Jiawen Chen

Smart Semantic Multi-channel Communication

A Master's Thesis
in Information Technology
February 9, 2015

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, quite many different media channels are used popoluarly. For instance, phone call, text message, email, website, and various mobile applications. Web technique plays a significant role in today's society, no matter where we are, what we are doing, we cannot live without it. The Web is quite functional to us, we surf on-line everyday for education, entertainment, or look for useful information. However the current web still cannot fulfil our demands. Now, consider a customer trying to buy a product first through visiting a website first in order to find out more information. Later this customer calls for a quote by phone, and finally this customer decides to buy this production by sending an email to the seller. This whole process is called multiple channel communications. 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 main targets of this thesis are firstly to seek how semantic technologies could be implemented for constituting a multi-channel communication system. Then specifically, this thesis attempts to review and introduce these semantic technologies from a deep and understandable perspective to readers. Literature review is used as the research approach for this thesis.

In conclusion, this thesis provides the solutions to each component of multi-channel communication system, however there are still some certain issues which need to be discussed with more details in the future. The author hopes this thesis could make a small contribution in semantic technologies fields for other research or in the practical world.
Suomenkielinen tiivistelmä: Abtract in Finnish
Keywords: Channel Communication, Semantic Web, Text Analysis, Customer Relation
Avainsanat: Kanavan tiedonvälitys, semanttinen web, tekstin analysointi, asiakassuhteet

Copyright © 2015 Jiawen Chen
All rights reserved.

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, I would like to state an example which briefly presents the main idea what Multi-channel communications is. A student is going to apply for a master program in the University of Jyväskylä(JYU). The student finds out application requirements and contact information through the university homepage. These contact information include a few email addresses, the university staff working phone number, and an on-line Q&A board where visitors can leave question. After reading the application requirements, the student is still confused about some prerequisites. Therefore he tries to look for help by leaving a few questions and his email address on the Q&A board. One week later, the admission office replies the student by sending an email. Nevertheless, the student does not check his email account on an regular basis, he directly makes a call to the university. In addition, before this student tried to apply for this program, he did a bit search through the Web to check if there are some other options. While he input his bachelor education background and interests orientation, the Web seemed to understand his intention and recommended this master program of JYU. This small example describes the concept of multi-channel communication from a everyday life. It also introduces a smart Web at the present day, the semantic Web.

As it is possible to use many different channels for communication, such as mobile messages, Internet advertisements etc., it can happen that the same information is sent through several channels or non is sent duet to confusion. The main goal of semantic multi-channel communication is to improve the efficiency of communication and to reduce unnecessary messages; Which could be advantages for future business purposes, but what components should be used to create the system?

The conception of multi-channel communication system was simply proposed in a previous research by Michael Nagy(2012).[1] In that research paper, he merely introduced each working principle but a brief introduction about the framework. Besides, he did not explain explicitly what technologies should be used for the framework components. Therefore, this thesis intends to explore and analyse what technologies could be used in multi-channel communication framework. Since there is

no similar framework or conception yet in semantic technology field, all the proposed technologies in this thesis are the results after finding and reading relevant research paper.

The overall structure of this thesis takes the form of nine chapters, including the introductory chapter. The thesis is divided into nine chapters. Following the introduction, chapter two begins by defining the concepts of communication and the semantic Web, as well as and laying out some relevant knowledge of core semantic Web technology. Moreover, this chapter introduces *ontology* and its relating approaches as it is a crucial part in semantic technology and essential for further understanding. The third chapter presents the main research findings and multi-channel communication framework proposal. Besides, the specified ontology models are discussed in this chapter. Chapter four analyses a smart channel selection mechanism and probability for its application to multi-channel communication framework. In chapter five, an unstructured text analysis approach, which is used for converting messages, 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 the Web and email communication. Several other commercial communication cases are also mentioned in this chapter. The next chapter seven gives a brief introduction about the knowledge management software Apache Stanbol. In the eighth chapter, some disadvantages of the semantic technology are presented. Finally, the conclusion gives a brief summary and critique of the findings, some ideas for possible future work are also shown in the final chapter.

2 Fundamental Knowledge

The current study in artificial intelligence field found that the semantic Web has the probability to become the new generation of the Web in the future. In addition, some key components of the semantic Web technology have been implemented in practise or have affected other relevant researches. This chapter is going to introduce the Web and communication history, the importance of the semantic Web is going to be discussed in Section 2.3. In order to have a better understanding on the multi-channel communication conception, a profound analysis concerning ontology is shown in section 2.5 and section 2.6. Finally information about the semantic Web and linked open data are presented as supplement.

2.1 Communication

The definition of *communication* is quite broad. Generally, this word is understood to mean exchanging information. In order to exchange these information not just verbally, media such as computers, radios, mobile devices etc., can be used. Therefore, humans need to communicate with these electrical devices, in a sense, to extract the information they need. It is an indispensable part of our society, and also plays a significant role in many fields.

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

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 the decoding process, the signal is delivered through the 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, the channel could be a cable, in the case of a speech, the channel is the air. Receiver could be con-

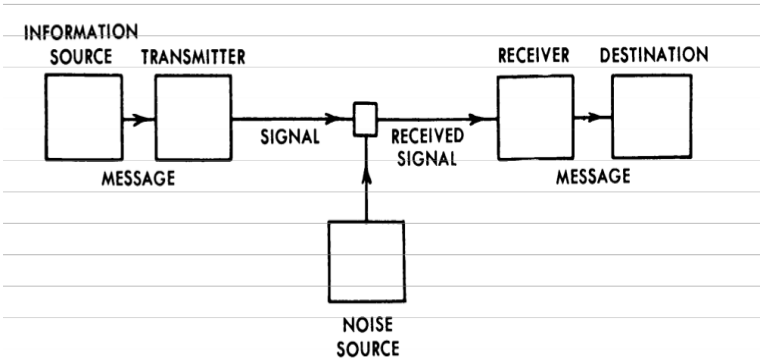

Figure 2.1: Communication System Model(Figure Owner: Claude Shannon) [2]

sidered as a reversed transmitter, it receives the signal then encodes them back to messages which are easily understood by machines or people. These messages will be sent to their destination afterwards. However in the process of communication, the disturbances can always happen in external or internal aspects. For example, while holding a presentation, the disturbance could be sounds from the audience, in a phone call, a wired or cable might be damaged causing errors during transmitting. All those disturbances are called noise.

Additionally, author Claude Shannon the book "Mathematical Theory of Communication"(1948) describes that, although the noise disturbs communication channel, it is still practicable to transmit separate signal or data in a nearly error-free level if the signal transmitting speed less than the signal channel capacity context. Corresponding to the topic of communication, previous studies from Shannon and Weaver's 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; The right interpretation of the incoming message by the receiver.

3. *Effectiveness:*
   Effectiveness level depends on the interpretation and is therefore interlinked with the semantic issue.

2.2 Internet VS Web

The Internet is an enormous and worldwide system of networks. It is a networking infrastructure which offers the possibility to connect to millions of computer worldwide. It forms a network which enable one or several computers to 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, who wanted to exchange data and results of experiments. However, they found that the exchange of those ideas and scientific results was difficult to achieve. Berners Lee understood the need for an improved system, and he realized the potential needs for many computers to connect. 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 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:

