An ontology for information system development methodologies

W.C. Shawa Hon BCom

Dissertation submitted in partial fulfillment of the requirements for the degree of *Master of Science* at the Potchefstroom Campus of the North West University

Supervisor: Mr. N. Mavetera
Co-Supervisor: Prof. H. M. Huisman

December 2009

ABSTRACT

As more organisations are expanding in their operations, there has been a strong emphasis on investing in Information Systems (IS). To ensure successful deployment of IS, software developers are advised to apply Information Systems Development Methodologies (ISDMs). The ISDMs provide a systematic approach to development of IS and ensure efficient use of resources during the development process. The stakeholders conducting the software development usually face problems communicating Systems Development (SD) terminologies. The software developers inconsistently use the SD terminologies during software development. Recently a new concept called ontology has been gaining a lot of popularity in Artificial Intelligence (AI), Computational Linguistics and Database theory. Ontology's potential is being realized in its ability to capture knowledge into a knowledge-base that could be shared with people within a particular domain area.

This study aims at helping the software developers in understanding the different types of terminologies applied when conducting software development. The main objective is to develop an ontology for Information Systems Development Methodologies. The ontology will be used to mediate the meaning of terminologies of SD. In cases where the development of software is undertaken through outsourcing, the software developers located in different countries can use a similar ontology in order to minimise the misunderstanding in the application of systems development terminologies during the development process.

To determine the terminologies to use in the development of the ontology, a qualitative research approach was used. The research is regarded qualitative, as the information gathered for this research will not be expressed in numbers. The research paradigm employed in this research in interpretive in nature. To gather the data, both primary and secondary data collection strategies were employed. Data was gathered through document review and it was coupled with interviews conducted on a software development practitioner and three academics. After the data was collected, it was analyzed using the agreed upon terminologies. The ontology was then developed using a software package called Protégé.

The study found that software developers still have a misunderstanding on the use of SD terminologies. By developing an ontology for Information Systems Development Methodologies, the software developers can use the ontology to mediate the meaning of the SD terminologies that would minimize the misunderstanding of systems development terminologies when conducting software development.

OPSOMMING

Met die verbreding van die werksaamhede van organisasies het daar sterk klem begin val op die belegging in Inligtingstelsels (IS). Om die suksesvolle ontplooiing van IS te verseker, word aanbeveel dat sagteware-ontwikkelaars gebruik maak van Inligtingstelselontwikkelingsmetodologieë (ISOM's). Die ISOM's verskaf 'n sistematiese benadering tot die ontwikkeling van IS en verseker die effektiewe gebruik van hulpbronne gedurende die ontwikkelingsproses. Die aandeelhouers wat die sagteware-ontwikkeling uitvoer, ondervind gewoonlik probleme rakende die kommunikasie van Stelselontwikkelings- (SO-) terminologieë. Die sagteware-ontwikkelaars gebruik die SO-terminologieë inkonsekwent tydens die ontwikkeling van die sagteware. 'n Nuwe konsep genaamd die ontologie het onlangs heelwat veld gewen in Kunsmatige Intelligensie (KI), Rekenaarlinguistiek en Databasisteorie. Die potensiaal van die ontologie blyk uit die vaardigheid van die ontologie om kennis in 'n kennisbasis vas te lê wat gedeel kan word met mense in 'n spesifieke domein.

Die studie poog om sagteware-ontwikkellaars te help om die verskillende tipes terminologieë, wat toegepas word te verstaan tydens sagteware-ontwikkeling. Die hoofdoel is om 'n ontologie te ontwikkel vit Inligtingstelselontwikkelings-metodologieë. Die ontologie gaan gebruik word om die betekenis van terminologieë van SO te bemiddel. In gevalle waar die ontwikkeling van sagteware plaasvind deur middel van uitkontraktering, kan die sagteware-ontwikkelaars wat in verskillende lande gesetel is 'n soortgelyke ontologie gebruik om misverstande in die toepassing van stelselontwikkelingsterminologieë gedurende die ontwikkelingsproses tot 'n minimum te beperk.

'n Kwalitatiewe navorsingsbenadering is gebruik om vas te stel watter terminologieë in die ontwikkeling van 'n ontologie gebruik moet word. Die navorsing word kwalitatief beskou, omdat die inligting wat vir hierdie navorsing versamel is nie in syfers uitgedruk word nie. Die navorsingsparadigma wat gebruik is, is verklarend van aard. Vir dataversameling is beide primêre en sekondêre dataversamelingstrategieë ingespan. Data is versamel deur middel van dokumentresensies en is gekomplimenteer deur onderhoude wat gevoer is met 'n sagteware-ontwikkelings-

praktisyn en drie akademici. Nadat die data versamel is, is dit geanaliseer deur gebruik te maak van ooreengekome terminologieë. Die ontologie is toe ontwikkel deur gebruik te maak van 'n sagtewarepakket genaamd Protégé.

