Jiawen Chen

Smart Semantic Multi-channel Communication

A Master's Thesis
in Information Technology
October 10, 2014

University of Jyväskylä

Department of Mathmatical information Technology

Jyväskylä

Author: Jiawen Chen
Contact information: jichen@student.jyu.fi
Title: Smart Semantic Multi-channel Communication
Työn nimi: Alykäs Semanttinen Monikanavakomunikaatio
Project: A Master's Thesis in Information Technology
Page count: 71
Abstract: Nowadays, there are various media channels which are in popular use. For instance, phone call, text message, email, website, and mobile applications. Web technique plays a significant role in this society, no matter where you are, what you are doing, nobody can live without it. Web is functional, people surf on-line everyday for education, entertainment, looking for useful information, however the modern web still cannot fulfil people' demands. Now, consider a customer trying to buy a product first through visiting a website first in order to find out more information. Then calling for a quote by phone and finally the customer decides to buy this product using email.

The main target of this thesis is to seek how a multi-channel communication system can understand and track customers shopping behaviours as well their habits. Design science is used as the research approach for this thesis, different theories and perspective are utilized to interpret the results.

Multiple channel communication system is a future vision for E-business, it is based on semantic web technique, autonomic computing and recommendation engine. So far, however, there has been little discussion about multiple channel communication in AI(artificial intelligence) field. The major objective of this thesis is to present proposed solutions for each multi channel communication component. It proposed the use of utility function for smart channel selection, LODifier approach and information system for message merge. It also provides an assumed design by using Apache Stanbol to modify core semantic enhancement engine. And this writing approach is called *design science.* In conclusion, this thesis provides some solutions for the multi-channel communication system, but there are still some certain issues which need to be discussed with more detail in the future. The author hopes this thesis could make a small contribution in the relevant fields for other research.
Suomenkielinen tiivistelmä: Abstract in Finnish
Keywords: Communication,Semantic Web,Customer Relation
Avainsanat: tiedonvälitys, semanttinen seitti, asiakas yhteys

Preface

The master thesis that is lying in front of you is the result of twelve months research at the Department of Mathematical Information Technology at the University of Jyävskylä. The writing of this thesis has gone through a lot of difficulties, especially in the beginning I had troubles to find a proper and define a suitable research question. With the help of my supervisors at the University of Jyväskylä, Professor Vagan Terziyan and Michael Cochez , I found out this interesting research topic. This thesis topic is motivated and inspired by Nagy previous research, also it is a part of a project from Steeri Oy, Sami Helin as a company supervisor provided quite useful information from Business side. Another reason I am involved in this topic is that the semantic technology,playing an important role in this thesis, is very promising and challenging. At the end, I would like thank you readers, I hope you could have a good time to enjoy reading this thesis. At least you have read one page of this thesis already.

Glossary

DRS	Discourse Representation Structure
HTML	HyperText Markup Language
HTTP	Hypertext transfer protocol
LD	Linked Data
LOD	Linked Open Data
NER	Named Entity Recognition
NLP	Natural Language Processing
NS	Name Space
OWL	Web Ontology Language
POS	Part of Speech
RDF	Resource Description Framework
RDBMS	Relational Database Management System
SMS	Short Message Service
SPARQL	SPARQL Protocol And RDF Query Language
SQL	Structured Query Language
URI	Uniform Resource Identifier
WWW	World Wide Web
WSD	Word Sense Disambiguation
XML	Extensible Markup Language

Contents

1 Introduction

Before introducing the structure and concept of the whole thesis, there is an example to briefly present the main idea what Multi-channel communications is. A student is going to apply for a master program in the University of Jyvaskyla(JYU). He discovered application requirements and contact information through the university homepage. These contact information include a few email addresses and mobile phone numbers of university staffs, and an on-line Q&A board where visitors can leave question. After reading the application requirements, the student was still confused about some prerequisites. Therefore he tried to look for help by leaving a few questions and his email address on the Q&A board. One week later, admission office replied the student by sending him an email, however at the same time, the student did not check his email on regularly, he directly made a call to the university. This small example describes how multiple channels can be used in reality.

In addition, before this student tried to apply for this program, he did a bit search through the Web. While he input his bachelor education background and interests orientation, the Web seemed to understand his intention and recommended this master program. This case introduces a smart Web at the present day, semantic web. The semantic Web is able to understand and interpret user intention in a right way, thus offering and reasoning satisfied search result back to users. Besides, semantic technology has become more and more significant in information technology area, it enables human-machine interaction more effective.

Nowadays, people could catch information from different communication channels, such as email, SMS, and Internet advertisement, etc.... However sometimes people get confused by what they have done through these various channels and get annoyed by some spam messages. Is that possible to solve these issue so people could be reached in a faster and more efficient way? The main goal of semantic multi-channel communications is to seek the answer and hopefully could be implemented for business purpose in the future. This thesis is motivated by previous research from Nagy(2012)[1].

This thesis will focus on combine semantic technology with multi-channel communications and propose a relevant framework which might be utilized to solve the

research question and for Business to Business(B2B) plan in the future.

The overall structure of this thesis takes the form of nine chapters, including the introductory chapter. Chapter Two begins by introducing communication and semantic Web, and laying out some relevant fundamental knowledge of semantic web components. This chapter also focuses on the ontology and its relating approaches since it is a crucial part in semantic technology. The third chapter presents the main research findings and multi-channel communication framework proposal. Besides, the specified ontology models are discussed in this chapter as well. Chapter Four analyses smart channel selection mechanism and probability for implementing its method to multi-channel communication framework. In chapter Five, an unstructured text analysis approach which is used for converting message, is going to be explained. Chapter Six firstly introduces a model that could automatically compose and send messages for the framework, also this chapter discusses the current popular technology recommendation engine to seek a solution for better user experience in Web and email communication. Several other commercial communication cases are also mentioned in this chapter. The next chapter Seven gives a brief introduction about knowledge management software Apache Stanbol. In the eighth chapter, some existing disadvantages of semantic technology are presented. Finally, the conclusion part gives a brief summary and critique of the findings, some assumptions for the future work are also shown in the final chapter.

2 Fundamental Knowledge

As mentioned in the introductory part, semantic web has become a hot issue in recent years, it is said it will be the new generation of Web. Besides some key components of semantic web have been implemented in practise or affected many researches. This chapter is going to introduce the Web and communication history, then talk about why semantic web is different in Section 2.3. And in order to have a better understanding on the multi-channel communication conception, a deep analysis concerning ontology is shown in Section 2.5 and Section 2.6. Finally some other information about semantic web and linked open data are presented as supplement.

2.1 Communication

The definition of communication is quite broad. To human, this word is generally understood to mean exchanging information between human or objects(for example: radios, computers, etc.. . .). It is an indispensable part of our society, and also plays a significant role in different fields.

According to the conception and theory from Shannon and Weaver[2], communication is a procedure of delivering and receiving message or information between two different parts through some channels. A communication system could be represented like the following way:

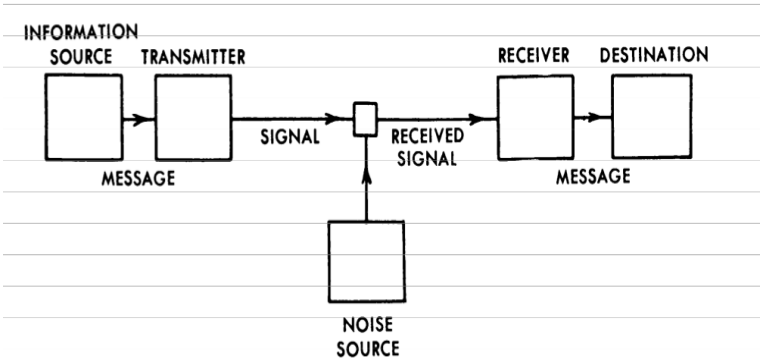

Figure 2.1: Communication System Model(Figure Owner: Claude Shannon) [2]

In a communication system, information sources refer to a machine or a person that produce messages, and these messages might be formed of text, spoken words, images or audio files. Then transmitter in Fig2.1 is prepared for decoding the message into signal. After decoding process, signal is delivered by some communication channels. A communication channel could be a physical transmitting medium or logical interconnection, it is generally seen as a bridge between sender and receiver. For instance, a case regarding the network, channel could be a cable, in the case of a speech, channel is the air. Receiver could be considered as a reversed transmitter, it receives the signal then encode them back to messages which are easily understood by machine or people. These messages will be sent to their destination afterwards. However in the process of communication, disturbance always exist, no matter from external or internal aspect. For example, in a presentation, the disturbance could be sounds from audience, in a telephone case, a wired or cable might be damaged so errors happen during transmitting. All those disturbances are called noise.

Additionally, in 1948, Claude Shannon, the author of book Mathematical Theory of Communication demonstrated a result in which describes that although noise disturbs communication channel, it is still practicable to transmit separate signal or data in a nearly error-free level within signal transmitting speed less than signal channel capacity context. Corresponding to the topic of communication, previous studies from Shannon and Weaver theory have reported that many issues are actually caused by the three following aspects[2].

1. *Technical Issue :*
   Technology could have positive or negative impact on accuracy of transference.

2. *Semantic :*
   This issue is about identity and understanding with regard that if receiver has right interpretation about the incoming message.

3. *Effectiveness:*
   Effectiveness level depends on the interpretation from semantic issue.

2.2 Internet VS Web

The Internet is an enormous and worldwide system of networks. It is a networking infrastructure which make millions of computers connecting together globally. It

forms a network which enable one or several computers could communicate with any other ones, as long as they are connected to the Internet.

The word of web is commonly understood to mean the abbreviation of World Wide Web. It is also widely known as WWW or W3. The initial concept of WEB is proposed by Berners-Lee in the early 1980, at the time, he was a software engineer at CERN, the large particle physics laboratory near Geneva, Switzerland. There were many scientists working for CERN at that moment, they wanted to exchange data and results of experiments, but they found it was difficult to achieve. Berners Lee found this need, and he understood the potential needs for so many computers connected together. Then he suggested to use hypertext for linking and accessing information between people, documents and institutions, thus people could exchange data in a more efficient way. Later in the 1990, he specified three main technologies which are still remain to be used for today's web in his proposal project[3]:

HTML: HyperText Markup Language. The publishing format for the web, able to format documents and resources to others.

URI: Uniform Resource Identifier. An "address" which is unique to each resource on the Web.

HTTP: HyperText Transfer Protocol. It allows computers to retrieve linked resources from the Web.

HTML use tags to represent text, hyper-links, documents, pictures and so on. For instance, in the following figure where tags are shown in bold:

```
<!DOCTYPE html>
<html>
<head>
This file demonstrates what HTML looks like
</head>

<body>
A very simple web page content
</body>
</html>
```

Figure 2.2: A graphical description of a very simple HTML document

In a nutshell, the Web is a system which uses interlinked hypertext documents, which could be accessed by user through the Internet. Also, a Web browser is the

bridge between web and Internet. The Web 2.0[4] is the second generation of the Web, it aims at improving ability to collaborate and share information by users. The Web 2.0 basically indicates the transition from those static HTML Web Pages to a more dynamic system. It focuses on serving web application to users in a better way. The other improved functionalities of Web 2.0 includes open communication with users, and more open sharing information. For instance, blogs, Wikipedia and web services could be all seen as components of Web 2.0. The Web 2.0 was previously used as a synonym for semantic web which is going to be introduced in the following section2.3. To sum up, the Web could be seen as a portion of whole Internet.

2.3 What is Semantic Web

The word semantic derives from ancient Greek, according to the explanation from Oxford dictionary, it is relating to meaning in language or logic. Then in computer science field, the term of semantic refers to the expression of vocabulary meaning. In other words, semantic is the interpretation of a language. A word could have very different meanings depending on the context, also there are denotations and connotations. The denotation of a word means its direct expression, whereas the connotation is an indirect or implied meaning. As an example of the difference between denotation and connotation, the *smell of the baking apple pie*[1] could directly mean the fragrance, but it might indirectly refer to happy memories at home. In addition, since semantic refers to the interpretation of natural language, so sometimes words coming out from a person might be "twisted" comparing to what he/she actually means. For example, when a person says *I love you* to different people, it might contain various meanings. It all depends on that how a person tries to understand it and this "twisted" could be seen as a form of semantics. However this interpretation later became the restriction for web developing, because machine do not have the human thinking pattern. How could let machine communicate with human and understand what people need, it has become a challenge. Therefore, it brings a new innovatory conception called semantic web.

The concept of semantic web was propose by Berners-Lee in 1988[5]. Although the semantic web is seen as an extension of the current web, its contents are mean-

[1]Apple pie sample originates from `http://www.answers.com/Q/What_are_some_examples_of_semantics`

ingful to computers. Semantic web is expected to interpret the exact meaning from users and could be used by machines afterwards. Five years ago, if someone said "I have found out that from the web", it means that someone found hyper links or web sites including information as they wanted. But semantic web converts the Web from simple keywords searching to a meaningful content query. The main purpose of semantic web is to enable the Web with "human" functionalities, such as identification, communication, self management, decision making and thinking. For example, if an user input "my mouse is dead, i need a new one", semantic web could recognize the explicit expression meaning from user, "my computer device is broken." Until now, semantic web is still a vision, it aims to allow data could be shared and reused in different environments or platforms, not just for displaying purpose. In order to have a better understanding on what semantic web could do, the following example could explain: John is a fan of basketball games. When he is surfing on the Internet, he types his favourite player's name into the searching engine. Would the result be exactly what he really wants? The answer might be negative. Normal web would just show John some links including the keywords that he typed. The Web cannot really understand what he is searching for. So according to the introduction part, semantic web is not a separate web, it is the extension of the Web which enables people to share contents. Also, semantic web offers a well defined data structure, it makes computers and people able to work in cooperation.