In a nutshell, the Web is a system which uses interlinked hypertext documents, which can 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 to improve the abilities 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

```
<!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

with users, and more information sharing. 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 the meaning in languages 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 might be "twisted" comparing to what a person actually meant. For example, when a person says *I love you* to different people, it might contain various meanings. It all depends on 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

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

to let a machine communicate with human and understand what people need, it became a challenge. Therefore, it brings about a new innovatory conception called the 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 meaningful to computers. The Semantic Web is expected to interpret the exact meaning from the users and could be used by machines afterwards. Five years ago, if a user 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 the Semantic Web converts the Web from a simple keywords searching to a meaningful content query. The main purpose of the Semantic Web is to equip the Web with "human" functionalities, such as identification, communication, self management, decision making and thinking. For example, if an user inputs "my mouse is dead, i need a new one", the Semantic Web can recognize the explicit meaning by the user, "my computer device is broken." Until now, the Semantic Web is still a unfinished vision, it aims to allow data to be shared and reused in different platforms. In order to have a better understanding on what the Semantic Web can 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 can only show John some links including the keywords that he typed, however, his intention cannot be understood in a right way. Therefore, as mentioned in the introduction, the Semantic Web is an extension of the Web which enables users to share contents. Moreover, the Semantic Web offers a well defined data structure, it makes computers and users able to work in cooperation.

Now take a look at the sample again, when John inputs his favourite player's name, the Semantic Web could give him back some relevant news concerning this player instead of just hyper links. Also, the Semantic Web can list the basic information(e.g. team, home town, career records.) about the player. In a nutshell, the Semantic Web is similar to a global intelligent database, tt offers an idea that anything could be linked with, known as everything as a service(EAAS). In the near future, with more development and researches about the Semantic Web, it will lead to significant functions 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 a Semantic Web

$$web \tag{2.1}$$

stack.

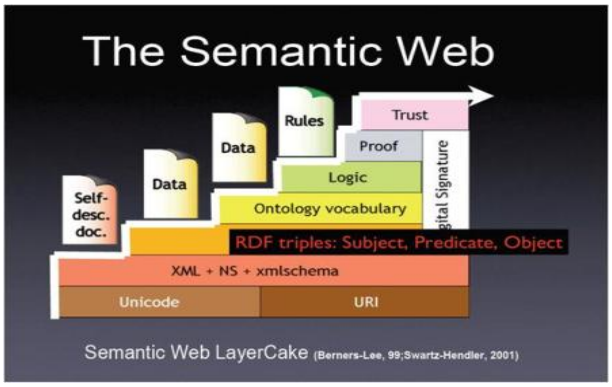

Figure 2.3: The Semantic Web Layer Cake(Figure Owner: Tim Berners-Lee) [5]

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;

  Unicode is an international standard for representing characters sets. All languages on the Web become accessible by use of the Unicode standard.[6] When the users surf on the internet, they have been using Unicode already. *URI*(Uniform Resource Identifier) is a string of characters, which helps the users point to a name of any resource type on the web, such as text, video, sound clip, or image.[7] Both components, Unicode and URI, together build the fundamental Semantic Web structure.

- Layer 2: XML+NS+XML Schema;

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

XML document structure. There are more detailed introductions on XML in subsection 2.3.3

- Layer 3: RDF+RDF Schema;

  This layer offers the 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 meanings between various concepts. Ontology vocabulary is a formal and explicit specification of a shared conceptualization.

- Layer 5: Logic;

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

- Layer 6 & 7:Trust and Proof;

  As mentioned in the logic layer part, the proof layer is responsible for executing the logics and then evaluating them with the trust layer which determines which the application should be trusted and 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 a 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. Since the diversity of languages and globalization, there are thousands of characters needed to be encoded on the Web. Another

issue is that 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 the computer to decrease the data quality or even ruin the data process.[6]

As introduced earlier in section 2.3, Unicode is also a system representing character sets. At first it was invented to merge all the 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 problems today.

2.3.2 Uniform Resource Identifier

As shortly mentioned in section 2.3, uniform resource identifier(URI) is a comprised sequence of strings. It identifies a resource on the Web by providing a simple and extensible method, and this resource can be identified by a location or a 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) can identify where an available resource is and retrieve it by describing a primary access mechanism (network location). A URL also defines how the resources can be obtained by providing their prefix names, the most common types are: *http://* and *ftp://*. It can 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 the URN scheme to identify the 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 the network locations of resources, while URN states a resource identity.

2.3.3 XML

As presented 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 about text, pictures etc. In fact, an **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>. The end tag, however, has the same structure but one slash in between those two brackets such as</body>. In the middle of these two tags, the contents are included. Besides, other elements can 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 it shows in the Figure 2.4, the person tag could be seen as the parent tag in XML document. Between the parent tag: sex, first and last names are the child elements. Nonetheless, sometimes there are too many child elements and, in this case, attributes with a name-value pair can replace the child elements, such as <person sex ="female">. Both child elements and attributes would provide the same information. Moreover, if there is nothing in the element, an empty element can be written as
. Lastly, in an HTML document, end tags are not necessary, but in an XML document, there must be one end tag, which is why XML is seen as well structured.

However, sometimes the elements in XML document will have some name conflict problems. For example, one element named *person* is defined twice in two different documents. In addition, when these two documents are combined, an application cannot deal with this situation. Therefore, XML Namespaces were invented

to provide some unique element and attribute names used in an XML document.[10] An XML Namespace is composed of two parts: a *Namespace prefix* and a *Namespace URI*. Take the example from Figure 2.4, to make a person tag unique, we can add an 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, the users and computers do not need to worry about the named conflict issue when merging the documents.

Another essential component in the 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 an XML documents. Besides, it performs the rules made by a programmer to define each part of XML documents. XML Schema is powerful because it supports XML Namespace and describes different data types, it can also work with a database.

2.3.4 Resource Description Framework

In recent years, the concept of Linked Data(LD) has become a remarkable pattern to represent 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 of the use of Semantic Web. LD methodology essentially consists of group practises and principles, it aims at publishing structured information on the Web. Its development is based on some standard web technologies such as Resource Description Framework(RDF), which is going to be discussed.[11] Additionally, 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, which elements consist of nodes and directed labelled arrows. The main goal of RDF is to provide a general description about the data which could be understood by applications on the Web, such description is often referred to meta-data. *Statement* is the basic unit for RDF, it is formed by three parts: **a subject**, **a predicate** and **an object**. Besides 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.

