Single noun	book, books	book	-
	Multi-nouns	book_title, BookTitle	title	book
	Multi-nouns with 'of'	titleOfBook	title	book
modifier-noun	Adjective-noun(s)	shortName	name	short
	Past participate -noun(s)	publishedBook	book	published
	Present participle-nouns(s)	increasingNumber	number	increasing
Verb	Single verb	uses	use	-
Verb-object	Verb-noun	hasSiblingsOf	have	siblings
	Verb-prep(-noun)	Submits_to_conf	submit	conf
	Passive form	writtenByAuthor	write	author

2.4.3 Semantic Similarity Measurement for Two Single Words

Before introducing the measurement method for compound word with patterns, a similarity measurement algorithm SMA for two single words is proposed. The pattern-based core word (PCW) similarity measurement is

based on Semantic MAtching (SMA), which combines two similarity measurement algorithms: Lin model and homonyms checker.

In both measurement algorithms, WordNet® [63] is adopted as the lexical database. WordNet is a large lexical database of English constructed by the connections between four types of Part-Of-Speech (POS): nouns, verbs, adjectives and adverbs. They are grouped into sets of cognitive synonyms (synsets), and each expresses a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The main relation among words in WordNet is synonymy, as between the words *shut* and *close* or *car* and *automobile*[8]. Some of these relations (hypernym, hyponym for nouns, and hypernym and troponym for verbs) constitute is-a-kind-of (holonymy) and is-a-part-of (meronymy for nouns) hierarchies [64]. In this thesis, WordNet is utilized from two aspects: (i) utilized as taxonomy tree in Lin Model and (ii) to retrieve synonyms (check if existing a synonym and get the number of synonyms) in homonyms checker. The resolution of WordNet is implemented with two external APIs: JWI and JWS (see Section 4.2.1, Table 4.1).

Lin model [65] is a reused and adapted method. Lin model [65] is a information content (IC) and taxonomy-based model for measuring semantic similarity of two concepts. Lin model takes the taxonomy as a tree and returns the semantic similarity by measuring commonality C between two words in the taxonomy tree (WordNet in our work). Seco et al. [66] compared eight methods to use their information content, which is based on WordNet. Lin Model showed good results in comparison with the others.

Assuming that the taxonomy is a tree, for two concepts s_1 and s_2, if $s_1 \in C_1$ and $s_2 \in C_2$, the commonality between s_1 and s_2 is $s_1 \in C_0 \wedge s_2 \in C_0$, where C_0 is the most specific class that subsumes both C_1 and C_2. $P(C)$ is the probability that a randomly selected object belongs to C. Lin model is defined in Eq. (2.1).

$$Lin(s_1, s_2) = \frac{2 \cdot log\, P(C_0)}{log\, P(C_1) - log\, P(C_2)} \qquad (2.1)$$

Homonyms checker: Homonym is a special case in semantic matching. The same word represents different meanings in different contexts, for instance, "*article*" may refer to a *publication* or refer to a *product*. Homonym checker is proposed to try to measure the similarity for this issue.

At first whether the two ontologies, where the homonyms are occurred, belong to the same context is measured. A semantic similarity indicator I_s helps to examine whether they belong to the same context. I_s is

[8] http://wordnet.princeton.edu/

computed based on the identified core words (not on the original labels). The calculation of I_s is defined in Eq. (2.2), where *#synonym* is the number of synonyms identified between *O* and *O'*, and *tcp* is the number of total concepts and properties. For a word in ontology *O*, if there is a synonym existing in *O'*, then *#synonym* count adds 1.

$$I_s = \#synonym/min\,(\#tcp_1, \#tcp_2) \tag{2.2}$$

A threshold *th* is set. If the indicator *Is* is greater than the threshold *th*, then the two ontologies are considered as belonging to same context. In this case, the two words are considered as identical and the similarity is assigned to 1.0. Otherwise, a formula (see Eq. (2.3)) is applied for computing the similarity of the pair of homonyms, where *#m* is the number of different explanations (retrieved from WordNet) that the word have. In our work, the threshold is set manually as th = 0.2, the value can be adjusted according to specific domain and experiments.

$$H(s) = \begin{cases} 1, I_s \geq th \\ (\#m - 1)/\#m, I_s < th \end{cases} \tag{2.3}$$

Semantic MAtching (SMA): An overall similarity measurement combining *Lin Model* and *Homonym Checker* for two single words s_1 and s_2 is defined in Eq. (2.4). If the two words to be matched are identical, then *Homonym Checker* is applied, otherwise, *Lin Model* will be applied.

$$SMA(s_1, s_2) = \begin{cases} LinModel(s_1, s_2), s_1 and\ s_2\ not\ identical \\ H(s_1), identical\ word \end{cases} \tag{2.4}$$

2.4.4 Pattern-based Core Word (PCW) Similarity Measurement

PCW is an algorithm for measuring the semantic similarity between a pair of compound words. The format of input is defined as follows:

(type[noun-based, verb-based], pattern, core word, <complementary info.$_1$, type$_1$>, <complementary info.$_2$, type$_2$>, ...).

Taking label "*_theShortTitle_OfBook*" for example, after a series of processes below, the generated input is *(Noun-based, JJ-NN-of-NN, title, <short, MODIFIER>, <book, MULTI_NOUN>).*

Original label		_theShortTitle_OfBook
a)	Elimination and tokenization	short title of book
b)	Pattern recognition	JJ-NNG >> JJ-NN-IN-NN >> JJ-NN1-of-NN2
c)	Core word	title
d)	Complementary information	short, MODIFIER; book, MULTI_NOUN

The PCW algorithm is based on *SMA* (see Eq. (2.4)) for computing the similarity between entities (e_1, e_2). SMA aims to measure the similarity between a pair of single concepts in a semantic context. PCW (see Eq. (2.7)) utilizes SMA as a component to compose the algorithm. There are two parts involved: core word part M_1 (see Eq. (2.5)) and complementary information part M_2 (see Eq. (2.6)). PCW combines these two parts using weighted method, while core word is considered more important in representing the semantic, thus the weights of M_1 and M_2 are set to 0.7 and 0.3 respectively.

In the equations, cw_1 and cw_2 denote the core words of entities e_1 and e_2 respectively. $CI = \{ci_1, ci_2, \ldots\}$ denotes the set of complementary information of entity e, the length of CI is l. CI_1 and CI_2 denote the sets of complementary information of entities e_1 and e_2 with length l_1 and l_2 respectively. For M_1, if the core words are the same, then its value is assigned to 1.0, otherwise its value is calculated using *SMA* (cw_1, cw_2). While for M_2, if it exists that two of the complementary information of two entities are identical, then its value is assigned to 1.0, otherwise, its value is accumulated based on each pair of them using *SMA*.

$$M_1(cw_1, cw_2) = \begin{Bmatrix} 1, cw_1 = cw_2 \\ SMA(cw_1, cw_2), cw_1 \neq cw_2 \end{Bmatrix} \quad (2.5)$$

$$M_2(CI_1, CI_2) = \begin{Bmatrix} 1, ci1_k = ci2_j \\ \sum SMA(ci1_k, ci2_j)/(l_1 * l_2), ci1_k \neq ci2_j \\ ci1_k \in CI_1,\ ci2_j \in CI_2, 0 < k < l_1, 0 < j < l_2 \end{Bmatrix} \quad (2.6)$$

$$PCW(e_1, e_2) = 0.7 * M_1(cw_1, cw_2) + 0.3 * M_2(CI_1, CI_2) \quad (2.7)$$

2.4.5 An Illustrative Example

The aim of *PCW* is to recognize core words from natural language alike compound word or short phrases that are used in labels of ontologies, thus the hypothesis to use and apply the method is that the description of ontology should be alike natural languages. The ontology, which is built by random strings or has few meaningful words, is less applicable to use the method.

A real-life ontology *ekaw* [9] is used to test the core word recognition part. The testing of the whole measurement algorithm *PCW* is presented in Chapter 4. This ontology contains 74 classes and 33 object properties. The ontology is in the domain of conference and publication.

[9] http://oaei.ontologymatching.org/2012/conference/data/ekaw.owl

There are total 106 entities recognized, and part of the results is kept without changing in Table 2.6. Most of the entities can be identified correctly as expected; however a few of them cannot be recognized correctly. We count manually the incorrectly identified core word and patterns regarding their real semantics. Table 2.6 listed the original label, identified pattern, core word and complementary information.

Table 2.6 Pattern and core word recognition on real ontology

Original label	Pattern	Core word	Complementary information
Abstract	JJ-	(Abstract, MODIFIER)	
Academic_Institution	NN-NN-	(Institution, MULTIPLE_NOUNS)	<MULTI_NOUN,Academic>
Accepted_Paper	VBN-NN-	(Paper, SINGLE_NOUN)	<MODIFER,Accepted>
Agency_Staff_Member	NN-NN-NN-	(Member, MULTIPLE_NOUNS)	<MULTI_NOUN,Agency> <MULTI_NOUN,Staff>
Camera_Ready_Paper	NN-NN-NN	Camera-Ready-Paper-	(Paper, MULTIPLE_NOUNS)
Conference_Banquet	NN-NN-	(Banquet, MULTIPLE_NOUNS)	<MULTI_NOUN,Conference>
Demo_Chair	NN-NN-	(Chair, MULTIPLE_NOUNS)	<MULTI_NOUN,Demo>
Early-Registered_Participant	O-NN-NN-	Early-Registered-Participant-	(Participant, MULTIPLE_NOUNS)
Individual_Presentation	JJ-NN-	(Presentation, SINGLE_NOUN)	<MODIFER,Individual>
Organising_Agency	NN-NN-	(Agency, SINGLE_NOUN)	< MODIFER,Organising>
Paper	NN-	(Paper, SINGLE_NOUN)	
Proceedings_Publisher	NN-NN-	(Publisher, MULTIPLE_NOUNS)	<MULTI_NOUN,Proceedings>
Programme_Brochure	NN-NN-	(Brochure, MULTIPLE_NOUNS)	<MULTI_NOUN,Programme>
Rejected_Paper	VBN-NN-	(Paper, SINGLE_NOUN)	<MODIFIER,Rejected>
Submitted_Paper	VBN-NN-	(Paper, SINGLE_NOUN)	<MODIFIER,Submitted>
Tutorial_Chair	NN-NN-	(Chair, MULTIPLE_NOUNS)	<MULTI_NOUN,Tutorial>
authorOf	NN-IN-	(author, SINGLE_NOUN)	
coversTopic	NN-NN-	(Topic, MULTIPLE_NOUNS)	<MULTI_NOUN,covers>
organisedBy	VB-IN-	(organised, VERB_BASED)	
paperPresentedAs	NN-VBN-O-	(paper, SINGLE_NOUN)	<MODIFER,Presented>
publisherOf	NN-IN-	(publisher, SINGLE_NOUN)	
referencedIn	VBN-O-	(referenced, MODIFIER)	
reviewerOfPaper	NN-IN-NN-	(reviewer, MUTIPLE_NOUN_WITH_OF)	<multi_noun_with_of, paper>
updatedVersionOf	VBN-NN-IN	(Version, SINGLE_NOUN)	<MODIFER,updated>
writtenBy	VB-IN-	(written, VERB_BASED)	
……	……	……	……

The labels that are not recognized correctly are marked with grey background. There are nine misidentified patterns out of 106, thus the recognition precision is 91.5% in this test. The reasons that cause the inaccuracy are twofold: (i) the proposed patterns do not cover all the cases, especially some special cases, and (ii) the precision of postagger to tag the words and WordNet to find synonyms are not 100%.

Regarding the first case, because of the complexity and diversity of language environment, the patterns can vary tremendously. The patterns defined in this thesis are commonly used in general domain. The patterns can be quite different on some specific domains.

Taking "*part-of-speech*" for example, following the pattern "*NN*-of-NNG*", the recognized core word will be "*part"*, it is obvious that this result is not accurate. Additionally, for example, for "*early-registered*" and "*camera-ready*", these words should be taken as one word, but in current proposed patterns, it is difficult to tokenize and recognize the core word automatically. To improve this issue, more patterns can be extended in order to adapt to different language environments and the special situations.

Concerning the second case, for the words that have several POS, the patterns cannot be recognized by postagger correctly. For instance, "*abstract*" is both noun and adjective, in this example, it is supposed to be taken as noun (see first row of Table 2.6), however the postagger recognizes it as a "*Modifier*", which is inaccurate.

Another issue about the precision is caused by the limitations of the lexical database, which is WordNet in this work. Some words and their special meanings are not be included in the database, so that the algorithm could not find proper synonyms. For example, the meaning of "*MS word*", which should be a name of word processing software, cannot be identified correctly with WordNet. A solution to this issue is to define a special name list, which contains the unusual meanings and uncommon words, such as, "*PDF*" and "*MS word*". And then assign these names with a commonly used equivalent concept, for example, using "*format*" to replace "*PDF*" and "*software*" to replace "*MS word*".

2.5 Non-semantic Based Matchers

In order to adapt the diverse contents of source ontologies and to improve matching ability, it is important to perform ontology matching from non-semantic aspects. Because *PCW* is not applicable for every case, especially, PCW is not applicable when no core words could be recognized. Therefore two matchers are proposed in the work to complement PCW: *Lexical - Based Matcher (LBM)* and *Structural-Based Matcher (SBM).*

These matchers are regardless of the semantics that the entities represent. LBM adopts two matching algorithms: *Edit Distance (ED)* [28] and *N-Gram (NG)* [37], while SBM applies *Similarity Flooding Algorithm (SFA)* [38] as the main algorithm.

2.5.1 Lexical-Based Matcher (LBM)

LBM seeks the correspondences from lexical level. It considers the label of entity as object of study. Lexical matcher is designed based on the string, which presents the labels of concepts and properties. The entities are treated only as a sequence of letters. The structure that the entity contains and the meaning that the entity represents are not investigated. Two similarity measurement algorithms: *Edit Distance* and *N-Gram* are reused. The selection of which algorithms to use depends on the length of labels of entities to be matched.

Cohen et al. [67] compared different string distance metrics methods, they stated that edit distance (Jaro Winkler distance) is intended for short strings. According to Smith [68] and Sojka [69], the average length of English words is around 9, and approximately 90 % of English words' lengths are less than 13, thus in the thesis, for the words that are longer than (or equal to) 13 are regarded as long strings. This survey is based on single words, but we adopted it in this matcher as a rule for general strings, including compound words. For both labels, whose lengths are longer than 13, N-gram, which is a token-based distance function, is applied; otherwise edit distance will be applied.

2.5.1.1 Edit Distance (ED)

String metric measures the similarity or distance between two plain strings. Distance function maps a pair of string s_1 and s_2 to a real number r, where a smaller value of r indicates greater similarity between s_1 and s_2 [67].

Edit Distance: Distance is the cost of operations, including insertion, deletion and substitution, for converting s_1 to s_2 in a best sequence. Edit distance [36] was proposed by Winkler based on Jaro distance [70, 71]. Jaro distance is defined in Eq. (2.8), where s_1 and s_2 are strings from entities of O and O', m is the number of *matching character* and t is half of the *number of transposition*.

Two characters are matched only when the distance is not beyond the match window. Match window is a range calculated based on the lengths of strings and their position, given $g=max\ (|s_1|,\ |s_2|)/2 - 1$, taking two characters a_i and b_j (i, j denotes the sequence in the string) from s_1 and s_2 respectively. For character a_i in string s_1, the window is $j - g \leq i \leq j+g$,

only character from b_{j-g} to b_{j+g} can be matched. The same principle is for character b_j from s_2. The number of matching (but different sequence order) characters defines the *number of transpositions*. For instance, two matching strings are "**T**DR**F**E" and "**F**DR**T**E", but the sequence of letter "T" and "F" is not the same, thus the transposition number is 2, and t = 2/2 =1.

$$Jaro(e_1, e_2) = \begin{cases} \frac{1}{3} * \left(\frac{m}{|e_1|} + \frac{m}{|e_2|} + \frac{m-t}{m}\right), m \neq 0 \\ 0, \mathrm{m} = 0 \end{cases} \quad (2.8)$$

Edit distance adds a weight for common prefix based on Jaro distance. It is defined in Eq. (2.9). P is the length of longest common prefix of s_1 and s_2, min (P, 4)/10 is for assuring the coefficient not exceeding 0.25, which may cause consequently that ED (s_1, s_2) is greater than 1.0.

$$ED(e_1, e_2) = Jaro(e_1, e_2) + \frac{min(P,4)}{10} * (1 - Jaro(e_1, e_2)) \quad (2.9)$$

For instance, given two strings s_1 = "*winkler*" and s_2 = "*wenklir*", then the lengths$|s_1|$ = 7, $|s_2|$ = 7 and g = *max* (7, 7) /2 – 1 = 2. Match window is j - 2 ≤ i ≤ j + 2, taking the first row as an example, j = 1, then -1≤ i ≤2, thus the first three cells are marked in grey as match window. There is a letter "*W*" in it, so "*W*" is matched. The rest matching process is shown in Table 2.7, the shadowed table cell represents the match window. For '*E*' and '*I*', they cannot be matched because they are beyond of the match window. Then m = 5, the matched strings are "*WNKLR*" and "*WNKLR*". The sequences of the two matched strings are the same, thus no transposition is needed, then t = 0.

Table 2.7 Example of Jaro-Winkler distance between "winkler" and "wenklir"

	W	I	N	K	L	E	R
W	1	0	0	0	0	0	0
E	0	0	0	0	0	0	0
N	0	0	1	0	0	0	0
K	0	0	0	1	0	0	0
L	0	0	0	0	1	0	0
I	0	0	0	0	0	0	0
R	0	0	0	0	0	0	1

The distance value *Jaro*("*winkler*" ,"*wenklir*")= 1/3*(5/7+5/7+ (5-0) /5)=17/21=0.809. The longest prefix is "*w*", then P=1, thus

ED ("*winkler*","*wenklir*") = 0.809 + 0.1*(1-0.809) = 0.828.

2.5.1.2 N-Gram (NG) Model

N-gram model [37] is originally developed in the domain of computational linguistic and probability. ***N-gram models*** *are probabilistic models of text that use some limited amount of history, or word dependencies, where n*

refers to the number of words that participate in the dependence relation [72]. For instance, "*unigram*" refers to n-gram with size of 1; "*bigram*" refers to n-gram with size of 2. *Tri-gram* (*s*) denotes the set of trigrams of string *s*. For example, *tri-gram* ("*between*") = {"bet", "etw", "twe", "wee", "een"}.

N-gram is applied to measure the similarity of a pair of strings (s_1, s_2). This method has been applied in some work to measure the similarity. A general measure is denoted in Eq. (2.10).

$$NG(s_1, s_2) = \frac{|ngram(s_1) \cap ngram(s_2)|}{min(|s_1|,|s_2|)-n+1} \quad (2.10)$$

In this thesis, tri-gram-based method is used to measure the similarity between strings s_1 and s_2, which is denoted in Eq. (2.11).

$$TG(s_1, s_2) = \frac{|trigram(s_1) \cap trigram(s_2)|}{min(|s_1|,|s_2|)-2} \quad (2.11)$$

Taking "*matcher*" and "*teacher*" for example, the calculating process is as follows. The generated similarity between them is 0.4.

1) trigram ("*matcher*") = {mat, atc, tch, che, her};
2) trigram ("*teacher*") = {tea, eac, ach, che, her};
3) trigram ("*matcher*") ∩ trigram ("*teacher*") = {che, her};
4) TG("*matcher*", "*teacher*") = 2 / (7 - 2) = 0.4.

2.5.2 Structure-Based Matcher (SBM)

Similarity Flooding (SF) proposed by Melnik et al. [38] is an algorithm for matching two data schemas based on *similarity propagation graph* and *fix point computation*. The algorithm takes two graphs as input and produces the mappings between corresponding nodes of graphs. In the work of Li et al. [48], they applied also similarity flooding in structure-based strategies. Our work applied SF differently in composing the similarity propagation graph, more edges are taken into account in order to improve the method.

2.5.2.1 Construct Similarity Propagation Graph (SPG)

The first step to apply SF is to construct a Similarity Propagation Graph (SPG). *A SPG is an auxiliary data structure derived from models A and B that is used in the fix point computation of this algorithm* [38]. A triple node (*s, p, o*) represents each edge in the graph, where *s* and *o* are the source and target nodes, while *p* refers to the label. Pairwise connectivity graph (PCG) is defined as follows:

$$\big((x,y),p,(x',y')\big) \in PCG(A,B) \leftrightarrow (x,p,x') \in A \; and \; (y,p,y') \in B.$$

Each node in PCG is an element from $A * B$. SPG is constructed based on PCG with additional edge going in the opposite direction. For ontology alignment, model A and model B are the two ontologies to be matched. To convert ontology (in format OWL) to SPG, the rules defined in Table 2.8 are used to construct the nodes. In Table 2.8, *c* denotes a class, *p* denotes a property, *e* denotes an entity and *i* denotes an instance of class. For example (see Figure 2.1), "*order contains product*" is denoted as (*order, l_object_property, product*).

Table 2.8 Relations for constructing SGP node in SF

Source node	Edge	Target node	Description
c_i	l_super_class	c_j	Class c_i has super class c_j
c_i	l_ sub _class	c_j	Class c_i has sub class c_j
p_i	l_sub_property	p_j	Property p_i has sub property p_j
c_i	l_object_property	p_j	Class c_i has object property p_j
c_i	l_data_property	p_j	Class c_i has data property p_j
c_i	l_domain	e_j	Class c_i has domain e_j
c_i	l_range	c_j	Class c_i has range with class c_j
c_i	l_ has_individual	i_j	Class c_i has instance i_j

2.5.2.2 Find Mappings

The second step is to find mappings with fix point computation. For node *(x, y)* $\in A * B$ in SPG, θ *(x, y)* denotes the similarity between *x* and *y*. θ-value is incremented by its neighbor pairs in SPG multiplied by the propagation coefficients on the edges going from the neighbor pairs to *(x, y)*. In general, similarity θ^{i+1} is computed from θ^i as defined in Eq. (2.12), where *a* and *b* are neighbor nodes of *(x, y)*, *w* is the coefficient (ranges from 0 to 1.0) between two nodes. The coefficient can be computed in many different ways. A method illustrated in the paper [38] is used. It is based on the intuition that each edge type makes an equal contribution of 1.0 to spreading of similarities from a given map pair. The θ^0 *(x, y)* is set to be 1.0 for all *(x, y)* $\in A * B$.

$$\theta^{i+1}(x,y) = \theta^{i}(x,y) + \sum_{\substack{(a_u,p,x)\in A,\\ (b_u,p,y)\in B}} \theta^{i}(a_u,b_u)\cdot w\big((a_u,b_u),(x,y)\big) + \sum_{\substack{(x,p,a_v,)\in A,\\ (y,p,b_v)\in B}} \theta^{i}(a_v,b_v)\cdot w((a_v,b_v),(x,y)) \quad (2.12)$$

A brief algorithm to show the matching process is given in Algorithm 1. At first two ontology *O* and *O'* are converted to two graphs *G* and *G'* with the relations defined in Table 2.8. Then, an initial mapping between the two graphs is needed in order to start the similarity propagation. In our work, the top node of ontology: "*Thing*" is taken directly as the initial mapping nodes. The details of the computation is not elaborated here, it can be followed in Melnik et al. [38]. In the implementation of SBM, an API[10] of similarity flooding algorithm is invoked. This API provides core computation of this algorithm.