In die studie is daar bevind dat sagteware-ontwikkelaars steeds misverstande toon in die gebruik van SO-terminologieë. Deur 'n ontologie vir Inligtingstelselontwikkelingsmetodologieë te ontwikkel, kan sagteware-ontwikkelaars die ontologie gebruik om die betekenis van die SO-terminologieë te bemiddel sodat die misverstande van stelselontwikkelingsterminologieë wanneer sagteware-ontwikkeling uitgevoer word, tot 'n minimum beperk kan word.

ACKNOWLEDGEMENTS

First and foremost I would like to thank God for giving me the strength to work throughout the period of my studies. I would also like to thank my parents for providing me with the moral support and financial support in all my endeavors. I would also like to thank the National Research Foundation (NRF) and the North West University for their financial support, and providing me with the facilities to enable me to conduct my research. I would also like to thank my supervisor and co-supervisor, Mr. N. Mavetera and Prof. H. M. Huisman respectively for providing me with the guidance in conducting my research. I would also like to thank Mrs. E. de Kock for editing my research. Last but not the least, I would like to thank my friends for the support they had provided me with when I was conducting my research.

This material is based upon work supported financially by the National Research Foundation.

TABLE OF CONTENT

CHAPTER ONE
INTRODUCTION

CHAPTER TWO
ONTOLOGY

CHAPTER THREE
INFORMATION SYSTEMS DEVELOPMENT METHODOLOGIES

CHAPTER FOUR
RESEARCH METHODS AND DESIGN

CHAPTER FIVE
DATA ANALYSIS AND INTERPRETATION

CHAPTER SIX
ONTOLOGY DESIGN

CHAPTER SEVEN
RESEARCH FINDINGS, CONTRIBUTIONS, LIMITATIONS AND FUTURE WORK

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER ONE
INTRODUCTION

1. 1 Introduction

When conducting software development, communication with the stakeholders plays a vital role during the development process. The stakeholders conducting the software development have different skills and levels of knowledge which often cause a lot of misunderstanding when applying Systems Development (SD) terminologies. An ontology can be created to assist software developers mediate the meaning of the SD terminology. The ontology will be used to capture the meaning of the SD terminologies, which can be shared among the stakeholder when conducting the software development processes. The development of the ontology would greatly minimise the misunderstandings of the SD terminologies.

This chapter provides an introduction regarding the study undertaken. The chapter also describes the problem statement, research motivation, research aim and objectives, main and minor research questions, scope and delimitation, and structure of the dissertation. The research involves the development of an ontology for Information Systems Development Methodologies (ISDMs). The ontology will contain agreed upon terminologies from SD that can be shared by the stakeholders when conducting software development processes.

1.2 Background of the study

Ontology is a relatively new terminology being used in Information Systems (IS) and many related fields of study. It is the researchers in Artificial Intelligence (AI) who are embracing this terminology more to describe the representation of things existing in the world by capturing them into a knowledge-base. The essence of their researches is to unearth better ways of representing human knowledge formally in a computer whilst still retaining its ability to be processed. Arnold & Bowie (1985) mention that computers cannot input and process knowledge internally themselves, but humans also have to play a role in what way knowledge can be represented inside the computer.

In respect of this research agreed upon terminologies relating to Information Systems Development Methodologies (ISDMs) will be collected and stored into an ontology. This will help software developers to have a better understanding of the terminologies applied in SD. Once the terminologies have been collected and stored into an ontology, they can be shared and reused by other software development teams when conducting software development projects.

1.3 Problem Statement

There is a limited amount of study that has been conducted in the field of ontology in South Africa. The IS community is yet to appreciate the potential it has to offer to software development. Ontologies have the ability to share information among people within a particular domain area. When conducting software development, the stakeholders are faced with a problem of communicating the SD terminologies, especially when they all have a different understanding of the SD terminologies. Many authors have their own definition of what constitutes an ISDM. Avison and Fitzergerald (2006), indicate that for a definition to qualify as an ISDM, the following aspects namely, *philosophical approach, system development methods* and *techniques and tools* have to be present. According to the author's knowledge, there is no ontology for ISDMs that has been developed in South Africa. The author's intension is to build an ontology to unify ISDM terminologies into a single knowledge-base to help software developers understand the different terminologies existing within that domain.

1.4 Research Motivation

The author has been motivated to conduct this research to help the software developers to have better understanding of different terminologies existing in SD. The SD terminologies could be unified into an ontology and used to share the knowledge among the stakeholders during the software development process. The ontology could help the stakeholders to strengthen their understanding of terminologies within the ISDM domain, in turn minimising their misunderstanding of SD terminologies during the software development process, especially in cases where the stakeholders have to conduct the software development processes whilst located in different geographical locations.

1.5 Research Aims and Objectives

1.5.1 Aim

The aim of this research is to develop an ontology for Information Systems Development Methodologies (ISDMs). The ontology will contain agreed upon terminologies from the ISDM domain. The captured knowledge will be shared and reused by the software developers. The ontology created will help the software developers to have a better understanding of SD terminologies to apply when conducting software development processes.