The subject of a RDF statement can be a resource of everything, it covers all physical and conceptual entities. A resource or its property is uniquely expressed by a

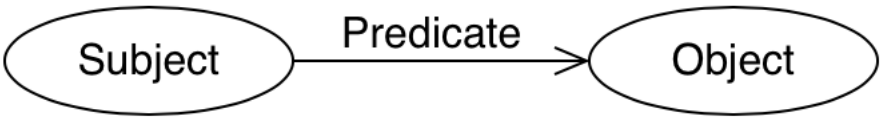

Figure 2.5: A standard statement of RDF(Figure Owner: Jeremy Carroll) [12]

Uniform Resource Identifier(URI). The predicate of a RDF statement is the property of a resource, it determines the relationship between subjects and objects. The object of a RDF statement is also a resource type like the subject, but sometimes it can be just a literal value like number or string as well. Some recent studies found that RDF has several possible benefits for the Semantic Web:

- RDF is a steady framework which focuses on meta-data about internet resources, which makes it increasingly convenient to identify data.
- RDF has standard rules for describing and querying data, allowing meta-data to be processed easier and faster.
- Users will gain more precise results due to meta-data.
- Intelligent software agents can work with more accurate data.

In most cases, there are a few methods to exchange RDF graphs and store the graphic presentation of RDF data, these methods are *serialization formats*. W3C specified an XML syntax for the serialization, it is called RDF/XML, which demonstrates RDF data in an XML form. Furthermore, this syntax uses the clearest data structure for RDF model, so machine can understand easily.

2.3.5 RDF Serialization

In order to understand how XML is implemented in 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, the subject(resource) is **http:// www.jyu.fi/mark**, the predicate(property) is **developer** and the object(literal:string or number) is **Mark**. Moreover, it can be shown like this in the RDF graph:

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

<rdf:RDF>

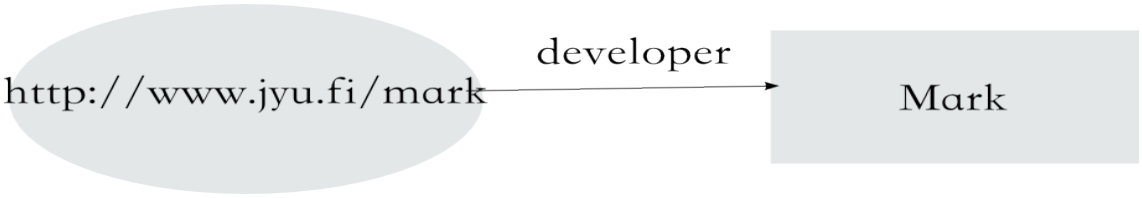

Figure 2.6: Example of RDF Graph

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

2.3.6 SPARQL

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

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

SPARQL syntax is close to RDF, because some concepts used for SPARQL syntax definition which are taken from RDF concepts and abstract syntax with some minor modifications.

A 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 datasets.

- 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:

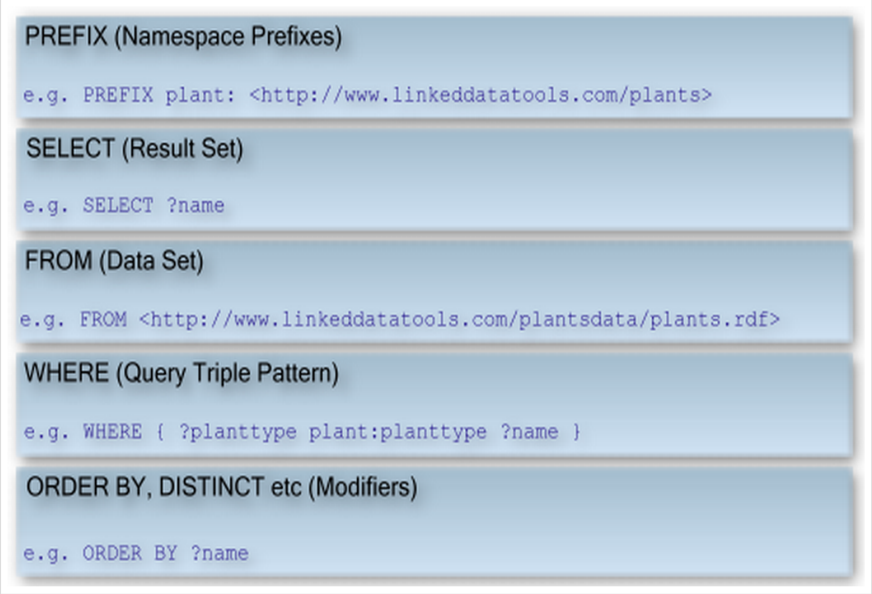

Figure 2.7: SPARQL query
[14]

2.4 Semantic Meta-data, Annotation and Named Entity

Semantic Meta-data: The term of meta-data can be defined as "data about data", it is a very popular topic in both academic and real world. With the development of Web, users are no longer satisfied with single HTML interlinked structure. Users expect a sophisticated approach, for example, that meta-data combined with pages or information resources could be indicated by URI. Besides, the creation of XML and RDF brings meta-data to the stage. Generally, meta-data can be used for two purposes, one concerns data construction and specification, the other one is data its self, the content. In the Semantic Web context, meta-data can interpret information and disambiguate it. It aims at achieving comprehensive management of documents by providing the formalization of content.

Semantic Annotation: First of all, semantic annotation is one type of meta-data; It is very specific. Since the Semantic Web is able to interpret information, semantic annotation is to annotate description on meta-data resource.[15] It provides classes and instances information(property values and relationships) with respect to the entities in a particular domain. In a nutshell, semantic annotation can be seen as a book, and the URIs are single pages inside the book. The following figure from Kiryakov(2004) demonstrate how semantic annotations work:

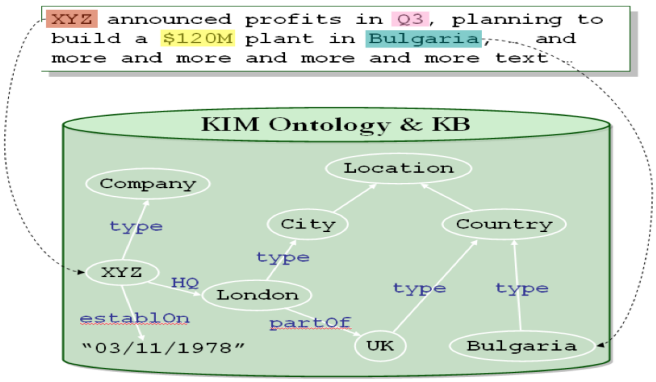

Figure 2.8: Semantic Annotation(Figure Owner: Atanas Kiryakov) [8]

Named Entity: Named entities are regarded as *places*, *people*, *organizations* and other things in Natural Language Process field. It is a description of an object with semantic characteristics that can be interpreted for future usages. Furthermore, named entities contain values such as *numberm* *address* and *time*. Comparing with vocabularies, named entities require more specific understandings of universal knowledge and conceptualization.

2.5 Ontology

The Web is an entity of documents for people, whereas the Semantic Web is an entity of documents for computers. At the present time, a web page is written in HTML. This language is easy for human to read and use, but its structure is complicated making it difficult for machines to gather useful information from it, which leads to fewer outcomes. Computers, on the other hand, read HTML documents like hieroglyphics. In order to make machines understand what users input. Computers

either need to be improved to a super intelligent level, or the structure of metadata needs to changed so that computers are able to understand. Based on current techniques, the second solution seems a bit easier. The Semantic Web collects data which are in a well-structured format for computers to read, understand and process. After data are input into computers, some useful information from the data are acquired. Then the acquired information can 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 which domain they are coping with, the general concepts applying in that domain and the reasoning policies. For instance, person A has a sibling, person B. From human perspective, users also understand person B is the sister or brother to person A, but machines can not understand this inverse relationship because of symmetry, whereas in ontology this issue has been solved.

In a nutshell, a specification provides shared and common understandings of a domain that can 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 exist. Now this concept has been applied to many different fields. For example, in autonomic intelligence aspect, the ontology is created to eliminate the conflicting definitions and understandings between literatures. Things described by an ontology in a domain of discourse by a formal and explicit way are called concepts(**classes**). The diverse features and attributes of concepts are slots(**properties**), the restrictions of properties are facets(**role restrictions**). In addition, a group of individual instances from classes along with an ontology can start to compose a knowledge base. Therefore, it can be considered that a completion of ontology implies an initiation of knowledge base.[16]

Class is an essential component of an ontology, it illustrates conceptions in a domain. For instance, a class of coffee can mean all coffees, and one specific kind from this class is called an instance, such as espresso coffee is an instance of the class coffee. What's more , classes can also be specified into subclasses which represent more detailed concepts than the superclasses. For example, black coffee, white coffee, espresso and cappuccino can be subclasses from the class of all coffees. Alternatively, dividing a superclass is very flexible, a class of all coffees can also be grouped into coffee with sugar or coffee without sugar.

Properties describe attributes and characteristics of classes and instances. For ex-

ample, Starbucks espresso is produced by Starbucks, hence *produce* is the attribute of Starbucks (instance). Moreover from classes perspective, flavour, milk level, etc., which can be the properties for instances of class coffee. The following figure illustrates classes, instances and properties by giving Starbucks as an example:

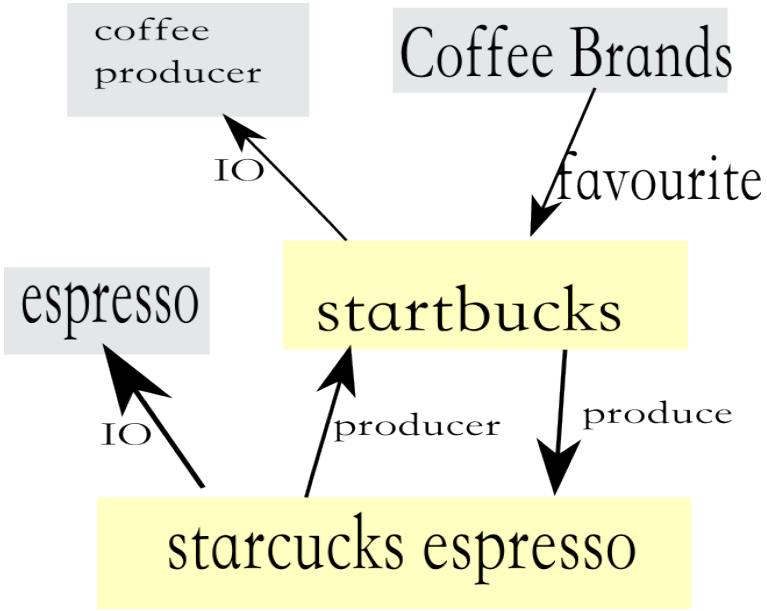

Figure 2.9: Classes, instances and relationship between them in coffee domain