Algorithm 1: **Similarity Flooding**

```
1G =ConvertToGraph (O) , G' = ConvertToGraph (O') ;
2 initialMap = StringMatch (G. thing , G' . thing) ;
3 mapping = SFA(G, G' , initialMap) ;
4 result = SelectThreshold ( mapping ) ;
5 correspondence = Wrap ( result) ;
```

2.5.2.3 *Illustrative Example*

A simple example is used for illustration as shown in Figure 2.7. (a) is the graph model of ontology from example in Figure 2.1. In (b), letters are used instead of meaningful names to focus on the structure. The graph is constructed with the edges listed in Table 2.8. (c) denotes the constructed pairwise connectivity graph (PCG). (d) is the induced propagation graph with coefficient. To implement the algorithm, core codes in Java are listed as follows. The comments explained the steps.

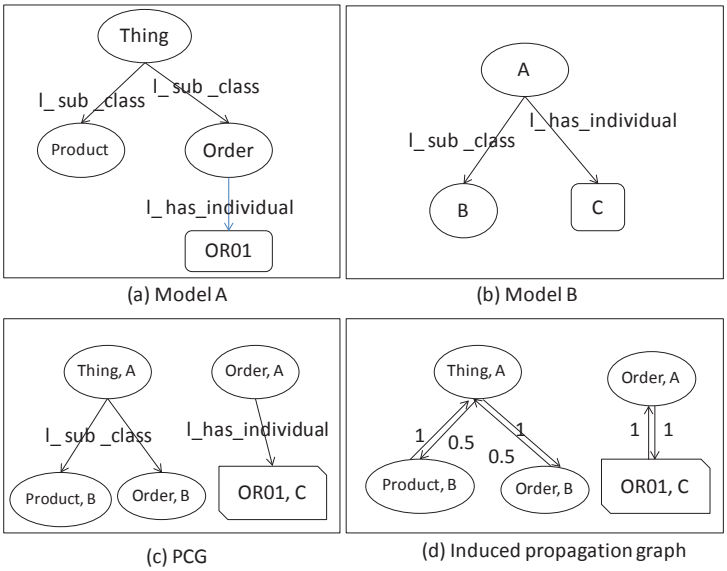


Figure 2.7 Illustration of similarity flooding algorithm

[10] http://www.jdom.org/downloads/source.html

```
    // STEP 1: create graph/model A
    Model MA = rf.createModel();
    Resource Thing = nf.createResource("Thing");
    Resource Product = nf.createResource("Product");
    Resource Order = nf.createResource("Order");
    Resource IDOS38 = nf.createResource("OR01");
    Resource l_class = nf.createResource("l_ sub _class");
    Resource l_indv = nf.createResource("l_ has_individual");
    MA.add(nf.createStatement(Thing, l_class, Product));
    MA.add(nf.createStatement(Thing, l_class, Order));
    MA.add(nf.createStatement(Order, l_indv, OR01));

    // STEP 2: create graph/model B
    Model MB = rf.createModel();
    Resource A = nf.createResource("A");
    Resource B = nf.createResource("B");
    Resource C = nf.createResource("C");

    MB.add(nf.createStatement(A, l_class, B));
    MB.add(nf.createStatement(A, l_indv, C));

    // STEP 3: create an initial mapping which is just a
cross-product with
    // 1's as weights
    List initMap = new ArrayList();
    initMap.add(new MapPair(Thing, A, 1.0));
    initMap.add(new MapPair(Thing, B, 1.0));
    initMap.add(new MapPair(Thing, C, 1.0));
    initMap.add(new MapPair(Product, A, 1.0));
    initMap.add(new MapPair(Product, B, 1.0));
    initMap.add(new MapPair(Product, C, 1.0));
    initMap.add(new MapPair(Order, A, 1.0));
    initMap.add(new MapPair(Order, B, 1.0));
    initMap.add(new MapPair(Order, C, 1.0));
    initMap.add(new MapPair(OR01, A, 1.0));
    initMap.add(new MapPair(OR01, B, 1.0));
    initMap.add(new MapPair(OR01, C, 1.0));

    // STEP 4: Start to match
    Match sf = new Match();
    sf.formula = Match.FORMULA_FFT;
```

```
sf.FLOW_GRAPH_TYPE = Match.FG_PRODUCT;

// STEP 5: Generate mappings
MapPair[] result = sf.getMatch(MA, MB, initMap);
MapPair.sort(result);
dump(result);
```

The generated mapping results are shown in Table 2.9. The first column lists the nodes from model A, namely, the entities from ontology. The second column shows the matched nodes from model B. The third column is the similarity obtained.

Table 2.9 Mapping results with SFA

Model A	Model B	Similarity
Thing	A	1.00
OR01	C	0.66
Order	A	0.66
Order	B	0.50
Product	B	0.50

2.6 Conclusion

In this chapter, a core word-based semantic similarity measurement approach has been proposed and presented. The core words are recognized based on pre-defined patterns of POS of words. This approach allows measuring the semantic similarity between a pair of short text or compound words. It can be applied not only for ontology alignment, but also in other domains, such as, semantic search and text categorization. An illustration example is presented to test the pattern recognition, and the results suggested that the recognition approach has high precision. Regarding future work to improve the method, more patterns can be extended to fulfill various needs and to adapt to some specific source ontologies.

The other two matchers have also been proposed by reusing and adapting existing algorithms. These two matchers seek correspondences from lexical and structural level of source ontologies, so that to complement semantic matching and improve the accuracy.

The whole alignment approach, combined with the other two matchers and the aggregation of matchers that will be discussed in Chapter 3, is implemented in Java and tested (Chapter 4). The experiment results showed that the proposed approaches obtained promising results and reached expected goals.

3> An Analytic Matcher Aggregation Approach

3.1 Introduction

To find correspondences, multiple matchers are applied from different aspects of ontologies. How to choose matchers and combine the different matching results becomes a concern in ontology alignment. Shvaiko and Euzenat [35] stated that one of the challenges facing in the domain of ontology alignment is matcher combination. In our work, three matchers applied at three levels: semantic, lexical and structural. An automatic and dynamic method to aggregate the matchers is expected.

Weighted mean method is selected to weight the matchers, since it can balance the various factors by respecting the final goal and produces globally optimized results. The key point to apply weighted mean method is the assignment of weight of each variable. In order to learn the weights automatically and dynamically, an analytic approach based on Analytic Hierarchy Process (AHP) to learn the weights is proposed. AHP is a method for organizing and analyzing complex decisions with a structured process. It

was developed by Thomas L. Saaty [73] in 1970s based on mathematics and psychology and has been continually studied and refined since then. AHP method balances various criteria against the goal and generates the priorities of each alternatives, so that to make decisions. It has been applied widely for decision making [74], the examples of application domain include government, industry, health care and education. A survey about applications of AHP was done by Vaidya and Kumar [75] in 2006, 150 application papers are referred, the applications cover almost every domains, such as, manufacturing, political, culture.

Besides applying AHP for decision-making, to compute the weights of variables is one of the applications. Some works for weighting have been done in the other domains, for instance, Zhao et al. [76] applied AHP for computing factor weights of network learning pattern recognition (NLPR) process, another work is that Shapira and Simcha [77] proposed AHP-based weighting for factors affecting safety on construction sites.

AHP is adapted by Mochol et al. [78] in the domain of ontology alignment for selecting ontology matchers, it aims to choose one to many matching approaches or techniques from macro level. Six categories of characteristics are taken as criteria to decide the matchers, including: input, approach, usage, output, costs and documentation. Based on these characteristics and AHP process, one to several matching approaches will be chosen according to the final generated priorities.

In this thesis, AHP is proposed to compute the weight of each matcher in order to aggregate the results that generated by different matchers. That is the main difference with work carried out by Mochol et al. [78]. A key point, somehow a difficult point, in applying AHP is the assignment of scale (intensity of importance). To automate this process and improve the precisions, three similarity indictors from whole-ontology level are proposed to facilitate the process.

In this chapter, the aggregation methods are investigated in Section 3.2. AHP is illustrated with an example in Section 3.3. Matcher selection process and the adaptation of AHP to learn the weights are presented in Section 3.4. Section 3.5 draws some conclusions.

3.2 Aggregation Method

The aggregation process involves two issues: (i) mapping cardinality, because one entity may find several matched entities in another ontology, namely, the policy to choose the mapping pairs between two ontologies, such as, one to one and one to many; (ii) how to aggregate the different

results obtained by different matchers in order to generate a final weighted value. In this section, firstly, the mapping cardinality is introduced in §3.2.1, *one-to-one* and *one-to-many* policies will be applied accordingly in our approach. Secondly, the different aggregation modes are investigated in §3.2.2, in this thesis, weighted mean method is applied in this thesis.

3.2.1 Cardinality

The first issue that needs to be considered in a mapping relationship is the mapping cardinality. A mapping cardinality constraints and specifies in a relation how many entities that an entity can be related to. In general, there are four possibilities for a binary relationship set as listed in Table 3.1.

Table 3.1 Mapping cardinality

Left	Right	Mode	Example
1	1	one to one	book - pages number
1	1..*	one to many	book - author
1..*	1	many to one	city - country
0..*	0..*	many to many	name - person

Between a pair of source ontologies O and O', for entity e_i in ontology O, the list of found correspondences from ontology O' by the matcher (one of the three matchers proposed in Chapter 2) is denoted as set $C = \{ c_i (e_i, e_{ij}), len \leq j \leq 0\}$, where len is the number of found correspondences in O'. The policy of cardinality applied in this thesis is twofold: if it exists more than one correspondences whose similarity equals to 1.0, policy *one to many (1:n)* will be applied. Otherwise, policy *one to one (1:1)* is applied

For the first case (*one to many (1:n)*), choose all the correspondences whose similarity equals to 1.0 as discovered correspondences, while for the second case (*one to one (1:1)*), choose the correspondence that has maximum similarity value as discovered correspondence. The final chosen correspondences CC_i of entity e_i is denoted in Eq. (3.1) in response to the two cases, where $|c_{ij}|$ refers to the similarity value of entity pair (e_i, e_{ij}).

$$CC_i = \begin{cases} c_{ij}(e_i, e_{ij}),\ where\ |c_{ij}| = 1 \\ c_{ij}(e_i, e_{ij}), where\ |c_{ij}| = \max\left(|c_{ij}(e_i, e_{ij})|\right) \end{cases}, 0 \leq j \leq len \quad (3.1)$$

3.2.2 Aggregation Modes

This section investigates some general aggregation methods. Given a list of similarities $s_1, s_2, \ldots, s_k$ for entities (e_1, e_2) generated by matcher $m_1, m_2, \ldots m_k$, the problem is how to produce the final combined similarity. Adapted from the surveys of similarity aggregation methods done by Ji et al. [79] and Houshmand et al. [80], the aggregation methods are summarized in Table 3.2. The first five methods try to generate the combined similarity by selecting part (or all) of data from the given data set according to certain conditions. The first three ones simply pick one maximum, minimum or average value from data set. The fourth and fifth method select part of data according to a given percentage range (*ps, pe*): start percentage and end percentage. In term of the percentage, there are two ways to limit it: number of selected data and range of similarity value. There are also some varieties of combination with these basic modes, they are not elaborated in the thesis.

Table 3.2 Aggregation modes

Method	Equation	Description
Max	$FS = \max(s_i)$	Maximum
Min	$FS = \min(s_i)$	Minimum
Average	$FS=\text{sum}(si)/k$	Average
By percentage of number	$FS = \frac{\text{sum}(s_i)}{(pne - pns) * k}, pns \leq \frac{i}{k} \leq pne$	In descending order; (pns, pne) is the range, e.g. (0.3, 08)
By percentage of value	$FS = \frac{\text{sum}(s_i)}{N}, pss \leq si \leq pse$	In descending order; (pss, pse) is the range, N is the number of values
Weighted mean	$FS = \sum_{i=1}^{k} s_i \times w_i$	Weighted mean

Another type of aggregation is weighted mean method, which is the mostly used method by most of current approaches. The weighted mean method allows the final weighted value reflecting relatively the importance of each variable that is being weighted. It balances the different factors and generates a final value by evaluating the importance of each factor. To find a vector of weights $w = (w_1, w_2, \ldots, w_k)$ is the key point to apply the weighted mean method. Some approaches learn the weights by using the different methods that have been introduced in the related work of Chapter 2 (see §2.3.1). In this thesis, we adopt innovatively AHP to learn the weights and aggregate the matchers. AHP and the relevant works are introduced in next section. How to use AHP for matcher aggregation is presented in Section 3.4.

3.3 Analytic Hierarchy Process (AHP)

Thomas L. Saaty [73] defines a systematic process to apply the approach with strict mathematics. Figure 3.1 illustrated the general process. The explanations of each step are as follows:

1) Define the problem (§3.3.1);
2) The expected goal, the criteria and the alternatives are defined to build the hierarchy (§3.3.1);
3) With a strictly defined process, the alternatives are compared to each other against one criterion and a specific intensity of importance (scale) is assigned (§3.3.2);
4) The results from each step are synthesized and the priority of each alternative is generated. The priority is used to make decision for choosing an alternative, which suits the goal best (§3.3.3).

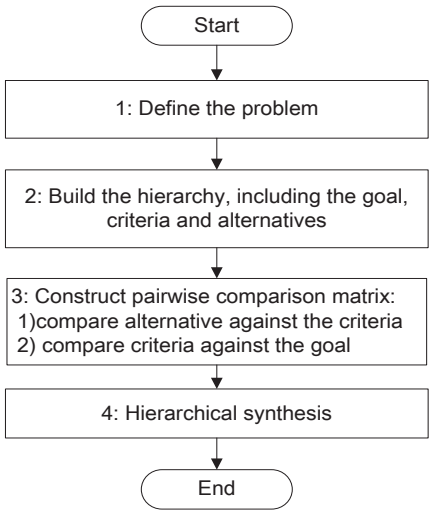


Figure 3.1 General AHP process

To explain the approach, an example for "*choosing a travelling destination*" is used. The process and example are illustrated at the same time in order to have a clear understanding and illustration. Relevant works using AHP for weighting are investigated in §3.3.4.

3.3.1 Description of Example

*"Charlie plans to travel during summer vacation. Charlie considers the cost, the procedures and the places of interests to make the decision. Currently he has three options: China, Greece and Netherlands. The decision problem facing is t**o choose an***

***ideal place to visit**. Charlie is French and he will need to request a visa to go to China. If he goes to Netherlands, he can take the train; otherwise, he needs to go by air for China and Greece."*

According to the above description, the AHP hierarchy is built as Figure 3.2 shows. The goal is to choose a country, while the criteria are low cost, less procedure and more places of interests. Three available alternatives are China, Greece and Netherlands.

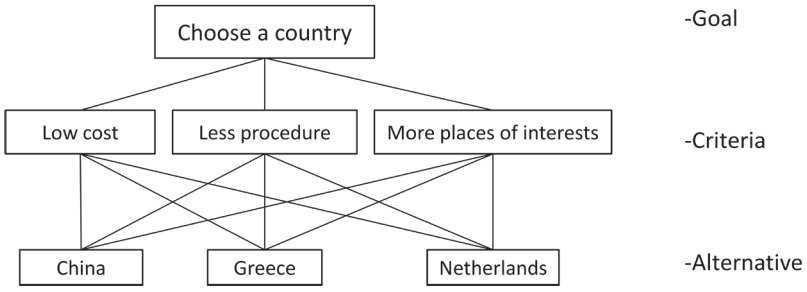


Figure 3.2 AHP hierarchy for the example

3.3.2 Pairwise Comparison

The next step of AHP is to determine the importance for the alternatives with respect to each criterion, and importance for each criterion with respect to reaching the goal. The priority is determined by pairwise comparisons with assigning a scale, which indicates the intensity of importance of this criterion. The comparisons are made by two steps: **(i) comparisons between alternatives against the criteria**, and **(ii) comparisons between criteria against the goal**. Each comparison is assigned with a scale to measure the intensity of importance. The fundamental scale [81] is listed in Table 3.3. In practice, the format of values can be flexible, such as, numerical or integer. It does not have to follow the format listed in Table 3.3. These values used need to be able to reflect the relative importance.

Table 3.3 AHP fundamental scales

Numerical scale	Verbal terms
1	Equally important
3	Moderately more important
5	Strongly more important
7	Very strongly more important
9	Extremely more important
2,4,6,8	Intermediate values to reflect compromise
Reciprocal scale i.e. 1/3, 1/5, 1/8 etc	Reciprocates the compared element in pair

3.3.2.1 Alternatives versus Criteria

Firstly, comparison is made between a pair of alternatives and the importance is evaluated by considering the major aspects. In Table 3.4, the importance of each alternative concerning *criterion "low cost"* is evaluated. The description helps to analyze and understand the reason of scale assignment. For instance, between "China" and "Greece", the cost for China is more because of: (i) the expense of applying visa, such as, application fees and transportation; (ii) the air flight normally is more expensive. Although the consumption level is relatively low in China, synthesizingly, it is reasonable to assign the scale as 2 for China and 5 for Greece. In the same way, we can compare the rest two pairs.

Table 3.4 Alternative comparison with respect to criterion "low cost"

Alternative	s	Alternative	s	Description
China	2	Greece	5	Air tickets for China are more expensive, and there are fees for applying visa, etc. The level of consumption is relatively lower than Greece.
China	1	Netherlands	2	Train tickets for Netherlands are relatively lower than air tickets. The expenses in Netherlands are relatively higher than in China.
Greece	3	Netherlands	2	The level of consumption in Greece is relatively lower than in Netherlands.

With mathematical processing, the data in Table 3.4 is transferred to a matrix as Table 3.5 for facilitating the calculation. The priority is generated in the last column of Table 3.5. It concerns only the criterion that has been evaluated: low cost. The other two criteria will be evaluated separately.

At the same time, an inconsistency factor is calculated. Inconsistency factor is a measure of the internal consistency of the judgments entered into the matrix. It is desirable that this factor is less than 0.100 [82]. If the inconsistency is less than 0.100, then the process is considered as good, otherwise the comparisons and scale assignment need to be adjusted and redone.

Table 3.5 AHP alternative comparison matrix with respect to criterion "low cost"

LOW COST	China	Greece	Netherlands	Priority
China	1	2/5	1/2	0.180
Greece	5/2	1	3/2	0.480
Netherlands	2	2/3	1	0.340

*Inconsistency Factor:0.0018

With the same process as evaluating the criterion "low cost", we evaluate the other two criteria: *less procedures and more places of interests* as listed in Table 3.6 and Table 3.7. For these two criteria, we just describe the final matrices and priorities, the comparison like Table 3.4 is omitted for simplification.

Table 3.6 AHP alternative comparison matrix with respect to criterion "less procedure"

LESS PROCEDURES	China	Greece	Netherlands	Priority
China	1	1/3	1/3	0.143
Greece	3	1	1	0.429
Netherlands	3	1	1	0.429

*Inconsistency Factor:0.0

Table 3.7 AHP alternative comparison matrix with respect to criterion "more place of interests"

PLACE OF INTERESTS	China	Greece	Netherlands	Priority
China	1	3/2	5/2	0.486
Greece	2/3	1	3/2	0.313
Netherlands	2/5	2/3	1	0.201

*Inconsistency Factor:0.0006

3.3.2.2 Criteria versus Goal

After comparing the alternatives with respect to each criterion, the following step is to compare *the criteria with respect to the goal*. In this case, we need to compare low cost, less procedure and more places of interests with respect to the goal "to choose a country". With the same comparison method, the final matrix is shown in Table 3.8. The priorities of cost, procedure and place of interests are 0.357, 0.141 and 0.502 respectively.

Table 3.8 AHP criteria comparison matrix with respect to the goal

CHOOSE A PLACE	Cost	Procedure	Places of interest	Priority
Low cost	1	3/1	3/5	0.357
Less procedure	1/3	1	2/6	0.141
More places of interest	5/3	6/2	1	0.502

3.3.3 Synthesis

The final step is to synthesize the priorities of the two levels: **alternative versus criteria**, and **criteria versus goal**. Taking the criterion "low cost" for example, the priority of criterion "low cost" with respect to the goal is 0.357 (see Table 3.8), the priorities (see Table 3.5) of alternatives: China, Greece and Netherlands, are 0.18, 0.48 and 0.34 respectively. We multiple the two priorities and then the final priority is obtained as listed in the second column of Table 3.9. With the same process, the priorities for criteria: less procedure and more places of interests are concluded as shown in Table 3.9. Then the final priorities of each alternative with respect to the goal are summarized in the last column of Table 3.9.

Table 3.9 Synthesis of all priorities

	Priority with respect to			
Alternative	Cost*0.375	Procedure*0.141	Places of interest*0.502	Goal
China	0.067	0.020	0.244	0.332
Greece	0.180	0.060	0.158	0.398
Netherlands	0.128	0.060	0.100	0.289
Total	0.375	0.141	0.502	1.000

The priorities of three options: China, Greece and Netherlands are 0.332, 0.398 and 0.289 respectively. Based on this result, Charlie will choose "Greece", which has a priority of 0.398, as the first choice of the visiting country. "China", which has a priority of 0.332 and is less than "Greece", will be taken as the second choice.

In this example, the priories are used for making decisions, however they can be adapted as weights of each alternatives. In next section, some works about using AHP for weighting are investigated and the proposal of this thesis is presented. More about AHP, Saaty and Vargas [83] analyzed the drawbacks of AHP and applying conditions. Vaidya and Kumar [75] reviewed the applications in different domains.

3.3.4 Relevant Works about Applying AHP for Weighting

Some relevant works about using AHP for weighting are listed in Table 3.10. The purpose is to investigate the application domain and the mode to assign the scale of pairwise comparisons. Scale assignment is a key point, somehow a difficult point, in applying AHP. Five applications are from different domains, such as, corridor analysis and E-learning. All of the five

applications assign the scale of pairwise comparisons *manually* according to domain experts, questionnaire or fundamental scales.

Berry [84] applied AHP to corridor analysis, the scales are assigned based on fundamental scales as listed in Table 3.3. The same for Zhao et al. [76], they applied AHP for computing factor weight of network learning pattern recognition (NLPR) process, and they assign the intensity of importance based on the fundamental scales. Tzeng et al. [85] applied AHP to e-learning to find out the weights of factors and to obtain each e-learning program score. The scales are assigned based on questionnaires from users. For both Shapira and Simcha [77] , and Abdullah and Azman [86], the scales are assigned manually by domain experts. The two applications are in the domain of construction sites and public health problem respectively.