1.5.2 Objectives

The main objective of this research is to develop an ontology that shares and reuses the captured terminologies in the ISDM domain.

1.6 Research Questions

1.6.1 Main Research Question

What role can ontologies play in Information Systems Development (ISD)?

The main research question could finally be answered by addressing the minor research questions.

1.6.2 Minor Research Questions

What is ontology?

Several definitions by different authors exist describing what an ontology is. They have different elements namely conceptualisation, explicit, formal, shared, semantic network and inference considered to constitute the definitions. The following definitions have been applied in this research.

Studer *et al.* (1998:162) and Hendler (2001:54) define ontology as *"a formal explicit specification of a shared conceptualisation of knowledge terminologies such as vocabulary exhibiting semantic interconnection and rules of inference in a particular* *domain area"*. *Conceptualisation* refers to an abstract model of some phenomenon. *Explicit* implies the concepts and constraints applied should be clearly defined. *Formal* implies the data requiring processing in the ontology should be converted in a form readable by a computer. *Shared* implies that the concepts used to create the ontology must be collectively accepted by stakeholders. The *semantic network* states that the association between the concepts of the ontology must be described. For *inference* an ontology must enable some form of reasoning applied to the categorisation of concepts.

The ontology's aim is to capture consensual knowledge that is shareable and reusable across software applications by different developers. The detailed description on this literature can found in section 2.4 of the research.

What is ISDM?

The research discovered that a lot of inconsistencies existed when giving an appropriate definition of a methodology. Other authors have inconsistently used the terminology method or approach when they intended to mention methodology (Jackson, 1981; Blum, 1994; Iivari *et al.*, 1998).

The definition provided by Avison and Fitzergerald (2006:24) closely captures the elements of what can constitute the definition of ISDM. They state that, "a methodology is a collection of procedures, techniques, tools and documentation aids, which will help the systems developers in their efforts to implement a new information system. A methodology consists of phases, themselves consisting of sub phases, which will guide the systems developers in their choice of the techniques that might be appropriate at each stage of the project and also help them plan, manage, control and evaluate information systems projects".

Additionally, Avison and Fitzergerald (2006) has indicated the term methodology to contain a more comprehensive meaning than method by possessing the characteristic philosophical view in its definition. The definition of ISDM should have the combination of the following components namely *Approach, Process Model, Systems Development Method, Tools* and *Techniques*. The detailed description on this literature can be found in section 3.2 of the research.

What is an ontology for ISDMs?

An ontology for ISDMs is a knowledge-base aimed at capturing consensual knowledge from the ISDM domain. The knowledge was captured by collecting terminologies of Systems Development through interviews conducted on one software development practitioner, three academics and an in-depth survey of the literature review on SD. The ontology can be shared by software developers to enhance their understanding of the terminologies applied in Systems Development (SD). The ontology for ISDMs can also be used in developing other types of ontologies that can be reused across software application by different developers. The detailed description of the ontology for ISDMs can be found in Appendix C.

Why is there a need for an ontology in ISDMs?

After conducting the study through interviews and literature reviews on SD, it was found that numerous inconsistencies existed when using SD terminologies. Different people misused terminology such as method to refer to a methodology. So in order to minimise the misunderstanding of SD terminologies, an ontology for ISDMs was developed. The purpose of the ontology would be to mediate the meaning of terminologies in SD. The ontology can then be shared across the network to minimise the misunderstanding of SD terminology, especially in cases where software development has to be conducted by stakeholders located in different geographical locations.

How do you develop an ontology for ISDMs?

To successfully develop an ontology for ISDMs, it will be required that you collect the agreed upon terminologies from the interview respondents and relevant SD literature. There after the appropriate methodology, method and tools have to be available and used to guide the developer during the ontology development. The methodology used in the development of the ontology was METHONTOLOGY (Fernandez-Lopez *et al.*, 1997). According to Fernandez-Lopez *et al.* (1997), METHONTOLOGY methodology consists of three developmental activities namely management, developmental and supportive activities. The detailed description on METHONTOLOGY can be found in section 2.5.2. The tool used in the development of the ontology is called Protégé. Protégé is an open source software package that is free and available for download on the Internet. It has extensible architecture and is platform independent. The developmental process of the ontology is described in section 6.3.

How will software developers use the ontology for ISDMs?

The software developers will use a share drive, where the ontology for ISDMs can be accessed to view the captured SD terminologies during the software development process.

Then in cases involving software development through outsourcing, software developers located in different geographical locations can use the same ontology in order to minimise the misunderstanding of SD terminologies.

1.7 Scope and Delimitations

The research looked at the role that ontologies play in ISD by developing an ontology for ISDMs. The ontology was developed using an ontology editor called Protégé. The software package is open source, has extensible architecture and is platform independent. The development of the ontology centres more on the management side rather than the technical side of development that involves the coding using Description Logics (DL). The terminologies of the ontology were collected through an extensive literature review and interviews held with one software development practitioner and three academics. The ontology for the ISDMs developed covered terminologies from Approaches, Methods, Process Models, Project Management, Training and Tools and Techniques. The Methodologies that have been fully described in the ontology for the ISDMs are from the Agile Methodologies only.