The espresso(instance) has a property *producer* which is the value from an instance of the subclass Starbucks espresso. The Starbucks can be an instance of class coffee producer, since coffee produce has one property named *produce*, hence all instances of coffee producer(class) also own this property. This is considered as consistency of data. The process of data consistency is to remain the information unchanged when data are transferred between various applications or networks. Data consistency can prevent information loss and ensure the data quality. As a result, in order to develop an ontology, here is the approach:

- 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, it has become a controversial topic for researchers how to correctly use ontologies for sharing and defining knowledge. Although there is no precise answer concerning what ontologies are exactly composed of, most ontologies are referring to one or two related things(e.g., stating that a cow is a mammal). Guarino(1998)[17] provided a definition about ontology in his research: A logical theory that accounts to the intended meaning of a formal vocabulary. One well known ability of ontology languages is to expand existed formal vocabularies based on logic truths. As a consequence, users are able to add or delete domain specifications 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, and it is mainly used for two purposes. Firstly, it intends to define terminologies and process data modelling in a flexible and fast way. Secondly, OWL is an efficient data query approach. OWL became a W3C recommendation in 2004, it can be seen as an extension of RDF because they are almost identical. Nevertheless, OWL has better computability, larger vocabularies and rigid constraints. Here is one diagram below which 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 shows, there is a header included in an ontology. An ontology header usually stores information that explain what this ontology contains. What's

more, it can also provide information about versions and whether it uses other ontologies elements.

- Instance: Generally, an instance is seen an object. In OWL, it is called *individual* in term of description logics. An individual is a member of one stated OWL class, it can be seen as a class extension. For example, there is one book called "1984", and a developer is designing a book review website and needs an ontology for the website. In this case, there is no need to concern any situation because any copy of the "1984" is the same, and for this reason, *"1984"* is an individual.

- Class: An OWL class is a collection of individuals which share common characteristics. One class can own infinite individuals, and one individual can belong to one or more classes, even none class. Besides, since OWL classes have subclasses, computers are able to easily infer the relationships between them, which can improve the working efficiency. For instance, there is one Class:*Mammal*, and it has one Individual(Instance):*Ape*. Meanwhile, class mammal is also a subclass of *animal*, and therefore computers can infer the ape is also a kind of animal.

  In addition, classes in OWL need to be explicitly declared since they sometimes cause confusion with individuals. Take the book "1984" as an example again, same books in different libraries might have their own item codes, locations and availabilities. For this reason, book "1984" is seen as a class in this case.

- Properties: *Properties* are the relationships between individuals in OWL. There are two types of properties: *Object property* and *Data type property*. Data type property is the literal values(name,number...) between individuals of OWL class. It is expressed as *OWL:DataTypeProperty*. Object property relates individuals of different OWL classes, for example hasChild can be an individual type property of class parent and class child. It is formulated as *OWL:ObjectProperty*.

In most cases, the Web Ontology Language is based on *Description Logic*[2], it can be used to tell what this world contains. Besides, comparing with RDFS, the OWL

[2]A language to express formal knowledge

languages provide a wider range of vocabularies which describe data model comprehensively and the OWL languages allow users to define relationships between ontologies by annotations.

2.5.2 Sub languages of OWL

The OWL languages are composed of three sublanguages, which respectively are: *OWL Lite*, *OWL DL* and *OWL Full*. All these three variants with different levels 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: To be strictly accurate, OWL Full cannot be deemed as a sublanguage, because it has all the OWL features and no limitations to use RDF constructs. For example, *owl:Class* in OWL Full documents and *rdfs:Class* have same functions, 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 can be executed in practise.[19].

OWL DL: OWL DL is a more computational complete and decidable alternative comparing to OWL Full. The aim of OWL DL is to support reasoning applications with description languages. The same as OWL Full, OWL DL includes all OWL language constructs, but have restrictions while using them. For example, classes cannot be viewed as instances of other classes.

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

Therefore, before users start to develop an ontology, they need to consider which sublanguage is perfect for their needs. Because of this, choosing an alternative has become an interesting issue. According to the specifications from Ontology Working Group, each OWL Lite ontology is also a OWL DL ontology, and therefore the choice between OWL Lite and OWL DL is determined by the degree of users expressive restriction. 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 of classes. There is one thing need to be noticed is that no complete OWL Full implementation currently exists, as a consequence, reasoners for OWL Full have less predictability comparing with OWL DL.[18]

In conclusion, OWL Lite and OWL DL are the extensions of RDF with more restricted terms, while OWL Full can 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. However, from the inverse direction, every RDF document can only be an OWL Full document. Since only some RDF documents are OWL Lite and OWL DL documents, when users are trying to import or change an RDF document to OWL format, some concerns need to be taken into account.[18] For instance, when defining suitable expressiveness of OWL DL and OWL Lite documents, which is expected 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 to users. In this thesis, OWL 1 is recommended for multi-channel framework proposal due to its function integrity comparing with OWL 2. 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 the Semantic Web and representing knowledge about things. 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 similar structures as OWL 1 ontology, and it comprises three notions[20]: *entities*, *expressions* and *axioms*. On the other hand, several new features are added in OWL 2, the following list provides a brief illustration[21]:

Syntactic sugar

This feature helps users make pattern design in a convenient way and it does not change any expressiveness, semantics and complexity. Additionally, reasoning processes develop more efficiently.

New constructs for property

This feature allows users to define additional restriction on properties, while at the same time, 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, seniors ages cannot be in a particular scale but fixed values. However in OWL 2, seniors can have ages over than 60.

Easy metamodelling ability

According to OWL 1 DL specifications, naming for classes should be used precisely, classes and individuals cannot share a same name. However OWL 2 allows users 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 a more precise explanation will be provided. Comparing with annotation for ontology entity in OWL 1, OWL 2 provides a new construct for annotation; It allows users 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

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

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 can be defined in terms of existed things with complicated descriptions. Moreover, this profile can capture the expressiveness of many large scale ontologies. For example, OWL 2 EL can provide a large scale class to define biomedical ontology *SNOMED CT*[5].[23] Additionally, the reasoning capability of this profile can be implemented in polynomial time based on the sizes of ontologies, therefore this profile is quite 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 the data. The most 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. Also, this process does not cause any affect to the data in relational database systems(RDBMS).[23]