Table 3.10 Relevant works of using AHP for weighting

Author	Domain	Scale assignment
Berry 2004, [84]	Corridor analysis	Manually (fundamental scales)
Tzeng et al. 2007, [85]	E-learning	Manually (questionnaire)
Shapira et al., 2009 [77]	Construction sites	Manually (domain experts)
Abdullah et al., 2011 [86]	Public health problem	Manually (domain experts)
Zhao et al., 2012 [76]	Network learning pattern recognition	Manually (fundamental scales)
Our approach	**Ontology alignment**	**Automatically(similarity indicators)**

In this thesis, an approach based on AHP for assigning the scales *automatically* is proposed. This method utilizes three similarity indicators, which are retrieved from whole-ontology level, to evaluate the importance of different criteria. This method allows assigning the scales automatically and maintaining certain high reasonability. It is elaborated in next section.

3.4 AHP-based Aggregation

In our work, several matchers are applied. Figure 3.3 illustrates the overall matcher selection. The selection of matchers stays at entity-level, namely, for different pairs of entities that from same pair of source ontologies (*O, O'*), the selections of matcher are different. A pair of entities from pre-processed ontology is taken as input, and then several conditions are examined to decide which strategy to apply, namely, which matcher(s) to use. First it is checked that whether the labels of entities are compound word and identical, if yes, the two labels are considered as identical and assigned similarity as 1.0 to the end. Otherwise, three matchers will be applied. The matching results are aggregated by the aggregation process, which is described in this section.

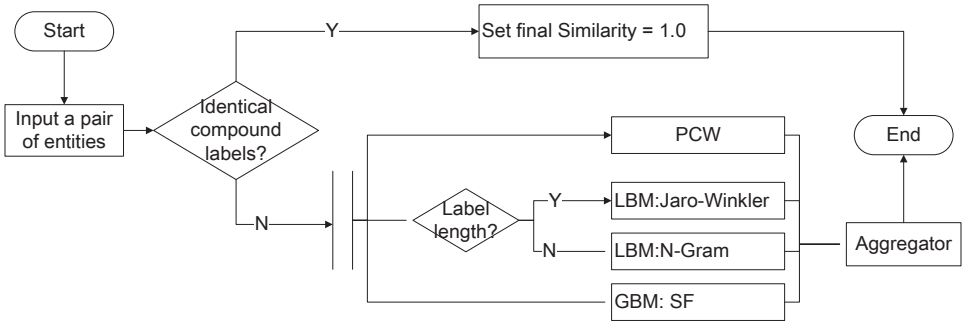


Figure 3.3 Matcher selection

A key issue to apply AHP is the assignment of scale during pairwise comparisons. Normally, the importance is quite subjective and it is not easy to assign a quantitative value. In order to facilitate and automate the assignment of scales, three similarity indicators are proposed from whole-ontology level.

The proposed aggregation process is shown in Figure 3.4. Firstly, three indicators are computed according to source ontologies (§3.4.1). Then AHP generates the priority of matcher, and the priority is adapted as weight of each matcher (§3.4.2). Finally, the final similarity is aggregated based on the weights and similarity values from each matcher (§3.4.3).

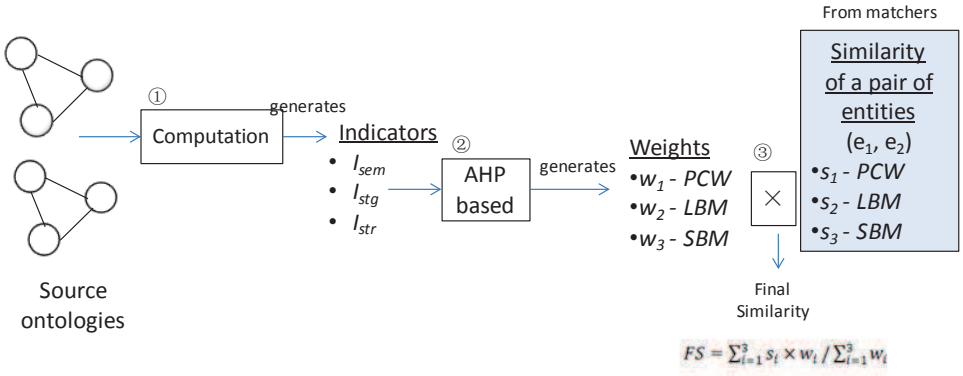


Figure 3.4 Aggregation process

3.4.1 Similarity Indicators

The indicators measure the similarity between two ontologies from a specific aspect for aiding assignment of scales during aggregation process. The three proposed indicators are corresponding to the three matchers for ontology alignment.

Lexical similarity indicator

The indicator I_{stg} reflects the similarity in string between two ontologies. This indicator is adapted from lexical affinity coefficient defined by Pirro and Talia [22]. The indicator is defined in Eq. (3.2), where *#ic* and *#ip* denote the number of classes and properties with identical labels, *#tcp* denotes the total number of classes and properties of one ontology.

$$I_{stg} = \frac{\#ic+\#ip}{min\ (\#tcp_1,\#tcp_2)} \tag{3.2}$$

Structural similarity indicator

The indicator I_{str} denotes the number of nodes with similar structure by investigating its subclasses (hierarchy) and relations (dependency) between two ontologies. Firstly the ontology is treated to a directed graph $G = <V, E>$, V is a set of vertices (or nodes), E is a set of edges with ordered pairs of vertices (v_i, v_j) from V. A vertex v in ontology is described by (*#indegree, #outdegree, #subclass*), for instance, v (*3, 2, 2*) denotes a node that has three in-degrees, two out-degrees and two sub-classes. For one node if the three values are all 0, then this node is regarded as *isolated* vertex, other nodes are called *non-isolated* vertices.

For vertex v in G (converted from source ontology O), if there is a non-isolated vertex v' in G' that has the same *#indegree*, *#outdegree*, and *#subclass*, then the two vertices v and v' are regarded as identical. The indicator I_{str} is denoted in Eq. (3.3**)**, where *#common_ds* denotes the number of identical vertices between G and G' (converted from source ontologies O and O'). $\#niv_1$ *and* $\#niv_2$ denotes the number of non-isolated vertices of G and G' respectively.

$$I_{str} = \frac{\#common\_ds}{min\ (\#niv_1,\#niv_2)} \tag{3.3}$$

Semantic similarity indicator

Semantic indicator I_{sem} is computed based on the tokenized single words (not on the original compound labels). For a tokenized word in entity of source ontology O, if there is a synonym (check with WordNet) in the entity of ontology O', then *#synonym* count adds 1, the number only counts once even there are several synonyms found. It is defined in Eq. (3.4), where *#synonym* is the number of synonyms identified between source ontologies O and O'. This indicator is also used for checking whether two ontologies belong to the same semantic context in homonym checker (see §2.4.3).

$$I_{sem} = \frac{\#synonym}{min\ (\#tcp_1,\#tcp_2)} \tag{3.4}$$

3.4.2 Aggregation Process

The motivation to apply AHP is to learn globally balanced weights for each matcher by considering some major factors, with this idea, in this thesis three indicators have been proposed to reflect the degree of similarity from whole-ontology level. It is believed that these indicators can reflect these factors correctly. One hypothesis is that if two ontologies to be matched have higher similarity (indicator) in one of the three aspect (lexical, structural or semantic), then the matcher proposed from this aspect, has stronger ability than the others in discovering more correspondence. For instance, two ontologies to be matched have many synonyms but with less identical labels, in this case, the semantic indicator I_{sem} will be higher than lexical indicator I_{stg}, then the matcher PCW will be considered having stronger ability in discovering more correspondences then matcher LBM.

This hypothesis is the base for the rules to assign scales using indicators during pairwise comparisons. It is not used as a direct rule to select matchers. The selection of matcher is based on the strategy defined in Figure 3.3. The whole process with AHP aims to learn the weight of each matcher in order to aggregate matching results.

Build AHP hierarchy

Following the AHP process, the goal is defined as: *to find as many as possible valid correspondences* for ontologies to be matched, since the purpose of matcher is to find correspondences. The alternatives are the matching methods described in Chapter 3: semantic-based matcher PCW and non-semantic based matcher: LBM (lexical) and SBM (structural).

As stated in Section 2.3.1, ontology matching performed from the three levels (lexical, structural and semantic) is regarded to be able to find all correspondences. Therefore, the criteria for evaluating the alternatives (three matchers) by respecting to the goal "to find more correspondences" can be defined from the perspective of "ability to find correspondences". Regarding the three levels, three criteria are defined as:

1) The Ability to solve lexical aspects Matching (*AM-stg*);
2) The Ability to solve structure aspects Matching (*AM-str*);
3) The Ability to solve semantic aspects Matching (*AM-sem*).

Each matcher is designed to discover correspondences by focusing on one aspect (lexical, structural and semantic), but in use and in reality these matchers are not completely independent, they are intersected. For example, especially for real ontologies, when two labels of entities are

similar (*bookTitle*, *book_title*), it is probably that they have same meanings and contain similar structures, of course not for all cases. For structural level, it is also this case, if two structures of entities are identified as equivalent, it is possible that the labels are similar (*book_title*, *book_of_title*) and the meanings are close.

Regarding PCW, when two entities are identified as equivalent, it is probable (and it happens often) that the two entities have similar labels and structures. Taking "*book_title*" and "*title_of_book*" for example, PCW will generate a similarity value from somatic level, but from lexical perspective, it will be also applicable. N-gram (see §2.5.1.2) will tokenize them as (*boo*, *ook*, ...) and (.....*boo*,*ook*), which will obtain a similarity value. In this perspective, we can also say that semantic matcher contributes to find lexical correspondences. On the contrary, we can say that lexical matcher contributes to find semantic correspondences.

This is the base and motivation of this work to apply AHP, for this phase, it is quantitative and only some general criteria are defined. In the aggregation process, quantitative criteria are transferred to qualitative values (with help of indicators). According to the above arguments, the AHP hierarchy is built as Figure 3.5 shows. Besides the three criteria, the other criteria of them are assumed equal, such as, accuracy and performance of algorithms.

Following the illustrative example (see Section 3.3.2), the first phase is pairwise comparison ,which involves two steps: (i) comparisons between alternatives (matcher PCW, LBM and SBM) against the criteria (AM-stg, AM-str, or AM-sem), and (ii) comparisons between criteria (AM-stg, AM-str, or AM-sem) against the goal (find as many as possible valid correspondences).

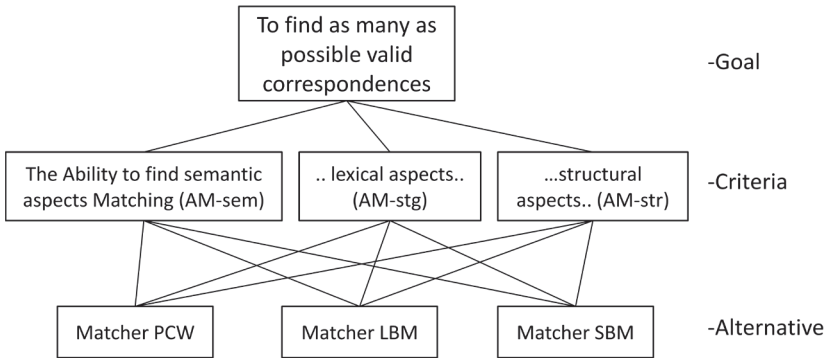


Figure 3.5 Description of goal, criteria and alternatives with AHP

Pairwise Comparison: Alternatives versus Criteria

For the first step, when comparing two alternatives (matcher PCW, LBM and SBM) with respect to one criterion (AM-stg, AM-str, or AM-sem), the value of corresponding similarity indicator (I_{stg}, I_{str}, or I_{sem}) is adopted as the value of scale for this alternative directly. But if the alternative is in the same category with the criteria to compare, then an additional base ratio *br* will be added on the scale of this alternative to enhance the importance of this matcher. The categories are listed in Table 3.11, for instance, AM-sem and PCW belong to the same category (semantic).

Table 3.11 Category of criterion and alternative

Category	Criterion	Alternative
Semantic	AM-sem	PCW
Lexical	AM-stg	LBM
Structural	AM-str	SBM

br is computed based on the three indicators, first we calculate the average value of three indicators, then get the absolute value of the difference between average value and one of the indicator. It is defined in Eq. (3.5), where I_x denotes the value of one of the three indictors (I_{stg}, I_{str}, or I_{sem}), and br_x denotes the basic ratio for this matcher. The calculation of *br* can be adjusted according to experiments.

$$br_x = \left| \frac{1}{3} \sum_{k=1}^{3} I_k - I_x \right| \qquad (3.5)$$

For instance, Table 3.12 shows an example of comparison between alternatives with respect to criterion AM-stg (the ability for solving lexical aspects matching), the assumed values are I_{stg} = 0.3, I_{str}=0.2 and I_{sem}=0.7, these values are taken as scales directly. Because AM-stg and LBM are in the same category, thus basic ratio br_{stg} = 0.1 (according to Eq. (3.5)) is added to its scale.

Table 3.12 Alternatives comparison with respect to criterion: the ability for solving lexical aspects matching (AM-stg)

Alter.	Scale	Alter.	Scale
LBM	$0.4(br + I_{stg})$	SBM	0.2 (I_{str})
LBM	0.4 ($br + I_{stg}$)	PCW	0.7 (I_{sem})
SBM	0.2 (I_{str})	PCW	0.7 (I_{sem})

The result in Table 3.12 is transferred to a matrix for calculating priority in Table 3.13. In this example, the priorities of LBM, PCW and SBM are 0.308, 0.538 and 0.154 respectively. From this example, we can see that matcher (PCW) has higher priority than matcher LBM (lexical) for

ability to match lexical aspects (AM-stg), that is because I_{sem} = 0.7 indicates that the two ontologies have very high similarity in term of semantic aspect. Thus even for the *ability to find lexical aspects*, semantic matcher PCW shows higher importance. Another argument is that the AHP method tries to generate the priorities by balancing all the factors globally, therefore, intermediate results might have small conflicts. This phenomenon is predicted and tolerant.

Table 3.13 AHP alternative comparison matrix with respect to criterion "AM-stg"

AM-stg	LBM	PCW	SMB	Priority
LBM	1	4/7	4/2	0.308
PCW	7/4	1	7/2	0.538
SBM	2/4	2/7	1	0.154

*Inconsistency Factor:0.00

Following the above described process, three criteria (AM-stg, AM-str, or AM-sem) can be evaluated respectively, the results of this step are three generated table (as Table 3.13) with priority of each alternative respect to one criterion. The details of each comparison are not presented in the thesis. The process is the same as above and the example illustrated in Section 3.3 can be referred. The synthesis of the results is presented in next step.

Pairwise Comparison: Criteria versus Goal and Synthesis

In AHP, the next process is to compare the importance between criteria (AM-stg, AM-str, or AM-sem) with respect to the goal, the scales of three matchers are set equally important (see Table 3.14). In this way, the priorities of each criterion (AM-stg, AM-str, or AM-sem) are all 0.333.

Table 3.14 Criteria comparison with respect to the goal " to find as many as possible valid correspondences "

Choose Matcher	AM-stg	AM-sem	AM-str	Priority
AM-stg	1	1	1	0.333
AM-sem	1	1	1	0.333
AM-str	1	1	1	0.333

A final process that synthesizes all these data (three tables generated in previous step and Table 3.14) will be done to generate the final priority of each matcher. Following the illustration described in §3.3.3, an example (the data are assumed for illustration, not linked to data in Table 3.13) is shown in Table 3.15. The priority of each alternative will be taken as the weight of the matcher, namely, the weights for matcher LBM, PCW and SBM are 0.261, 0.178 and 0.261 respectively in this example.

Table 3.15 Synthesis of all alternatives

	Priority with respect to			
Alternative	AM-stg	AM-str	AM-sem	Goal/prio.
LBM	0.1257	0.0071	0.1287	0.2615
SBM	0.0314	0.0185	0.1287	0.1786
PCW	0.1257	0.0482	0.3860	0.5599
Total	0.2828	0.0738	0.6434	1.0000

3.4.3 Calculate Final Similarity

In the final step, the weight of each matcher is obtained. Therefore, the final similarity value *FS* is produced by Eq. (3.6), where v_x is the intermediate value of similarity obtained by each matcher, and w_x is the weight generated with the method introduced in this section.

$$FS = \sum_{i=1}^{3} s_i \times w_i \,/\, \sum_{i=1}^{3} w_i \qquad (3.6)$$

Taking the data from Table 3.15 as an example, the final similarity *FS* equals to (0.560 * S_{LBM} + 0.179 * S_{PCW} + 0.261 * S_{SBM}) / (0.560 + 0.179 + 0.261).

3.4.4 Similarity Cut-off

Matchers are used for measuring the similarity between entities and generating all candidate correspondences. In order to decide which level of correspondences is kept, a limitation is needed to cut off the candidate correspondences. Similarity threshold[11] is a lower limit for the similarity of two entities that belong to the same set. If the similarity is greater than the defined threshold, then the correspondence is valid, otherwise the correspondence is invalid. For instance, if the threshold $t = 0.9$, then all the discovered correspondences, whose values of similarity are greater than 0.9, are taken as valid correspondences, the others are omitted.

The way to decide the value of threshold depends on the particular needs. Usually two ways are used: expert (experiences) - based and calculation - based. A domain specialist can decide the threshold based on his/her experiences, for instance, 0.5 is probably appropriate in the domain

[11] http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.im.model.doc/c_similarity_threshold.html

of wood manufacturing. One of the calculation methods is N-percent. A certain percentage based on the maximal similarity is taken, $th = max(sim(e_1, e_2)) * (1 - np)$, where np is a fixed percentage defined by user. More methods about threshold cut-off can be followed in [87].

3.5 Conclusion

In this chapter, a novel analytic matcher aggregation approach based on AHP has been proposed for matcher aggregation in the domain of ontology alignment. This approach takes the advantage of AHP to assign proper weight of each matcher by evaluating their importance of the specific source ontologies to be matched. AHP uses systematic mathematics computing process to generate the results by taking both quantitative factors and qualitative factors into account.

Compared with the others approaches that applied AHP for weighting (see Table 3.10), the thesis proposed an automatic method to assign the scales (intensity of importance). Scale assignment in AHP is a key and somehow difficult step, since qualitative factors need to be measured with quantitative values. This approach assigns the weight of each matcher automatically and dynamically with three whole-ontology level similarity indicators. This proposed approach facilitates the aggregation process, meanwhile assuring certain precisions.

The aggregation approach, with the matchers described in Chapter 2, is implemented and tested in Chapter 4. The experiment results suggested that the proposed aggregation method improved considerably the precision and recall of the combined matchers, and it has reached expected goals. Concerning the future works on this topic, the ways to calculate the similarity indicators could be improved according to specific needs and application fields.

4> Implementation and Testing

4.1 Introduction

Chapter 2 and Chapter 3 have described the proposed approaches for ontology alignment and matcher aggregation. In order to validate the results and to apply the proposal in the future, a prototype system has been implemented in Java. Following the software engineering life cycle, in this chapter, the implementation and testing of the proposed approaches are described. Section 4.2 presents how the proposals are implemented, including the development environment, data models and implementation of each component. Section 4.3 describes the testing of the implemented system. Firstly the overall testing plan is presented. Secondly, the test cases used are described. The test cases are from OAEI 2012[12]. Thirdly, we analyze and discuss the testing results and draw some conclusions by comparing with the other approaches. Section 4.4 concludes this chapter.

[12] http://oaei.ontologymatching.org/2012/benchmarks/

4.2 Implementation

An alignment prototype system is implemented in Java, in order to test and validate the proposed matching approaches, as well to apply the approach in applications. Approximately five thousands lines of codes have been coded in the system. The system is based on command line, no graphical user interface (GUI). It takes two ontologies in format RDF/OWL and some parameters as input to generate the alignment in XML format. In this section, firstly, it is introduced that the development environment to implement the system and external APIs used. Secondly, considering each main component, class diagrams, sequence diagrams and flowcharts are used to illustrate the implementations. Thirdly, the format of input and output are presented. The command line interface is displayed to show how the system is used.

4.2.1 Development Environment

The prototype system is developed in Java JRE 7, with IDE Eclipse Indigo Sr 1. In the implementation, the external APIs listed in Table 4.1 are applied. These APIs deal with certain specific algorithms and ontology processing. For instance, JDOM is used for XML processing, such as, reading data from XML file and writing data into files. The main functions used of each API are described in the last column of Table 4.1. Two lexical databases are used as listed in Table 4.2. WordNet is for homonyms checker and Lin model. Stanford POS tagger is for tagging words in PCW similarity measurement.

Table 4.1 APIs used in implementation

API	Version	Author(s)	Main functions
JWI [13]	2.2.2	MIT CSAIL	Resolve WordNet
JWS [14]	beta.11	University of Sussex	Resolve WordNet, Lin Model
POS tagger [15]	3.1.5	Stanford NLP	Resolve POS tagger
SFA [38] [16]	2003-11-17	S. Melnik, Stanford	Similarity flooding algorithm
JDOM [17]	1.1.3	jdom.org	XML processing
OWL API [18]	4.2	INRIA	Ontology wrap, read

[13] http://projects.csail.mit.edu/jwi/
[14] http://www.sussex.ac.uk/Users/drh21/
[15] http://nlp.stanford.edu/software/tagger.shtml
[16] http://infolab.stanford.edu/~melnik/mm/sfa/
[17] http://www.jdom.org/downloads/source.html
[18] http://alignapi.gforge.inria.fr/

Table 4.2 Corpora used in implementation

Name	Version	Author	Description
WordNet[19]	2.2.1	Princeton University	English lexical database
Pos tagger	3.1.5	Stanford	english-left3words-distsim

4.2.2 Data Models

The first step to implement a system is to design the data models. UML[20] class diagram is used to model the structure of major components, including entity, correspondence, matcher and alignment. In the class diagram, it shows the attributes, methods and dependency with the other classes.

4.2.2.1 Entity

Figure 4.1 is the class diagram of *entity*. Entity (see Definition 2.1) is the basic element to compose ontology. The considered types of entity in this system are *individual*, *datatype property*, *object property* and *class*, which are defined in enumeration *EntityType*. An entity is composed of some basic information, namely, *label*, *entity type* and *filtered label*. It contains two objects: *core word object* and *vertex object*.