1.8 Structure of the Dissertation

Chapter One: Introduction

This chapter describes the beginning to the research under study. It contains a background of the study, the problem statement, research motivation, aim and objective for conducting the research, main and minor research questions, scope and delimitations of the research and the structure of the dissertation.

Chapter Two: Ontology

In this chapter the researcher examined the available literature on ontology. The literature focused on the description of ontology from a philosophical view, description of ontology from an ontology engineers view, main components of an ontology, types of ontologies, ontology lifecycles and lastly ontology developmental tools.

Chapter Three: Information Systems Development Methodologies

This chapter examines the available literature on Systems Development Methodologies. The chapter contains several definitions of Systems Development Methodology. Some examples of formal Systems Development Methodologies are described as categorised by Avison and Fitzgerald (2006). Lastly the current direction in the use of Systems Development Methodologies is described.

Chapter Four: Research Methods and Design

This chapter outlines the research strategies employed. The research approach employed is qualitative. The research conducted is qualitative because the information that has been gathered is not expressed in numbers. The research paradigm employed is interpretive in nature. The interpretation of the research involves the sound relationship between the researcher and what is being explored, and the situational constraints shaping the process. The research method employed is narrative. The narrative research method involves collecting stories and writing narratives about individual experiences. Narrative research is a literary form of qualitative research with strong ties to literature. The data collection strategies employed was both primary and secondary. In the primary data collection strategy, interviews collected from four respondents were used. The secondary data collection strategy involved literature on SD gathered from SD textbooks and academic journals. The data analysis methods employed in the research was content analysis, the SD terminologies collected were tested for consistency by using the categorisation process prescribed by Avison and Fitzgerald (2006). After the data collection and analysis was completed, the ontology was developed using an ontology development editor called Protégé.

Chapter Five: Data Analysis and Interpretation

In this chapter, data collected from the interviews conducted with the four respondents and how the data collected has been analysed is described.

Chapter Six: Ontology Design

This chapter indicates the steps on how to develop the ontology for Information Systems Development Methodologies and the results of the development are presented in Appendix C.

Chapter Seven: Research Findings, Contributions Limitations and Future Work

This chapter contains the findings, the contribution made by the research, limitations and future work that can be done on the research.

1.9 Summary of the chapter

The aim of this research is to develop an ontology for Information Systems Development Methodologies (ISDMs). This chapter has outlined the background of the study, problem statement, research aims and objectives, main and minor research questions, scope and delimitations and dissertation structure. The next chapter provides detailed description on ontologies.

CHAPTER TWO
ONTOLOGY

2. 1 Introduction

Chapter two reports on the study relating to ontologies. The literature gives an insight to the background on ontologies, ontology's description from a philosophical perspective, ontology's description from an ontology engineer's perspective, developing ontologies using an ontological engineering approach, ontology method, methodology, and tool suites applied in the research and summary of the chapter.

2.2 Background on Ontologies

In the recent past, ontologies have been gaining extensive popularity within the Computer Science community. The terminology ontology obtains its origin from an ancient Greek philosopher Aristotle in the years 350 BC who had concerns over the study of things in existence. Guarino (1998) advises the reader to pay attention to a preliminary terminological classification to distinguish between "Ontology" with the capital letter 'O', which implies having an uncountable reading and refers to a particular philosophical discipline, whilst 'ontology' in the lower case implies a countable reading and is commonly referred to within the Artificial Intelligence (AI) community. In philosophy, ontology is considered as a system of categories accounting for certain vision of the world that is independent of a particular language. The independence of the *language* is what is also used to describe Aristotle's ontology from a philosophical perspective. On the other hand, ontology from AI perspective refers to an *engineering artefact*, constituted by specific *vocabulary* used to describe a certain reality, coupled with a set of explicit assumptions regarding the *intended meaning* of the vocabulary (Guarino, 1998).

The 'ontology' described above clearly have a minimal distinction in differentiating between their relationships. By using the word *conceptualisation*, two ontologies can be different in vocabularies, while sharing the same conceptualisation (Guarino, 1998). The definition of conceptualisation provided by Guarino (1998) states that it is a structure <D, **R**>, where D is a domain and **R** is a set or relevant relations on D. It is this definition stating an ontology as a specification of a conceptualisation that has been adopted by Gruber (1993b).

According to Neches *et al.* (1991), ontology research originated as the Defence Advanced Research Projects Agency (DARPA) vision to invest in building modern intelligent systems. There vision was to create knowledge-based systems using reusable components interoperating with existing systems capable of performing some reasoning functionalities. The resulting effort of this endeavour led to the development of ontologies that represented knowledge semantically without losing their ability to be machine processable (Zhang, 2007).