OWL 2 RL:

The abbreviation of RL reflects the relation to Rules Language, this profile was designed to make applications use proper expressivity to do scalable reasonings, and describe rules in ontologies. OWL 2 RL could be seen as a perfect option for companies which have RDF applications. What's more, some restrictions in this profile make rule based reasoning engine possible to use by defining customer own business logics. Some individuals which contain implicit meanings in knowledge bases will not be shown during reasoning process because of the restrictions.

[4]EL++ is a lightweight description logic, 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

2.5.4 Ontology Personalization

This section briefly explains what ontology personalization is and how it can help this thesis. A key aspect of ontology personalization named user profiles, which can be used to help understand this part. In the process of Web information collecting, user profiles are created to reflect what users need and their preferences, also, it helps interpret semantic meanings.[24] User profiles usually can be classified and shown in two schematic diagrams: *data diagram* and *information diagram*. Data diagram is obtained through database analysis while information diagram acquired by questionnaires and interviews.

Thus, ontology personalization[24] refers to a conceptualization model. To distinguish on-line users probably have individual expectations from identical things, a personalized ontology is created to develop user profiles with formal descriptions and specifications. An useful example of personalization is Helsinki journey. Users may search Helsinki on the Web and look for different information. Some users will travel to Helsinki, therefore, they care about local weather, history places, etc. Some others going there for studying, so these users concern more about education, student accommodation, etc. Even though the same user, he might expect diverse results according to different situations. Therefore, a user model constructing personalized ontology is needed. Future investigation in ontology personalization is strongly recommended.

2.6 Ontology Matching

It is necessary to clarify the semantic heterogeneity problem before introducing ontology matching. The term of *heterogeneity* refers to the differences between different things, even in a same domain. For example, when independent developers are designing database schema for a same domain, yet the results can be quite different because the developers have their own comprehension. As a result, these differences are seen as *semantic heterogeneity*.[25] Semantic heterogeneity can also exist in some other occasions, such as enterprise information integration, XML documents, and ontologies etc. At present, multiple data systems have been implemented widely in many fields, in order to make the systems understand each other schema, semantic heterogeneity must be eliminated.

In semantic technologies area, *ontology matching* is a way to solve semantic het-

erogeneity issue. First of all, matching functions take ontologies as input sources, and then the functions determine the relationships(correspondence) between ontologies as outputs.[26] The correspondence can be addressed for different tasks, for example, ontologies combination, data interpretation and query etc. As a result, the goal of matching is to have ontologies interoperated.

2.6.1 Motivation

When users try to describe ontologies, semantic heterogeneity issue can happen because different languages and model concepts are used. Firstly, there is one example from Shvaiko's(2005)[26] research. Figure 2.11 illustrates the ontologies matching problem.

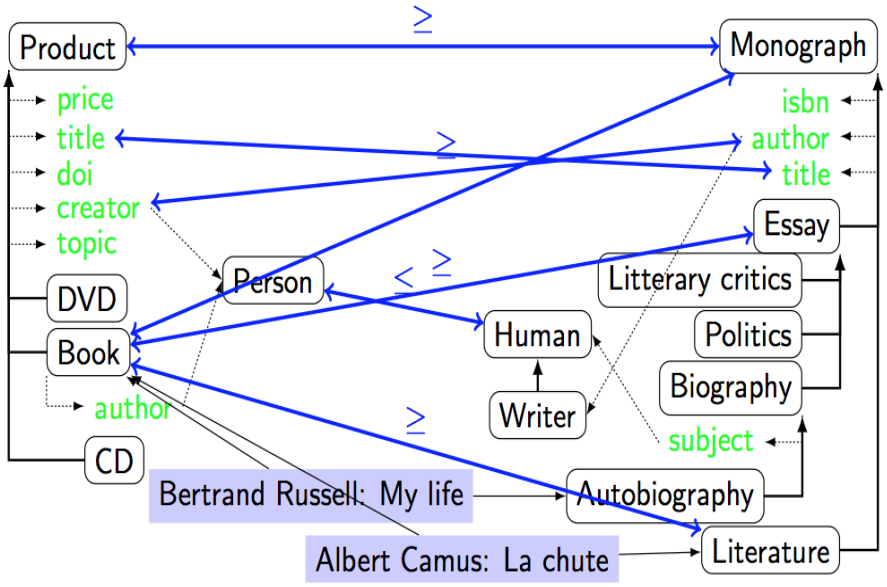

Figure 2.11: Ontologies Matching (Figure Owner: Pavel Shvaiko) [26]

There are two ontologies, ontology *product* and *monograph*. 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 relationships(correspondence) between classes or properties are shown by the line with relation symbols, such as *book*(the subclass of product) is greater($\geq$) than *essay*(subclass of monograph). *Bertrand Russell* and *Albert Camus* are two shared individuals of the subclass book.[27]

Now take the following case as assumption, when two companies start coop-

erating and try to expand their business together. It requires both companies to integrate their products and client data, which are saved in ontology documents. Since these ontologies contain classes relationships, descriptions for properties and instances, the ontologies integration will probably cause the semantic heterogeneity problem. However, once correspondence are determined after the merge, they can be used for many purposes, such as reasoning and inferring. As it can be seen in Fig 2.11, the property title in both product and monograph can be merged, then systems can recognize class product contains but greater than class monograph. [27]

2.6.2 Matching method

Usually, semantic heterogeneity issues need two steps to be solved. Determining the alignments by matching methods is the first step. An *alignment* is a set of correspondences among the merged ontologies(entities), but how to present correspondence? Shvaiko's tutorial on ontology matching(2006) showed that correspondence can be seen as a tuple[6]. For instance, the correspondence between the given ontologies can be shown like this: {id, e , e' R, n }. [26]

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

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

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 matching operations, 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

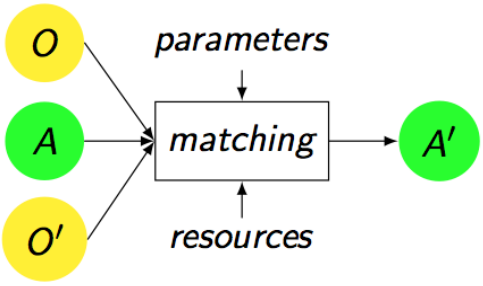

Figure 2.12: Matching Operation (Figure Owner: Pavel Shvaiko) [26]

2.7 Linked Open Data

As shortly mentioned in previous section 2.3.4, generating machine readable data and connecting all documents on the Web became a popular topic in computer science field. In order to achieve the goal, a new type Web named Linked Data Web is being created and under development. This section discusses the essential of Linked Data Web, *Linked Data*. The concept of linked data came from Berners-Lee's article which described the future trends about the Web. [9]

The term *linked open data* is technically understood to mean that data with explicit definitions for machines reading. Its distinctive attribute is to connect or be connected by external datasets, it is proposed that it might be the ideal solution for web data publishing and data connection. The concepts of linked data and the Semantic Web have become exchangeable in the past few years, because 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 models and to interchange RDF links with other links from different data sources. In a consequence, this new type of data can be seen as the fact of the Semantic Web. Here is a list below that demonstrates the comparisons between modern used data and linked data.

Flexibility Both types of data can be published on the Web at any time by users. However, the format of linked data needs to fit RDF document.

Browser Usability It will be a better idea to load linked data by some specific browsers. However, most of current browsers are developed for HTML documents.

Connectivity Linked data aims at connecting everything in the world, comparing with traditional Web which only connect HTML documents, linked data Web has a wider scope.

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

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

1. Assign a distinctive or universal URI name for sources or concepts, which can disambiguate meanings for documents.
2. In order to ensure that URIs are unique, one suggestion is to put HTTP restriction on URIs.
3. When users input URIs into browsers, users will get respond with relevant useful information.
4. For the purpose of expanding information, the related links can be connected and explored.

3 Smart Multi-Channel Communication