Now take a look at the basketball player sample again, when John inputs his favourite player's name, semantic web would give him back some connecting relation or news about this player instead of just hyper links. Also semantic web would list the basic information(e.g. team, home town, career records.) about the player. In a nutshell, semantic web is similar to a global connecting intelligent database. It offers an idea that anything could be linked with, also known as everything as a service(EAAS). In the near future, with more developments and researches about semantic web, it will lead some significant function and better process ability to machines. However, as a matter of fact, semantic web is not a fast growing technique, it will take years to develop it successfully.

At the XML Conference Meeting in 2000, Berners-Lee represented this semantic web stack.

The figure shows the proposed layers of the semantic web with higher level languages using the syntax and semantics of lower levels.

- Layer 1: Unicode and URI;

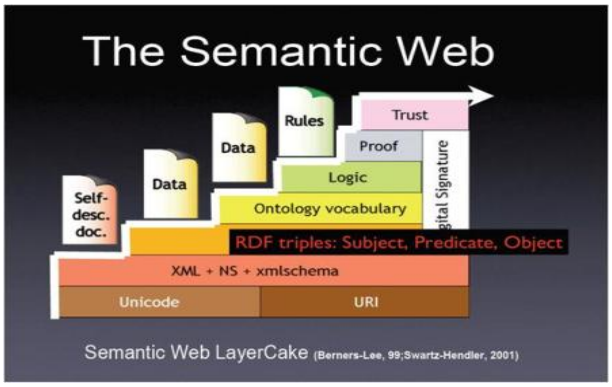

Figure 2.3: The Semantic Web Layer Cake(Figure Owner: Tim Berners-Lee) [5]

Unicode is an international standard for representing characters sets. Through the use of Unicode standard, all written and read languages on the web become accessible.[6] Unicode has spread through out the world at present and been used in our daily life everywhere. When user surf on the internet, they have been using Unicode already. *URI*(Uniform Resource Identifier) is a string of characters, it helps user point to a name of any resource type on the web, such as text, video, sound clip, or image.[7] Both of these two components are the fundamental of whole semantic web structure.

- Layer 2: XML+NS+XML Schema;

  This layer is responsible for representing contents and data in a well formed structure. *XML* is a common markup language used to contain information in documents. Since XML is one of the popular document formats used for developing web, there are always some names overlap or conflict problems when developing web. But XMLNS(XML Name Space) is the solution for those issues. Lastly, XML Schema provides the description about an XML document structure. There are more detailed introduction about XML in subsection 2.3.3

- Layer 3: RDF+RDF Schema;

  This layer is used to offer semantic models which describe resources, resource type and data interchange on the Web. The term of RDF is used to describe objects relationship by stating a triple graph, and RDFS is an extension of RDF to give the meaning of elements of RDF.[8]

- Layer 4: Ontology Vocabulary;

This layer aims at describing the relationships and meaning between various concepts. Ontology vocabulary is a formal and explicit specification of a shared conceptualization.

- Layer 5: Logic;

  It is developed to define a collection of logics for semantic web when proof layer performs these logics.[9] This layer could be very various and flexible because it depends how the users decide to develop semantic web.

- Layer 6 & 7:Trust and Proof;

  As mentioned in Logic layer, proof layer is responsible for executing the logics and then evaluating them with Trust layer which determines application should trust the given proof or not.[9] However, these three layers are still being under research, more investigation are needed to understand these aspects in the future.

2.3.1 Unicode

The primary task of a computer is to deal with digital numbers, and these numbers together compose characters which could be handled by the processor. This process is called encoding. In the early age of computer science development, there were hundreds of encoding systems for assigning these numbers. However, due to their storage restriction, it is not sufficient to contain characters for some languages or even one language, for example, Chinese. But since the diversity of languages and globalization, there are thousands of characters needed to be encoded on the web. Moreover, these encoding systems cannot exist concurrently. For instance, two encoding systems might need to assign a same number for two different characters or different numbers point to a same character. In that case, it might lead computer to decrease the data quality or even ruin the data process.[6]

As introduced early in Section 2.3, Unicode is also a system for representing character sets. But it is invented to merge all encoding systems into one universal encoding standard for text representation. Unicode standard assigns a distinctive number to every character so that most platforms, programs and languages could be implemented without any problem at today.

2.3.2 Uniform Resource Identifier

As shortly mentioned in Section 2.3 before, uniform resource identifier(URI) is comprised of a compact sequence of strings. It identifies a resource on the Web by providing a simple and extensible method, and this resource could be identified by the location or name, even both of them. One URI contains two subsets which are commonly used, **Uniform Resource Locator(URL)** and **Uniform Resource Name(URN)**.

A Uniform Resource Locator (URL) could identify where an available resource is and retrieve it by describing the primary access mechanism (network location). A URL also defines how the resources could be obtained by providing their resources prefix names, the most common types are: *http://* and *ftp://*. It could be considered as a street address in real life, here are a few examples about URL from RFC 3986 URI specification document.[7]

1. ftp://ftp.is.co.za/rfc/rfc1808.txt
2. http://www.ietf.org/rfc/rfc2396.txt
3. URL: mailto:John.Doe@example.com
4. telnet://192.0.2.16:80/

A Uniform Resource Name(URN) refers to a URI which uses URN schema to identify resources. Therefore URN dose not indicate the availability of identified resources, it is similar to a person's name. For example:

1. urn:oasis:names:specification:docbook:dtd:xml:4.1.2
2. tel:+1-816-555-1212

In a word, URI is responsible for providing network locations, while URN states a resource identity.

2.3.3 XML

As introduced in the section 2.3, the term of XML is abbreviated from extensible markup language, it is a mechanism to identify structure in documents. Tags have been already introduced in HTML, it is also used in XML to show information about text, pictures etc. In fact, **element** is the basic unit for XML syntax, each element usually contains two tags as start and end symbols. Start tag is displayed with two

angle brackets such as <body>, and end tag has the some structure but one more slash in those two brackets such as</body>, between these two tags, contents are included. Besides, other elements could also be enclosed between the start and end tags, then these elements are called child elements, for example:

```
<person>
  <sex>female</sex>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

Figure 2.4: child elements in XML

As we can see in the Figure 2.4, person tag could be seen as the parent tag in XML document, and between those: sex, first and last names are the child elements. But sometimes if there are too many child elements, attributes with name-value pair could replace the child elements, such as <person sex ="female">, both child elements and attributes would provide the same information. Also, if there is nothing in the element, an empty element could be written like this
. Lastly, in HTML documents, end tags are not necessary, but in XML document, there must be one end tag, that is why XML is in well structured form.

However, sometimes elements in XML document would have name conflict problems. For example, one element named *person* is defined twice in two different documents. And when these two documents are combined together, an application cannot deal with this situation. Therefore, XML Namespaces are invented to provide some unique elements and attributes names used in XML documents.[10] A XML Namespace is composed of two parts: *Namespace prefix* and *Namespace URI*. Take the example from Figure 2.4, to make person tag unique, we could add XML Namespace like this:

<personxml: person xmlns: personxml = "http://www.example.com/person">

Therefore, personxml is the Namespace prefix and "http://www.example.com/person" is the Namespace URI. With the help of xmlns, users and computers do not need to worry about named conflict when mixing documents.

Another essential component in XML layer of Semantic Web architecture is XML Schema. XML schema is a recommendation from World Wide Web Consortium, it provides a standard to define structures in XML documents. Besides, it could per-

form rules made by programmers to define each part of XML documents. XML Schema is powerful because it could support XML Namespace and describe different data types, it could also work with a database.

2.3.4 Resource Description Framework

In recent years, the term of Linked Data(LD) has become one typical pattern for representing information on the web. It is capable of querying and achieving unparalleled web search through integrating global data and information. This data type has brought a dramatic increase to semantic web. The essential of LD methodology consist of a group practises and principles, it aims at publishing structured information on web, its development is based on some standard web technologies such as Resource Description Framework(RDF), which is going to be discussed in the following.[11] Additionally the more detailed information about Linked Open Data will be introduced in the next Section 2.7.

RDF is a form which encodes structured information as a directed labelled graph, similar to the Web of Linked Data. RDF is a flexible graph based model, it has nodes and directed labelled arrows as its elements. The main goal of RDF is to provide general description about data which could be understood by applications on the Web, such description is often referred as meta-data. *Statement* is the basic unit for RDF, it is formed by three parts: **a subject**, **a predicate** and **an object**. And in RDF, a statement is basically the same as a set of triples. Each statement is visualized as a node-arc-node link in the following figure.

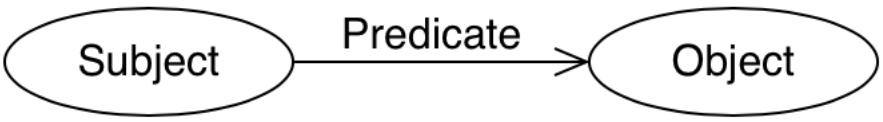

Figure 2.5: A standard statement of RDF(Figure Owner: Jeremy Carroll) [12]

The subject of a RDF statement could a resource of everything in this world, it covers physical and conceptual entity. And a resource or its property is uniquely expressed by a Uniform Resource Identifier(URI). The predicate of a RDF statement is the property of a resource, it determines the relationship between subject and object. The object of a RDF statement is also a resource type like the subject, but

sometimes it could just be literal value like number or string. The current study found that RDF has several possible benefits for semantic web:

- RDF is a steady framework which focuses on meta-data about internet resources, data could be easier identified.
- RDF has standard rules for describing and querying data, meta-data could be easier and faster to process.
- Users will gain more precise results due to meta-data.
- Intelligent software agents could work with more accurate data.

Generally, there are a few methods to exchange RDF graphs and store the graphic presentation of RDF data, these methods are serialization. W3C recommendation specified an XML syntax for one serialization, it is called RDF/XML, which demonstrates RDF data in XML form. And this syntax uses the most clear data structure for RDF model so machine could understand easily.

2.3.5 RDF Serialization

In order to understand that how XML is implemented for RDF serialization, here is one example to illustrate. *Mark is the developer of http:// www.jyu.fi/mark* is one RDF statement. In this statement, subject(resource) is **http:// www.jyu.fi/mark**, predicate(property) is **developer** and object(literal:string or number) is **Mark**. And it could be shown like this in RDF graph:

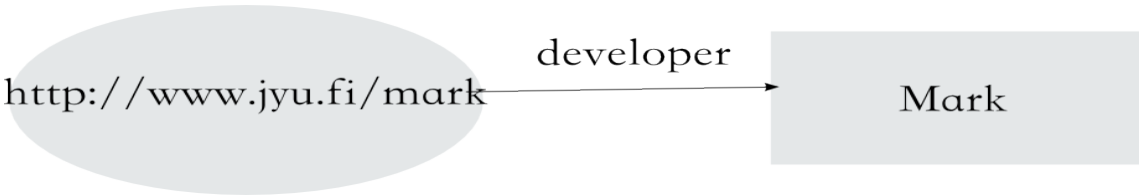

Figure 2.6: Example of RDF Graph

Therefore the RDF graph could be serialized to RDF/XML syntax to computers like this:

```
<rdf:RDF>
<rdf: Description about = "http://www.jyu.fi/mark">
<s:developer>Mark</s:developer>
```