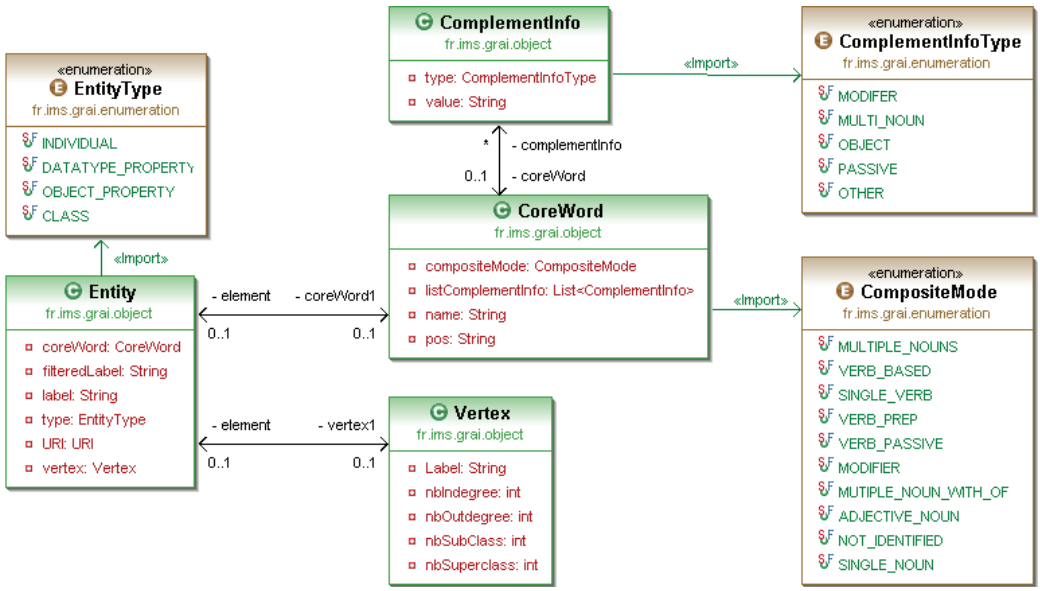


Figure 4.1 Class diagram of "Entity"

[19] http://wordnet.princeton.edu/
[20] http://www.omg.org/spec/UML/2.2/

A *Vertex* is a graphical representation of *entity*, and it will be used to compute structural similarity. An object of *CoreWord* encapsulates the data of a core word. It includes *name*, *pos*, *composite mode* and *a list of complementary information*. The composite modes are defined in enumeration *CompositeMode*, namely, the patterns of composition of labels. One core word includes one to many *ComplementInfo*, which contains the type that is defined in enumeration *ComplementInfoType* and the value of this information. This diagram illustrated how the relations and compositions of entity in ontology are implemented in this system.

4.2.2.2 Matcher

Figure 4.2 shows the class diagram of *matchers*. In order to have high extendibility, the design and implementation of matcher applies an interface *IMatch* and an abstract class *Matcher*. All the concrete matchers must extend the abstract *Matcher* and implement the method *getConfidence()* defined in interface *IMatch*. The method *getConfidence()* returns the value of similarity. The concrete matchers are *LinModel*, *JaroWinkler* (Edit distance), *Ngram*, *PCW*, *SimilarityFlooding* and *HomonymsChecker*. Each of them implements the algorithm that is presented in Chapter 2. Some utility classes are added to aid implementing the algorithms. *CoreWordUtility*, *SFProcessor* and *WordNetUtility* are three utility classes for matcher *PCW*, *SimilarityFlooding* and *HomonymsChecker* respectively.

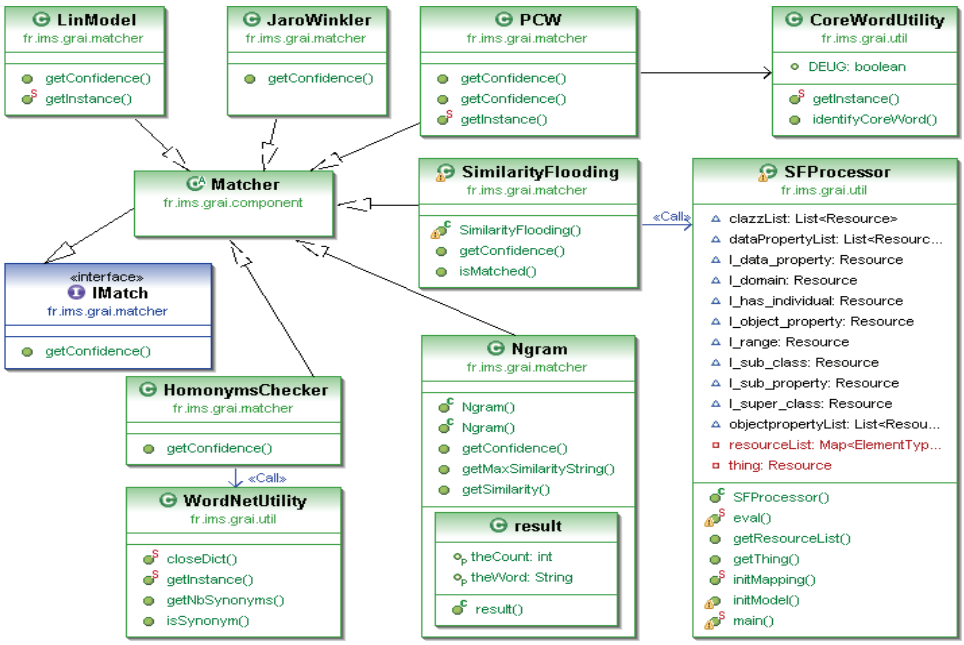


Figure 4.2 Class diagram of "Matcher"

4.2.2.3 Correspondence

Adapted from the representation format of correspondence in Euzenat [88], there are two types of correspondences for multiple matchers - based approach: *intermediate* and *final*. Intermediate correspondence is discovered by a specific matcher, and several intermediate correspondences are combined into a final correspondence. Namely, final correspondences are used for aligning, whereas intermediate correspondences are used for generating final correspondences. An intermediate correspondence *ic* is defined as $ic = \{e_1, e_2, r, v, M, id\}$, where e_1 and e_2 are elements from ontology O_1 and O_2 to be matched. v denotes the confidence between e_1 and e_2 identified by matcher M with a relation r. *id* is a unique identifier for this correspondence.

A final correspondence *fc* is similar to an intermediate correspondence without the information concerning a specific matcher M, defined as $fc = \{e_1, e_2, fr, fv, fid\}$, where *fr* is the relation derived from relations in intermediate correspondences and *fv* denotes a confidence combined from intermediate correspondences' confidences. For both types of correspondences, the relation *r* and *fr* refer to *equal* in this thesis.

Figure 4.3 illustrates the class diagram of *correspondence*. There are two types of correspondences: *InterCorrespondence* and *FinalCorrespondence*. They are extended from basic *Correspondence* (see Definition 2.3), which is a basic class with common data. *Correspondence* includes *unique id*, *confidence*, *relation*, *source entity* and *target entity*.

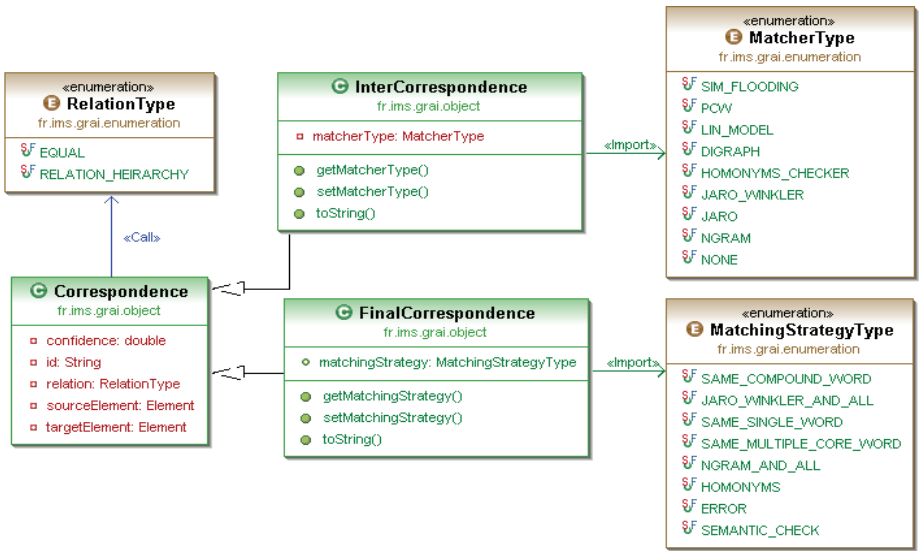


Figure 4.3 Class diagram of "Correspondence"

The types of relations are defined in enumeration *RelationType*: "*equal*" or "*hierarchical*". *InterCorrespondece* is generated by a specific matcher, thus there is an attribute *MatcherType* relating to a specific matcher. This attribute refers to the name of a specific matcher, which generates the intermediate correspondence. *FinalCorrespondece* is aggregated by several *InterCorrespondece*. The attribute *MatchingStrategyType* is used to control the policy of matcher selection and aggregation according to the specific conditions.

4.2.2.4 Alignment

The result of output alignment is represented in XML with certain format. In this thesis, the alignment format refers to Align API [21] provided by INRIA France. The structure of XML document contains mainly three nodes: *onto1*, *onto2* and *map*.

- *onto1*: the URI of the first aligned ontology;
- *onto2*: the URI the second aligned ontology;
- *map*: a correspondence between entities of the ontologies, its value is *Cell*.

The structure of *Cell* is composed of four attributes: *entity1*, *entity2*, *measure* and *relation*. *Entity1* and *entity2* refer to the URIs of matched entity, *measure* is the value of similarity between the two entities with *relation*. An example of *Cell* is as follows:

```
<Cell>
  <entity1 rdf:resource="http://ekaw#Document" />
  <entity2 rdf:resource="http://openconf#Text" />
  <measure rdf:datatype=" XMLSchema#float">0.8639</measure>
  <relation>=</relation>
</Cell>
```

An example of output alignment fragment is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns="http://knowledgeweb.semanticweb.org/heterogeneity/align
ment" xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

<Alignment>
```

[21] http://alignapi.gforge.inria.fr/format.html

```
<onto1>http://nb.vse.cz/~svabo/oaei2011/data/ekaw.owl</onto1>
<onto2>http://oaei.ontologymatching.org/2011/conference/data/O
penConf.owl</onto2>
<map>
    //cells
</map>
</Alignment>
</rdf:RDF>
```

In the implementation, the output alignment is represented by class *MyAlignment*, which is composed of *Cell*. *Cell* is the basic element for denoting one record of alignment; it contains *entities of source ontologies*, *confidence* (measure) and *relation*. *MyAlignment* can be converted to and from XML file with utility class *XMLUtility*. The illustration is shown in Figure 4.4.

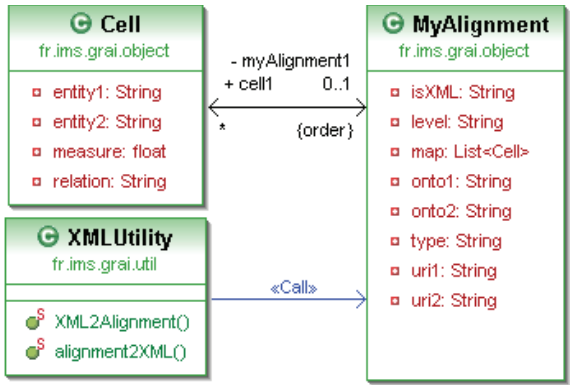


Figure 4.4 Class diagram of "Alignment"

4.2.3 Component Implementation

The prototype system consists of three main components (see Figure 2.5). Sequence diagram in Figure 4.5 is used to illustrate the execution process at component level. Mainly two types of illustration diagrams are used: UML sequence diagram and flow chart.

- *Sequence Diagram* is used to illustrate the sequences of invocation between major classes;
- *Flow chart* [22] is mainly used to illustrate the matching algorithms inside one "*Matcher*" class. The formalism representation can be followed in Chapter 2.

[22] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=11955

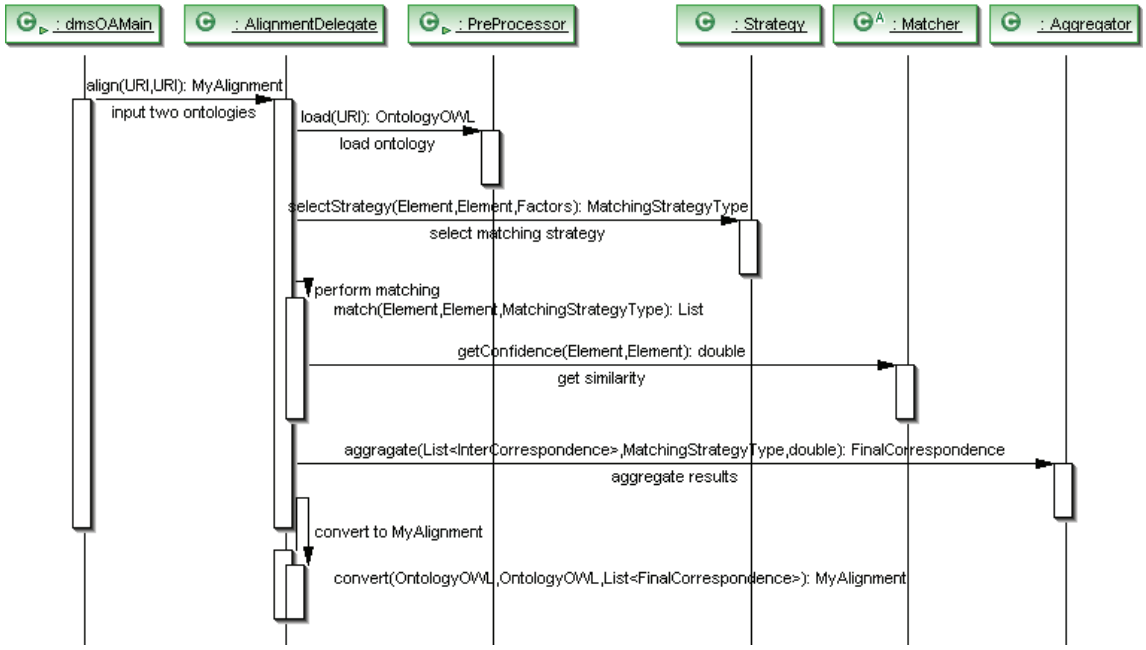


Figure 4.5 Sequence diagram of the main alignment process

1) *dmsOAMain* is the main program. In this program, it inputs two ontologies to be matched and invoke method *align (URI, URI)* from *AlignmentDelegate* to perform ontology matching;
2) *AlignmentDelegate* is a portal class to invoke each component. Firstly, it calls *PreProcessor* to load ontology from file. *PreProcessor* is a class to process source ontology and encapsulate it into an *OntologyOWL* object with method *load (URI)*;
3) Next step is to select a matching strategy according to the inputted pair of entities. The strategy determines which match(s) will be applied and how the results are aggregated. It invokes method *selectStrataegy (Element, Element, Factors)* from class *Strategy*. It returns an object with type *MatchingStrategyType*;
4) *AlignmentDelegate* then will call method *match (Element, Element, MatchingStrategyType)* to perform matching. In this step, specific matchers are selected according to the strategy type. *Matcher* is an abstract class, which is extended by the concrete matching algorithms. Concrete matchers implement method *getConfidence(Element, Element)* to return the value of similarity. The return value of *match (Element, Element, MatchingStrategyType)* is a list of intermediate correspondences, whose similarities are greater than 0;
5) The list of intermediate correspondences is used as the input of method *aggregate (List<InterCorrespondence>)* in *Aggreagtor*, it will generate a final correspondence;

6) In last step, the generated final correspondence is converted to *MyAlignment*. *MyAlignment* follows XML format, it can be exported to XML file and can be used by users according to their own specific needs.

The details of each component can be found in Chapter 2 and Chapter 3. In the follows of this section, the implementation of each part is elaborated.

4.2.3.1 Ontology Pre-processing and Core Word Recognition

Pre-processing concerns mainly loading and encapsulating ontology from file (in format RDF/OWL) into a list of entities. Figure 4.6 shows the process of loading ontology from file into a list of entities, which are for further processing. The format and composition of entity are shown in the class diagram of Figure 4.1. In this process, the major tasks include: (i) loading ontology with OWL API; (ii) setting label, URI and filtered label of entity; (iii) setting entity type;(iv) identifying core word, and (v) composing vertices. In the flowchart, a square with grey background (Identify core word, add to entity) denotes a pre-defined sub-process. The process of core word recognition is illustrated in Figure 4.7.

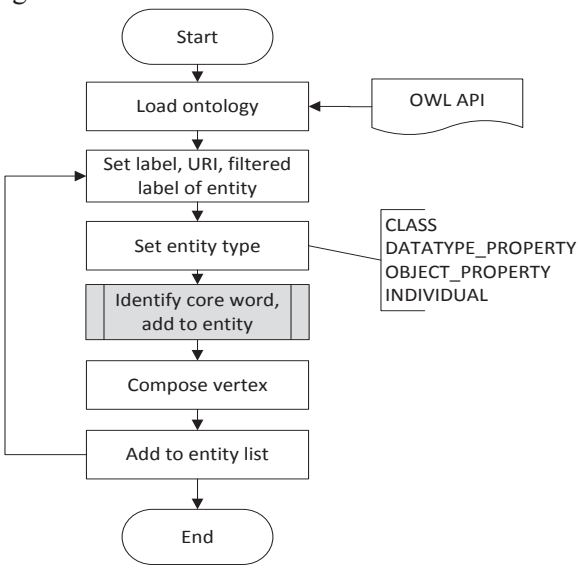


Figure 4.6 Flowchart of loading ontology from file to list of entities

The core word recognition process presented in Figure 4.7 corresponds to the description in Chapter 2 (see Section 2.4). An original label of entity is taken as input. Most of labels are composed of several words with stop words and separators. Firstly, elimination helps to eliminate the unnecessary information that could confuse the matching task. And then tokenization makes the compound word splitted into single ones.

The compound word is tokenized by rules: (i) stop words, e.g. dash, underscore, and dot; and (ii) capitalized word, for instance "*numberOfTelephone*" is tokenized into *number, of, telephone*.

With the help of lexical database POS tagger and API:POS, the tokenized words are tagged and their POS are identified. The tagging rules of POS are defined in Chapter 2 (see Section 2.4). Meanwhile, the position of each tagged word and number of each category are noted. The formalizing representation is as follows:

$[(w_1, pos_1, position_1), (w_2, pos_2, position_2), \ldots, (w_n, pos_n, position_n)]$.

The final step is to note the complementary information of each label in order to calculate the semantic similarity.

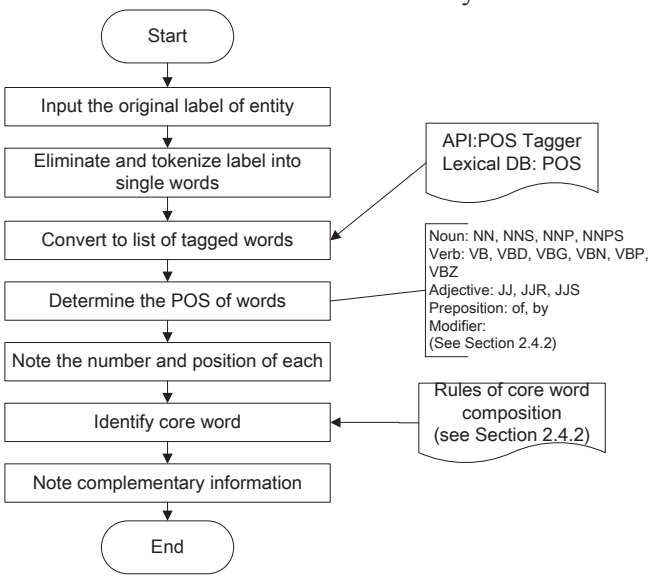


Figure 4.7 Flowchart of core word recognition

4.2.3.2 PCW Similarity Measurement

Matcher PCW uses core word to measure the similarity between a pair of entities. To illustrate the implementation of similarity algorithm of PCW, three flowcharts (Figure 4.8, Figure 4.9, and Figure 4.10) are used. Figure 4.8 describes the algorithm of homonyms checker, which is part of matching algorithm SMA. SMA is the major component in PCW. The three parts are presented one by one as follows.

Homonyms Checker (see Figure 4.8) is used to measure the similarity of homonyms. Homonyms are the same words that represent different meanings in different context. Firstly, we check whether the semantic similarity indicator is greater than threshold. If yes, it refers that the two words represent the same meaning, then the similarity is assigned as 1.0. Otherwise, we calculate the similarity using the number of homonyms

that it has. *#m* is the number of homonyms retrieved from WordNet. In order to read this number, API:JWI is used. Finally, we use formula (*#m*-1) / *#m* to compute the similarity.

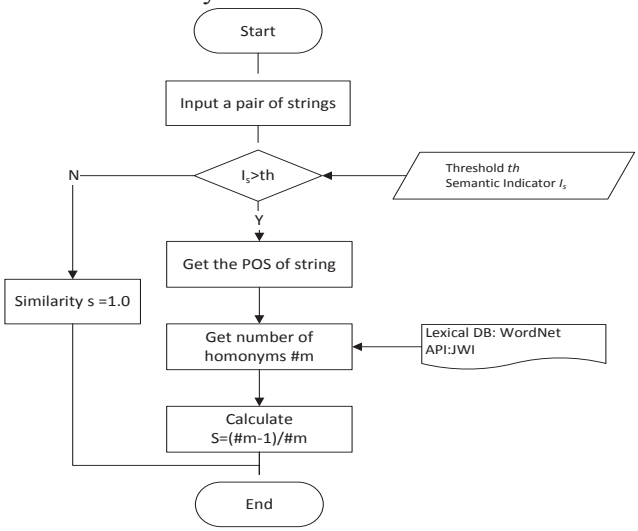


Figure 4.8 Flowchart of homonyms checker

SMA (see Figure 4.9) is used to measure the similarity between a pair of single words s_1 and s_2. First step is to check whether the two strings are homonyms, if yes, the process illustrated in Figure 4.8 will be applied. Otherwise, Lin Model is used to measure the similarity. In the implementation of Lin Model, WordNet and API:JWS are applied.

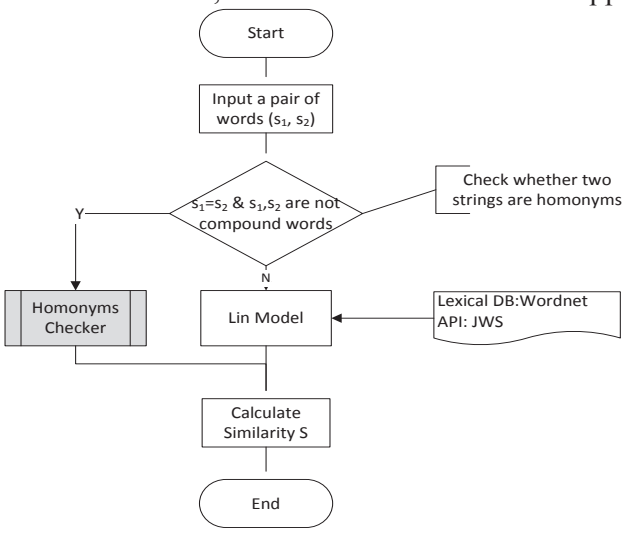


Figure 4.9 Flowchart of matching algorithm SMA

PCW (see Figure 4.10) is for measuring the similarity between a pair of entities. It contains two parts: core word part M_1 and complementary information part M_2. The two parts take up respectively 70% and 30% weight of final result respectively. M_1 represents the similarity between

core words and M_2 represents the part of complementary information. M_2 is produced by accumulating results of each pair of complementary information.

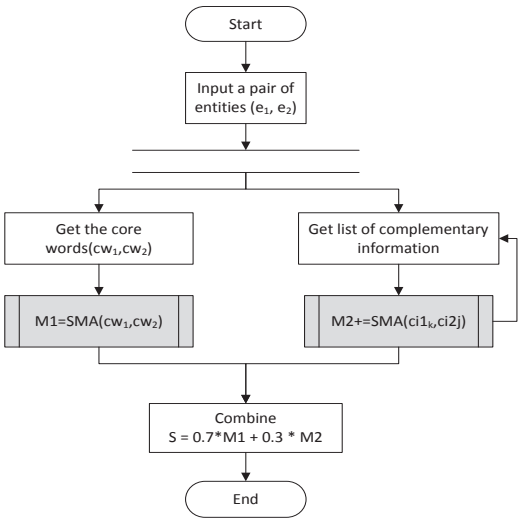


Figure 4.10 Flowchart of matching process of matcher PCW

4.2.3.3 Edit Distance(ED) and N-Gram (NG)

ED and NG measure the similarity of a pair of entities from lexical level. The process is illustrated in flowchart of Figure 4.11. A condition to check the length of strings is examined to decide which algorithm to use. The details of matching algorithms about N-gram and edit distance have been described in Chapter 2 (see Section 2.5).

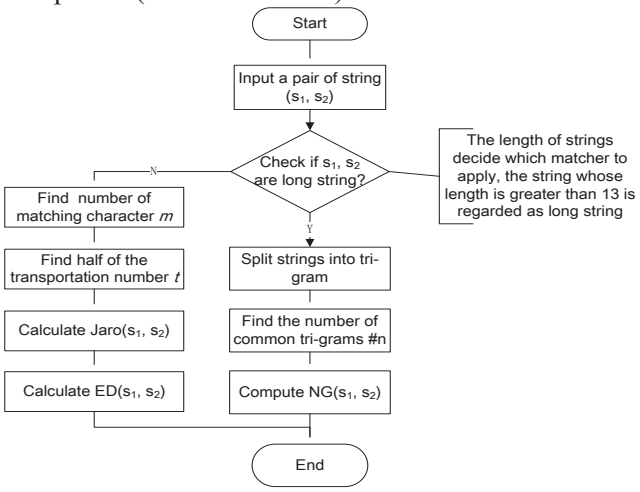


Figure 4.11 Flowchart of lexical matchers ED and NG

4.2.3.4 Similarity Flooding (SF)

Similarity flooding algorithm (SFA) is applied to perform ontology matching from structural level. The approach is described in §2.5.2. Flowchart of Figure 4.12 illustrates the implementation of SFA. Two ontologies are taken as input and converted to similarity propagation graph (SPG). With an initial mapping node set, the algorithm starts to seek the correspondences. In this process, API:SFA is invoked to execute the algorithm and to generate the mapping results. The mapping results are encapsulated into correspondences of ontology alignment.

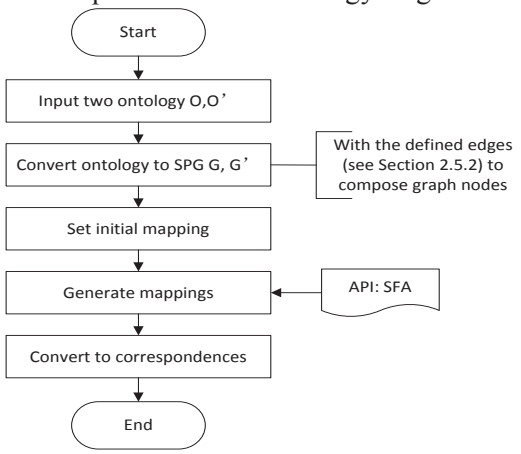


Figure 4.12 Flowchart of similarity flooding algorithm

4.2.3.5 Aggregation Process

The classes implemented for aggregation process are shown in Figure 4.13. The sequence of invocation is shown in Figure 4.14. *AlignmentDelegate* calls method *getFactors()* in *FactorCalculator* to generate *Factors (indicators)* according to the inputted source ontologies. *AHP* uses the generated indicators to calculate the weights of each matcher. It returns object *Weight* to aggregator. *Aggregator* uses the weights to combine the similarity of intermediate correspondence and generates the final correspondence.

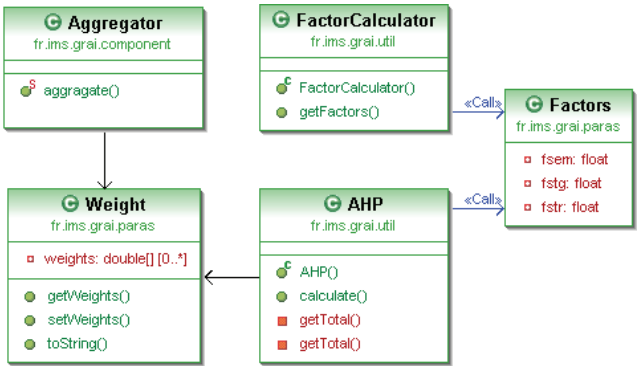


Figure 4.13 Class diagram of aggregation process

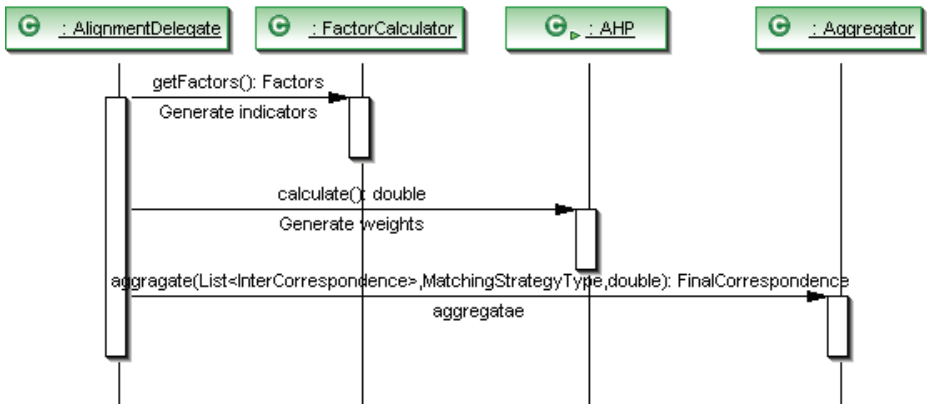


Figure 4.14 Sequence diagram of aggregation process

4.2.4 Use of System

The use of this system is relatively simple. The application takes a few parameters and generates the alignment file. The matching process is encapsulated in the program and is transparent to users. No graphical interface is available to use. The program accepts the parameters and runs the application. The format of parameters is as *[source ontology o_1, source ontology o_2, threshold, generated alignment file path]*. The first two parameters are the file path of ontologies to be matched. The files should be in RDF/OWL format. The third parameter is the threshold for filtering the discovered correspondences. An example of input parameters is as follows. A screen snapshot is shown in Figure 4.15.