Currently ontologies are playing a significant part in Artificial Intelligence (AI), Computational Linguistics, and Database theory. Research areas namely, knowledge engineering (Gruber, 1993a; Uschold & Gruninger, 1996; Gaines, 1997; Gomez-Perez, 1997), knowledge representation (KR) (Guarino, 1995; Artale *et al.*, 1996; Sowa, 1999), qualitative modelling (Gotts *et al.*, 1996; Borgo *et al.*, 1997; Casati & Varzi, 1997, language engineering (Lang, 1991; Bateman, 1995), information retrieval and extraction (Guarino, 1997; Benjamins & Fensel, 1998; McGuinness, 1998; Welty, 1998) and object-orientated analysis recognise the importance of using ontologies within the Information Systems (IS) field. The section below gives a description of the fundamental aspects regarding ontologies from a philosophical point of view.

2.3 Ontology's Description from a Philosophical Perspective

The origin of ontology as a technical word that philosophically describes the existence of things in the world and their relationship arose from the ancient Greeks Aristotle in the years 350 BC. Aristotle was concerned with the question 'what is the essence of things through the changes' (Corcho *et al.*, 2006). The findings of Corcho *et al.* (2006) reported that all entities in the world that existed had some form of relationship in one way or the other, or is said to exhibit some universal link.

The universals play an important role in the popularity of ontologies. It is essential to understand what problems postulating universals can be, when applied to examine and provide a solution to address meta-ontological problems (Rodriguez-Pereyra, 2002). The supposition of universals clearly provides a better understanding of the subject matter. Rodriguez-Pereyra (2002) indicates that the theory of universals have a resemblance to nominalism. If his reports can be shown to be wrong to this claim, then most of the resemblance of nominalism will collapse (Rodriguez-Pereyra, 2002).

Some philosophers such as Fraassen (1989) have attacked the very existence of valid inference to the best explanation, claiming that a theory can never be justified just by showing that it would explain something. According to Rodriguez-Pereyra (2000), universals should explain various phenomena, such as how general terminologies can apply to different individuals or even how different individuals can be similar or have the same property. Armstrong (1978) already suggested the scope of the problem when he said:

"The problem of universals is the problem of how numerically different particulars can nevertheless be identical in nature, all be of the same type".

The problem of universals was first posed by Porphyry, who formulated it but did not offer a solution. When Porphyry posed the problem, he did not formulate it as a problem of how something can be, but as a problem of whether something is. The definition of universals provided by him includes entities accidentally predicated of something. Aristotle defines a universal as *that which is by its nature predicated of a number of things.* The terminology *predication* implies the relation between the linguistic expression in Greek and other natural languages.

In essence, the universals have become the main emphasis in knowledge modelling, providing the basis for classes or concepts. Marias (2001) came across many recent studies closely linking the base of the ontology theory. There discussions emphasised ways in which people's minds capture reality and in what ways their mind structure is employed in visualising reality. Once reports on people's perception of reality were researched and established, it was also essential to identify the various forms of categorisation that identified anything in existence in the world.

The concept of frame was proposed by Minsky (1975). His proposition explained that a frame in a frame system represents a concept or an objective. Additionally, Minsky (1975) reported that a collection of properties or concepts existed on frame, and they contained initial values. He also noted that the concepts could be used to represent the possible raised questions in hypothetical situations. This is then represented by the frame and changing the values of concepts relating the frame to the particular situation. Based on Minsky's concepts, the frame system subsequently gained ground as a basic tool for knowledge representation. According to Cocchiarella (1996), only three forms of states have been studied to date, namely nominalism, conceptualism and realism. These states, known as formal ontologies, are believed to provide an architectural foundation of modern Information Systems (Cocchiarella, 1996).

Cocchiarella (1996) later reported on the categorisation of ontologies by describing them in detail. He firstly described nominalism as being primarily *words*. They are the most restrictive of the three, because of the universals that can be predicated of things other than the predicated expression of the language. Secondly conceptualism was described primarily as *concepts*. Lastly realism was described as primarily being *things* (Cocchiarella, 1996). Information representation systems can also be used to categorise forms of the state of an ontology.

For a classification to be an ontology for a computer, there must be an ability to reason with the ontology that has been created. The finding by Corcho *et al.* (2006) lead to the establishment of the major difference between ontology from a philosophical point of view and from a Computer Science point of view. According to Corcho *et al.* (2006), an ontology has to be codified in a machine interpretable language. The description in the following section provides some documented definitions of ontologies from an ontology engineering perspective.

2.4 Ontology's Description from an Ontology Engineering Perspective

Gomez-Perez *et al.* (2004a) indicates ontologies can capture consensual knowledge that can be shared and reused across software applications by different developers. Several authors namely, Gruber (1993b), Guarino (1995), Studer *et al.* (1998), Swartout and Tate (1999) and Hendler (2001) have given numerous but almost similar explanations to the terminology ontology which can be applied as a definition. Each depends on what the individual wants to derive from it. These definitions provided help software developers in establishing a most suitable definition of an ontology from an ontology engineering perspective. To people with little knowledge about ontologies, it would be best described as some form of explanation related to nature of reality. To the Computer Scientists, several definitions below have been documented to enhance their comprehension of the subject matter (Uschold & Gruninger, 1996).