Generally, multichannel communication is to send or transmit messages from sources to goal sites respectively. Messages are sent from one channel to another or some others, like driving a car can have several options at a cross or water spread into different rivers. Multichannel communication is commonly used in the following terms:

1. cross media publishing and communication
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 can offer more preferable patterns for customers. Briefly, messages text are integrated or translated into proper versions, which sent or received by right channels What's more, contents from different types of media should be sent at a appropriate time directly to the right person. Therefore, multichannel communication can increase response rates, market awareness,revenues and profit for investors.

3.1 Framework Overview

The initial proposal of semantic multichannel communication came from a previous research by Michael Nagy. In that research, the author offered a sketch, which can be seen in the following Figure 3.1.

The structure of framework is composed by two important parts: *Knowledge Base* and *Message Process Engine*. Knowledge base can be seen as a universal database, it is used to save and update information, which come from the ontologies. Message process engine is responsible to interpret and merge messages. Also, it can choose a proper channel for sending or receiving messages.

As Figure 3.1 shows, five specified ontologies are essentials of the knowledge base. Commodity ontology contains all the information about commercial goods

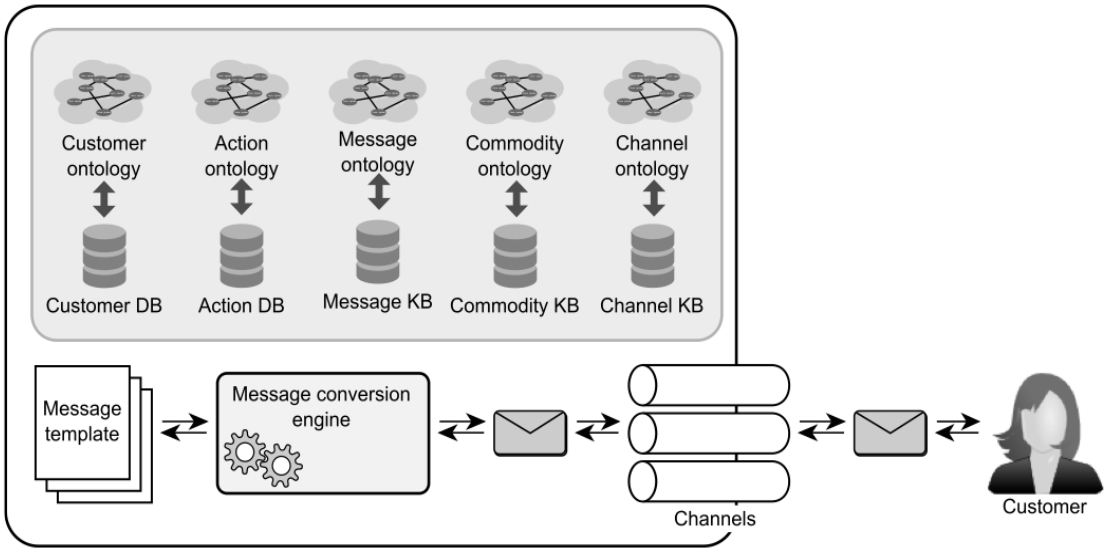

Figure 3.1: Multiple Channel Communication Framework(Figure Owner: Michael Nagy)

[1]

and business services. Channel ontology describes all available communication channels. Message ontology expresses two types of messages for the framework: *concrete message* and *abstract message*. Customer ontology is similar to user profiles which is 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 buyers and sellers perform. More detailed explanations about these five ontologies will be discussed later in the thesis. Also, these five ontologies could be connected to each other, when administrators modify any part of the knowledge base, the rest parts could give correct respondings to adjust the modifications. From customer perspective, when customers send messages through preferred channels, the key information that relating to the business will be abstracted by the message conversion engine. Moreover, information about customers and the preferred communication channels would be stored in message template, which could be invoked when needed. Therefore, customers will not receive annoyed messages from the company.

Commodity Ontology

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

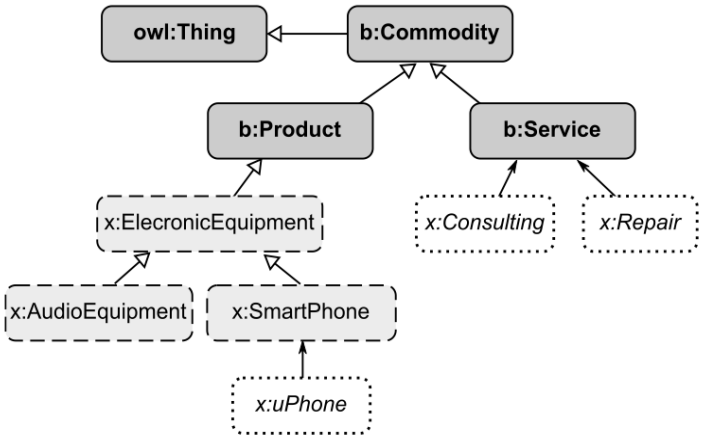

Figure 3.2: Commodity Ontology Structure(Figure Owner: Michael Nagy) [1]

Products and services are all subclasses of class commodity, but at the same time products and services could also own various subclasses according to real business scenarios. As an example, products could have subclass *electronic equipment*, and service could own subclass *consulting*. Furthermore, each subclass could define its own instances like uPhone in Figure 3.2. Due to the flexibility of ontologies, commodity ontology could also import or be imported to integrate with other ontologies.

Communication Ontology

The original name of this ontology is called channel, however, channel could only be seen as a part of this ontology. Since this ontology is responsible for communication domain, it is entitled to communication ontology. The structure of communication ontology is shown below:

Communication ontology has class *channel* as its core, users could define class *channel* by adding or reducing different communication approaches in terms of individual business requirements. For instance, SMS and Email in Figure 3.3 are two subclasses. Class *channel handler* describes how the message should be formed and

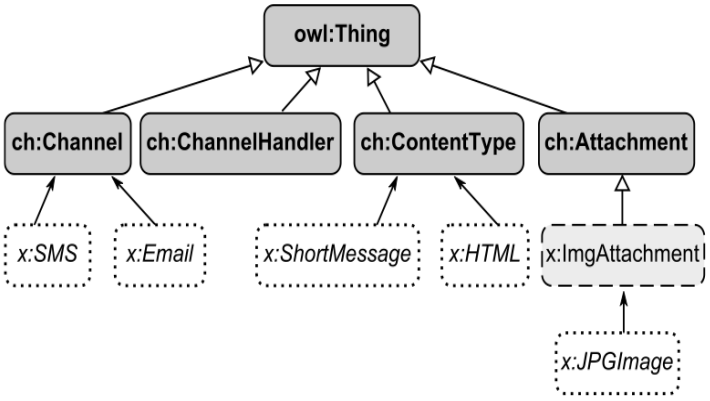

Figure 3.3: Channel Ontology Structure(Figure Owner: Michael Nagy) [1]

operated. 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 formats of attaching files in messages, such as image, voice and video etc. Lastly, class *channel* could be given some properties, such as *speed*, *reliability* and *cost*. However, these properties should depend on the real scenarios, there could be more specific properties in the future.

Customer Ontology

As mentioned in section 2.5.4, user profiles could be seen as the origin of customer ontology. Customer ontology is a model which stores and describes all relating information about users. The central part of customer ontology is class *contact*, which means all communication methods of one user. Furthermore, there is no limitation on how many reaching communication channels could be preferred by one customer. Also, a data type property with value could be defined in contact class, it is called *preference.* Class *customer* has a property *hasContact*, which 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.

Lastly, this ontology is expected to help companies know better about their cus-

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]

tomers, 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, for instance, description about senders and receivers, messages content, and more important is which channel the message used. Besides, action ontology could also describe some business actions, such as information consulting, service complaining, etc. Unfortunately, since action ontology is quite dependent on real cases, it is hard to give detailed classes and properties. There is one figure below which explains relationships between action, message and products.

Message Ontology

Message ontology plays a key role in the whole framework. 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 necessary to include sender or receiver. Abstract messages could be viewed as it is responsible for things which concrete message cannot cover, for example contact information, channel preferences and attachments, etc. As can be seen from Figure 3.6, message ontology is represented in terms of All Thing, it has four subclasses.

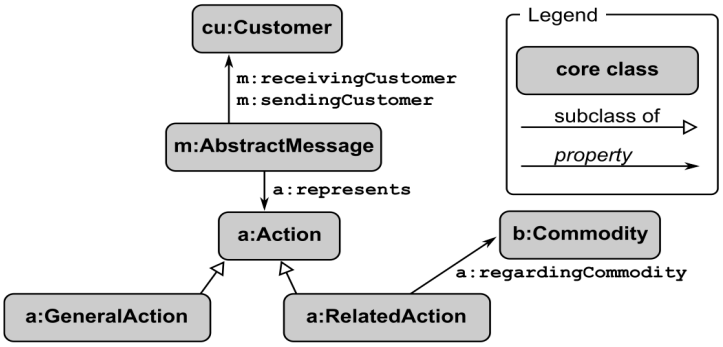

Figure 3.5: Action,Message and Products Relation(Figure Owner: Michael Nagy) [1]

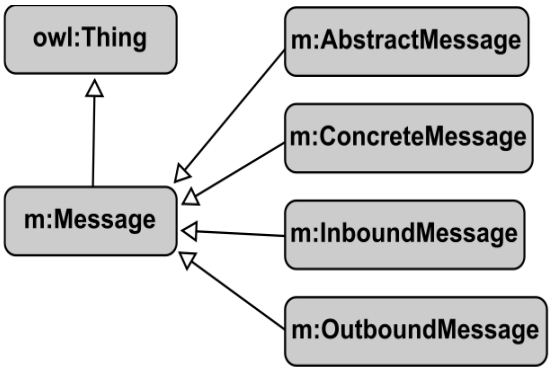

Figure 3.6: Message Ontology(Figure Owner: Michael Nagy) [1]

Since concrete message only shows the central information from messages, it could be defined to have several data type properties with string values, such as contact information, subjects and primary contents. Besides, concrete message class should 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 in order to reach customers for administrators. Some customers do not like using SMS because of character number limitations, by contrast, email systems let users input information as much as they want to. Therefore, the email system channel will have a higher probability to be used according to customer preferences such as 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 becoming increasingly 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 presented in 2001 by IBM. Autonomic computing is a system which can control the functioning of computer applications and manage its own with high level policies from users. Also, this system can 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 can generate several polices and rules to define how the system should work. In other 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 automatically. For this purpose, the 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. Almost in the same manner as if 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 can help the system to perform more efficiently and punctually. Systems seek for improvement through searching, identifying and applying.[29] In a way it is like fitness training, which transform the body to a better state.

Self-Healing:

Nowadays, computer system functions are remarkable and impressive. They can do various things for human, even replace people to finish suitable tasks. However, there is an interesting phenomena; Computer systems look powerful but in fact they are weak. An operational character error, an additional comma, or bracket can cause systems 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 a long time for system developers to identify errors. Self-healing function can detect 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 provides a good example to show this function.[31] Database can detect fail occurrences automatically and repair them by installing some patches. There is no need for the administrator 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 administrators can 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 to detect problems from the early system logs or running exceptions, and then systems can find a step to resolve them.

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 the artificial intelligence field. Utility is a number which can show the level of one state, if the value of utility is higher, it represents that state is better. To objects(human administrator and intelligent agents), utility function can be used to detect possible states of themselves. Also, human administrators or intelligent agents choose the practical state to maximize the utility as desired results. Therefore, in this context, utility can 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 can present customer 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 can have advantages in different attributes or purposes. As a result, customers choose their preferred laptops with maximum utility of each attribute. On the other hand, if an accurate result of customer's utility function is

handed to an intelligent agent, the agent can filter those unmatched laptops on the behalf of customers. For that reason, the process improves the efficiency and save time.

In autonomic computing context, a human administrator can list all values of possible system states through using utility functions. To an intelligent agent, it can obtain these values via an agreement or another relevant utility function. Then, the agent would make the best choices on the customer's behalf to maximize the utility. Besides, as 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 have emphasized the use of utility function. In order to explain why the 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:

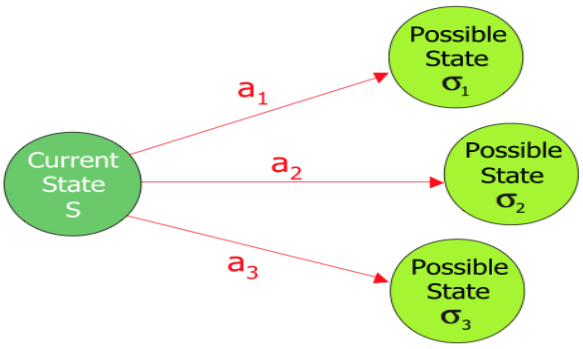

Figure 4.1: States and actions for automatic computing (Figure Owner: JO Kephart) [33]

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 can 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 the case that 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 the system, it tells the 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 the 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 in action policy where administrators make decisions, the system will generate proper actions or behaviours by itself based on goal policy. This move can be seen as self-optimization of the system, then it will understand what it needs to do.

Utility Function: Utility function policy can be seen as the extension from goal policy. It assign a desirability with real value scalar to different states. The administrator does not make decisions or specify to the systems which action should be reached in advance, the system itself chooses 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 can 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 the semantic multichannel communication framework, the 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. However, how to share data in different platforms or applications, it is challengeable. For example, one international IT company has numerous applications for sharing data which are implemented in different platforms with various languages. In that case, how to combine all these applications and make them work together or share information? The solution can be messaging. Messaging is a method which transfer data by using specific formats, it is reliable, immediate and asynchronous. As can be seen in Figure 5.1, it shows

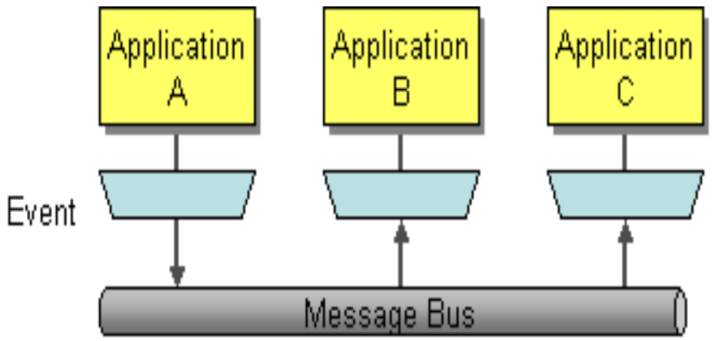

Figure 5.1: Messaging to different applications(Figure Owner: Claude Shannon) [2]

how each application receive messages from one or several other sources, and then one problem coming up with; How could the message or data can be delivered precisely? The answer is message routing. During this chapter, I will use the e-mail message routing analysis as an assumption due to its popularity in current business field, also, some brief introduction about other message routings will be given.

5.1 Message Routing

In general, message routing is an approach that 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, message routing helps the senders to reduce their working load.[35]

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

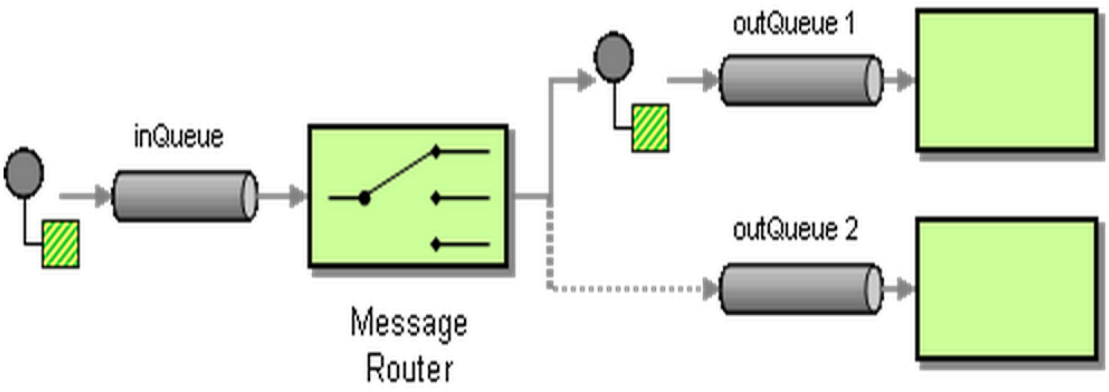

Figure 5.2: Message Router Figure Owner: G Hohpe
[35]

5.2 Information Filtering

Generally speaking, information filtering is a system which uses machine methods automatically to remove or add information based on the user 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 deals with unstructured or semi-structured data, the most common example is email message.[36]

Figure 5.3 below shows an architecture how the system will work. The primary parts are: information filtering system, different types of database and recommendation engine. The system will receive incoming messages from multiple channels 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 saved in the database. Then system will do the syntactical analysis to form the decent response with the help of human and recommendation engine.

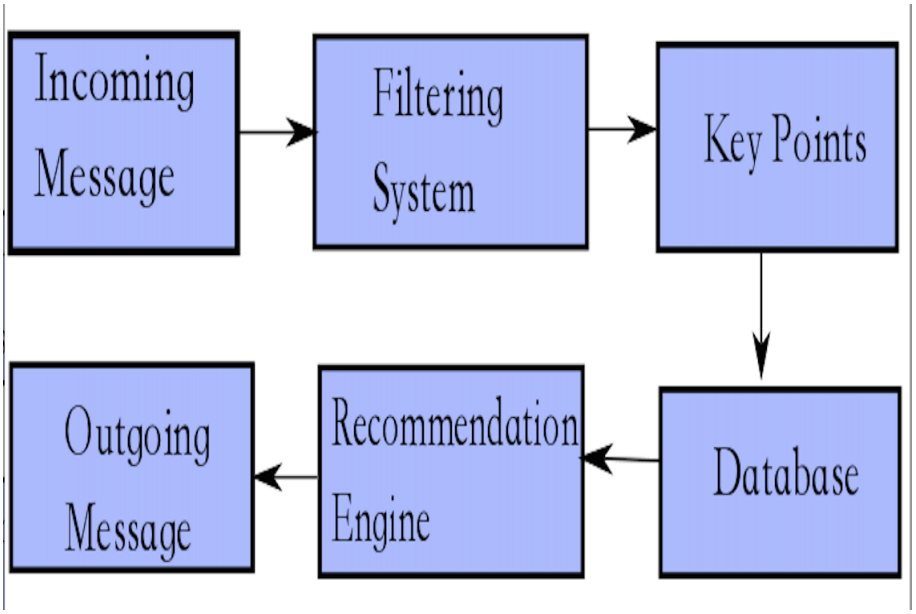

Figure 5.3: Flowchart of Message Merge Model

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.

5.3 Message Conversion Engine

The multichannel framework should be able to defer and review emails automatically according the business rules setting by the administrators. The framework

keeps 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[37]. These business rules along with business policies are stored in message conversion engine. As a result, this engine should own a plurality of actions to deal with each incoming and outgoing message. What's more, message conversion engine could provide these actions to other distribution engines, to help other engines enforce higher priority actions. 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, 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.

Gated messages will be forwarded to the framework administrator, who will check and review according to the business policies. The gated messages in this framework means that messages only contain useful information extracted from sender with different channels. This gated message is the converting results from abstract to concrete. After that, gated messages would be deleted, forwarded or returned manually based on administrator decisions.

Information involved in gated message could be easily used and recognized by machines. Besides, in the Semantic Web research and computational linguistics, information extraction is always a challenge topic. However, how could these gated message be generated; Message conversion engine with LODifier approach is introduced in next section.

5.3.1 LODifier for input text semantic analysis

The extracted information could be applied to deal with different kinds of tasks, for example, customer service can directly know the issues, similar content could be grouped into the same category for management. Besides, combing the extracted information with the message template, companies could know customer preferences, and what the best reply message content should be. Moreover, extracted information could be used to retrieve text and generate ontologies for semantic researches.

LODifier is firstly introduced by a few researchers from Karlsruhe Institute of Technology[11]. Simply speaking, LODifier is an approach which extract entities from unstructured text and discover the relationships between them, and then con-

vert these named entities into RDF representation. Some current researches or strategies can extract semantic relations from text and transform them into RDF representation. However, the existing drawback is the selectivity of text input, which 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. LODifier 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.

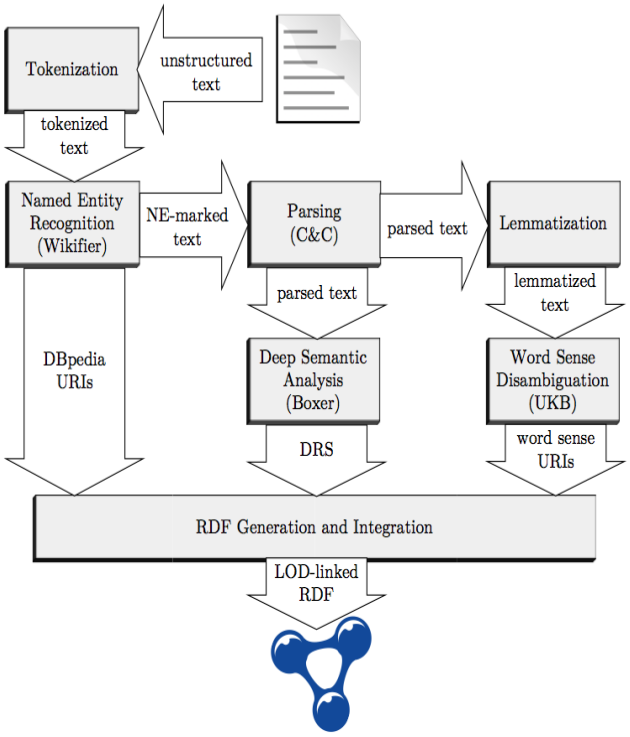

Figure 5.4: LODifier working flowchart(Figure Owner: Augenstein) [11]