```
[http://oaei.ontologymatching.org/2012/conference/data/ekaw.owl,
http://oaei.ontologymatching.org/2012/conference/data/OpenConf.owl,
0.7,
D:\\alignment.xml]
```

Figure 4.15 Screen snapshot of input parameters

A XML file following RDF format is generated and the fragment of the file is shown as follows.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns="http://knowledgeweb.semanticweb.org/heterogeneity/align
ment" xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
    <Alignment>
        <xml>yes</xml>
        <level>0</level>
        <type>11</type>
        <onto1>http://ekaw</onto1>
        <onto2>http://openconf</onto2>
        <uri1>http://ekaw</uri1>
        <uri2>http://openconf</uri2>
        <map>
            <Cell>
                <entity1 rdf:resource="http://ekaw#Document" />
                <entity2 rdf:resource="http://openconf#Text" />
                <measure   rdf:datatype="http://www.w3.org/2001/
XMLSchema#float">0.9</measure>
                <relation>=</relation>
            </Cell>
        </map>
        <map>
            <Cell>
            <entity1 rdf:resource="http://ekaw#Industrial_Paper" />
            <entity2 rdf:resource="http://openconf#Paper" />
                <measure   rdf:datatype="http://www.w3.org/2001/
XMLSchema#float">0.7</measure>
                <relation>=</relation>
            </Cell>
        </map>
……
    </Alignment>
</rdf:RDF>
```

The generated alignment in XML format can be reused and processed according to the specific demands of users, for instance, reading correspondences from the file and setting up link between entities. In the section of experiment, the alignment files are used to compare with reference alignment and generate the evaluation results. In Chapter 5, the implemented prototype system is applied to address the heterogeneous data sources querying. The generated alignment file is used to link the different ontologies with the help of built-in constructs in OWL.

4.3 Experiment

Experiment is carried out based on test cases from OAEI 2011 benchmarking[23]. An overview of testing plan is presented in Figure 4.16. The basic testing case unit is a "*Test Case*" that contains *source ontology 1, source ontology 2* and *reference alignment*. A test set of test cases includes one to many (1..*) test cases that are categorized into one group.

The testing adopts black-box testing [89] to run the test cases and observes the results. Black-box testing is a software testing technique to examine the functionality of system without peering into the internal structure of it. As shown in Figure 4.16 for each test case in a set, the system will call *AlignmentDelegate* to load "source ontology 1" and "source ontology 2" and generate the discovered alignment. Then, the system evaluates the discovered alignments by comparing the results with "reference alignment" and generates the precisions and recalls. Reference alignments are provided by OAEI. The results of all test cases in one set will be aggregated via an average or harmonic mean method, eventually to get evaluation results of one test case set.

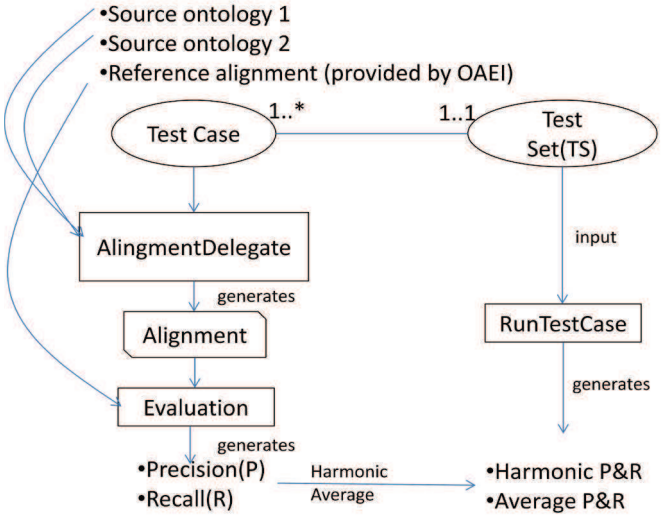


Figure 4.16 Testing plan

4.3.1 Measurements

Three measurements [90] are used to evaluate the experiments results: harmonic precision (*HP*), harmonic recall (*HR*) and harmonic F1-measure

[23] http://oaei.ontologymatching.org/2011/benchmarks/

(*HF1*). *Precision* measures the ratio of correctly *found correspondences* over the total number of *returned correspondences*, and *recall* measures the ratio of *correctly found correspondences* over the total number of *expected correspondences*. In logical term, precision and recall are supposed to measure the *correctness* and *completeness* of method respectively. F1-measure combines and balances between precision and recall. These measurements are defined in Eqs. (4.1). The set of alignments identified by the approach described in this thesis is denoted as A_d, and the set of reference alignments provided by OAEI is denoted as A_r. $|A_r|$ denotes the number of correspondences.

$$P = \frac{|A_d \cap A_r|}{|A_d|} \quad , \quad R = \frac{|A_d \cap A_r|}{|A_r|}, \text{and} \quad F1 = \frac{2 \times P \times R}{P + R} \tag{4.1}$$

Harmonic mean[24] is used in OAEI for comparison among different approaches. Harmonic mean measurements are denoted in Eqs. (4.2), where A_{di} and A_{ri} are the i^{th} set of alignment discovered by the approach described in this thesis and reference alignment from OAEI respectively.

$$HP = \frac{\sum_{i=1} |A_{di} \cap A_{ri}|}{\sum_{i=1} |A_{di}|}, \quad HR = \frac{\sum_{i=1} |A_{di} \cap A_{ri}|}{\sum_{i=1} |A_{ri}|}, \text{and} \quad HF1 = \frac{2 \times HP \times HR}{HP + HR} \tag{4.2}$$

4.3.2 Test Cases

OAEI is an initiative contributing to assess the ontology alignment systems and approaches. Since 2004, it has organized 11 campaigns using systematic testing data set. Data set **biblio** has been used since 2004 and the seed ontology concerns bibliographic references, which contains 33 named classes, 24 objet properties, 40 data properties, 56 named individuals and 20 anonymous individuals. The data sets are systematically generated based on the seed reference ontology by discarding a number of information in order to evaluate how the algorithm behaves when this information is lacking. The information includes name, comment, specialization hierarchy, instance, property and class.

Test sets are grouped into five test cases TS1 to TS5 as listed in Table 4.3. Test case TS1 contains three ontologies with small changes in labels and structure. Test case TS2 contains ten ontologies with same structure and different lexical labels. Test case TS3 has many variations in

[24] http://mathworld.wolfram.com/HarmonicMean.html

structure. Test cases #248 to #266 (TS4) have variations in both aspects, especially the labels are randomly generated strings. Test cases #301 to #304 (TS5) are four real-life ontologies created by different institutions.

Table 4.3 Benchmarking data set "biblio"

Test Set (TS)	Tata cases	No. of onto.	Description
TS1	#101 - #104	3	Simple ontology
TS2	#201 - #210	10	Variations in lexical aspect
TS3	#221 - #247	18	Variations in structural aspect
TS4	#248 - #266	15	Both aspects
TS5	#301 - #304	4	Real ontology

4.3.3 Results and Discussions

The computer for running the test is a laptop, model Dell E5510, OS Windows 7, 2G RAM, P4500 1.87GHz. The tests are performed and discussed in terms of three aspects: (i) run the tests using each single matcher, and then aggregate the results with two kinds of methods: simple average and AHP-based method, in order to observe the results of each matcher and AHP-based aggregation approach; (ii) analyze the result of each test set; (iii) compare the results with the other approaches that have used the same benchmarking test sets. The results data of the other approaches are from the published OAEI 2011 report [91].

Single matcher and AHP-based aggregation

In order to measure the results of combined method with AHP, at first each single matcher is evaluated. Only one matcher is applied to run the test cases. As shown in Table 4.4, the first column is the number of test cases, and then the following three columns represent the results of each single matcher: LBM, PCW, and SBM. The fifth column is the simple average value of the results obtained by three matchers without applying AHP. The last column represents the results generated by combining all the matchers with AHP. The running time of all five test cases is listed in the last row of Table 4.4. The running time of each matcher is 2 m 33 s, 5 m 33 s, and 9 m 24 s respectively. The AHP-based method uses 13 m 17 s.

Generally, considering all test sets, the table suggests that SBM has better matching results than PCW, and PCW has better matching results than LBM. Regarding *HF1* of each test set, PCW has better performance on TS5 (0.55) that is based on real ontologies and TS4 (0.32) that has both variations in label and structure. SBM obtained better results on test set TS2

(0.74). For TS1 and TS3, the three matchers obtained approximate results. The results suggest PCW has good matching ability, especially on real ontologies, it is one of the expected goals when proposing this approach.

Table 4.4 Results of single matcher and combined matchers

#	LBM (ED+NG)			PCW			SBM(SF)			Simple average			With AHP		
	HP	HR	HF1	HP	HR	HF1	HP	HR	HF1	HP	HR	HF1	HP	HR	HF1
TS1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TS2	0.69	0.45	0.54	1.00	0.44	0.61	1.00	0.59	0.74	0.90	0.49	0.64	0.99	0.70	0.82
TS3	0.96	1.00	0.98	1.00	1.00	1.00	0.99	1.00	1.00	0.98	1.00	0.99	1.00	1.00	1.00
TS4	0.25	0.02	0.03	0.58	0.22	0.32	0.96	0.19	0.31	0.60	0.14	0.23	0.96	0.34	0.50
TS5	0.60	0.44	0.51	0.99	0.38	0.55	0.72	0.35	0.47	0.77	0.39	0.52	1.00	0.60	0.75
HM	0.70	0.58	0.64	0.91	0.61	0.73	0.93	0.63	0.75	0.85	0.61	0.71	0.99	0.73	0.84
R. time	2m33s			5m33s			9m24s			17m30s			13m17s		

With simple average method, the results of *HP*, *HR* and *HF1* are 0.85, 0.61 and 0.71 respectively. The results are better than LBM and worse than both SBM and PCW. However, with AHP aggregation, the results are much better improved. The value of HF1 on each test set is better than the results of any matcher. The overall *HF1* reached 0.84. The results suggest that AHP-based aggregation improved the combined matching results obtained by different matchers. The expected goal is achieved.

Analysis on each test set

Table 4.5 summarizes the results concerning all the data sets with three proposed matchers and AHP aggregation. The results on TS4 are lowest on all matchers. On the contrast, TS1 and TS3 obtained quite high matching results, the HF1 measure is approximately 1.0. The results of combined matchers are better than each single matcher on all test cases. Its precision remains very high and the average value is 0.99. The recalls of TS4 and TS5 are relatively low, consequently, the HF1 measure on the two data sets are relatively low. The average HF1 of all test cases is 0.84. Compared with the results of each single matcher, the value is higher by 0.19, 0.11 and 0.09 respectively.

The table suggests that the overall proposed matching approach (including three matchers and AHP-based aggregation) has good matching ability to solve lexical aspect (TS2), structural aspect (TS3) and real ontologies (TS5). The matching ability to address both lexical and structural aspects (TS4) at the same time is relatively weaker.

Table 4.5 Results of test cases TS1 to TS5 with AHP

Test Set	HP	HR	HF1
TS1	1.00	1.00	1.00
TS2	0.99	0.70	0.82
TS3	1.00	1.00	1.00
TS4	0.96	0.34	0.50
TS5	1.00	0.60	0.75
Average	0.99	0.73	0.84

Comparisons with the other approach

HF1-measure is used to compare with the other approaches. The data of the other approaches is from OAEI campaign 2011 [91]. There were 15 participants, and two of which, AgrMaker and MapSSS, had no results in **biblio** benchmarking. The comparison is displayed in Figure 4.17. The last column (msOA) is the result obtained by the approach presented in this thesis. It ranges second (out of 14), the HF1 is 0.84 and is lower than the first (YAM++ [52], 0.86) by 0.02 and higher than the third (CSA [53], 0.83) by 0.01. YAM++ and CSA have been surveyed in Chapter 2 (see Table 2.1).

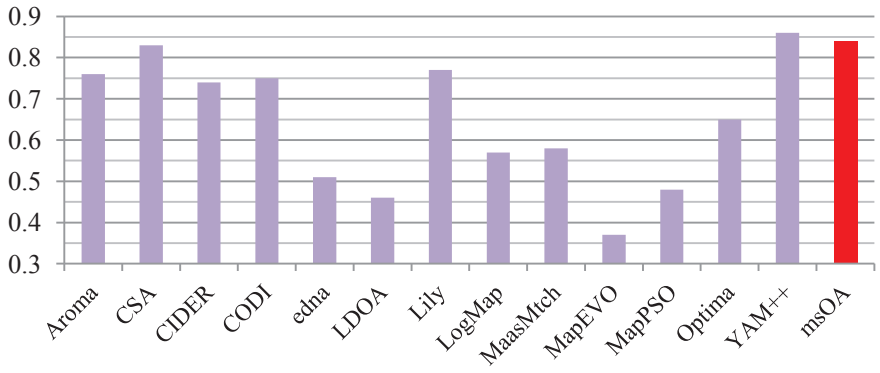


Figure 4.17 Comparison of different approaches with HF1

YAM++ is a matching system of extension YAM [92] based on machine learning and similarity flooding. Figure 4.18 illustrates the architecture of YAM++, after parsing and processing, the source ontologies are passed to matching systems that consist of two levels: element level and structure level. The results obtained from the two levels are combined by weighted mean aggregation after filtering the discovered correspondences by a few concrete rules. At last, in term of consistency a verification step is carried out to check again the combined correspondences by two rules.

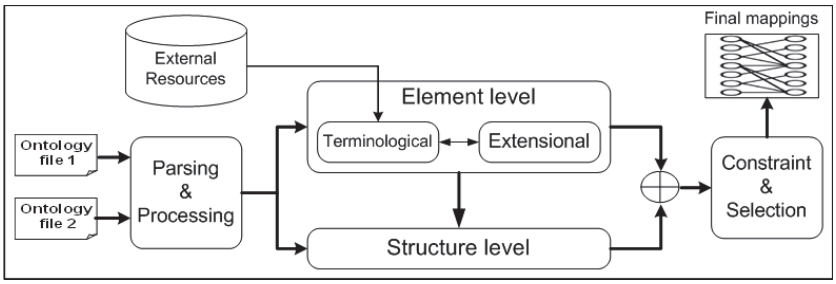


Figure 4.18 YAM++ system architecture ([52])

According to the above description and architecture shown in Figure 4.18, some comparisons are discussed as follows:

- YMA++ seeks the correspondences from terminological level and structural level, no matching techniques from semantic level (such as, WordNet, semantic learning) are used. While the approach proposed in this thesis focus on discovering the correspondences from semantic level, as well two matchers from lexical level and structural level are proposed to complement the semantic matching. Thus it is believed that our approach has stronger matching ability in term of semantic level, which is one of the main objectives of ontology alignment on real-life ontologies;
- YAM++ involves many pre-defined rules to constraint the process of selection and combination, such as, rules in extensional matcher, rules for filtering the discovered correspondences and final selection. The side effects of many rules involved are less flexible in term of adapting diverse situations and complicated to use the system. In this thesis, the whole matching and combination process is automatic in terms of use and adaptation;
- The combination of discovered results by matchers in YAM++ depends on weighted aggregation method on two filtered list. The combination process is somehow controlled by rules and pre-defined constraints. The combination process in this thesis is automatic based on AHP considering multiple factors and tries to generate a globally optimized weighting result.

4.4 Conclusion

The first part of this chapter elaborated the implementation of the prototype system for the proposed matchers and the AHP-based aggregation method. The prototype system is implemented in Java and consists of approximate 5000 lines of codes. The implemented system ran stably and generated expected alignment results.

The second part of this chapter carried out the testing and discussed the results. The evaluation results suggested that the proposed matchers have good matching ability, especially on real-life ontologies, and the AHP-based aggregation method contributes significantly to improve the matcher aggregation. The expected goals of these proposals are fulfilled. Compared with 13 other approaches based on the OAEI 2011 benchmarking data sets, the measurement HF1 of the proposed approach ranges second, its value is 0.84 and is lower than the first (YAM++ [52], 0.86) by 0.02. These results suggest that the proposed approach processes good matching ability.

Concerning the further work that could be done to improve the proposed approach, the possible tracks include: (i) the rules in PCW can be enriched and extended to adapt specific context and needs; (ii) to reduce the system running time, the Java algorithm implementation can be re-factored and improved to make the system more efficient; (iii) as for the AHP-based aggregation process, the initial scale for pairwise comparisons and basic ratio (br) can be adjusted based on the experiments and training data, instead of manual assignment, in order to improve the aggregation results; (iv) the computation method of similarity indicators can be extended to adapt various source ontologies and context.