- Gruber (1993a:199) defines ontology as *"an explicit specification of a conceptualisation"*. Conceptualisation means an abstract, simplified view of the world. Then specification means a formal and declarative representation.

- Guarino (1995:626) defines ontology as *"the study of the organisation and the nature of the world independently of the form of our knowledge about it"*. Guarino's emphasis is the formal ontology, theory of a priori distinctions between the entities of the world.

- Studer *et al.* (1998:162) define ontology as *"a formal explicit specification of a shared conceptualisation"*. The definition is the most concise but requires clarification. Conceptualisation refers to an abstract model of some phenomenon. Explicit implies the concepts and constraints applied should be clearly defined. Formal implies the data requiring processing in the ontology should be converted in a form readable by a computer. Shared implies that the concepts used to create the ontology must be collectively accepted by stakeholders.

- Swartout and Tate (1999:18) define ontology as *"a basic structure or armature around which a knowledge-base can be built"*. They indicate that like an armature in concrete, an ontology should provide a firm and stable knowledge skeleton to which all other knowledge should be attached.

- Hendler (2001:54) defines ontology *"as a set of knowledge terminologies, including the vocabulary, the semantic interconnections and some simple rules of inference and logic for some particular topic"*. Firstly it states that ontology specifies the meaning of relations between the concepts used. Lastly, an ontology must enable some form of reasoning.

The definition applied in this research is provided by combining Studer *et al.* (1998:162) and Hendler's (2001:54) to define ontology as *"a formal explicit specification of a shared conceptualisation of knowledge terminologies such as vocabulary exhibiting semantic interconnection and rules of inference in a particular* *domain area". Explicit* implies the concepts and constraints applied should be clearly defined. *Formal* implies the data requiring processing in the ontology should be converted in a form readable by a computer. *Shared* implies that the concepts used to create the ontology must be collectively accepted by stakeholders. The *semantic network* states that in an ontology meaning of relations between the concepts are used. To *inference* an ontology must enable some form of reasoning applied to the categorisation of concepts.

The above definitions all focus on different aspects that constitute an ontology. Gruber's (1993a) emphasis of the definition is on the formal explicit specification of a conceptualisation. Guarino's (1995) emphasis is on the distinction between entities of the world. Studer *et al.*'s (1998) emphasis of the definition is on reusability and shareability. Swartout and Tate's (1999) emphasis is on providing a stable knowledge skeleton to which other knowledge should be attached. Hendler's (2001) emphasis is on semantic interconnections and rules of inference and logic of a particular topic.

For an ontology to satisfy the requirements in the definitions as provided above, vital components have to be present to ensure its completeness and conciseness. In the following section some of the vital components required for the development of ontologies are described. Although the terminology ontology is adopted from the Artificial Intelligence (AI) domain, it is regarded to be outside the boundary of fulfilling the objective of this research.

2.4.1 Main Components of Ontologies

There are numerous knowledge representations of ontologies in existence sharing at least the following elements described below (Corcho *et al.*, 2006).

Classes describe the conceptual representation of concepts. The classes follow a hierarchical structure from which new classes can be formed from already existing ones.

Relationships describe the association between concepts of the domain. The relationships indicate the characteristics existing on the concepts.

Formal axioms is a preposition that is not easy to demonstrate, but is apparently necessary to making a decision. .

Instances represent elements or individuals in an ontology.

The components of the ontologies described above will assist in the development of the ontology for Information Systems Development Methodologies (ISDMs) in section 6.3 of this research.