```
</rdf:Description>
</rdf:RDF>
```

2.3.6 SPARQL

This section follows from the previous subsection 2.3.4 , it will introduce SPARQL briefly. As explained earlier, RDF is a flexible data model for representing information on the Web. In order to retrieve and handle RDF data, SPARQL was then created by the RDF Data Access Working Group. And in 2008 it became an official W3C Recommendation.[13]

SPARQL is able to obtain values from structure and semi structured data, and it could detect data by unknown relation queries. In addition, it could transform RDF data and accomplish complicated database joint.

SPARQL syntax is close to RDF, because some concepts used in definition of SPARQL syntax are taken from RDF concepts and abstract syntax with minor modification.

One standard SPARQL query is composed of five parts: *prefix declaration*, *dataset description*, *a SELECT clause*, *query pattern* and *query modifiers*.

- Prefix Declaration is used for URI abbreviation.
- Dataset Description(A FROM clause) specifies the sources or datasets to be queried.
- A SELECT clause identifies what information from query should be returned to user.
- Query Pattern(A WHERE clause) specifies filtered values of underlying dataset.
- Query Modifier indicates ordering querying results and preserve duplicate solutions.

At the end of this section, Figure 2.7 below indicates a general form of SPARQL query:

2.4 Semantic Meta-data, Annotation and Named Entity

Semantic Meta-data: The terms of meta-data can be defined as "data about data", it is a very popular topic in academic and real world. With development of the

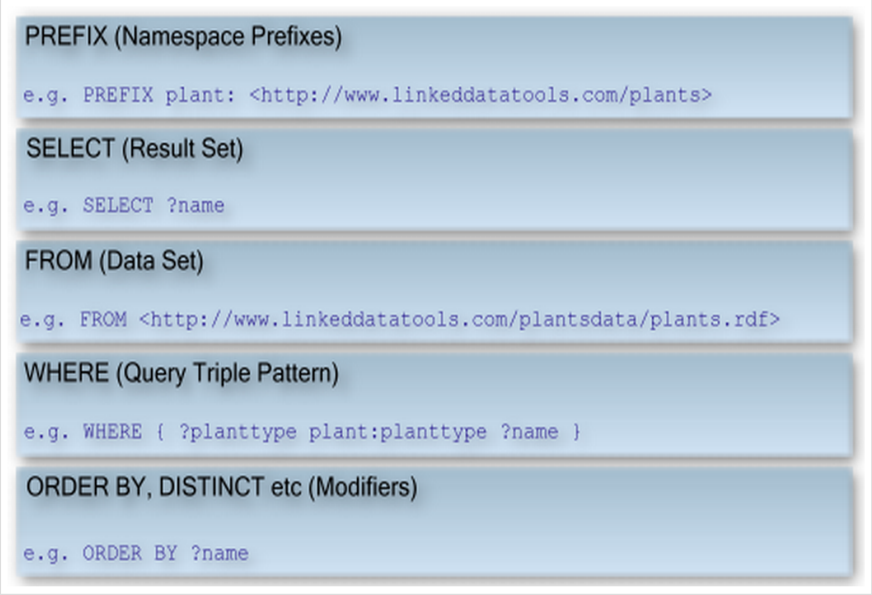

Figure 2.7: SPARQL query
[14]

Web, people are no longer satisfied with single HTML interlinked structure. People expect a sophisticated approach that meta-data combined with pages or information resources could be indicated by URI. Besides, the creation of XML and RDF also bring meta-data to the stage. Generally, meta-data could be used for two purposes, one is about data construction and specification, the other one is data its self, the content. In semantic web context, meta-data could interpret information and disambiguate them. It aims at achieving comprehensive management of document by providing the formalization of content.

Semantic Annotation: First of all, semantic annotation is one type of meta-data, it is very specific. Since semantic web is able to interpret information on web, where the data needs to be understood by computers. Semantic annotation is to annotate description on meta-data resource.[15] It provides class and instances information(property values and relationship) with respect to entities in a particular domain. In a nutshell, semantic annotation could be seen as book, and the URIs are each page inside. The following figure from Kiryakov(2004) demonstrate how semantic annotation works from a general view:

Named Entity: Named entities are regarded as *places*, *people*, *organizations* and other

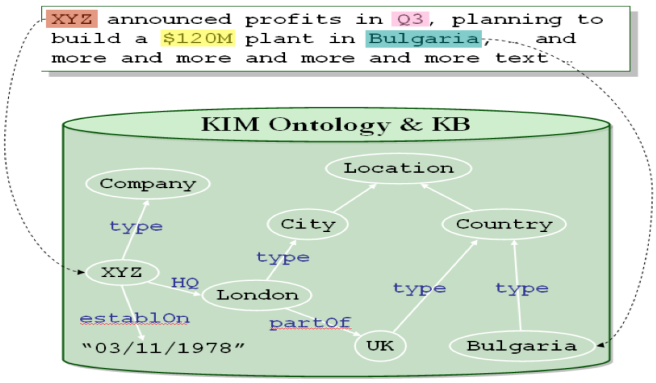

Figure 2.8: Semantic Annotation(Figure Owner: Atanas Kiryakov) [8]

things in Natural Language Process field. It is a description of an object with semantic characteristics that could be interpreted for future usage. Besides, named entities contain values such as *number*m *address* and *time*. Comparing with vocabulary, named entities require more specific understanding of universal knowledge and conceptualization.

2.5 Ontology

The Web is an entity of documents for people, whereas semantic web is an entity of documents for computers. Current a web page is written in HTML, this language is easy for human to read and use, but its structure is complicated so that machines can only gather few useful information from it. Computers read HTML documents like hieroglyphics, so how could we make machines understand what users input? Either improve computers to a super intelligent level, or change the structure of meta-data so that computers are able to understand. Based on current techniques, second solution seems a bit easier. So semantic web is to collect data which are in a well-structured format for computers to read and understand in an easy way. After data are input into computer, computing machine will acquire useful information from the data, then the acquired information would be utilized for determining logical truth. For example, Mary is the mother of Gary, then Mary can be inferred she is a female. This process is so called reasoning. In order to obtain information with logical facts, computers should firstly understand in which domain they are coping with, the general concepts in that domain and the reasoning policies. For instance,

person A has a sibling, person B. From human perspective, we also understand person B is the sister or brother to person A, but machine might not understand this inverse relationship because it has no idea about symmetry, whereas in ontology this issue has been solved.

In a nutshell, a specification provides shared and common understanding of a domain that could be used both by people and machines, it is called ontology. The term of ontology originates from philosophy,it refers to the study of things which are existed. And now this concept has been applied in many different fields. For example, in autonomic intelligence aspect, the ontology is created to eliminate the conflicting definition and understanding between literature. The things described by an ontology in a domain of discourse by a formal and explicit ways are called concepts(**classes**). And the diverse features and attributes of concepts are slots(**properties**), the restriction of properties are facets(**role restrictions**). In addition, a group of individual instances from classes along with an ontology could start to compose a knowledge base. Therefore, in real life, the completion of ontology implies the initiation of knowledge base.[16]

Class is the essential component of an ontology, it illustrates conception in a domain. For instance, a class of coffee could mean all coffees, and one specific kind from this class is called instance, such as espresso coffee is an instance of class coffee. Besides, class could also be specified into subclass which represent more detailed concept than superclass. For example, black coffee, white coffee, espresso and cappuccino could be the subclasses from the class of all coffees. Alternatively, how to divide a superclass is very flexible, a class of all coffees could also be grouped into coffee with sugar or coffee without sugar.

Properties describe attributes and characteristics of classes and instances. For example, Starbucks espresso is produced by Starbucks, hence *produce* is the attribute of Starbucks (instance). Moreover from class perspective, flavour, milk level and so on could be the properties for instances of class coffee. The following figure illustrates class, instance and property by giving Starbucks as an example:

The espresso(instance) has a property *producer* which is the value from an instance of the subclass Starbucks espresso. The Starbucks could be the instance of class coffee producer, since coffee produce has one property named *produce*, hence all instances of coffee producer(class) could also own this property. This could be considered as consistency of data. The process of data consistency is remaining the information unchanging when data are transferred between various applications

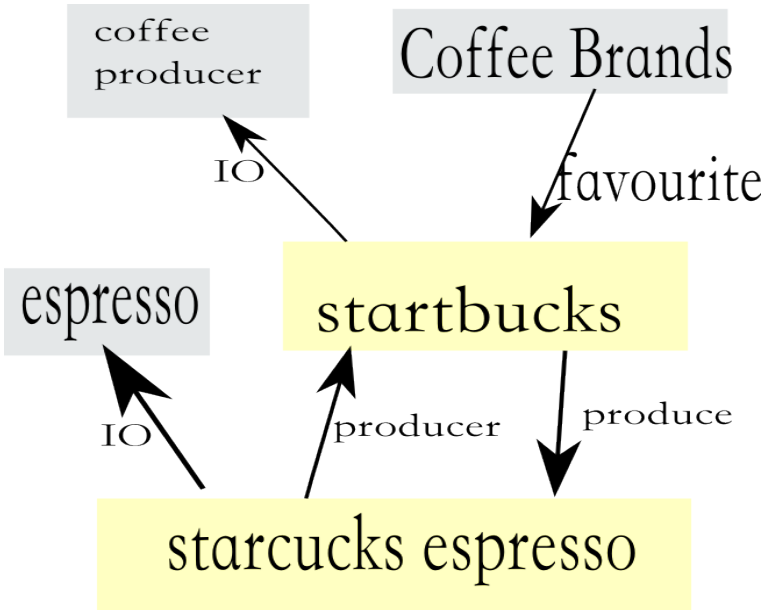

Figure 2.9: Classes, instances and relationship between them in coffee domain

or networks. Data consistency could prevent information lost and ensure the data quality.

Therefore, in order to develop an ontology, the approach could be like this:

- define class of ontology.
- put classes in a taxonomic hierarchy.
- define property and its value.

2.5.1 Web Ontology Language

As mentioned earlier, the definition and use of ontologies to the semantic web are important and crucial. Over the past decade, how to correctly use ontologies for sharing and defining knowledge has become a controversial topic for researchers. Although there is no precise answer about what ontology is exactly composed of, most ontologies are referring to one or two related things(e.g., stating that a cow is a mammal). So Guarino(1998)[17] gave a definition for ontology in his research, a logical theory that accounts for the intended meaning of a formal vocabulary. One well known ability of ontology language is to expand existed formal vocabulary based on logic truth. As a consequence, user is able to add or delete domain specification for modifying ontology, which it is beneficial to exchange or make use of

information.

The OWL(Web Ontology Language) was designed to be interpreted by machines instead of human. It is mainly used for two purposes. First it intends to define terminology and process data modelling in a flexible and fast way. Second OWL is an efficient data query approach. OWL became a W3C recommendation on 2004, it could bee seen as an extension from RDF because they are almost alike, but OWL has better computability, larger vocabulary and rigid constraints. Here is one diagram below shows what OWL looks like:

```
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:owl="http://www.w3.org/2002/07/owl#"
        xmlns:dc="http://purl.org/dc/elements/1.1/">

        <!-- OWL Header Example -->
        <owl:Ontology rdf:about="http://www.linkeddatatools.com/plants">
                <dc:title>The LinkedDataTools.com Example Plant Ontology</dc:title>
                <dc:description>An example ontology written for the LinkedDataTools.com RDFS & OWL introduction tutorial</dc:description>
        </owl:Ontology>

        <!-- OWL Class Definition Example -->
        <owl:Class rdf:about="http://www.linkeddatatools.com/plants#planttype">
                <rdfs:label>The plant type</rdfs:label>
                <rdfs:comment>The class of plant types.</rdfs:comment>
        </owl:Class>