The results would be improved with these above potential work implemented. The proposed ontology alignment approach and implemented prototype system have been applied to an application for querying data from multiple relational databases. This part is elaborated in Chapter 5.

5> Ontology Alignment to Support Enterprise Interoperability

5.1 Introduction

Ontology alignment, as a method to find semantic correspondences, needs to be combined with the other processes and components for developing interoperability of enterprise information systems. Depending on different requirements, ontology alignment can be applied in different ways. In this chapter, an ontology-driven architecture is developed using the proposed ontology alignment approach as one of the main components. The objective is to query data from multiple Relational Database (RDB) systems that are used in enterprises, so that to enable enterprise data interoperability at semantic level. The architecture focuses on RDB as data sources, since RDB is the main data storage type that is being widely used in enterprises.

In order to implement the solution that queries data from multiple RDBs, the main task is to construct *Semantic Information Layer (SIL)* from RBD. Three main steps are involved to develop SIL: ontology extraction, ontology enrichment and ontology alignment. The relevant suggested methods and techniques are presented in this chapter. This general overview is introduced in Section 5.2, including the general architecture, architecture components and suggested techniques. Section 5.3 describes an illustrative

example in the domain of mobile phone to show the feasibility and steps of the approach, also it draws some discussions of the proposed architecture in terms of potential applications and challenges. Section 5.4 summarizes this chapter.

5.2 Construct Semantic Information Layer with Ontology Alignment

Relational Database (RDB), as a main storage of information systems in industry and enterprise, often has interoperability issues when collaborating with the other systems. Concerning the issue of heterogeneity between information systems, there are two significant issues: (i) semantic interpretation of same concepts between two systems; (ii) common mediation that allows communication between two sides. Referring to the roles of ontology played in enterprise interoperability (see §1.3.3), ontology is adopted as a solution to address these issues.

To extend the narrow scope of pure ontology and adapt to enterprise systems, an enlarged information layer with semantics representation called *Semantic Information Layer (SIL)* is proposed. SIL refers to an information layer with semantic representation, interface to interact with upper level applications and access to lower data source. SIL differs from a single pure ontology, besides taking ontology as main storage of data, SIL possesses the ability to be accessed from user level and maintains the connections to data sources.

During creation of SIL, ontology alignment plays two key roles: (i) building semantic links between equivalent concepts from two sides; (ii) creating a common media for data querying from multiple data sources. These two roles correspond exactly to the two issues in solving heterogeneity among information systems.

A practical ontology-driven architecture is proposed for building SIL to enable heterogeneous data querying among multiple information systems, eventually to achieve interoperability among different enterprise systems. The data source is considered to be relational database (RDB), the other kinds of data sources are not discussed in this thesis. The architecture serves as reference methodology to develop SIL. In real implementation, the specific techniques used can vary depending on particular demands.

In the following sections, first it is presented that the background, the general structure of SIL, major components and steps to build SIL in §5.2.1. §5.2.2 and §5.2.3 describe ontology extraction from databases and ontology enrichment respectively. §5.2.4 presents the OWL built-in constructs, which use the correspondences generated by the proposed ontology alignment approaches and prototype system, to build links

between the extracted ontologies. §5.2.5 illustrates the mapping path and query implementation.

5.2.1 Overview

To build SIL, the first issue is developing semantics from RDB. Ontology extraction, ontology enrichment and ontology alignment are three main steps involved. Ontology extraction and enrichment will map the schema and populate records into ontology. Ontology alignment links and maps concepts between the extracted ontologies. The second issue is how to make the SIL accessed by upper applications. Mapping representation and query implementation are two components to enable this. Mapping representation maintains connections between SIL and RDB, and with query language, the information will be retrieved either from SIL directly or from RDB via mapping representation.

An architecture, which includes the main components and steps to develop SIL, is proposed in Figure 5.1. It is designed in the context of a solution for EIS interoperability using semantic web technologies [93]. In this architecture, the main components are data sources (RDB), SIL, query and mapping path, user interface and upper application. Except SIL, mapping path and query implementation, the other parts are not detailed in this thesis. The main steps to develop SIL are ontology extraction, ontology enrichment and ontology alignment. In the follows of this section, firstly the three steps to develop SIL are presented, and then the mapping path and query implementation are elaborated.

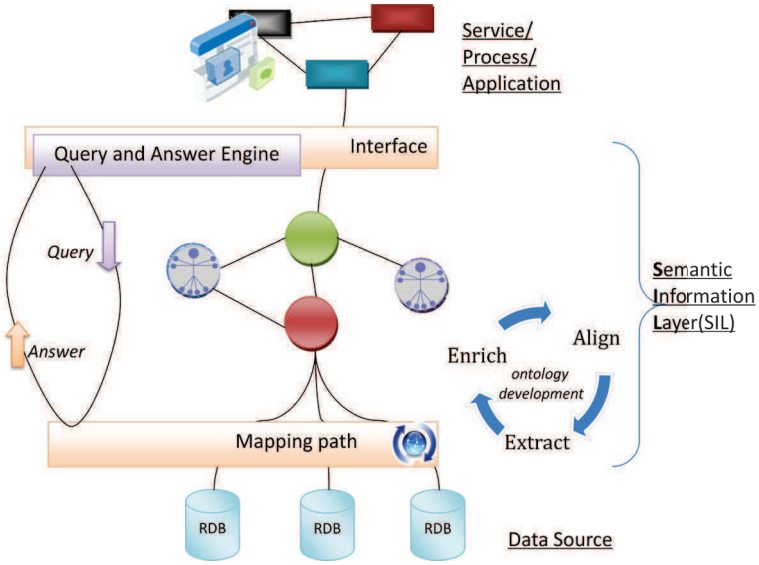


Figure 5.1 Architecture for building Semantic Information Layer (SIL)

5.2.2 Ontology Extraction

Ontology extraction is a pre-process for developing the semantics. The extracted ontology from RDB is regarded as "*raw ontology*", which needs to be further enriched and elaborated by reference ontology or manual adjustment. The two processes are illustrated in Figure 5.2. Raw ontology is extracted with pre-defined rules, and then enriched with upper or reference ontology to obtain final ontology. Ontology extraction mainly involves two parts: schema extraction and instance population.

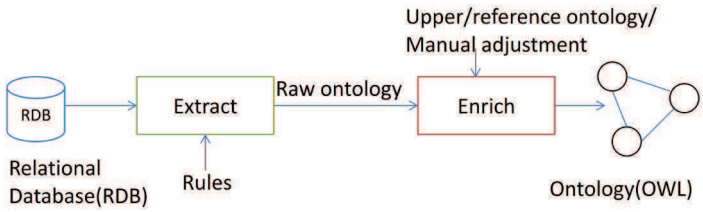


Figure 5.2 Ontology extraction and enrichment

5.2.2.1 Schema Extraction

Schema extraction rules are classified as explicit and implicit. Explicit rules are the rules defined from structural aspects. The mappings between RDB and ontology are set up directly. For instance, a *table* in RDB is mapped to *class* in ontology, and *field* in RDB to *property* in ontology. In addition, some implicit rules are defined in [94, 95]. The implicit rules mine the hidden information from RDB, which is not explicitly available. For instance, Cerbah [96] uses hierarchy mining to discover relation *subClassOf* and Rodriguez [97] extract *sameAs* property using similarity-based method. We summarized and adapted some commonly used rules from [98-100] as listed in Table 5.1, which could describe most of the semantic relation in database schema extraction.

Besides these rules, a few mapping languages are proposed to enable the mappings between RDB and different ontology representation. Hert et al. [101] gave a comprehensive comparison among different main mapping languages. W3C is working on standardizing the mapping language from RDB to RDF, and a named R2RML[25] ongoing work is being carried out. This mapping language would contribute to the ontology extraction from RDB in the future.

[25] http://www.w3.org/TR/r2rml/

Table 5.1 Adapted ontology extraction rules

Rule type	From RDB	To ontology(OWL)
explicit	Table	class
explicit	Column	property
explicit	Row	individual/instance
explicit	Primary key	owl:inversFunctionalProperty owl:minCardinality rdf:datatype="&xsd:int" 1/
explicit	Foreign key	owl:objectProperty
explicit	Check	owl:hasValue
explicit	Unique	owl:inversFunctionalProperty
explicit	Not null	owl:minCardinality rdf:datatype="&xsd:int" 1/
implicit	Similarity check	owl:sameAs
implicit	Hierarchy mining	rdfs:subClassOf

5.2.2.2 *Instance Population*

Ontology population is the process to transform data from database into ontology instances. There are two general approaches to do it, normally regarded as static and dynamic or massive dump and query driven [102] as illustrated in Figure 5.3. Previous case uses a batch process to transform all the database records into ontology instances (Figure 5.3 a)), whereas dynamic approach (query driven) only transforms part of database data in response to certain queries when requests are made (Figure 5.3 b)). Byrne [103] and Green et al. [104] are such application cases for static and dynamic population respectively.

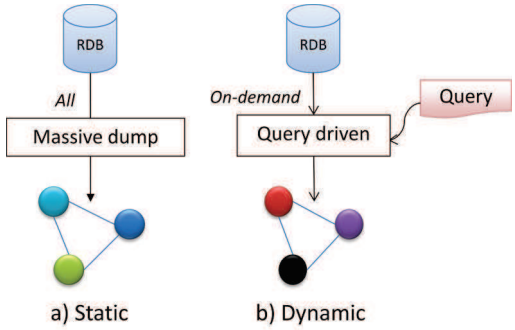


Figure 5.3 a) Static mode and b) Dynamic mode in instance population

Current research work is towards enabling automatic extraction, since manual ontology building is time-consuming and error-prone. Even though many promising extraction rules and methods have been proposed and defined, the full-automatic extraction tools are unsatisfactory when applying to real enterprise databases. Normally a hybrid way is used,

so-called semi-automatic approach by combining automatic process and manual involvement. More details could be found from previous work, Park et al. [105] evaluated some tools with a comprehensive evaluation framework. Sahoo et al. [106] surveyed current approaches and tools implemented for mapping from RDB to RDF.

5.2.3 Ontology Enrichment

In engineering and manufacturing domain, some reference ontologies have been developed for conceptualizing domain knowledge. These ontologies can be used as base to enrich and elaborate the extracted ontology. Suggested Upper Merged Ontology (SUMO) [107] and Object-Centered High-level Reference Ontology (OCHRE) [108] are two ontologies for general concepts in any domain. Infrastructure Product Ontology (IPD) [109] and ONTO-PDM [18] emphasize on the domain of product and service. In IPD, products span five sectors of utilities: water, wastewater, gas, electricity, and telecom. In manufacturing domain, MASON [110] as well as Lagos and Setchi [111] are two propositions.

A few methodologies have been proposed to enable automatic enrichment. Navigli and Velardi [112] proposed an approach to enrich ontology and annotate documents automatically based on semantic annotation of on-line glossaries. Zouaq et al. [113] used upper ontology SUMO as a base to extract knowledge and analyze semantics from text. Oltramari and Stellato [114] presented a framework to evaluate integrations between ontological and linguistic resources, it defines sets of shared vocabularies for the knowledge about heterogeneous linguistic resources.

5.2.4 Ontology Alignment

In this step, the extracted ontologies will be used as source ontologies to perform ontology alignment. The alignment approaches are the methods that have been described in Chapter 2 and Chapter 3. The prototype system implemented in Chapter 4 is used to generate the correspondences of alignment in format of XML.

OWL provides some built-in constructs to link equivalent entities as listed in Table 5.2. *owl:equivalentClass*, *owl:equivalentProperty* and *owl:sameAs* are such kinds of constructs. *owl:equivalentClass* links a class description to another class description, and this axiom requires that the two class descriptions contain the same class extension. *owl:equivalentProperty* is used to state that two properties have the same extension. *owl:sameAs* links an individual to another individual. It means that the two URI

references refer to the same entity and they have the same identity. The three constructs will be used to setup the links between different entities with the discovered final correspondences.

Table 5.2 OWL constructs used for linking entities

OWL Constructs	Entity type
owl:equivalentClass	Class
owl:equivalentProperty	Property
owl:sameAs	Individual

5.2.5 Mapping Path and Querying Implementation

Mapping path

In order to query data from RDB via SIL, links are maintained between SIL and data sources. XPath - based rules [106] (in format XSLT) could be used to represent the mappings between RDB and OWL ,such as D2R MAP [115] and R_2O [116] built-in XML-based declarative language. D2R MAP uses three elements: *ClassMap*, *DataTypePropertyBridge* and *ObjectPropertyBridge* to describe the mappings. An example of mapping product information is illustrated as follows. Three fields: *product code, product name,* and *product family name* from two tables: *product_table* and *product_family_table* are mapped to datatype property: *code, name* and object properties *family_of* in ontology.

```
<Map>
<DBConnection odbcDSN="ProductDB" />
<ProcessorMessage outputFormat="RDF/XML-ABBREV"/>
<Namespace prefix="ex" namespace="http://example.org#"/>
<ClassMap type="ex:Product" sql="SELECT product_table.code,
product_table.name,product_family_table.name product_table.FROM
product_table, product_family_table WHERE
product_table.family_id = product_family_table.id;" groupBy="
product_table.code">
  <DatatypePropertyBridge property="ex:productCode" column="code" />
  <DatatypePropertyBridge property="ex:productName" column="name" />
  <ObjectPropertyBridge property="ex:family_of" referredClass =
"ex:Product" referredGroupBy="code"/>
 </ClassMap>
</Map>
```

R_2O allows the description of arbitrarily complex mapping expressions between ontology entities and relational elements (relations and

attributes). R_2O provides conditions and operations and the rule-style mapping definition for attributes, which allow extendable ability to define mappings that are more complex.

Query Implementation

Once the semantics and mappings have been built, there are three means to implement queries and retrieve data. The first two means correspond to the methods of static and dynamic data population respectively, and the last mean uses the created mappings to query data: (i) query with semantic querying language, such as SPARQL[26], to execute queries against SIL. It does not access the data sources since all the data are dumped to SIL; (ii) transform query language (such as, SPARQL) to SQL (Structured Query Language) and execute SQL against the RDB directly. Cyganiak [117] discussed the transformation from SPARQL to algebra and furthermore into SQL; (iii) query from the mapping path, such as, ODEMapster [116].

5.3 Illustrative Example

A scenario of virtual enterprise is assumed and created to illustrate the architecture and application of proposed ontology alignment approaches. Firstly, the scenario and relevant data sources are described in §5.3.1. Following the SIL developing steps, raw ontologies are extracted from each relational database and enriched with the help of domain ontology. These two steps are illustrated in §5.3.2. §5.3.3 present in details the ontology alignment process with showing the correspondences and aggregation process. In §5.3.4, a brief assumed query example is demonstrated using the generated alignment results. §5.3.5 discusses the contributions of the proposed architecture to enterprise interoperability regarding the illustrated example.

5.3.1 Scenario Description

This enterprise is a mobile phone manufacturing company. Among many of their information systems, there are two information systems: order management system and human resource management system. Now they are expanding and planning to open up new market in film industry, in order to take up the filming market on mobile devices. As a start-up, they took over a film management and rental company.

[26] http://www.w3.org/TR/rdf-sparql-query/

In order to keep the current operations unchanged and observe if the information systems can interoperate without re-developing existing ones, they plan to build a mediation system to query information from these information systems. They hope the query can be executed through a single interface by different departments. The long-term strategy is to merge the business of the current film management company into their conventional business, such as processing the film rental business and online watching. In addition, they plan to use current human management systems to manage the staffs in the new merged company.

The illustration of the scenario is show in Figure 5.4. Three information systems are located in two organizations and a mediation query interface is required.

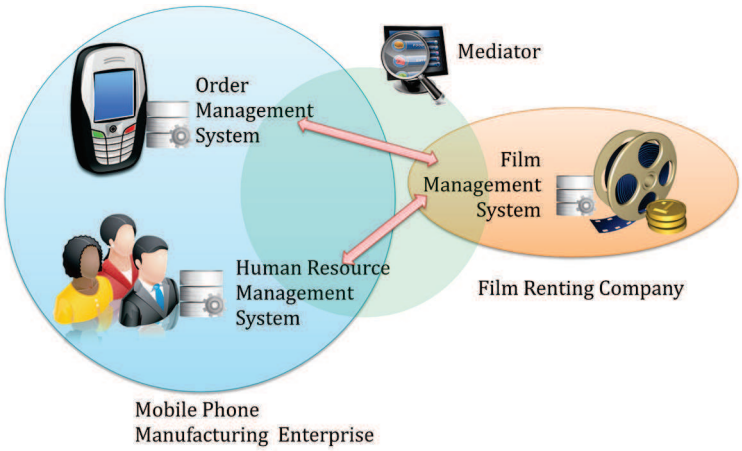


Figure 5.4 Scenario of virtual enterprise

The databases used are from the other demo projects. *Order DB* database is a minimal scale database of production order management, including the knowledge of order, product, customer and employee. *HR DB* is the database of human resources management. *Film DB* is the database from Film Company for managing film information and sales of films. The information of databases is listed in Table 5.3, including number of tables, number of fields and number of records.

Table 5.3 Information of source databases

RDB name	Tables	Fields	Records	Description
Order DB	8	59	3864	Production order management
HR DB	9	44	78635	Human resources management
Film DB	16	98	47273	Film rental management

5.3.2 Ontology Extraction and Enrichment

Tool RDBtoOnto[27] [96] is used to extract ontology from RDB. RDBtoOnto maps relation to class, field to datatype property, foreign key relationship to functional object property and composite key relation to object property. It also populates the records into instances with the RTAXON method, which can mine the hierarchical relations from data stored in database. Figure 5.5 shows the snapshot of this application and the settings. Three ontologies extracted in RDF/XML with instances are listed in Table 5.4.

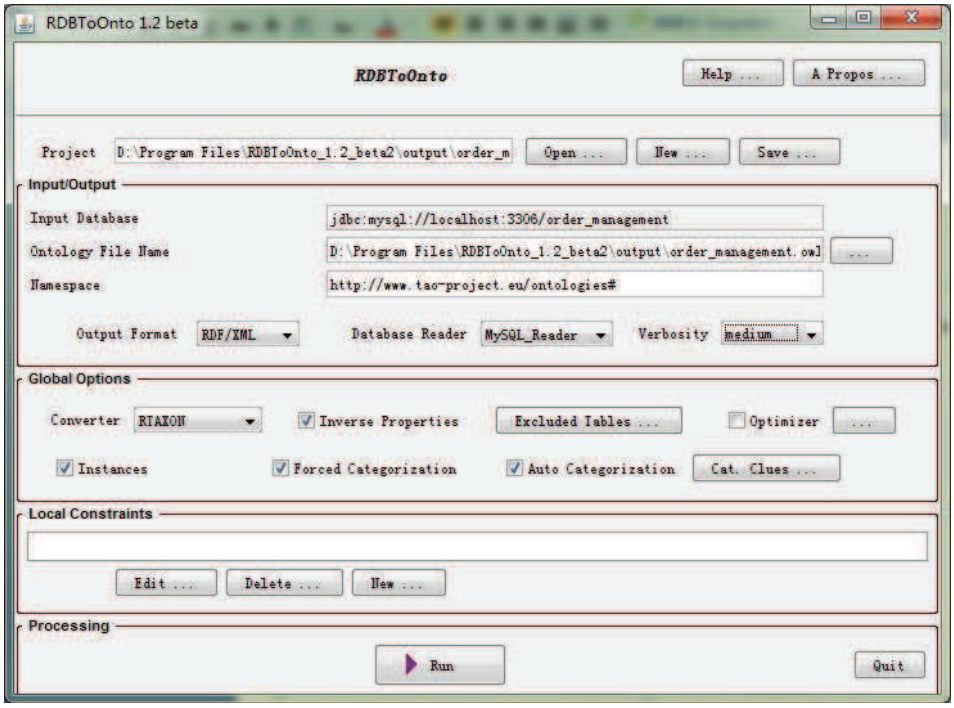


Figure 5.5 Snapshot of RDBtoOnto to extract ontology from RDB

Table 5.4 Information of extracted ontologies

RDF/OWL	Class	Object property	Datatype property	Instances
orders.owl	8	14	52	3864
hr.owl	10	12	18	78635
film.owl	16	42	67	42274

[27] http://sourceforge.net/projects/rdbtoonto/

The tool for editing and displaying ontologies is Protégé[28] v 4.1.0. Part of the extracted object property and data property are shown in Figure 5.6 and Figure 5.7. Figure 5.6 displays a snapshot of extracted class and instances. In Figure 5.7, left two columns are class and instance, while the right column is the information linked to each instance.

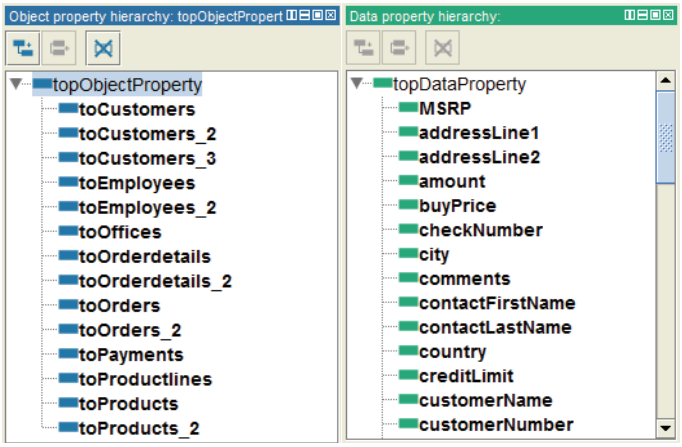


Figure 5.6 Extracted object property and data property

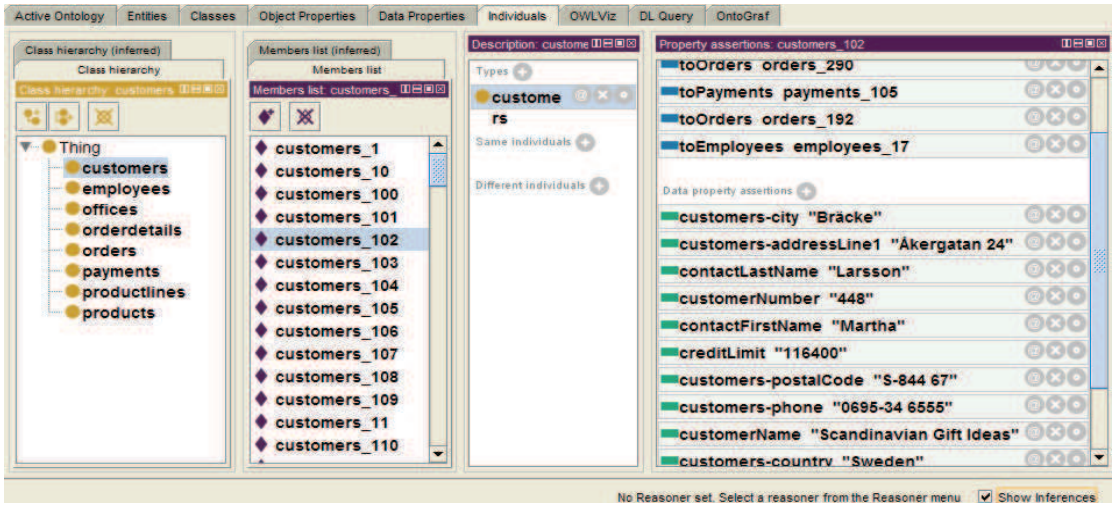


Figure 5.7 Snapshot of extracted instances in Protégé v4.1.0

The extracted ontology is further enriched with reference ontology SUMO and manual adjustment. The manual adjustment includes: (i) rename the object property. The extracted object property in RDBtoOnto follows certain naming rules: *to + class name,* such as, *toCustomers*. That is because the relation information cannot be retrieved from the schema of

[28] http://protege.stanford.edu/

database. Thus it is necessary to adjust the names of object property manually, for example, adjusting *toCustomer* to *belongs_to*; (ii) enrich the meaning by manual adjustment, such as using full word *department* to replace abbreviation *dept*.