Semantic Network

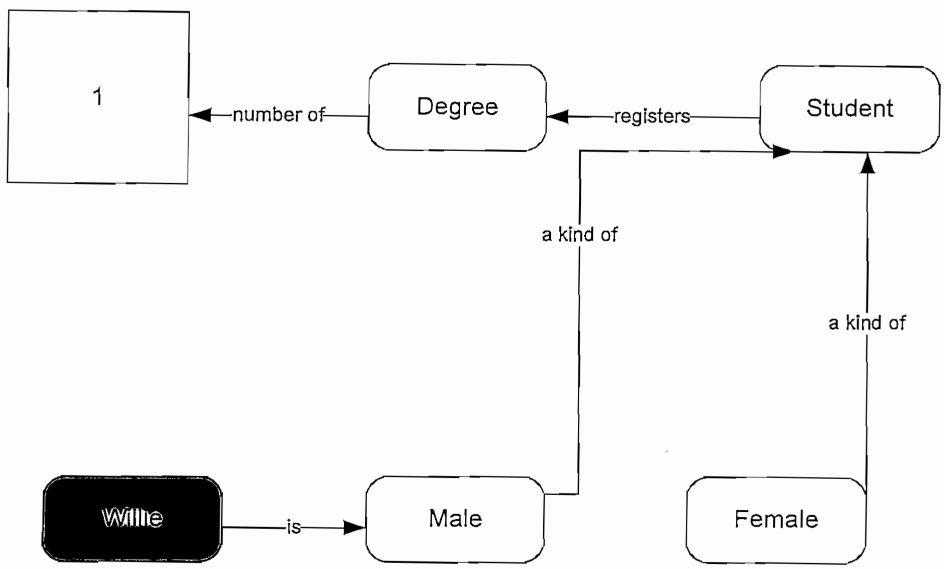

Figure 2.1: A simple Semantic Network

The diagram represented in figure 2.1 is a Semantic Network. Such networks represent the components of the ontologies namely, classes and its attributes and association between the classes. The concepts shown in the figure 2.1 represent classes. These classes are linked by a form of relationship. For example, Willie is linked to male by relationship type *is*. The Male is linked to Student by the relationship type *a kind of*. Then Student is related to Degree by the relationship type *registers*. Lastly Degree is related to the instance 1 by the relationship type *number of*. So Willie is a Male that is a kind of a student that is registered for a single Degree Programme.

2.4.2 Types of Ontologies

Numerous authors have viewed ontologies from different perspectives, hence the diverse literature existing on ontology classification.

From a *generality level*, Guarino (1998) describes the following ontology types to exist:

High-level ontologies describe concepts at a very abstract and general level. These concepts are often shared across several domains and application. There description of the concepts is restricted to abstract level.

Domain and Task ontologies describe knowledge specific to a domain of task of interest. The description of the concepts is much more in-depth.

Application ontologies have a more refined scope, which provides a specific vocabulary required to describe a certain task being performed in a particular context of application. They make use of both the domain and task ontologies and describe the roles that some domain entities play in specific tasks.

2.4.3 Characteristics of Ontologies

The following are useful characteristics of an ontology based on literature by Chandrasekaran *et al.* (1999); Gruber (1993b); Guarino (1995); McGuiness (2002) and Schreiber *et al.* (1994).

- Vocabulary

An ontology has a data dictionary providing the meaning of terminologies in the subject area.

- Taxonomy

A taxonomy is a hierarchical categorisation or classification of entities within a domain.

- Content Theory

The ontology is responsible for classifying the classes of objects, their relations and concepts.

- Knowledge Sharing and Reuse

The major purpose of ontologies is knowledge sharing and knowledge reuse by application.

2.5 Developing Ontologies using an Ontological Engineering Approach

Since a software product is produced at the end of the development process, a disciplined approach is followed to ensure a monitored design process of ontologies. The developers should have knowledge on processes, life cycles, principles and methodologies to successfully build the ontology. The developers have to also adhere to quality control standards to aid the maintenance process of the ontology in cases where major changes are made to the requirement specifications.

2.5.1 Principles for the Design of Ontologies

In this section, a set of principles that are significant in the development process of an ontology is reported. According to Gruber (1993b), ontology design principles are the objective criteria that aid an ontology engineer in building and evaluating the ontology design.

The following five design principles in the development of ontologies are identified (Gruber, 1993b):

Clarity: The principle indicates that an ontology should communicate effectively the intended meaning of the defined terminologies. The definitions should be objective and indicated by formal axioms. The definitions should be documented with a natural language. This principle has to be reflected in the ontology for ISDM developed.

Minimal encoding: The conceptualisation should be specified at the knowledge level without depending on a particular symbol-level encoding.

Extendibility: The ontology should be able to define new terminologies for special users based on existing vocabulary without making any changes to the already existing definitions.

Coherence: The ontology should only permit inferences that are consistent with the definitions. If a sentence inferred from the axiom contradicts a definition, then the ontology is incoherent.

Minimal ontological commitments: Many of the entities quantified in everyday speech do not really exist. So our ontology should contain simple objects. These things do exist but, we should not commit to admitting things into our ontology.

2.5.2 Development Process and Life Cycle for Ontologies

Similar to any software development, a development process starts with inception and ends with completion. The development process is divided into stages, which the software developer can use to check if the development process is adhering to the stipulated requirements.