Unstructured text firstly are tokenized, in other words, important text are abstracted with tokens. After that, tokenized text as input are identified by a NER tool named *Wikifier*[1] to get entities text and *DBpedia* URIs[2]. Then the mentioned entities

[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

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

Entity Recognition: The first step is 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. Because all the Wiki information are created and revised by verification users, thus 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 proceeding Wikifier outputs. C&C parser and boxer play important roles during the process. The NE-marked text are firstly labelled with tags according to

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.

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 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*. In addition, RDF WordNet can provide 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.

[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 structures with annotations, it also uses POS tags to annotate inputs.

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

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 and it is, assumed to be able to filter and pick up proper information and automatically send it to users, considering their preferences and action history. If message merge will be finished in the near 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 are interested in, whereas information users are not into 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, in order to get rid of trash mails, now users can set the tags or frequent words from same type emails. However, the trash mail could change the title of email or its address to avoid being rejected.

Here is another example why the business aspect needs this technique. An online shopping website reopened after being updated, at the same time, they have many kinds of items with good and reasonable prices.

In order to attract more customers, the website decided to send newsletters each day to their members.[38] Additionally, the website is able to record all the member's actions, such as viewing history, purchased items, user profiles. Customers are not into receiving newsletters everyday, they just hope to receive some useful advertisement or something they are interested. For this reason, 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, the commodity ontology will identify what products are new arrival, and then it will attempt to find and filter customers, who have interests according to their profiles and shopping history. In the next step, the message merge system will filter newsletter according to customers preferences and history. Then the message conversion engine is going to format the messages and get it sent to customers by smart channel selection engine.

6.2 Recommendation Engine

In business models, recommendation engine is a system or an application usually to provide items which customers do not notice. It is also based on customers preference profiles and purchase history. Therefore, a recommendation system can 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.[39] 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 purchased items. For example, if you are surfing Amazon, it will recommend plus products based on other customers searching history which probably are matching yours.

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

Although recommendation system contains various technologies, it still can be classified into three 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 reads a number of books relating about romance, then in the database there will be a classification with tags *romance* and stored in user profile.

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

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]

knowledge based recommendation

This system is based on explicit knowledge about the item assortment, user preferences, and recommendation criteria. For example, which item should be recommended in which context?

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.

6.2.1 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.

6.2.2 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.[40] 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.[41] 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 is 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 conven-

tional usage based similarity metrics because of repeated tags. Also they tried and examined to use meta-data content to recommend interesting things.

6.2.3 Knowledge Based Recommendation

The third type of recommendation system firstly sets up a knowledge foundation with a model about both the users and items, then the system makes an recommendation to user through reasoning if users and items are matched or fitting each requirements.[42] Knowledge based recommendation is prior preferred in marketing comparing to other recommender systems, this might be due to its several great features.[43]

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: the system knows what and such why the user needs 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

6.3 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, the 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 that, in order to communicate with their customers, the company in our assumption could use several channels. 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 5.3, 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 the sender to the 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, a 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 service 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 a laptop and worked, as mentioned earlier enhancement engine cannot be inserted with personalized engine, so we cannot manipulate it and achieve 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. There still has 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 the semantic Web will store the 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 the 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 already happens today with many people receiving the Web advertisements. Traditional web ads are only based on the websites content in order to attract customer attention, 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.[44] 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.

Concluding it can be said that the growing privacy invasion is a complicated issue because there are many groups get involved. It needs more on-line par-

ties to participate 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 many researches investigate this topic(e.g., ontology alignment), there is still a need for a robust and open set of vocabulary in semantic field. Besides, technologies 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.[45]

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

As introduced at the beginning of this thesis, the previous research from Nagy only presented a brief conceptual framework, not every component has been discussed in depth. Therefore, this thesis provides ideal technologies for implementing 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 the reader to have an overall understanding of the semantic Web technique. With the increase 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 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 multichannel communication system. 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 the company tries 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 big data, 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 could be explored as a subtopic for the future work.

Last but not the least, 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 thank 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] 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.

[37] 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.

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

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

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

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

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

[43] 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.

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

[45] 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.