5.3.3 Ontology Alignment

Based on the three extracted ontologies, this section introduces the ontology alignment with the proposed approaches in Chapter 2 and Chapter 3 as well as the implemented prototype system in Chapter 4. As illustrated in Figure 5.4, ontology alignment will be carried out between (*orders.owl* & *film.owl*) and (*hr.owl* & *film.owl*). In the follows, the illustration includes four steps: (i) pre-processing to load ontology from file and identify the entities, part of the fragment of *orders.owl* is presented; (ii) match with single matchers and generate intermediate correspondences, some results of matching between *orders.owl* and *film.owl* are presented; (iii) aggregate the intermediate correspondences into final correspondence with AHP process; (iv) demonstrate the final correspondences filtered by threshold.

Pre-processing

The extracted ontology will be used as input for ontology alignment. The matching task is carried out only between classes and properties, instances are not included in this example. First, an input ontology is pre-processed and entities are identified. Following is a fragment of the processed entities of ontology *orders.owl*, including the *type of entity*, *label*, *filtered label name (F)*, and *core words (C)*. For instance of the second line, the original label of entity is *checkeNumber*, its type is *datatype property* and the core word is *number* with complementary information <*MULTI_NOUN, check*>.

- `Entity [CLASS,customer, F:customer, C:(customer, SINGLE_NOUN) ]`
  `....`
- `Entity [DATATYPE_PROPERTY,checkNumber, F:checkNumber, C:(Number, MULTIPLE_NOUNS) <MULTI_NOUN,check>]`
- `Entity [DATATYPE_PROPERTY, employees-officeCode, F:employeesofficeCode, C:(Code, MULTIPLE_NOUNS) <MULTI_NOUN,employees> <MULTI_NOUN,office>]`
- `Entity [DATATYPE_PROPERTY,htmlDescription, F:htmlDescription, C:(Description, MULTIPLE_NOUNS) <MULTI_NOUN,html>]`
- `Entity [DATATYPE_PROPERTY, salesRepEmployeeNumber, F:salesRepEmployeeNumber, C:(Number, MULTIPLE_NOUNS) <MULTI_NOUN,sales> <MULTI_NOUN,Rep> <MULTI_NOUN,Employee>]`
  `....`

Intermediate correspondences

After pre-processing, the entities between two source ontologies will be matched by one or several matchers (according to the selection strategy, see Figure 3.3). In total, 67 correspondences are found between source ontologies *order.owl* and *film.owl*. Table 5.5 listed part of the intermediate correspondences, including *serial number*, *the entities matched*, and *the similarity values* generated by three matchers. The table suggests that structural matcher SBM obtained no results, the reason is that the extracted ontologies contain very few structure related information (see Table 2.8) for learning the structure similarity, such as, domain, range and sub-class, thus SBM is not very applicable to this situation. For rows #11, #15, #18 and #19, no specific matcher is applied, because they are compound words and they are identical (filtered labels) (see Figure 3.3). Take #3 for example, between *contactFirstName* and *staff-first_name*, the similarity generated by lexical matcher LBM is 0.58 and by PCW is 1.00. For #13 between *image* and *picture*, the similarity generated by PCW is 1.00 and is 0.00 for LBM.

Table 5.5 Intermediate correspondences

#	e_1 (order.owl)	e_2 (film.owl)	LBM	SBM	PCW
1	addressLine1	address2	0.83	0.00	0.00
2	checkNumber	return_date	0.00	0.00	0.68
3	contactFirstName	staff-first_name	0.58	0.00	1.00
4	country	country_id	1.00	0.00	0.00
5	creditLimit	payment_date	0.00	0.00	0.65
6	customer-country	country-country	0.83	0.00	0.76
7	customer-postalCode	postal_code	0.49	0.00	1.00
8	customer-state	country-country	0.00	0.00	0.76
9	customerName	customer-first_name	0.80	0.00	1.00
10	employees-officeCode	address-address	0.00	0.00	0.66
11	firstName	first_name	1.00		
12	htmlDescription	description	1.00	0.00	0.70
13	image	picture	0.00	0.00	1.00
14	jobTitle	film_text-title	0.50	0.00	0.81
15	lastName	last_name	1.00		
16	officeCode	address-address	0.00	0.00	0.70
17	orderDate	return_date	0.29	0.00	0.88
18	paymentDate	payment_date	1.00		
19	postalCode	postal_code	1.00		
20	priceEach	amount	0.00	0.00	0.73
21	productDescription	film_text-description	0.63	0.00	0.85
22	shippedDate	rental_date	0.25	0.00	0.70
23	state	district	0.55	0.00	0.91
24	territory	district	0.00	0.00	1.00
25	textDescription	film_text-description	1.00	0.00	1.00
..	……	……	…	…	…

AHP aggregation

The first step to use the proposed AHP-based aggregation approach (see §3.4) to combine the intermediate correspondences is to calculate the three indicators. Regarding this example between *orders.owl* and *film.owl*, the three indicators are computed as follows, which indicate that the two ontologies have higher similarity in semantic than lexical and structural aspects.

$$I_{stg} = 0.12,\ I_{sem} = 0.10 \text{ and } I_{str} = 0.32$$

Following the AHP aggregation synthesis process, the final results are presented in Table 5.6, the last column illustrates the values of priorities, which are taken as weights of each matcher to aggregate the intermediate correspondences, namely, the weights of matcher PCW, LBM and SBM are

$$W_{LBM} = 0.231,\ W_{SBM} = 0.195 \text{ and } W_{PCW} = 0.574 \text{ respectively.}$$

Table 5.6 Synthesis and weights of each matcher

	Priority with respect to			
Alternative	AM-stg	AM-str	AM-sem	Goal (weight)
LBM	0.122	0.063	0.047	**0.231**
SBM	0.051	0.105	0.039	**0.195**
PCW	0.161	0.166	0.247	**0.574**
Total	0.333	0.333	0.333	**1.000**

According to Eq. (3.6), the similarity values of intermediate correspondence displayed in Table 5.5 are aggregated to final similarity as listed in Table 5.7. The last column presents the similarity values of final correspondence.

Final correspondences

After the previous steps, the final correspondences are generated. The final correspondences are stored to file in format XML following the format defined in Section 4.2.4.

Following fragment showed a pair of identified entities with *a confidence* and *a unique identifier*. In the fragment, the similarity between *film.owl# addressLine1* and *order.owl# address2* is 0.833, for *film.owl# first_name* and *product.owl# firstName* is 1.0.

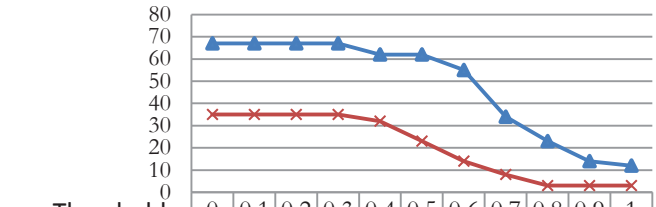
```
<Map>
<entity1 rdf:resource="http://ims-bordeaux.fr/film.owl# address2 "/>
<entity2 rdf:resource="http://ims-bordeaux.fr/order.owl#
addressLine1 "/>
<confidence rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.
833</confidence>
```

```
<relation>=</relation>
<identifier>AL-FILM-ORDER-00001</identifier>
</Map>
……
<Map>
<entity1 rdf:resource="http://www.ims-bordeaux.fr/grai/film.owl#
return_date "/>
<entity2 rdf:resource="http://www.ims-bordeaux.fr/grai/order.owl#
checkNumber "/>
<confidence rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.
682</confidence>
<relation>=</relation>
<identifier>AL-FILM-ORDER-00002</identifier>
</Map>
```

Table 5.7 Final correspondences

#	e_1 (order.owl)	e_2 (film.owl)	LBM	SBM	PCW	Final sim
	weights		*0.231*	*0.195*	*0.574*	
1	addressLine1	address2	0.83	0.00	0.00	0.833
2	checkNumber	return_date	0.00	0.00	0.68	0.682
3	contactFirstName	staff-first_name	0.58	0.00	1.00	0.880
4	country	country_id	1.00	0.00	0.00	1.000
5	creditLimit	payment_date	0.00	0.00	0.65	0.648
6	customer-country	country-country	0.83	0.00	0.76	0.783
7	customer-postalCode	postal_code	0.49	0.00	1.00	0.854
8	customer-state	country-country	0.00	0.00	0.76	0.763
9	customerNumber	username	0.64	0.00	0.00	0.643
10	employees-officeCode	address-address	0.00	0.00	0.66	0.657
11	firstName	first_name	1.00			1.000
12	htmlDescription	description	1.00	0.00	0.70	0.786
13	image	picture	0.00	0.00	1.00	1.000
14	jobTitle	film_text-title	0.50	0.00	0.81	0.723
15	lastName	last_name	1.00			1.000
16	officeCode	address-address	0.00	0.00	0.70	0.695
17	orderDate	return_date	0.29	0.00	0.88	0.706
18	paymentDate	payment_date	1.00			1.000
19	postalCode	postal_code	1.00			1.000
20	priceEach	amount	0.00	0.00	0.73	0.734
21	productDescription	film_text-description	0.63	0.00	0.85	0.786
22	shippedDate	rental_date	0.25	0.00	0.70	0.571
23	state	district	0.55	0.00	0.91	0.808
24	territory	district	0.00	0.00	1.00	1.000
25	textDescription	film_text-description	1.00	0.00	1.00	1.000
..	……	……	…	…	…	…

With threshold ranged from 0.1 to 1.0, the number of filtered correspondences is displayed in Figure 5.8. In general, when the threshold increases the number of discovered alignments decreases. The figure suggests that when threshold is less than 0.4, the number remains stable, around 67 and 35 respectively. When the threshold is greater than 0.4, the number decreases smoothly.



Threshold	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
orders-film	67	67	67	67	62	62	55	34	23	14	12
hr-film	35	35	35	35	32	23	14	8	3	3	3

Figure 5.8 Discovered correspondences filtered by threshold from 0.0 to 1.0

The threshold is set manually to cut-off the correspondences, *th* = 0.7 for alignment between *orders.owl* and *film.owl*, while *th* = 0.6 for alignment between *hr.owl* and *film.owl*. The number of available correspondences and the percentage are shown in Table 5.8.

Table 5.8 Available correspondences filtered by threshold

Source 1	Source 2	No. of Corres.	Total	Threshold	Percentage
orders.owl	film.owl	34	67	0.7	50.7%
hr.owl	film.owl	14	35	0.6	40%

5.3.4 Data Query Sample

The generated alignment file can be used to fulfill different requirements accordingly. Here an illustrative sample is presented to query data from the linked ontologies. Assumingly that the three ontologies are linked using the OWL equality constructs (see §5.2.4) and discovered correspondences, the alignments of class and property will be connected between different ontologies. SPARQL can be used to retrieve the data, for example, *Find the name and age of employees, who are responsible for the film rental orders?* The querying and retrieved data are as follows. The retrieved data could be used by upper level applications. The way to use the data depends on specific requirements, such as, for returning a report to user or being processed to generate more complex results. The part is not implemented in this thesis, because it is beyond the scope of this thesis and time consuming to implement.

Query:	Result:
PREFIX alinged: <http://www.ims-bordeaux.fr/aligned/> SELECT ?name ?age WHERE { ?Employee aligned:responsible_for ?Order. ?Order aligned:contain ?Rental.}	'1088', 'Patterson', 'William' '1102', 'Bondur', 'Gerard' '1143', 'Bow', 'Anthony' '1165', 'Jennings', 'Leslie' … …

5.3.5 Discussion

Following the proposed architecture that adopts ontology alignment as key component, a relatively simple illustrative example is presented to show the feasibility and the steps to apply the architecture. The ontology alignment played a key role in this architecture in terms of enabling data interoperability at semantic level and addressing part of technical barriers, from our point of view, this architecture possesses high extendibility and reusability, with the help of ontology alignment it can address effectively the following issues to support enterprise interoperability:

- By establishing a global semantic layer upon multiple RDBs, it could solve the issues of querying data from various data sources. Taking conventional data integration approach as an example, if we need to query data from different RDBs, we need to connect to each RDB system and use joint query to retrieve information. However, with SIL and the query mechanism in the framework, it is unnecessary to deal with each data source directly. In addition, current in-use systems will remain unchanged and unaffected, so that no extra resources and cost concerning maintaining legacy systems will be needed;
- Semantic information layer overcomes the weakness of structured data storage, since the ontology can represent knowledge in a semantic and machine-readable way. It contributes to fulfill the gap of semantic heterogeneity among different engineering systems. Besides, as the increasing development in semantic web, the SIL can enable rapid transition and connection to the emerging technologies;
- The architecture can be implemented as mediation system to interoperate among systems, since usually there are many information systems isolated in one organization. Additionally, there are demanding requirements to enable collaboration between the systems from various partners. To achieve these goals, a mediation system could be built based on the architecture accordingly.

Due to great complexity and diversity of real situations in enterprises, some challenges that may encounter when applying the architecture are predicted.

- Firstly, the architecture relies much on automatic extraction and automatic alignment, especially for large-scale databases. The current tools are not mature enough to deal with complicated enterprise systems, this may lead difficulties to apply for databases with huge amount of data. This issue has been attempted by Shvaiko and Euzenat [118], many researchers are still working on it, such as Algergawy et al. [119] and Swartout et al. [120].
- Secondly, since the RDB has been used for decades and is dominant in most enterprise systems. Many enterprises rely on it and are not likely to make changes on it. This may obstruct the application of this solution.
- Thirdly, the performance of mapping accessibility between semantic layer and RDB could become a bottleneck in this architecture when there are huge and complex mappings. Nevertheless, we believe in foreseeable few years, as the development of periphery techniques and approaches, these issues will be improved.

5.4 Conclusion

In this chapter, an ontology-driven architecture has been proposed by adopting the proposed ontology alignment approach as key component to support enterprise data interoperability at semantic level. An illustrative example is presented by utilizing the implemented prototype systems and relevant results are illustrated steps by steps. Some discussions are elaborated to analyze in which way the ontology alignment can contribute to enterprise interoperability.

By establishing a global semantic information layer (SIL) with ontology alignment upon multiple RDBs, it can address the issues of querying data from various data sources. Some suggested methods and techniques are discussed in developing SIL, and they can be chosen according to specific demands. In this chapter, the data source of proposed architecture focuses on relational database (RDB), to extend the work more generic, the data resources can be free texts, documents and semi-structured data. Some researchers are working on this topic for extracting ontology from free texts, documents. The architecture is also applicable for these cases.

General Conclusion

With the objective to develop enterprise data interoperability at semantic level, this thesis focused on adopting ontology alignment technique to contribute to a federated enterprise interoperability approach. The thesis has improved ontology alignment and matcher aggregation approaches for a better performance to facilitate federated enterprise interoperability between different heterogeneous enterprise information systems. With proposed approaches, regarding EIS context, the performance can be considered improved at the two levels: (i) increased possibility to find appropriate mappings, and (ii) increased precision of the alignment results.

Chapter 1 stated the problem, scope of the research, investigation of ontology-based approaches and the specific proposals of this thesis to improve. Investigations of using semantic web technologies to contribute to enterprise information system interoperability are described in this chapter and details can be found in Song et al. [93]. The roles that ontology plays in enterprise interoperability have been discussed in Song et al. [121].

Chapter 2 has elaborated a novel core word-based semantic similarity measurement method for ontology alignment. This approach measures the similarity based on recognized core word, which represents main meaning in a compound word or short phrase. A specific algorithm is proposed to compute the value of semantic similarity [122]. Additionally two matchers from lexical level and structural level are designed by reusing existing algorithms to enhance the matching ability. The evaluation results obtained in Chapter 4 suggested that the proposed approach possesses good matching ability and has reached expected goals. It is argued that the proposed semantic similarity measurement approach also can be applied to the other fields besides ontology alignment, such as, semantic search, semantic web and information extraction.

Chapter 3 has developed a new analytic matcher aggregation method that allows combining the multiple proposed matchers [123, 124]. This method is developed based on AHP and three similarity indicators aiming to automate the aggregation process and to improve the combined results. The experiments results in Chapter 4 showed that the proposed aggregation method improved considerably the combined results. In addition, the method is facile to apply due to the automatic process and also can be applied to the other domains for addressing weighting problems that involve multiple variables and complex factors.

Chapter 4 has implemented the proposed approaches in Java and tested with OAEI benchmarking data sets. The experiment results suggested that the proposed ontology alignment approaches and aggregation method made some improvements and obtained promising results. Compared with the other 13 approaches that participated to benchmarking *biblio* of OAEI 2011, the HF1 of our approach is 0.84 and ranked second. It is lower than the first (YAM++, 0.86) by 0.03 and higher than the third CSA [53] by 0.01. The major advantage of our approach is the compromise between the simplicity to apply and a good matching ability, especially the good matching ability on real-life ontologies, as well as the potentials to be applied to other domains. Possible improvements of our approaches have been already identified and discussed in the conclusion section of Chapter 4, and once they are implemented, the approach should allow obtaining better result.

Chapter 5 has presented an ontology-driven architecture for querying data from multiple relational databases [40], in order to apply the proposed approaches and implemented prototype system to develop enterprise interoperability. This architecture focuses on addressing data interoperability at semantic level. It can be applied accordingly to several possible applications to support enterprise data interoperability.

The work presented in this thesis is also concerned with several other research tracks and questions that can be considered as open issues for future research as follows:

- Regarding the rules for core word recognition, in order to adapt to specific knowledge domain and special cases, one future research can be done is to extend these rules, such as defining more general rules and some special rules. Currently the rules defined in this work are limited for the general cases and maybe less applicable for special cases;
- The PCW matcher and lexical matcher perform alignment task mainly based on the labels of entities at the moment, if the comments and additional annotations of entities are also taken into account as semantic sources to aid the alignment, the results could be improved;
- In the matcher aggregation method, three similarity indicators are utilized to automate the assignment of scales in applying AHP. In order to apply this proposed weighting method in other fields, the way to calculate the similarity indicators can be adapted accordingly, such as using some parameters that could reflect the importance of alternatives;
- Current implemented prototype system utilizes command line interface that takes a few parameters as input and generates the alignments in XML as output. Although we argue that it is sufficient for users to

utilize the results on their own specific needs, it can be improved to make the system easier to use by developing a GUI and graphical display for the alignment results;

- Concerning the solution for building semantic information layer (SIL) for developing data interoperability, the rules for extracting ontologies from relational databases are mainly defined at schema-level. More rules regarding instance-level for extracting records in RDB to ontology instances can be extended in order to enrich the semantics of data;
- Considering the semantic interoperability needs at conceptual level in enterprises, an extensional ongoing work [125], which proposes to apply ontology alignment for developing the interoperability between Model-Driven Architecture (MDA) and simulation models, is carried out based on the proposed ontology alignment approaches. The general methodology has been proposed and elaborated [125]. The application of ontology alignment to exchange information provides a semantic and loose connection between models and simulation. At current stage, the work that has been done by focusing on defining a general methodology and framework. Remaining work mainly concerns elaborating the method and operational application steps, the work includes: (i) the rules and formalism of information exchange, and (ii) the way of exchanging information between the two sides using ontology alignment.

References

[1] Chen D., Doumeingts G., Vernadat F., Architectures for enterprise integration and interoperability: Past, present and future, Comput. Ind., 59 (2008) 647-659.

[2] Chen D., Daclin N., Framework for Enterprise Interoperability, in: EI2N 2nd International Workshop on Enterprise Integration, Interoperability and Networking, In I-ESA 2006, Bordeaux, France, 2006, pp. 77-88.

[3] He K.-Q., Wang J., Liang P., Semantic Interoperability Aggregation in Service Requirements Refinement, Journal of Computer Science and Technology, 25 (2010) 1103-1117.

[4] Buccella A., Cechich A., Brisaboa N.R., Ontology-based data integration methods: A framework for comparison Revista Colombiana de Computación, (2005).

[5] Zhan C., O'Brien P., Domain Ontology Management Environment, in: System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on, 2000, pp. 9 pp. vol.1.

[6] Yahia E., Aubry A., Panetto H., Formal measures for semantic interoperability assessment in cooperative enterprise information systems, Computers in Industry, 63 (2012) 443-457.

[7] Vernadat F.B., Technical, semantic and organizational issues of enterprise interoperability and networking, Annual Reviews in Control, 34 (2010) 139-144.

[8] Shukair G., Loutas N., Peristeras V., Sklarß S., Towards semantically interoperable metadata repositories: The Asset Description Metadata Schema, Computers in Industry, 64 (2013) 10-18.

[9] Heery R., Johnston P., Fulop C., Micsik A., Metadata schema registries in the partially Semantic web: the CORES experience, in: Proceedings of the 2003 international conference on Dublin Core and metadata applications: supporting communities of discourse and practice-metadata research & applications, Dublin Core Metadata Initiative, Seattle, Washington, 2003, pp. 1-8.

[10] Heery R., Gardner, T., Day, M., Patel, M., DESIRE Metadata Registry Framework. DESIRE Deliverable 3.5, in, 2000.

[11] Alasem A., An overview of e-government metadata standards and initiatives based on Dublin Core, Electronic Journal of e-Government, 7 (2009) 1-10.

[12] Lee–Smeltzer K.-H., Finding the needle: controlled vocabularies, resource discovery, and Dublin Core, Library Collections, Acquisitions, and Technical Services, 24 (2000) 205-215.

[13] Maali F., Cyganiak R., Peristeras V., Enabling Interoperability of Government Data Catalogues, in: Wimmer M., Chappelet J.-L., Janssen M., Scholl H. (Eds.) Electronic Government, Springer Berlin Heidelberg, 2010, pp. 339-350.

[14] Halevy A.Y., Ashish N., Bitton D., Carey M., Draper D., Pollock J., Rosenthal A., Sikka V., Enterprise information integration: successes, challenges and controversies, in: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, ACM, Baltimore, Maryland, 2005, pp. 778-787.