</rdf:RDF>
```

Figure 2.10: OWL Sample
[14]

As Figure 2.10 demonstrates, there is a header included in an ontology. An ontology header usually stores information that explains what this ontology contains. What's more, it could also provide information about version and whether it uses elements from other ontologies.

- Instance: Generally, an instance is seen an object. In OWL, it is called *individual* in term of description logic. The individual is a member of one stated OWL class, it could be regarded as class extension. For example, there is one book called "1984", and someone is creating one book review site and needs one ontology for the site. In this case, there is no need to concern any situation because any copy of the "1984" is the same. So *"1984"* is called an individual.

- Class: An OWL class is a collection of individuals which share common characteristics. One class could own infinite individuals, at the same time, one individual could belong to one or more classes, even none class. Besides, OWL

classes also have subclasses, it help machine reader easily infer the relationship between them hence improve the working efficiency. For instance, there is one Class:*Mammal*, and it has one Individual(Instance):*Ape*. Meanwhile, Class:Mammal is also the Subclass of *Animal*, so machine now can infer the Ape is also a kind of Animal.

In addition, classes in OWL should be explicitly declared since it sometimes cause a wrong impression with individual. Take the book "1984" as an example again, the book in different libraries might have its own item code, location and availability. In this case, calling "1984" a class makes sense.

- Properties: *Properties* are the relationship between individuals in OWL. There are two types of property: *Object property* and *Data type property*. Data type property is the literal value(name,number...) between individuals of OWL class. It is expressed as *OWL:DataTypeProperty*. Object property relates individuals of two OWL classes, for example hasChild could be an individual type property of Class:Parent and Class: Child. It is formulated as *OWL:ObjectProperty*.

Briefly speaking, Web Ontology Language is based on *Description Logic*[2], it could be used to tell what this world can contain. Besides, comparing with RDFS, OWL provides a wider range of vocabulary which could describe data model comprehensively and OWL allows users to define relationship between ontologies by annotation.

2.5.2 Sub languages of OWL

OWL is composed of three sublanguages, which respectively are: *OWL Lite*, *OWL DL* and *OWL Full*. All these three variants with different level of expressiveness can describe instance, classes and property, they aim at supporting different users with their demands. Expressiveness is the expressive power of one language, the stronger expressiveness a language has, the more precise and various process to represent an idea. It is generally accepted that OWL could be used to develop complex computational ontologies, each of its sub languages can handle with different ontology requirements.

OWL Full: Strictly speaking, OWL Full cannot be deemed as a sublanguage, because it has all the OWL language features and no limitation to use RDF con-

[2]A language to express formal knowledge

structs. For example, *owl:Class* in OWL Full document and *rdfs:Class* are equal in function, whereas *owl:Class* in OWL Lite or OWL DL document might be a subclass of *rdfs:Class*. Besides, a class in OWL Full could be regarded as an individual, and both object properties and datatype properties of the individual are composed of all resources because *owl: Thing* is equivalent to *rdfs: Resource*. These two properties in OWL Full are connected, *owl: ObjectProperty* is equal to *rdfs: Property* and datatype property could be seen as a subclass of object property.[18]

Although OWL Full allows expressivity of OWL and metamodelling features of RDF to be associated, OWL Full is not possible to perform all reasoning features from various relevant applications. In conclusion, it is still under discussion whether a complete implementation of OWL Full could be executed in practise.[19].

OWL DL: OWL DL is a more computational completeness and decidability alternative to OWL Full. The aim of OWL DL is to support reasoning applications with description language. Also as OWL Full, OWL DL includes all OWL language constructs, but have restriction when using them. For example, a class cannot be viewed as an instance of another class.

OWL Lite: OWL Lite complies with all the constructs of OWL DL. It is used for simple data modelling, it is even simpler than OWL DL because of lower complexity. But it comes up with a positive reasoning efficiency for OWL Lite.[18]

Therefore when developers try to develop one ontology, they need to consider which is the most suitable to their needs. But how to make a choice from these three alternatives? According to the specification from Ontology Working Group, each OWL Lite ontology is also a OWL DL ontology, the choice between OWL Lite and OWL DL is determined by the degree of user expressive restriction provided by OWL DL. The selection between OWL DL and OWL Full is based on that how much meta-modelling abilities of RDF Schema users want to demonstrate, for example: defining a class within another class and properties to classes. There is one thing need to be noticed is that no complete OWL Full implementation currently exist, so reasoner for OWL Full have less predictability comparing with OWL DL.[18]

In conclusion, OWL Lite and OWL DL are the extensions of RDF but with restricted terms, while OWL Full could be viewed as a transformation from RDF. In addition, all three kinds of OWL documents(Lite, DL and Full) are and must be RDF

documents. But from the inverse direction, every RDF document could only if be an OWL Full document. Since only some RDF documents are OWL Lite and OWL DL documents, when developers are trying to import or change an RDF document to OWL, there are some concerns which need to be taken into account.[18] For instance, when defining the suitable expressiveness of OWL DL and OWL Lite documents, there are some cares should be taken to make sure that RDF documents abided by restrictions required from OWL Lite or OWL DL.

In fact, OWL not only has these three sublanguages, it also has a new generation *OWL 2*. OWL 2 has better abilities to deal with computational complexity, however it comes with more restrictions for developers to use. In this thesis, OWL 1 is recommended for multi-channel framework proposal due to its function integrity comparing with OWL 2. But in the next section 2.5.3, OWL 2 will be shortly introduced.

2.5.3 OWL 2

The OWL 2 Web Ontology Language is the latest version for defining semantic web and representing knowledge about things. It became a W3C Recommendation on October 2009. OWL 2 is an extension of OWL 1, as a result it inherits all the features from OWL 1 and enhances the reasoning capability. An OWL 2 ontology has the similar structure as OWL 1, it comprises three notions[20]: *entities*, *expressions* and *axioms*. On the other hand, several new features are added to OWL 2, the following list provides a brief illustration[21]:

Syntactic sugar

This feature helps developer make pattern design in a easier way and it does not change any expressiveness, semantics and complexity. Besides, reasoning processing becomes more efficient.

New constructs for property

This feature allows user to define additional restriction on properties, meanwhile, express new characteristics of properties. In addition, the incompatibility is strengthen in OWL 2.

Datatype extension

To provide a wide range of datatype property in OWL 2 now is available, for

example in OWL 1 a senior could be defined with an age without range restriction. In OWL 2, a senior could have an age over than 60.

Easy metamodelling ability

According to OWL 1 DL specification, the name(a thing) should be used precisely, it cannot be both a class and an individual. However OWL 2 allow user to define the same term for classes and individuals via *punning*[3]. For example, father could be both an instance of a class and a class of all fathers. Also, an object property and a class can have the same name for use. But a name for both a class and a datatype in OWL 2 is forbidden, each kind of property can only be given with one name.

Enlarged annotation ability

In Web Ontology Language, annotation consists of unofficial information, in the Section 2.4 more precise explanation will be introduced. Comparing with annotation for ontology entity in OWL 1, OWL 2 provides a new construct for annotation, it allows user to annotate axioms and annotation itself.

As introduced earlier, OWL 1 has three sublanguages for different ontology purposes. In OWL 2, there are also three variants, but they are called *profiles*. Each OWL 2 profile could be seen as a slim version of OWL 2 and able to handle with specific application requirements in an efficient way. Besides, every OWL 2 profile is defined by placing restriction on the structure of OWL 2 ontology.[20] The three OWL 2 profiles are: *OWL 2 EL*, *OWL 2 QL* and *OWL 2RL*.

OWL 2 EL:

The design of OWL 2 EL is based on the EL family of description logic(EL++[4]). This profile aims at developing ontologies to deal with cases where users need to describe a large number of classes and/or properties, the classes could be defined in terms of existed things with complicated descriptions. Also, this profile could capture the expressiveness of many large scale ontologies. For example, OWL 2 EL could be provide a large scale class to define biomedical ontology *SNOMED CT*[5].[23] Moreover, the reasoning capability of this pro-

[3]Pun means that a joke exploit the different possible meanings of a word.

[4]EL++ is a lightweight description logic which admits sound and complete reasoning in polytime, it became a syntactic component of OWL 1 DL.[22]

[5]It is the most comprehensive and precise clinical health terminology product in the world

file could be implemented in polynomial time based on the size of ontologies, therefore this profile is pretty suitable for inference tasks.

OWL 2 QL:

QL is abbreviated from query language, it is based on the DL-Lite family of description logic. The purpose of this profile is to process a large number of instance data, and efficiently reason on top of it. The important reasoning characteristic of OWL 2 QL is relating query answering, for example, information from an ontology could be captured by rewriting a query into a simple SQL query. And this process would not cause any affect to data stored in the relational database system(RDBMS).[23]

OWL 2 RL:

The abbreviation of RL reflects the relation to Rules Language, this profile has been designed for applications that could use proper expressivity to do scalable reasoning, and describe rules in ontology. OWL 2 RL could be seen as a perfect option for companies which have RDF applications. Also some restrictions in this profile make it possible to use rule based reasoning engine by defining customer own business logics. Some individuals which contain implicit meaning in knowledge base will not be shown during reasoning because of these restrictions.

2.5.4 Ontology Personalization

This section briefly explains what ontology personalization is about and how it could help this thesis. A key aspect of ontology personalization called user profiles could help understand this part. In the process of web information collecting, user profiles are created to reflect what users need and their preference, it also helps interpret semantic meanings.[24] User profiles usually could be classified and shown in two schematic: *data diagram* and *information diagram*. Data diagram is obtained through database analysis while information diagram acquired by questionnaire and interview as well as machine learning technique.

Ontology personalization[24] refers to a conceptualization model. To distinguish on-line users might have individual expectation from identical things, personalized ontology is created to develop user profiles with formal description and specification. Take *Helsinki* as an example, tourists might search it and look for some interesting places to visit. But others might demand information that differ from tourists,

such as local weather, history, etc. Even sometimes same user in different situation are expecting diverse results. Therefore a user model constructing personalized ontology according to different situation query is needed. Future investigation in ontology personalization is strongly recommended.

2.6 Ontology Matching

It is necessary here to clarify what semantic heterogeneity problem is before introducing ontology matching. The term of *heterogeneity* refers to the differences between different things, even in a same domain. For example, when independent individuals are developing database schema for the same domain, the results could be quite different because developers have their own comprehension, and those differences are seen as *semantic heterogeneity*.[25] Semantic heterogeneity could also exist in other occasions, such as enterprise information integration, XML documents, and ontologies etc.... At present, multiple data systems have been utilized quite wide in may fields, in order to make them understand each other schema, the semantic heterogeneity must be eliminated.

In semantic technologies area, *ontology matching* is a way to solve semantic heterogeneity issue. Matching function takes ontology as input source and determines the relationship(correspondence) between ontology entities as output.[26] These correspondence could be addressed for different tasks, for example ontologies combination, data interpretation and query. Therefore, the goal of ontology matching is interoperation.

2.6.1 Motivation

When describing ontologies, semantic heterogeneity issue might happen because using different languages, dissimilar terminologies and modelling, etc... Firstly, there is one simple example from Shvaiko's(2005)[26] study in Figure 2.11 to illustrate ontologies matching problem.

There are two ontologies, ontology *Product* and *Monograph*presented in Figure 2.11. The classes appear in rectangle without corner, and link to their properties by dash lines with arrow, for example, *title* as an attribute is defined in String domain. The relationship(correspondence) between classes or properties are shown by the line with relation symbols, such as *Book* in ontology Product is greater($\geq$) than

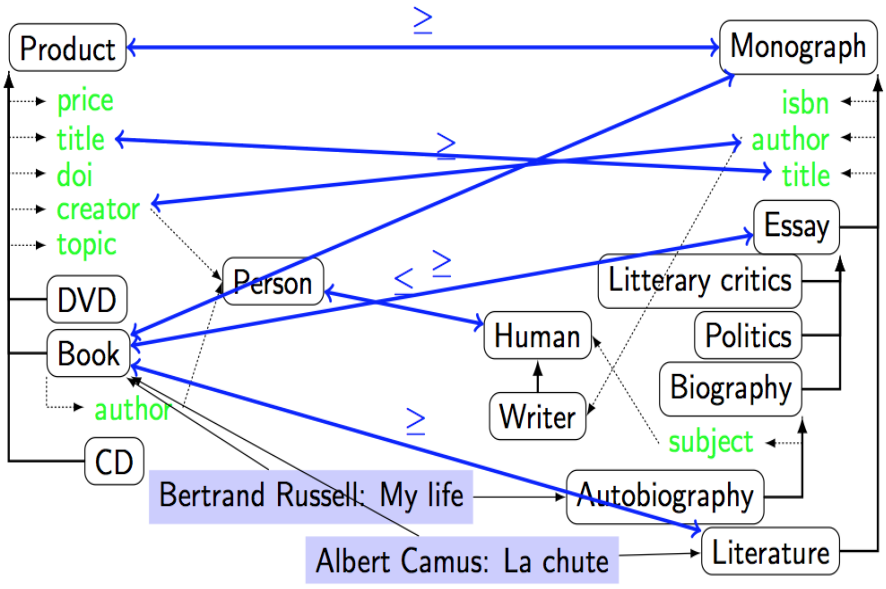

Figure 2.11: Ontologies Matching (Figure Owner: Pavel Shvaiko) [26]

Essay in ontology Monograph. *Bertrand Russell* and *Albert Camus* are two shared individuals.[27]

Now let's take the following case as assumption, when two companies start co-operating to expand their business, it requires both companies to integrate their products or client data which are stored in ontology documents. Since these ontologies contain class relationship,descriptions for properties and instances, the ontologies integration might cause semantic heterogeneity problem. However, once the correspondence is determined after merge, it could be used for many purposes, such as reasoning. For example, from the Fig 2.11, the property title in both ontology product and monograph can be merged, then it could tell class product contain but greater than class monograph. [27]