[15] Ziegler P., Dittrich K., Three Decades of Data Integration - All Problems Solved?, in: In 18th IFIP World Computer Congress (WCC 2004), Volume 12, Building the Information Society, 2004, pp. 3-12.

[16] Halevy A., Rajaraman A., Ordille J., Data integration: the teenage years, in: VLDB'2006: Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment, 2006, pp. 9-16.

[17] Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., Hübner S., Ontology-based integration of information - a survey of existing approaches, in: Stuckenschmidt H. (Ed.) IJCAI01 Workshop Ontologies and Information Sharing, 2001, pp. 108-117.

[18] Panetto H., Dassisti M., Tursi A., ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment, Advanced Engineering Informatics, 26 (2012) 334-348.

[19] Martínez-Costa C., Menárguez-Tortosa M., Fernández-Breis J.T., An approach for the semantic interoperability of ISO EN 13606 and OpenEHR archetypes, Journal of Biomedical Informatics, 43 (2010) 736-746.

[20] Fallahi G.R., Frank A.U., Mesgari M.S., Rajabifard A., An ontological structure for semantic interoperability of GIS and environmental modeling, International Journal of Applied Earth Observation and Geoinformation, 10 (2008) 342-357.

[21] Quang T., Barker K., Alhajj R., Semantic Interoperability Between Relational Database Systems, in: Database Engineering and Applications Symposium, 2007. IDEAS 2007. 11th International, 2007, pp. 208-215.

[22] Lu Y., Panetto H., Ni Y., Gu X., Ontology alignment for networked enterprise information system interoperability in supply chain environment, International Journal of Computer Integrated Manufacturing, 26 (2012) 140-151.

[23] Oberle D., How Ontologies Benefit Enterprise Applications, Semantic Web Journal, (2013).

[24] Bürger T., Simperl E., Measuring the Benefits of Ontologies, in: Meersman R., Tari Z., Herrero P. (Eds.) On the Move to Meaningful Internet Systems: OTM 2008 Workshops, Springer Berlin Heidelberg, 2008, pp. 584-594.

[25] Küçük D., Salor Ö., İnan T., Çadırcı I., Ermiş M., PQONT: A domain ontology for electrical power quality, Advanced Engineering Informatics, 24 (2010) 84-95.

[26] Naudet Y., Latour T., Guedria W., Chen D., Towards a systemic formalisation of interoperability, Computers in Industry, 61 (2010) 176-185.

[27] Liao Y., Lezoche M., Panetto H., Boudjlida N., Semantic annotation model definition for systems interoperability, in: OTM 2011 Workshops 2011 - 6th International Workshop on Enterprise Integration, Interoperability and Networking (EI2N), Springer-Verlag, Crete, Greece, 2011, pp. 61-70.

[28] Zouggar N., Chen D., Vallespir B., Semantic Enrichment of Enterprise Modelling - Use of Ontology, in: Interoperability for Enterprise Software and Applications China, 2009. IESA '09. International Conference on, 2009, pp. 252-258.

[29] Lim S.C.J., Liu Y., Lee W.B., A methodology for building a semantically annotated multi-faceted ontology for product family modelling, Advanced Engineering Informatics, 25 (2011) 147-161.

[30] Fernandes R.P., Grosse I.R., Krishnamurty S., Witherell P., Wileden J.C., Semantic methods supporting engineering design innovation, Advanced Engineering Informatics, 25 (2011) 185-192.

[31] Rezgui Y., Wilson I.E., Miles J., Hopfe C.J., Federating information portals through an ontology-centred approach: A feasibility study, Advanced Engineering Informatics, 24 (2010) 340-354.

[32] Colombo G., Mosca A., Sartori F., Towards the design of intelligent CAD systems: An ontological approach, Advanced Engineering Informatics, 21 (2007) 153-168.

[33] Maedche A., Motik B., Silva N., Volz R., MAFRA - A MApping FRAmework for Distributed Ontologies, in: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, Springer-Verlag, 2002, pp. 235-250.

[34] Choi N., Song I.-Y., Han H., A survey on ontology mapping, SIGMOD Rec., 35 (2006) 34-41.

[35] Shvaiko P., Euzenat J., Ten Challenges for Ontology Matching, in: Meersman R., Tari Z. (Eds.) On the Move to Meaningful Internet Systems: OTM 2008, Springer Berlin / Heidelberg, 2008, pp. 1164-1182.

[36] Winkler W.E., The state of record linkage and current research problems, in, Statistical Research Division, U.S. Bureau of the Census, 1999.

[37] Brown P.F., deSouza P.V., Mercer R.L., Pietra V.J.D., Lai J.C., Class-based n-gram models of natural language, Computational Linguistics, 18 (1992) 467-479.

[38] Melnik S., Garcia-Molina H., Rahm E., Similarity flooding: a versatile graph matching algorithm and its application to schema matching, in: the 18th International Conference on Data Engineering, IEEE Computer Society, Washington, DC, USA, 2002, pp. 117-128.

[39] Studer R., Benjamins V.R., Fensel D., Knowledge engineering: Principles and methods, Data & Knowledge Engineering, 25 (1998) 161-197.

[40] Song F., Zacharewicz G., Chen D., An ontology-driven framework towards building enterprise semantic information layer, Advanced Engineering Informatics, 27 (2013) 38-50.

[41] Euzenat J., Shvaiko P., Ontology matching, Springer, Heidelberg, 2007.

[42] Stoilos G., Stamou G., Kollias S., A String Metric for Ontology Alignment, in: 4th international conference on The Semantic Web, Springer, Galway, Ireland, 2005, pp. 624-637.

[43] Ehrig M., Staab S., QOM – Quick Ontology Mapping, in: McIlraith S.A., Plexousakis D., Harmelen F.v. (Eds.) The Semantic Web – ISWC 2004, Springer, Heidelberg, 2004, pp. 683-697.

[44] Hu W., Jian N., Qu Y., Wang Y., GMO: A Graph Matching for Ontologies, in: K-CAP Workshop on Integrating Ontologies, 2005.

[45] Huang J., Dang J., Vidal J.M., Huhns M.N., Ontology Matching Using an Artificial Neural Network to Learn Weights, in: 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, 2007.

[46] Granitzer M., Sabol V., Onn K.W., Lukose D., Tochtermann K., Ontology Alignment—A Survey with Focus on Visually Supported Semi-Automatic Techniques, Future Internet, 2 (2010) 238-258.

[47] Alasoud A., Haarslev V., Shiri N., An empirical comparison of ontology matching techniques, Journal of Information Science, 35 (2009) 379-397.

[48] Li J., Tang J., Li Y., Luo Q., RiMOM: A Dynamic Multistrategy Ontology Alignment Framework, Knowledge and Data Engineering, IEEE Transactions on, 21 (2009) 1218-1232.

[49] Pirró G., Talia D., UFOme: An ontology mapping system with strategy prediction capabilities, Data & Knowledge Engineering, 69 (2010) 444-471.

[50] Mao M., Peng Y., Spring M., An adaptive ontology mapping approach with neural network based constraint satisfaction, Web Semantics: Science, Services and Agents on the World Wide Web, 8 (2010) 14-25.

[51] Tu K., Yu Y., CMC:Combining multiple schema-matching strategies based on credibility prediction, in: 10th International Conference on Database Systems for Advanced Applications Springer, Beijing, China, 2005, pp. 888-893.

[52] Ngo D.H., Bellahsene Z., Coletta R., YAM++ -- Results for OAEI 2011, in: Pavel Shvaiko J.E.T.H.C.Q.M.M.I.C. (Ed.) ISWC'11: The 6th International Workshop on Ontology Matching, 2011, pp. 228-235.

[53] Tran Q.-V., Ichise R., Ho B.-Q., Cluster-based similarity aggregation for ontology matching, in: Proc. of 6th Ontology Matching Workshop, 2011, pp. 142-147.

[54] Akbari I., Fathian M., A novel algorithm for ontology matching, Journal of Information Science, 36 (2010) 324-334.

[55] Xu P., Wang Y., Liu B., A differentor based adaptive ontology matching approach, Journal of Information Science, (2012) 1-17.

[56] Muslea I., Extraction Patterns for Information Extraction Tasks: A Survey, in: 6th National Conference on Artificial Intelligence Workshop on Machine Learning for Information Extraction, 1999, pp. 1-6.

[57] Ceausu V., Desprès S., A semantic case-based reasoning framework for text categorization, in: 6th international The semantic web and 2nd Asian conference on Asian semantic web conference, Springer-Verlag, Busan, Korea, 2007, pp. 736-749.

[58] Maynard D., Funk A., Peters W., W.: SPRAT: a tool for automatic semantic patternbased ontology population, in: International Conference for Digital Libraries and the Semantic Web, Trento, Italy, 2009.

[59] Sari Y., Hassan M.F., Zamin N., Rule-based pattern extractor and named entity recognition: A hybrid approach, in: Information Technology (ITSim), 2010 International Symposium in, 2010, pp. 563-568.

[60] Ritze D., Meilicke C., Sváb-Zamazal O., Stuckenschmidt H., A Pattern-based Ontology Matching Approach for Detecting Complex Correspondences, in: Shvaiko P., Euzenat J., Giunchiglia F., Stuckenschmidt H., Noy N.F., Rosenthal A. (Eds.) OM, CEUR-WS.org, 2008.

[61] Šváb-Zamazal O., Svátek V., OWL Matching Patterns Backed by Naming and Ontology Patterns, in: 10th Czecho-Slovak Knowledge Technology Conference, Stara Lesna, Slovakia, 2011.

[62] Toutanova K., Klein D., Manning C.D., Singer Y., Feature-rich part-of-speech tagging with a cyclic dependency network, in: NAACL-Human Language Technology Conference, Association for Computational Linguistics, Edmonton, Canada, 2003, pp. 173-180.

[63] Fellbaum C., WordNet and wordnets, in: Brown K. (Ed.) Encyclopedia of Language and Linguistics, Elsevier, Oxford, 2005, pp. 665-670.

[64] Li L., Xiao H., Xu G., Finding Related Micro-blogs Based on WordNet, in: Yu H., Yu G., Hsu W., Moon Y.-S., Unland R., Yoo J. (Eds.) Database Systems for Advanced Applications, Springer Berlin Heidelberg, 2012, pp. 115-122.

[65] Lin D., An Information-Theoretic Definition of Similarity, in: Shavlik J.W. (Ed.) 5th International Conference on Machine Learning, Morgan Kaufmann, Wisconsin, USA, 1998, pp. 296-304.

[66] Seco N., Veale T., Hayes J., An Intrinsic Information Content Metric for Semantic Similarity in WordNet, in: ECAI'2004, the 16th European Conference on Artificial Intelligence, 2004.

[67] Cohen W., Ravikumar P., Fienberg S., A comparison of string distance metrics for name-matching tasks, in: IJCAI-03 Workshop on Information Integration, 2003, pp. 73-78.

[68] Smith R.D., Distinct word length frequencies: distributions and symbol entropies, arXiv preprint arXiv:1207.2334, (2012).

[69] Sojka P., Notes on compound word hyphenation in TEX, TUGboat, 16 (1995) 290-296.

[70] Jaro M., Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida, Journal of the American Statistical Association, 84 (1989) 414-420.

[71] Jaro M.A., Probabilistic linkage of large public health data files, Statistics in Medicine, 14 (1995) 491-498.

[72] Hiemstra D., N-Gram Models, in: Liu L., ÖZsu M.T. (Eds.) Encyclopedia of Database Systems, Springer US, 2009, pp. 1910-1910.

[73] Saaty T.L., How to make a decision: The analytic hierarchy process, European Journal of Operational Research, 48 (1990) 9-26.

[74] Saaty T.L., Peniwati K., Group Decision Making: Drawing Out and Reconciling Differences, Rws Publications, 2008.

[75] Vaidya O.S., Kumar S., Analytic hierarchy process: An overview of applications, European Journal of Operational Research, 169 (2006) 1-29.

[76] Zhao X., Zhao Q., Chen G., AHP Method in Computing Factor Weight of the Network Learning Pattern Recognition, in: Jin D., Lin S. (Eds.) Advances in Electronic Engineering, Communication and Management Vol.1, Springer Berlin Heidelberg, 2012, pp. 193-197.

[77] Shapira A., Simcha M., AHP-Based Weighting of Factors Affecting Safety on Construction Sites with Tower Cranes, Journal of Construction Engineering and Management, 135 (2009) 307-318.

[78] Mochol M., Jentzsch A., Euzenat J., Applying an Analytic Method for Matching Approach Selection, in: Shvaiko P., Euzenat J., Noy N.F., Stuckenschmidt H., Benjamins V.R., Uschold M. (Eds.) 1st International Workshop on Ontology Matching, CEUR Workshop Proceedings, 2006.

[79] Ji Q., Haase P., Qi G., Combination of Similarity Measures in Ontology Matching Using the OWA Operator, in: Yager R., Kacprzyk J., Beliakov G. (Eds.) Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice, Springer Berlin Heidelberg, 2011, pp. 281-295.

[80] Houshmand M., Naghibzadeh M., Araban S., Reliability-based similarity aggregation in ontology matching, in: Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on, 2010, pp. 744-749.

[81] Saaty T., What is the Analytic Hierarchy Process?, in: Mitra G., Greenberg H., Lootsma F., Rijkaert M., Zimmermann H. (Eds.) Mathematical Models for Decision Support, Springer Berlin Heidelberg, 1988, pp. 109-121.

[82] Saaty T.L., Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process, RWS Publications, 2000.

[83] Saaty T.L., Vargas L.G., The Seven Pillars of the Analytic Hierarchy Process, in: Models, Methods, Concepts & Applications of the Analytic Hierarchy Process, Springer, 2012, pp. 23-40.

[84] Berry J.K., Optimal Path Analysis and Corridor Routing: Infusing Stakeholder Perspective in Calibration and Weighting of Mode of Model Criteria, in, University of Denver, 2004.

[85] Tzeng G.-H., Chiang C.-H., Li C.-W., Evaluating intertwined effects in e-learning programs: A novel hybrid MCDM model based on factor analysis and DEMATEL, Expert Systems with Applications, 32 (2007) 1028-1044.

[86] Abdullah L., Azman F.N., Weights of Obesity Factors Using Analytic Hierarchy Process, International Journal of Research & Reviews in Applied Sciences, 7 (2011) 6.

[87] Ehrig M., Sure Y., Ontology Mapping – An Integrated Approach, in: Bussler C., Davies J., Fensel D., Studer R. (Eds.), Springer Berlin / Heidelberg, 2004, pp. 76-91.

[88] Euzenat J., An API for Ontology Alignment, in: 3rd International Semantic Web Conference, Springer, Hiroshima, Japan, 2004, pp. 698-712.

[89] Beizer B., Black-Box Testing: Techniques for Functional Testing of Software and Systems, Wiley, 1995.

[90] Euzenat J., Semantic precision and recall for ontology alignment evaluation, in: 20th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, Hyderabad, India, 2007, pp. 348-353.

[91] Euzenat J., Ferrara A., Hage W.R.v., Hollink L., Meilicke C., Nikolov A., Ritze D., Scharffe F., Shvaiko P., Stuckenschmidt H., Sváb-Zamazal O., Santos C.T.d., Results of the ontology alignment evaluation initiative 2011, in: the 6th International Workshop on Ontology Matching, CEUR Workshop Proceedings, Bonn, Germany, 2011.

[92] Duchateau F., Coletta R., Bellahsene Z., Miller R., YAM: a schema matcher factory, in: Proceedings of the 18th ACM conference on Information and knowledge management, ACM, Hong Kong, China, 2009, pp. 2079-2080.

[93] Song F., Zacharewicz G., Chen D., An Architecture for Interoperability of Enterprise Information Systems Based on SOA and Semantic Web Technologies, in: Zhang R., Cordeiro J., Li X., Zhang Z., Zhang J. (Eds.) 13th International Conference on Enterprise Information Systems, SciTePress, Beijing, 2011, pp. 431-437.

[94] Astrova I., Rules for Mapping SQL Relational Databases to OWL Ontologies, in: Sicilia M.-A., Lytras M.D. (Eds.) Metadata and Semantics, Springer US, 2009, pp. 415-424.

[95] Hu W., Qu Y., Discovering simple mappings between relational database schemas and ontologies, in: 6th international The semantic web and 2nd Asian conference on Asian semantic web conference, Springer-Verlag, Busan, Korea, 2007, pp. 225-238.

[96] Cerbah F., Learning Ontologies with Deep Class Hierarchies by Mining the Content of Relational Databases, in: Guillet F., Ritschard G., Zighed D., Briand H. (Eds.) Advances in knowledge discovery and management, Springer-Verlag, Heidelberg, 2010, pp. 271-286.

[97] Rodriguez M.A., Egenhofer M.J., Determining semantic similarity among entity classes from different ontologies, Knowledge and Data Engineering, IEEE Transactions on, 15 (2003) 442-456.

[98] Sane S.S., Shirke A., Generating OWL ontologies from a relational databases for the semantic web, in: International Conference on Advances in Computing, Communication and Control, ACM, Mumbai, India, 2009, pp. 157-162.

[99] Xu Z., Zhang S., Dong Y., Mapping between Relational Database Schema and OWL Ontology for Deep Annotation, in: Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on, 2006, pp. 548-552.

[100] Dou D., LePendu P., Ontology-based integration for relational databases, in: Proceedings of the 2006 ACM symposium on Applied computing, ACM, Dijon, France, 2006, pp. 461-466.

[101] Hert M., Reif G., Gall H.C., A comparison of RDB-to-RDF mapping languages, in: 7th International Conference on Semantic Systems, ACM, Graz, Austria, 2011, pp. 25-32.

[102] Ghawi R., Cullot N., Database-to-Ontology Mapping Generation for Semantic Interoperability, in: 3rd International Workshop on Database Interoperability (InterDB 2007), 2007.

[103] Byrne K., Having Triplets – Holding Cultural Data as RDF, in: Larson M., Fernie K., Oomen J., Cigarran J. (Eds.) ECDL 2008 Workshop on Information Access to Cultural Heritage, Aarhus, Denmark, 2008.

[104] Green J., Hart G., Dolbear C., Engelbrecht P.C., Goodwin J., Creating a Semantic Integration System using Spatial Data, in: Bizer C., Joshi A. (Eds.) International Semantic Web Conference (Posters & Demos), CEUR-WS.org, 2008.

[105] Park J., Cho W., Rho S., Evaluating ontology extraction tools using a comprehensive evaluation framework, Data & Knowledge Engineering, 69 (2010) 1043-1061.

[106] Sahoo S.S., Halb W., Hellmann S., Idehen K., Jr T.T., Auer S., Sequeda J., Ezzat A., A Survey of Current Approaches for Mapping of Relational Databases to RDF, in, W3C, 2009.

[107] Niles I., Pease A., Towards a standard upper ontology, in: Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001, ACM, Ogunquit, Maine, USA, 2001, pp. 2-9.

[108] Schneider L., How to build a foundational ontology: The Object-Centered High-level REference ontology OCHRE, in: Günter A., Kruse R., Neumann B. (Eds.) 26th Annual German Conference on AI, Springer, Hamburg, Germany, 2003, pp. 662.

[109] Osman H., Ei-Diraby T., Ontological Modeling of Infrastructure Products and Related Concepts, Transportation Research Record: Journal of the Transportation Research Board, 1984 (2006) 159-167.

[110] Lemaignan S., Siadat A., Dantan J.Y., Semenenko A., MASON: A Proposal For An Ontology Of Manufacturing Domain, in: Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006. DIS 2006. IEEE Workshop on, 2006, pp. 195-200.

[111] Lagos N., Setchi R., A manufacturing ontology for e-learning, in: IPROMS, 2007.

[112] Navigli R., Velardi P., Ontology enrichment through automatic semantic annotation of on-line glossaries, in: 15th international conference on Managing Knowledge in a World of Networks, Springer-Verlag, Podebrady, Czech Republic, 2006, pp. 126-140.

[113] Zouaq A., Gagnon M., Ozell B., A SUMO-based Semantic Analysis for Knowledge Extraction, in: the 4th Language & Technology Conference, Poznań, Poland, 2009.

[114] Oltramari A., Stellato A., Enriching Ontologies with Linguistic Content: An Evaluation Framework, in: Proceedings of OntoLex (Hosted by Sixth international conference on Language Resources and Evaluation), 2008.

[115] Bizer C., D2R MAP - A database to RDF mapping language, in: WWW (Posters), Citeseer, 2003.

[116] Rodríguez J., Pérez A., Upgrading relational legacy data to the semantic web, in: Carr L., De Roure D., Iyengar A., Goble C., Dahlin M. (Eds.) WWW, ACM, 2006, pp. 1069-1070.

[117] Cyganiak R., A relational algebra for SPARQL, in, HP Laboratories Bristol, 2005.

[118] Shvaiko P., Euzenat J., Ontology Matching: State of the Art and Future Challenges, Knowledge and Data Engineering, IEEE Transactions on, 25 (2013) 158-176.

[119] Algergawy A., Massmann S., Rahm E., A clustering-based approach for large-scale ontology matching, in: Proceedings of the 15th international conference on Advances in databases and information systems, Springer-Verlag, Vienna, Austria, 2011, pp. 415-428.

[120] Swartout B., Ramesh P., Knight K., Russ T., Toward Distributed Use of Large-Scale Ontologies, in: AAAI Symposium on Ontological Engineering, 1997, pp. 138-148.

[121] Song F., Zacharewicz G., Chen D., Ontology for Contributing to Enterprise Interoperability, in: Jardim-Concalves R., Stefanova K. (Eds.) UNITE 2nd Doctoral Symposium: R&D in Future Internet and Enterprise Interoperability, AVANGARD PRIMA, Sofia, Bulgaria, 2012, pp. 127-134.

[122] Song F., Zacharewicz G., Chen D., Pattern-Based Core Word Recognition to Support Ontology Matching, International Journal of Knowledge-Based and Intelligent Engineering Systems, 17 (2013) 167-176.

[123] Song F., Zacharewicz G., Chen D., Multi-strategies Ontology Alignment Aggregated by AHP, in: Graña M., Toro C., Posada J., Howlett R.J., Jain L.C. (Eds.) Advances in Knowledge-Based and Intelligent Information and Engineering Systems IOS Press, San Sebastian, 2012, pp. 1583-1592.

[124] Song F., Zacharewicz G., Chen D., An Analytic Aggregation-Based Ontology Alignment Approach with Multiple Matchers, in: Tweedale J.W., Jain L.C. (Eds.) Advanced Techniques for Knowledge Engineering and Innovative Applications 16th International Conference, KES 2012, San Sebastian, Spain, September 10-12, 2012, Revised Selected Papers, Springer, 2013, pp. 143-159.

[125] Song F., Zacharewicz G., Chen D., Adapting Simulation Modeling to Model-Driven Architecture for Model Requirement Verification, in: 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH) SciTePress Reykjavik, Iceland, 2013, pp. 302-309.