2.6.2 Matching method

There are two steps to solve the problem from subsection 2.6.1, determine the alignment between ontologies by matching method is the first step. An *alignment* is a set of correspondences among the merged ontologies(entities).

But how correspondence could be represented? Shvaiko's tutorial on ontology

matching(2006) shows that correspondence could be seen as a tuple[6]. For instance, the correspondence between given ontologies could be shown like this: {id, e , e' R, n }. [26]

- *id* is the individual name for correspondence.
- *e* and *e'* represent the entities of given ontologies respectively.
- *R* explains the relationship from *e* to *e'*, such as, greater or equal to($\supseteq$)[7].
- *n* is confidence measure in the correspondence, it varies between 0 to 1, higher value of confidence states higher relation probability.

Therefore, the correspondence in Figure 2.11 can be shown like this, {id01, product, monograph, $\geq$, 0.8}. After some correspondence are found, they will form an alignment for matching process.

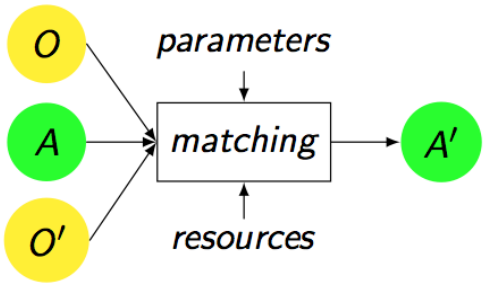

Figure 2.12: Matching Operation (Figure Owner: Pavel Shvaiko) [26]

Figure 2.12 above can describe how matching is operated. *A'* is the sequent alignment for ontologies O1 and O2, and *A* is an input alignment which can affect the matching operation, it might come from other resources or exist in the same merged ontologies. Besides a set of parameters(datasets) and resources can also determine the output alignment. Lastly, the number of alignments between ontologies range from 1:1 to n:n.[27]

[6] A tuple is an ordered list of elements

[7] It is the same as the operator $\geq$ in Figure 2.11

2.7 Linked Open Data

As introduced in the previous Section 2.3.4 about semantic web in this thesis, generating machine readable data and connecting all documents on web have become very attractive. In order to achieve the goal, a new type Web named Linked Data Web is being created and under development. Therefore this section of thesis discusses the concept behind the Linked Data Web, *Linked Data*. The concept of linked data came from Berners-Lee's article which describe the future trends about web. [9] And now this concept has become a popular research and development topic in academic field and reality world.

The term *linked open data* is technically understood to mean that data with explicit definition are published on web for machine reading. Its distinctive attribute is to connect or be connected by external dataset, it is proposed that it might be the ideal solution for web data publishing and data connection. And in fact, in the past few years, the concepts of linked data and semantic web have become exchangeable. Both of them have the same goals concerning machine readable data generation. Besides, the main ideas of linked data is to create structured data by using RDF data model and interchange RDF links with other links from different data sources. In a consequence, this new type of data might be seen as the fact of semantic web. Here is a list below that demonstrates the comparison between modern used data and linked data.

Flexibility Both types of data could be published on the web at any time by anyone. Except the format of linked data has to fit RDF document.

Browser Usability It might be a better idea to use specific browser for linked data because it is developed for machine understanding. Most of the current browsers are developed to manipulate HTML documents.

Connectivity Linked data aims at connecting everything in the world, comparing with traditional web which only connect HTML documents, it has a wider range.

Scalability The current study found that linked data web is able to develop application based on unbound dataset.[28] It means the semantic applications could perform in a more efficient way.

Berners-Lee(2006)[9] offered a draft proposal when developing a web of linked data.

1. Assign a distinctive or universal URI name for source or concept, it could disambiguate meanings for documents.

2. In order to make sure when publishing data where URIs should be unique, one suggestion is to put HTTP restriction on URIs.

3. When user input URIs into web browser, user should get respond with relevant useful information.

4. For the purpose of expanding information and linked data achievement, the related links should be connected and explored.

3 Smart Multi-Channel Communication

Multi-Channel Communication is an innovation technology which has its own theoretical foundation in the work of Guilford's Structure of Intellect. However this thesis has no direct relationship with human intelligence. Generally speaking, multiple channel communication is to send or transmit message from resource to goal sites respectively. Messages would be sent from one channel to another or some others, just like driving a car could have several options at a cross or water spread into different rivers. Multiple channel communications are commonly used in the following terms:

1. cross media publishing and communications
2. multi-touch-point campaigns
3. Integrated marketing campaigns

In Business-to-Business and Business-to-Customer(B2C) models, multichannel communication is the fundamental, it would offer more preferable patterns for consumers when they purchase items. In a word, among several channels, message text are integrated or translated into proper version to fit the right channels which are going to receive them. The contents from different types of media would be send at a appropriate time directly to the right person,in this case,multichannel communication could increase response rates, market awareness,revenue and profitability for the investors.

3.1 Framework Overview

The initial proposal of multiple channel communication for business comes from work by Nagy. In that research, Nagy offered a sketch, multiple channel communication framework could be seen in the below Figure 3.1.

The structure of framework is composed by two important parts: *Knowledge Base* and *Message Process Engine*. Knowledge base could be seen as a universal database, it is used to save and update information. These information will come from *ontologies*. Knowledge base is used to save and update information, so it could be

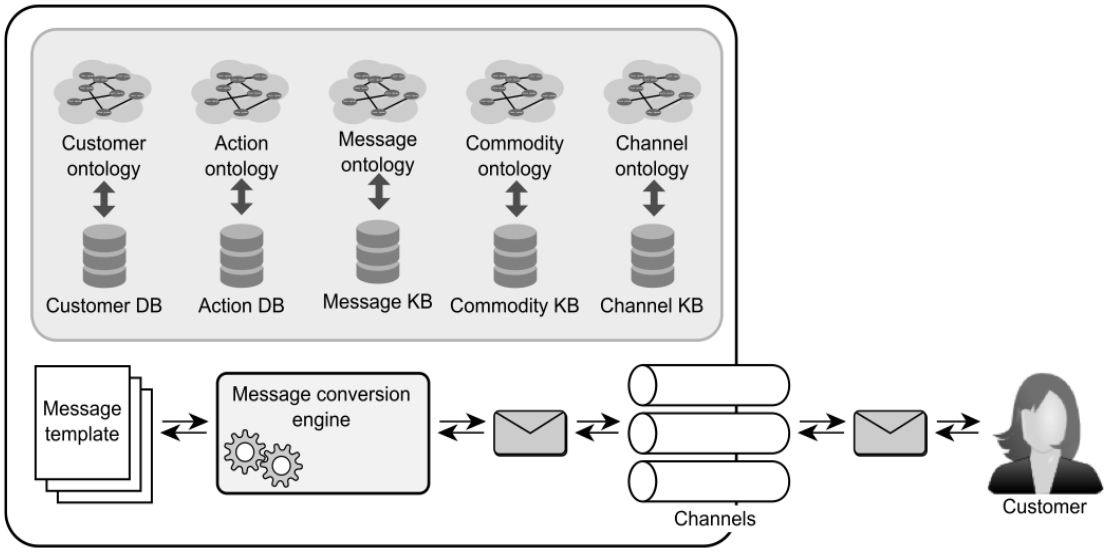

Figure 3.1: Multiple Channel Communication Framework(Figure Owner: Michael Nagy)

[1]

seen as a universal database. Message process engine is responsible to interpret and merge messages, it could also choose a proper channel for sending and receiving the messages. The flowchart of message process engine could be seen in the Figure 3.1.

As Figure 3.1 shows, five specified ontologies which compose a knowledge base are proposed for the framework. Commodity ontology contains all the information about commercial goods and business services. Channel ontology describes all available communication channels. Message ontology expresses two types of messages in the framework: *concrete message* and *abstract message*. Customer ontology is similar to user profiles mentioned earlier, it is a customer diagram that includes all personal information such as contact number, ID, age, profession and preferences etc.. . . Action ontology refers to the actions which buyer and seller perform. The more detailed explanation about these 5 parts will be discussed later in the thesis. Also these five ontologies could be connected to each other, so when administrator modify any part of the knowledge base, the rest parts could give correct responding to adjust the modification. From customer perspective, when customers send some messages through preferred channels, the key information that relating to business will be abstracted by message conversion engine. Also customer's important information and preferable communication channel would be stored in message tem-

plate at where could be used for next time. From company perspective, sometimes customers will feel annoyed by useless messages. In order to avoid that situation, framework could reach their customers with preferred channels and information by implementing message process engine.

Commodity Ontology

In this framework, commodity ontology is represented in terms of business domain. It is composed of two main parts: products and business service. This initial ontology could be infinitivally extended based on the user needs. The following diagram explains basis and what extensions might be included in commodity ontology.

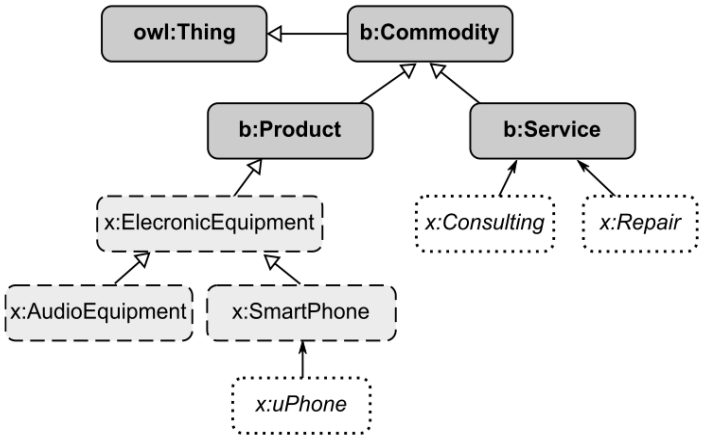

Figure 3.2: Commodity Ontology Structure(Figure Owner: Michael Nagy) [1]

Product and service are all subclasses of main Class:Commodity, at the same time, product and services could also own various subclasses according to real business scenarios, for instance product might have subclass electronic equipment and service might own subclass consulting. Furthermore, each subclass then could define its own instances like uPhone in Figure 3.2. Due to the flexibility of ontology, commodity ontology can also import or be imported to integrate with other ontologies.

Channel Ontology

Channel ontology is represented in terms of communication domain. The structure of channel ontology is shown below:

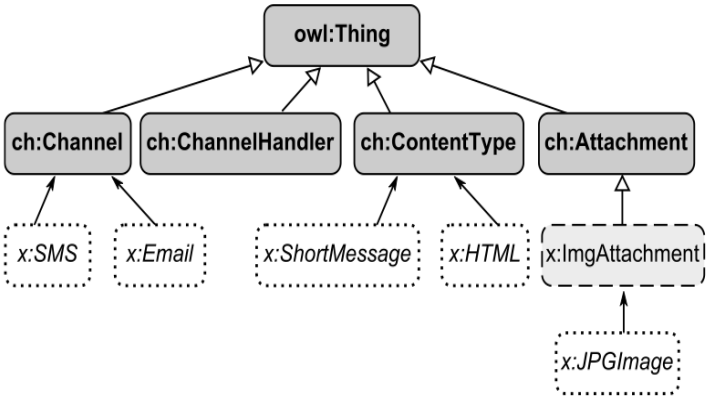

Figure 3.3: Channel Ontology Structure(Figure Owner: Michael Nagy) [1]

The core of channel ontology is Class: Channel, user could define Class: Channel by adding or reducing different communication approaches in terms of individual business requirement. For instance, SMS and Email in Figure 3.3 are two communication channels(subclasses) of Class:channel. Class: Channel Handler describes how the message should be formed and operated. And Class: Content Type could distinguish message type and it could be connected with message conversion engine for future information analysis. Class: Attachment is responsible for recognizing the format of attaching file in messages, such as image or voice. Lastly, channel class could be defined with various properties, like *speed*, *reliability* and *cost*. But these properties are too limited for real scenario, there will be more properties defined in the future.

Customer Ontology

As mentioned earlier in Ontology Personalization section, customer ontology is a model which store and describe all relating information about people. It is represented in terms of Thing domain, the central part of customer ontology is Class: Contact. Furthermore, there is no limitation on how many reaching communication channels could be preferred by one customer. Besides a data type property with value could be defined in contact class, it is called *preference*. And customer class has a property named *hasContact*, it defines how many contact ways that one customer can have. The value in float type arranges from 0 to 1, 0 means customer do

not want to be reached by any communication channel, and increasing numerical value express the percentage of willing to be contacted. There is one table shows all customer ontology properties and corresponding property value.

URI	Min. card.	Max. card.	Domain	Range
cu:hasContact	0	n	cu:Customer	cu:Contact
cu:correspondingChannel	1	1	cu:Contact	ch:Channel
cu:contactAddress	1	1	cu:Contact	xsd:string
cu:preference	1	1	cu:Contact	xsd:float

Figure 3.4: Customer Ontology Property Table(Figure Owner: Michael Nagy) [1]

Lastly, this ontology is used to help companies know better about their customers, but customer privacy will not be stored.[1]

Action Ontology

In this framework, action ontology is represented in terms of business rules. It mainly describes a whole process of message working, who are the sender and receiver, what is the content about, which communication channel this message could use. Besides, action ontology could also describe some business actions, such as asking for information, complaining about services, etc. There is one figure below explains relationship between action, message and products.

However, future research might need to investigate more specifies on property of action ontology.

Message Ontology

Message ontology plays a key role in the whole framework proposal. In the framework, both incoming and outgoing messages are categorized into two types: *concrete* and *abstract* messages. A concrete message only contain crucial information which people want to know, and to multiple channel communication framework, it is not

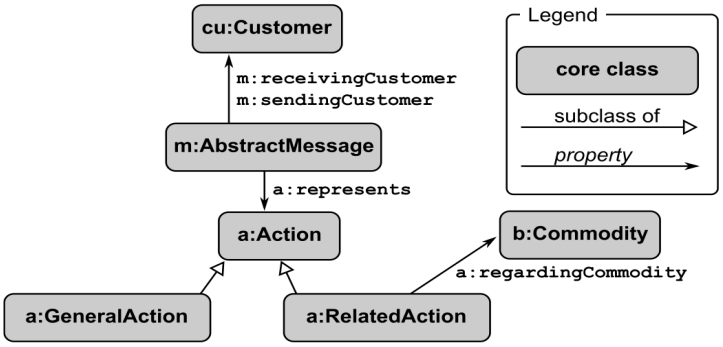

Figure 3.5: Action,Message and Products Relation(Figure Owner: Michael Nagy) [1]

necessary to include who send or received. Abstract messages could be viewed as it is responsible for things which concrete message cannot cover, for example contact information, channel preferences. As can be seen from Figure 3.6, message ontol-

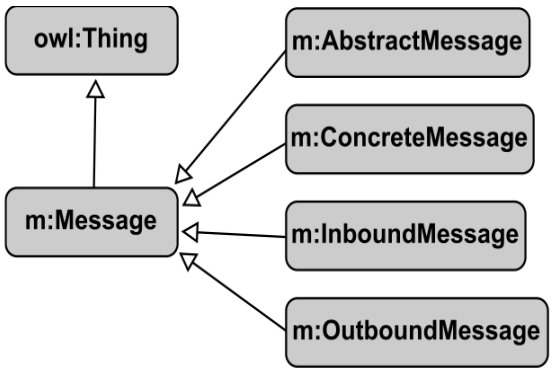

Figure 3.6: Message Ontology(Figure Owner: Michael Nagy) [1]

ogy is represented in terms of All Thing, it has four subclasses as mentioned earlier. Since concrete message only show the central information from message, it could be defined to have several data type property with Value: String, such as contact information, subject and primary contents. Besides, concrete message class could also have two object properties: *channelConnect* and *hasAttachment*. First property is used to connect with channel ontology for sending and receiving message, and the second property is to detect if there is attachment along with message.

4 Smart Channel Selection

In the framework, *channel selection* is expected to pick up a preferred communication way automatically to reach customers in real business scenarios. Some customers do not like using SMS because the character number limitations, by contrast, email system let user input information as much as they want to. Therefore, the email system channel will have a higher probability to be used according to consumer preference as well speed, reliability and availability etc.

An approach for selecting smart channel is proposed in this chapter according to autonomic computing technology.

4.1 Autonomic Computing

In the past decades, computer systems have been substantially developed. With the computing systems has become more sophisticated and diverse, the current system architectures face more and more problems about interacting between its components. For instance, some environments for operating systems need over 4,000 programmers to create about over 30 million lines of code. In order to deal with rapid growing complexity of systems, the concept of autonomic computing was proposed in 2001 by IBM. Autonomic computing is a system which could control the functioning of computer applications and manage by its own with high level policies from users. Also, this system would make optimization for its current status and adapt itself to the fluctuated conditions.[29]

In autonomic computing system, administrators do not need to control the system directly, they could generate several polices and rules to define how the system should work. In another words, these polices and rules lead systems for self management procedure. For this procedure, IBM company defines four functional parts[29]:

1. Self-Configuration
2. Self-Optimization
3. Self-Healing

4. Self-Protection

Self-Configuration:

Autonomic system should be capable of installing and setting up software in a automatic way. For this purpose, system will identify the changes on a configurable component. When a new component is added or registered, the system will integrate it smoothly and make sure it can be used, so that the other parts of the system accept its existence and cooperate with it. Just like when a new flash disk inserted to a computer, the computer should recognize and integrate it immediately.[30]

Self-Optimization:

An autonomic system will always look for an upgrade or modification, it will not stay with one status forever. Meanwhile, with the business level objectives changing and demands from customers, self optimization could help system itself perform more efficiently and punctually. Systems always look for improvement through searching, identifying and applying. Like fitness training, body will become in a better way.[29]

Self-Healing:

Nowadays, computer system functions are remarkable and impressive. They can do various things for human, even replace people to finish suitable tasks. But there is an interesting phenomena, computer systems look powerful but in fact they are weak. An operational character error, an additional comma, or bracket could cause the system to interrupt work or even lead to a breakdown. So the term of self-healing system means that autonomic systems should have the ability to discover, locate and fix bugs or potential failures in software and hardware during the runtime. Besides, with the sophisticated system architecture, it might take such a long time for system developers to identify the error. Self-healing function could know where the error happens, and could resolve it based on systems self-configuration or log files analysis, it help developers save time and efforts. The SMART(Self Managing And Resource Tuning) database from IBM company have a good example to show this function.[31] Database would detect fail occurrence automatically and repair it by installing some patches. Administrator is no need to get involved in the whole

process.

Self-Protection

With the development of computer system, malicious attacks, hostile cooperate, and potential virus from Internet have also increased gradually. Although administrator could use firewalls, anti-virus software and intrusion detection tools to deal with those cases, they still have to make the right decision when systems get attacked. Autonomic systems could protect themselves in two ways, reaction and pro-actions. Reactive protection would be that systems address the whole platform, find the errors and cope with it. Proactive protection allows systems detect problems from the early system logs or running exceptions, then systems would find a step to resolve it.

However, it is difficult to build an absolute autonomic system, because it requires developers with new technical skills and fresh innovation ideas. Therefore, achieving 100 percent intelligent behaviours are still a significant challenge for the future.

4.2 Utility Function and algorithms

In economic field, utility is famous for accurately measuring the desirability of different product types and services. Later, the concept of utility" has been used to help building multiple agents system in artificial intelligence field. Utility is a number which could show the level of one state, if the value of utility is higher, it represents that state is better. To object(human administrator and intelligent agents), utility function could be used to detect possible states of those themselves. Also, human administrator or intelligent agents should choose the practical state to maximize the utility. Therefore, in this context, utility could be understood as an object's preferences.[32]

Utility function is welcomed in autonomic field because it could help intelligent agent make decision in a rational way. A laptop example could tell about customers preferences. From external aspect, colour, price, and brand(manufacturer) could be included in multi-attribute utility function. Then from internal side, system version, CPU(central processing unit), hard disk space and memory are also added to utility function. Different laptops could have the advantages in different attributes

or purposes. So customer choose their preferred laptop with its maximum utility of each attribute. On the other hand, if an accurate result from customer's utility function is handed to an intelligent agent, it could filter those unmatched laptops on the behalf of customer. In autonomic computing context, a human administrator could list all values of possible system states through using a utility function. To an intelligent agent, it would obtain these values via an agreement or another relevant utility function. Then, the agent would make the best choice on the customer's behalf to maximize the utility. Besides, the systems in autonomic context are intelligent and dynamic, optimal actions are possible to change over time because of varying workloads or some other parts, so agent would optimize themselves in a repeated way.

In the past decade most researches in autonomic computing field has emphasized the use of utility function. In order to explain why utility function is used for representation and management of objectives, one previous work from Kephart which contains an defined framework will be introduced.[33] The term of policy from that research refers to any kind of formal behavioural guide, it is defined to be a significant role to play in autonomic computing field. That research introduces three types of policies to control systems, **Action**, **Goal** and **Utility Function**.

In order to explain these three types of policies in details, here is one figure which describes the concept:

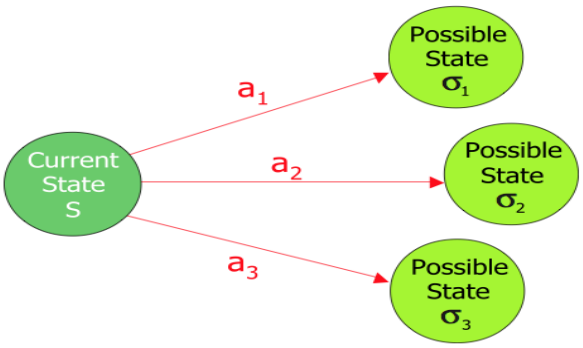

Figure 4.1: States and actions for automatic computing (Figure Owner: JO Kephart) [33]

In the Figure 4.1, **State:S** means the given or specific moment in a time of a system or part of the system. A policy given by administrator or system itself could trigger an action to the system, the action α would conduct systems to make a change or determination for turning into a new state σ in a direct or indirect way. So Figure

4.1 illustrates one case where an intelligent agent needs to make a choice among the three actions, each action is playing a role for conducting the current state **S** to a new different state.

To distinguish these three types of policies, they will be explained as following[34]:

Action Policy: This policy gives orders of actions to system, it tells system to take and make some changes based on these orders no matter what states that system owns now. Generally speaking, it is alike that IF THEN structure in logic field. *IF* represents the term of condition, and *THEN* is the action. Instead that policy will not tell system which actions should take for a given state, the administrator does. Administrators could determine which actions should be reached for achieving the goal state.

Goal Policy: The purpose of goal policy is to conduct one given state to a specified or desired state. Any action which could achieve the desired state is acceptable. Unlike administrator makes decision in action policy, the system will generate proper actions or behaviours by itself based on goal policy. This move could be seen as self-optimization of the system, then it will understand what it needs to do.

Utility Function: Utility function policy could be seen as the extension from goal policy. It assign a desirability with real value scalar to different states. Administrator will not make decisions or specify systems which action should be reached in advance, system itself choose proper state with high value of utility. Utility function provides a better, flexible and feasible solution than the other policies.

In a nutshell, utility function could be used as the main algorithm for an autonomic computing application. And this application might be capable of selecting channels and storing messages in the proper database for framework.

5 Messaging

In this smart multi-channel framework, there is one proposed message conversion engine which is used to convert incoming and outgoing messages. Incoming messages contain abstract information that will be extracted out and converted into concrete messages that only keep the useful information for users and machines. This process could help organizations improve their working efficiency and filter unnecessary messages.

In worldwide networks, data communication between machines and human has become a standard part with endeavours. But how to share data in different platforms or applications, it becomes challenging. For example, one international IT company that has numerous applications for sharing data, they are implemented in different platforms with various languages. So how to combine all these applications and make them work together or share information? The solution could be messaging. Messaging is a method which transfer data by using specific format, it is reliable, immediate and asynchronous. As can be seen in Figure 5.1, it shows

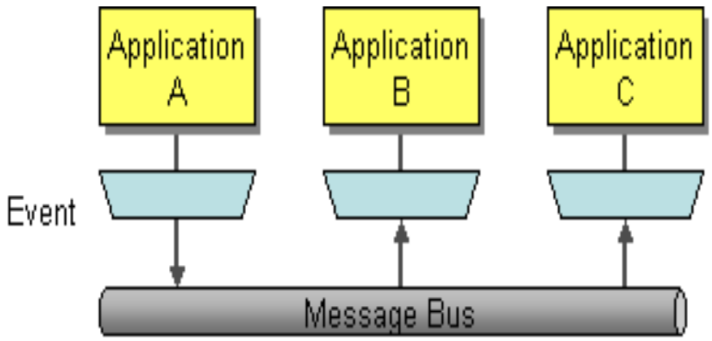

Figure 5.1: Messaging to different applications(Figure Owner: Claude Shannon) [2]

how each application receive message from one or several other resources, and one problem coming up with, that is how could the message or data could be delivered precisely. It is called message route. During this chapter, we will use the e-mail message routing analysis as an assumption due to its popularity in current business field, also will give some brief introduction about other message routings.

5.1 Message Routing

In general, message routing is a approach which messages are delivered from one channel to others. The *Content Based Router* is a simple variant of message router. It can examine the message content so as to choose the best channel for delivering. Alternatively, it helps the sender to reduce their working load.[35]

A message router can be set with fixed or flexible rules according to the real business cases. User can change logic rules of dynamic router. The following Figure 5.2 show working principle of one message router:

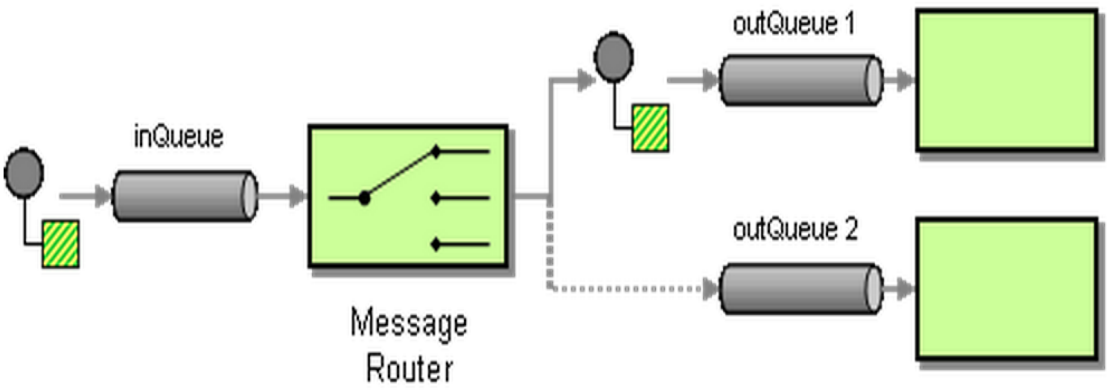

Figure 5.2: Message Router Figure Owner: G Hohpe
[35]

5.2 Message Conversion Engine

The framework should be able to defer and review email messages automatically according the business rules setting by administrator. It saves a large number of business rules acquired from business communication policies. These rules perform as post office in the framework, messages from clients or from other post office could be received here[36]. These business rules along with business policies are stored in message conversion engine. So this engine owns a plurality of actions could deal with each incoming and outgoing message. Also message conversion engine could provide these actions to other distribution engines, that helps other engines enforce a higher priority action. Message conversion engine is able to release, delete, forward, return and gate the messages. Besides it could be seen as the bridge between

different components in the framework, it is not only used for sending messages to external world, such as customers, it could also be used to connect each component and make them work. For example, when a concrete message is ready to be sent, router could send a broadcast to autonomic computing system to check which channel the receiver might prefer.

Message that are gated will be forwarded to the framework administrator who will check and review according to the business policies. Then gated messages would be deleted, forwarded or returned manually based on administrator decisions.

The gated messages in this framework means that messages only contain useful information extracted from sender with different channels. This gated message is the conversion from abstract to concrete. Information involved in gated message could be easily used and recognized by machine like mentioned before. Besides, in semantic web research and computational linguistics, information extraction is still a very popular challenge topic. But how could these gated message be generated, message conversion engine using LODifier approach would be introduced in this section.

5.2.1 LODifier for input text semantic analysis

The extracted information could be applied to deal with different kinds of tasks, for example, when receiving customer email, the useful information could be extracted and stored. When answering customer, we could apply the message template with these information then give customer preferred answer automatically. Also, extracted information could be used to retrieve text and generate ontology for semantic research.

LODifier is firstly introduced by a few researchers from Karlsruhe Institute of Technology[11]in the paper 'Generating Linked Data from Unstructured Text'(2012). Simply speaking, LODifier is an approach which extract entities from unstructured text and find the relationships between them, then convert these named entities into RDF representation. Some current research or strategies can extract semantic relation from text and transform them into RDF representation.[37] But the existing drawback is the selectivity of text input, it means that text should already be specified with typical information. LODifier is targeted at transforming whole text input into organized RDF representation. Hence, LODifier is relatively perfect to be proposed in this thesis due to its comprehensiveness and robust technique. LODi-

fier system applies methods based on named entity identification(NER), word sense disambiguation(WSD) and deep semantic analysis.[11] The following figure would illustrate how LODifier system architecture.

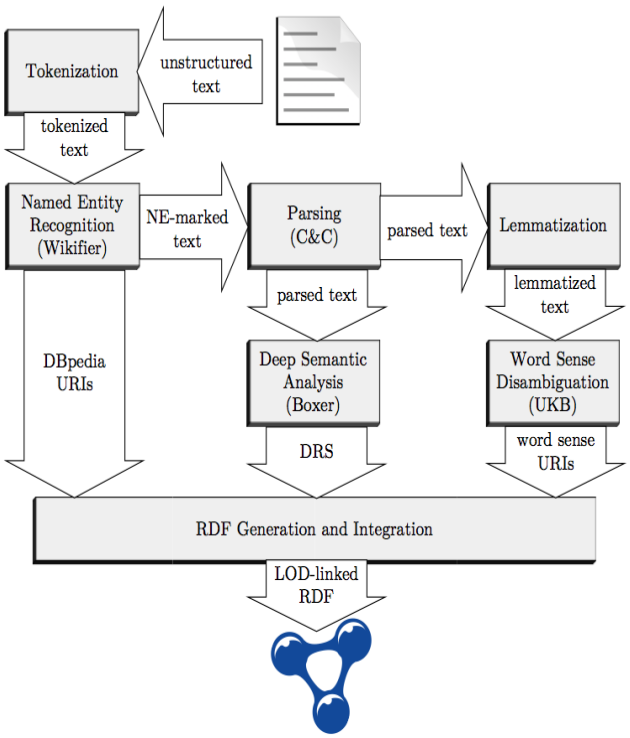

Figure 5.3: LODifier working flowchart(Figure Owner: Augenstein) [11]

Unstructured text firstly are tokenized and get text with tokens. After that, tokenized text as input are identified by *NER tool(Wikifier)*[1] to get entity marked text and *DBpedia* URIs[2]. Then the mentioned entities text are analysed by *C&C* which generate parsed text. Parsed text are detected by deep semantic analysis tool-kit *Boxer*[3] for obtaining discourse representation structure, at the same time, parsed text are lemmatized. After lemmatization, Word Sense Disambiguation tool *UKB*[4] will get word sense URIs for future processing. The final step is that RDF graph is generated

[1]The Wikifier identifies entities and concepts in text, disambiguates them and links them to Wikipedia.

[2]DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web.

[3]Boxer is developed by Bos, Curran, and Clark for generating semantic representations

[4]UKB is a collection of programs for performing graph-based Word Sense Disambiguation and lexical similarity using a pre-existing knowledge base.

by DRS output and enhance with DBpedia URIs and WordNet 3.0.

Entity Recognition: The first step is called entity recognition. Wikifier is a NER tool which can identity English text with corresponding Wikipedia pages. When Wikifier detects the input text, it first locates named entity, then entity is replaced by the link of matching Wikipedia page. Take the following sentence as an example:

James Gosling invented the Java.

Then after entities are identified by Wikifier, the generated output could be like this:

```
James Gosling(computer scientist) invented the Java(programming
language).
```

In order to remove uncertainty meaning from Wikipedia links, Wikifier employs one machine learning method. This method is to employ Wikipedia information links as training set. And because all the Wiki information are created and revised by verification users, so the training set is full of high flexibility and credibility as well as disambiguation choice.[11]

DBpedia URIs Assigned: The following step is that DBpedia URIs are created based on Wikifier outputs. Following this, the DBpedia URIs will be connected with Boxer classes. As noted by Augenstein(2012), every DBpedia page can match with a relative Wikipedia page.

Identify Relations: The third step is to decide relationships between entities from preceding Wikifier outputs. C&C parser and boxer play important roles during the process. The NE-marked text are firstly labelled with tags according to POS[5](part of speech) from the Penn Treebank[6] tag set.[11] Then parse trees are produced in a typical pattern called *Combinatorial Categorial Grammar*(CCG)[7]. CCG contains two kinds of categories: atomic and complex, both of them can be shown in XML tags. Besides, C&C parser has function to identity different

[5]A *part of speech* is also called a word class in grammar, it is a class of lexical items in linguistic. Ordinary linguistic classes comprise *noun* and *verb*.

[6]*Penn Treebank* is a project which shows syntactic and semantic information with a bank of linguistic trees. It analyse natural text for linguistic structure with annotations, it also uses POS tags to annotate inputs.

[7]CCG is an efficiently parseable, yet linguistically expressive grammar formalism

entity types such as: person(*per*), title(*ttl*),organization(*org*),quotation(*quo*), location(*loc*), first name(*fst*), surname(*sur*), URL(*url*), email(*ema*) and unknown name(*nam*). In addition, its success rate at recognition exactness and recalling is over 80%.

Boxer expands the parsed output of C&C and generates discourse representation structures for further process. DRS displays the meaning of text according to the relevant entities and relationship between them. Plus, DRS and RDF structure are quite alike, therefore it can used for transforming text into RDF as a suitable option.

RDF WordNet allocation: The forth step is to map Boxer relations onto LOD entities. DBpedia is not suggested to be used at this step due to its restriction at property definition. Then a better choice is proposed : WordNet. *WordNet* is an on-line large lexical database for English language, it has an abundant words over 15,000 in the latest version. Various types of vocabularies(nouns,verbs,adjectives and adverbs) are divided into sets of cognitive synonyms which are called synsets. Every synset express one distinct concept, but each synset can also be linked to another one by conceptual semantic relation. For example, the noun of *actor* is linked to another verb *act*. RDF WordNet is an extended version, so it could be seen as a LOD of WordNet. In addition, RDF WordNet gives the URI of word sense to its corresponding one. Lastly, words need to be disambiguated so that instances of words can be mapped onto URIs.

RDF Generation and Integration: The final step is to build an RDF graph. The URIs are defined first for the predicate of grammar and relation types. Then the URI are assigned to corresponding Boxer class and used to translate DRS.

6 Message Merge

In the following chapter, a concept named message merge will be introduced. Message merge in this thesis refers to the dynamic multichannel content delivery, it will be used in business part to save company's cost and reduce working labour. It is still a developing technique based on cloud computing, we assumed this technology could filter and pick up proper information then automatically send to users by their preferences and action history. If message merge will be finished in the soon future, it could help people drive data more efficiently and save a great amount of time. Also through this technique, customers will just receive news they have interests, information which users have no interests will be denied.

6.1 Message merge Model

With rapid information development in modern society, information has become significant and necessary part in people's life. However, sometimes people are annoyed by different types of information from various types of channels. To solve this problem, people are considering several solutions. For example to get rid of trash mails, now you could set the tags or frequent words from same type email. But here is the problem, the trash mail could change the content of email or its address to avoid be rejected.

Here is another example why the business aspect is relevant to this technique. An on-line shopping website has reopened after update, at the same time, they have many kinds of items with good and reasonable price. In order to attract more customers, website decide to send newsletters each day to their members.[38] The website could record all the members actions. Customers are not into receiving newsletters everyday, they just hope to receive some useful advertisement or something they are interested. Hence, what the website should do if they don't want to annoy their customers and lose them?

Message merge could resolve this issue mentioned above, its main function is to integrate message which contains useful messages according to customers shopping habits and personal data. First, system will identify what products are new

arrival,then it will look for and filter customers who have interests according to their profiles and shopping history. In the next step, system will filter newsletter according to customers preferences and history. Then system is going to format the message and get it sent to customers by decent channel.

6.2 Information Filtering

Generally speaking, information filtering is a system uses machine method automatically to remove or add information according to users own interests or behaviours. It automatically finish works like abstract, classification and summary on machine. It plays as a mediator role between resources and users. Information filtering normally deal with unstructured or semi-structured data, the most common example is email message.[39]

Figure 6.1 below shows an architecture how system will work. The primary parts of consisting the whole system are: information filtering system, different types of database and recommendation engine. System will receive incoming information or message from multiple channel first, usually the preferred channels are through phone text or email. During the filtering process, system will match items by semantic contents and sub string index according to user's interests and action history in database. Then system will do the syntactical analysis to form the decent response with the help of human and recommendation engine.

In the following, here are some reasons why we choose IF system as foundation technique.

1. Information Filtering commonly deal with textual contents.

2. Recommendation engine is one subclass from IF system.

3. IF system has great relating to meta-data.

We propose this technique could use semantic language in the future, when it is mature and applied in the Web context, it could filter incoming message so as to get the key contents from users.

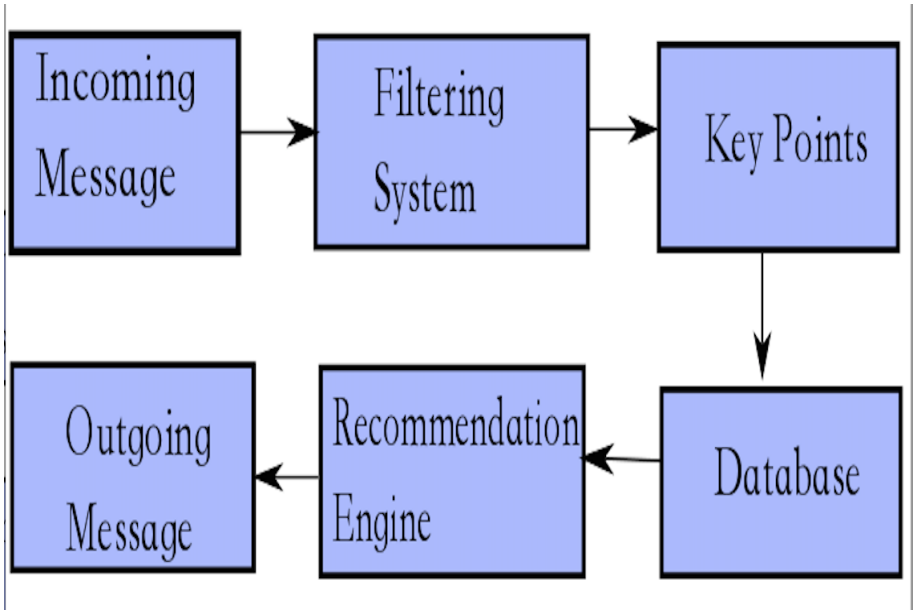

Figure 6.1: Flowchart of Message Merge Model

6.3 Recommendation Engine

In business models, recommendation engine is a system or application usually to be used in machine to provide items which customers do not notice. It is also based on customers preference profiles and purchase history to work. Therefore, a recommendation system could be seen as a subclass from information filtering system.

Since the concept was proposed in the mid 1990s, this technique has been common and popular in the past decades.[40] Here are two good samples to illustrate this technology:

1. Offering suggestions to on-line customers about what they might like, based on their history of searching and purchase. For example,if you are surfing Amazon, it will recommend plus products based on other customers searching history which probably matching with yours.

2. Offering new articles to on-line readers according to their interests in profiles. For instance, there is one website[1] from China, user could purchase and read books after registration. If users fill the reading habits form, each time when they log in, the website would recommend the matched books to them.

[1]The website is: www.dangdang.com

Although recommendation system contains various technologies, it still could be classified into two groups based on purposes.

Content-based information filtering

It is mainly used to examine attribute of recommended items. Take a look at the Chinese book website again, if the user read a number of books relating about romance, then in the database there will be a classification with tags *romance* and stored in user profile.

Collaborative information filtering

It offers suggestions according to similarity between users and items. Applications like email, calendar and social booking marking belong to this field.[30]

If this technology could be used in our case, it probably will increase efficiency of working labor and decrease the chance of merging trash mails.

6.3.1 Content-based information Filtering

In content-based filtering system, each item needs a profile which contains characteristics of the item, in other words, it tries to recommend items to users who might have similar interests. In our message merging case, we assume this technology is possible to be used for filtering incoming message from customers, then it could find the similarity from customer profile in database. For this reason, it probably is the first primary preparing step for recommending items.

6.3.2 Collaborative Filtering

Traditional collaborative filtering is to gather and analyse information from users behaviours instead of using properties from items, it is to determine similarity when comparing with other users.[41] For example, collaborative filtering could be explained like this: If preferences of user group A is similar with a single user X, system will determine to recommend items for user X in the situation. This process is collaborative filtering. Also, the common algorithm which calculating user similarity and item similarity is called k-nearest neighbourhood in collaborative filtering.[42] In message merge model, this technique will probably be beneficial for recognizing group customers shopping pattern, thereby decreasing time on searching customers information and habits.

Recently, an extended concept which based on k-nearest neighbourhood was proposed by two researchers from Tilburg University, they are doing some experiments to investigate how they could use nearest-neighbour filtering algorithm to comprise tags and other meta-data. This algorithm might take the place of conventional usage based similarity metrics because of repeated tags. Also they tried and examined to use meta-data content to recommend interesting things.

6.3.3 Knowledge Based Recommendation

The third type of recommendation system firstly set up a knowledge foundation with a model about both users and items, then system make an recommendation to user through reasoning if users and items are matched or fitting each requirements.[43] Knowledge based recommendation is prior preferred in marketing comparing to other recommender systems, this might be due to its several great features.[44]

1. Simplicity: large amount of data is not necessarily required in knowledge based type.
2. Quickness: new user with personal detailed profile could receive recommendation at once.
3. Humanity: system know what and why user need this item.

Knowledge based recommender system has already been used in some on-line stores such as Amazon..., here is an simple case from a book named Proceedings of the Workshop on Recommendation and Personalization in E Commerce could illustrate:

1. Gathering information details about items
2. extract characteristics
3. Set data with label like the second step
4. Set up users profile in terms of semantic characteristics
5. match the product with user visited

6.4 Business Case

In modern society, on-line shopping and cooperating has become the main trend in business sector. To buyers, it has brought plenty of benefits, such as convenience. For instance, there is no need for customers to go to a distance store or in a bad weather. To company, on-line shopping stimulate the activities of business. However there existed one problem in the shopping pattern, lack of communication and comprehension to their customers. For example, some proportion of customers are just only one-time shopper, it was more beneficial if company could deliver some proper advertisements in a decent time.

In our case, companies are willing to provide a consistent and dynamic communication through various channels to their customers. This communication is based on customers expense calendar and profile, it will not bring too much negative emotional effects to customers, such as trash mail. Here is an simple example to illustrate our case, a shopper wants to buy a new TV from an on-line shop, as he hopes he could receive some information on TV discounts through electronic advertisement. Meanwhile the company has a new TV product launch. As the company can merge these messages, to achieve increased efficiency and save time, there is no need to send a message to each customer individually. But how company know and communicate with customers, in our assumption, we suggest it could be fulfilled by several channels.

1. Social Communication Websites followers
2. Surf on-line shopping store
3. SMS notifications
4. newsletters via email or postal mail
5. Phone Call

As introduced by message conversion engine in Section 5.2, it seems possible that we could apply LODifier approach for practical commercial goal. Therefore, in this case, the do-ability of merging different communication systems with LODifier are assumed and discussed in the following three popular cases:

- Email and On-line shopping form: In general, a email system is the most common and popular way for internal an external communication in companies. It

could be simply seen as composed of following parts: post agents, servers and applications. A post agent is a distributed mechanism which receive message from client and then transfer it to other post agents with respect to some specified receivers. Besides, a traditional post agent is a store and forward model where message is saved only temporarily before it is dealt with next agent or administrator. In addition, the essential operation rules of conventional email system is an unabated delivery approach. It means to deliver an email message from sender to receiver as quickly as possible without any interference. Therefore, with an increase of demands on email for different types of communications, it has become desirable to have personalized email system. Also company hope to define and perform communication or business rules in their email system for handling with message contents. Altogether, from the previous Section 2.7, it seems possible that LODifier could be implemented into post agents.

To begin with, company could create different email accounts for coping with separate things. Each account could be seen working in their own thread and have personalized LODifier model. For instance, accounts can roughly grouped into: *information consult customer service* and *consumer complaint*. Information consult account is responsible for that customer find out more specific details about product they have interests. And these detailed information could include size, colour , shape, warranty, discount and so on. Thus agents could be applied with LODifier and predefine vocabulary, then agents receive incoming email from information consultant, where it might be able to extract the key information and pass them to the prepared email template or the administrator.

Customer service deal with consumer information collection and storage. Besides, they could send preferred advertisement content of recommendation products according to analysis from information consult emails records and consumer info. Later the template for customer service will regularly send these preferred promotion ads.

Consumer complaint account is charge of receiving customers' unsatisfied affair, it could extract the key points from the email content then forward to support staff. This might help company improve their working efficiency.

- SMS: SMS(Short Message Service) is one of most popular communication ser-

vice for sending small text in a global range to other enabled devices. Customer send the message to designation number, then these messages are transmitted as input text to LODifier for analysis. In addition, these message might be also filtered by content based approach, it aims at not sending spam message. In the end, the response message with preferred information in a new template is delivered to mobile user.

- Phone Call: Telephone network is the most welcomed communication way, because it is easy and convenient. But in our case, we have not discovered an effective way to employ LODifier or other automatic input analysis technique combing with phone network. So the information which customer want to send are going to be recorded by manual.

7 Apache Stanbol

Apache Stanbol is an open source HTTP software with semantic features for content management. It can also be used for many other aspects such as delivering email in terms of extracted entities. The most important point of Apache Stanbol is that it let users develop their own content management system with their core. It uses Java as its programming language and RESTful as its interface.

Apache Stanbol has four primary features:

Content Enhancement This service enables users to add semantic information into other information pieces, it is processed by enhancement engines. However by now, the enhancement engines cannot be modified by user to achieve a higher level.

Reasoning This service is combined with content enhancement service to retrieve additional semantic information.

Knowledge Models This service can be used to define and manipulate semantic data models. Furthermore, Apache Stanbol has one component named *Ontology Manager* can be used to manage ontologies and ontology networks.

Persistence This service is generally used to store semantic information which could be searched. Its component *Apache Stanbol Contenthub* is able to store text based documents and customize semantic searching engine.

Apache Stanbol could be a very important component for multiple channel communication framework. It has been installed in laptop and worked, as mentioned earlier enhancement engine cannot be inserted with personalized engine, so we cannot manipulate it and achieve for our goal by now. Therefore in the future, it is strongly recommended to have further research in Apache Stanbol.

8 Privacy and Security

This proposed framework is mainly built upon semantic web, but semantic web in our current society is still not a mature technology. People still have some arguments about it such as its privacy and security. In this chapter, there is a list that shows personal opinions about negatives aspects in terms of semantic issue.

Anonymous Aspect The Web is a place where people do not want to leak their personal information. However semantic web will store user information such as user identity, hobbies, habits and home address, etc. With increased amount of available information as well as *web is a huge database* idea, these information could become transparent and discoverable. For example signing up for an account on some websites, while filling in account information to register on websites and to sign in.

Privacy Invasion This issue could be considered from anonymous problem on semantic web. The advantage of semantic web is that it is capable of storing vast amounts of information, and the drawback is the same. It might be an easy access for someone if they want to misuse these information for monetary goal.

This situation has already happened in people's life, for example web advertisement. Traditional web ads are only based on websites content in order to attract user interests, but now it has been gradually replaced by 'smart' advertisement which targets at user preferences. 'Smart' Advertisement is named targeted advertising. The targeted advertising technique was used to track data for security reasons, but now it is developed to reach certain customers. It generally could be divided into two types:*demographic based* advertising and *content based* advertising.[45] Demographic based advertising aims at reaching customers in terms of shared characteristics(age, gender). And content based advertising is created to reach customers with individual interests. Both types of advertising are designed according to users browsing history or information during registration.

In a nutshell, the growing privacy invasion is a complicated issue because there are many groups get involved. It needs more on-line parties to partic-

ipate in developing web security.

Incompatible Vocabulary In semantic web field, vocabularies are used to describe the concepts and relationship in a domain. From a general view, it could be seen that vocabularies constitute ontologies. Developers classify ontologies by using these vocabularies(terms), therefore the definition of one ontology could vary from simple to difficult.

As mentioned in Section 2.6, ontology matching is used to solve the issue that people use different vocabularies(terms) to define the same concept. However, vocabulary incompatibility refers to people using the same word containing different meanings for individual purposes. For instance, one developer use the term *bank* to describe the financial establishment in finance field, whereas the other developer use the same term(bank) for hydraulic purpose. Another type of example is to use the same vocabulary(e.g., bank) for query, computer cannot always interpret user's original meaning right.

Although there are already a lot researches investigating about this topic(e.g., ontology alignment), there is still a need for a robust and open set of vocabulary in semantic field. Besides, technology become more automate and intelligent, we need to ask ourselves, are people really happy to communicate with completely automatic machines?

Description Logic Drawbacks In knowledge Representation, description logic could be viewed as a dominant formalism. But some fundamental properties of description logic might have negative affect for semantic and its future.[46]

There are some uncertainties about logic description, the main drawback, however is to deal with prior possibility and confused knowledge vocabulary. For example, when consumer sends a message through one channel, this message might include some important information which need to be solved immediately, such as *I cannot wait until tomorrow*. But machine cannot understand the deep meaning from literal text, it might interpret into *tomorrow is not available for this user*. Then it will not label this message with a higher priority tag, therefore it might cause some serious faults.

Generally, there are two ways of solving this type of issue, one is manual(technician expert) and the other is automatic(machine learning). However, the requirements for technician expert is to have profound professional knowledge, and

the algorithm of machine learning will be a huge work.

9 Conclusion

The previous research from Nagy only presents a brief conceptual framework, not every component has been discussed in depth. Therefore, this thesis is set out to explore the multi channel communication system in a particular way and provided ideal solutions for each framework component. The author also hopes this thesis will offer enough help in developing a real and successful multi channel communication system in business field. This thesis firstly provides underlying information for reader to have an overall understanding of semantic web technique. With increasing and various demands in electronic business field, more sophisticated techniques are required to support company development as well as customer satisfaction. Semantic technology could bring users a smarter and more flexible experience, some key techniques such as personalized ontology, linked open data model make people redefine what Web is and certainly have some sustainable affect for future development. Moreover, this thesis presents utility function and main principle of autonomic computing as the best solution for selecting communication channel, because channel ontology(agent) could make a precise and rational decision by implementing utility function. In addition, the thesis has found that the LODifier approach could be utilized to extract useful information from incoming and outgoing messages. It is highly recommended because it is a mature technique and could convert extracted entities into RDF formalism. Furthermore, message merge model is suggested by proposing information filtering and recommendation engine techniques, which three types of filtering methods might effectively combine individual messages according to consumer shopping habits.

However, this thesis is not specifically designed to integrate each component, these results might not be applicable to constitute a mature multi channel communication. In addition, due to my own limited knowledge scope, the thesis still lacks of information on the following aspects:

Ontology Matching

As discussed in Section 2.6, in practise when company try to manage ontologies for data integration, which application would be the option for a large volume of data? Since this thesis does not focus on the big data part, further

work might explore big data area for a proper answer.

Ontology Personalization

Information about developing an ontology based on user profile is still blank in this thesis, the future research might need to find which technology could be used in this part. Besides, when developing ontology according to the real business cases, more properties should be added to the specified ontologies in knowledge base.

Component Integration

Although there are several techniques introduced in this thesis, unfortunately I cannot find a good solution to merge them. Therefore, this issue might be explored as a subtopic for the future work.

There is still one thing which needs to be noticed, if we assume this multiple channels communication assumption is not working or removed in the future. Every technology of component is still worthy to be explored. Nevertheless, this proposed semantic multi-channel communication has already offered some positive ideas in practice and academic field.

ACKNOWLEDGMENTS

I would like to express my gratitude to my two supervisors Professor Vagan Terziyan and Michael Cochez from University of Jyväskylä. Thanks for the useful comments, remarks and engagement through the learning process of this master thesis. Furthermore, I would like to thank Sami Helin from Steeri Oy who provided those useful comments as a supervisor in business area as well. Also, I would like to thanks the University of Jyväskylä and technical staff of Steeri Oy for their support. Last but not th least, I would also like to thank the reviewer for helpful comments and improvement suggestions.

10 References

[1] M. Nagy, "On the problem of multi-channel communication," *ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer*, p. 128, 2012.

[2] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[3] T. Berners-Lee and R. Cailliau, "Worldwideweb: Proposal for a hypertext project," *Retrieved on February*, vol. 26, p. 2008, 1990.

[4] T. O'reilly, "What is web 2.0," 2005.

[5] T. Berners-Lee, J. Hendler, O. Lassila, *et al.*, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.

[6] "Unicode official." `http://www.unicode.org/standard/WhatIsUnicode.html`.

[7] T. Berners-Lee, R. Fielding, and L. Masinter, "Rfc 3986: Uniform resource identifier (uri): Generic syntax," *The Internet Society*, 2005.

[8] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff, "Semantic annotation, indexing, and retrieval," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 2, no. 1, pp. 49–79, 2004.

[9] T. Berners-Lee, W. Hall, J. A. Hendler, K. O'Hara, N. Shadbolt, and D. J. Weitzner, "A framework for web science," *Foundations and trends in Web Science*, vol. 1, no. 1, pp. 1–130, 2006.

[10] T. Bray, D. Hollander, and A. Layman, "Namespaces in xml," *World Wide Web Consortium Recommendation REC-xml-names-19990114. http://www. w3. org/TR/1999/REC-xml-names-19990114*, 1999.

[11] I. Augenstein, S. Padó, and S. Rudolph, "Lodifier: Generating linked data from unstructured text," in *The Semantic Web: Research and Applications*, pp. 210–224, Springer, 2012.

[12] G. Klyne and J. J. Carroll, "Resource description framework (rdf): Concepts and abstract syntax," *W3C Recommendation*, 2005.

[13] E. Prud'Hommeaux, A. Seaborne, *et al.*, "Sparql query language for rdf," *W3C recommendation*, vol. 15, 2008.

[14] "Owl sample." `http://www.linkeddatatools.com/introducing-rdfs-owl`. Accessed: 2014-09-19.

[15] E. Oren, K. Möller, S. Scerri, S. Handschuh, and M. Sintek, "What are semantic annotations," *Relatório técnico. DERI Galway*, 2006.

[16] N. F. Noy, R. W. Fergerson, and M. A. Musen, "The knowledge model of protege-2000: Combining interoperability and flexibility," in *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*, pp. 17–32, Springer, 2000.

[17] N. Guarino, *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, vol. 46. IOS press, 1998.

[18] D. L. McGuinness, F. Van Harmelen, *et al.*, "Owl web ontology language overview," *W3C recommendation*, vol. 10, no. 2004-03, p. 10, 2004.

[19] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler, "Owl 2: The next step for owl," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no. 4, pp. 309–322, 2008.

[20] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, "Owl 2 web ontology language primer," *W3C recommendation*, vol. 27, no. 1, p. 123, 2009.

[21] C. Golbreich, E. K. Wallace, and P. Patel-Schneider, "Owl 2 web ontology language: New features and rationale," *W3C working draft, W3C (June 2009) http://www. w3. org/TR/2009/WD-owl2-new-features-20090611*, 2009.

[22] F. Baader, S. Brandt, and C. Lutz, "Pushing the el envelope," in *IJCAI*, vol. 5, pp. 364–369, 2005.

[23] B. Motik, B. C. Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz, "Owl 2 web ontology language: Profiles," *W3C recommendation*, vol. 27, p. 61, 2009.

[24] X. Tao, Y. Li, and N. Zhong, "A personalized ontology model for web information gathering," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 23, no. 4, pp. 496–511, 2011.

[25] A. Halevy, "Why your data won't mix," *Queue*, vol. 3, no. 8, pp. 50–58, 2005.

[26] P. Shvaiko and J. Euzenat, "Tutorial on schema and ontology matching," *PowerPoint Presentation ESWC*, pp. 05–29, 2005.

[27] P. Shvaiko and J. Euzenat, "Ontology matching: state of the art and future challenges," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 1, pp. 158–176, 2013.

[28] L. Yu, "Linked open data," in *A Developer's Guide to the Semantic Web*, pp. 409–466, Springer, 2011.

[29] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[30] T. Bogers and A. Van den Bosch, "Collaborative and content-based filtering for item recommendation on social bookmarking websites," *Submitted to CIKM*, vol. 9, 2009.

[31] J. H. M. Nogueira and J. C. Júnior, "Autonomic forensics a new frontier to computer crime investigation management," *The International Journal of Forensic Computer Science*, vol. 1, pp. 29–41, 2009.

[32] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[33] J. O. Kephart and W. E. Walsh, "An artificial intelligence perspective on autonomic computing policies," in *Policies for Distributed Systems and Networks, 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on*, pp. 3–12, IEEE, 2004.

[34] J. O. Kephart and R. Das, "Achieving self-management via utility functions," *Internet Computing, IEEE*, vol. 11, no. 1, pp. 40–48, 2007.

[35] G. Hohpe and B. Woolf, *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.

[36] F. J. Geiger, S. T. Tandon, and W. K. Wood, "Automated post office based rule analysis of e-mail messages and other data objects for controlled distribution in network environments," June 6 2000. US Patent 6,073,142.

[37] Y. Li, V. Chu, S. Blohm, H. Zhu, and H. Ho, "Facilitating pattern discovery for relation extraction with semantic-signature-based clustering," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 1415–1424, ACM, 2011.

[38] J. C. Michael Cochez, Sami Helin, "Cloud communication service," 2013.

[39] N. J. Belkin and W. B. Croft, "Information filtering and information retrieval: two sides of the same coin?," *Communications of the ACM*, vol. 35, no. 12, pp. 29–38, 1992.

[40] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*, pp. 1–35, Springer, 2011.

[41] A. Rajaraman and J. D. Ullman, *Mining of massive datasets*. Cambridge University Press, 2011.

[42] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system-a case study," tech. rep., DTIC Document, 2000.

[43] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[44] D. Dell'Aglio, I. Celino, and D. Cerizza, "Anatomy of a semantic web-enabled knowledge-based recommender system," in *Proceedings of the 4th international workshop Semantic Matchmaking and Resource Retrieval in the Semantic Web, at the 9th International Semantic Web Conference*, 2010.

[45] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas, "Adnostic: Privacy preserving targeted advertising.," in *NDSS*, 2010.

[46] A. Sheth, C. Ramakrishnan, and C. Thomas, "Semantics for the semantic web: The implicit, the formal and the powerful," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 1, no. 1, pp. 1–18, 2005.