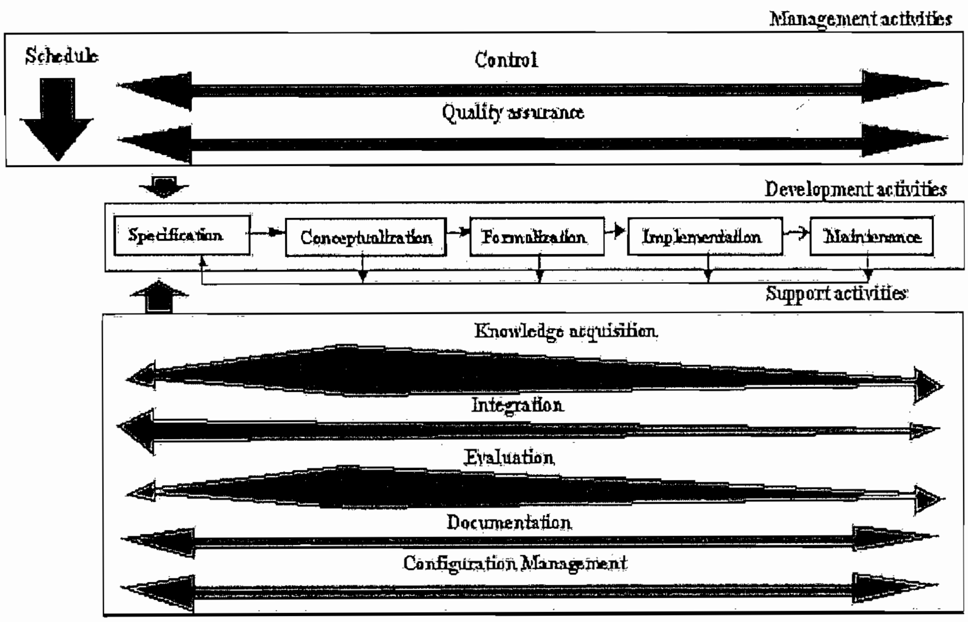

Figure 2.2 METHONTOLOGY Lifecycle (Adopted from Fernandez-Lopez *et al.*, 1997)

With the emphasis on the design principles mentioned in the previous section, developing an ontology can eventually deliver a software product that is consistent with a proposed design by software developers (Gruber, 1993b). According to Fernandez-

Lopez *et al.* (1997), a process for the development of an ontology was created in the form of a framework of the METHONTOLOGY methodology for ontology construction. The ontology development process consists of activities that are performed during the ontology construction and are classified into three categories namely management, development and support activities. These activities are discussed next with the aid of the diagram in figure 2.2.

The **ontology management activities** are performed throughout the development process to help the developers in ensuring that the project does not run off course. This category of activities includes scheduling, control and quality assurance. The scheduling activity identifies the time and resources required for completion. The control activity ensures tasks are completed accordingly. The quality assurance activities assure product output remains satisfactory (Corcho *et al.*, 2006).

Ontology development-oriented activities are grouped into pre-development, development and post-development activities. The pre-development activities identify the problem to be solved with the ontology and the applications where the ontology will be integrated.

During the development stage, the objective and the intended users are identified. The conceptualisation activity structures the domain knowledge as meaningful models at the knowledge level either from scratch or by reusing existing models. The implementation activity builds the computable model in an ontology language (Corcho *et al.*, 2006).

The post-development activities involve maintaining, updating and correcting the ontology if needed.

Finally, **ontology support activities** include a series of activities that can be performed during the development-oriented activities. They include knowledge acquisition, evaluation, integration, merging alignment, documentation and configuration management (Corcho *et al.*, 2006). The lifecycle process model of METHONTOLOGY proposes to start with scheduling, then the specification on the ontology. Once the specification is completed, conceptualisation commences. The objective of conceptualisation is to organise and structure the required knowledge. Once the conceptual model is completed, METHONTOLOGY proposes to automatically implement the ontologies using translators (Gomez-Perez *et al.*, 2003).

2.5.3 Methods and Methodologies used in Developing Ontologies

The increased popularity in the development of ontologies compelled research groups to provide systematic steps for ontology development. The steps include following methods and methodologies to aid the development process.

Concerning the methods and methodologies for developing ontologies, several approaches have been described detailing the activities involved during the process. The approaches described are Uschold and King's (1995) method, the Gruninger and Fox's (1995) methodology, the Bernaras *et al.*'s (1996) KACTUS approach, the Swartout *et al.*'s (1997) SENSUS method, the Staab *et al.*'s (2001) On-To-Knowledge methodology and Fernandez-Lopez *et al.*'s (1997) METHONTOLOGY.

Uschold and King (1995) developed a method to be used in developing ontology that was regarded as an extensive contribution to ontology engineering. In order to achieve an effective and efficient development process, four fundamental phases were proposed. The phases were firstly to identify the main purpose of developing the ontology, secondly to build the ontology, thirdly, to evaluate the ontology and lastly for documentation of the ontology.

The **Gruninger and Fox** (1995) methodology proposes building a knowledge-based systems using first-order logic. It identifies intuitively the main scenarios. Then, a set of natural language questions called competency questions are used to determine the scope of the ontology. The questions and their answers are used to extract the main concepts and their relations and axioms of the ontology.

In the **KACTUS project** methodology, the ontology is built on the basis of an application knowledge-base (KB) by means of a process of abstraction. Bernaras *et al.* (1996) state this methodology is initiated by building the KB for a specific application. When a new KB in a similar domain is needed thereafter, they generalise the first KB into an ontology and adapts it for both applications. Applying this method recursively, the ontology would represent the consensual knowledge needed in all the applications.

Swartout *et al.* (1997) came up with the **SENSUS** method. The method follows a top-down approach which links domain specific terms to a large scale ontology and later prune the large-scale ontology to suit the requirements. This approach promotes the shareability of knowledge, since the same base ontology is used to develop ontologies in a particular domain.

Staab *et al.* (2001) introduced the **On-To-Knowledge** methodology. The methodology includes techniques that ensure that knowledge management within an organisation improves its quality. The methodology firstly conducts preliminary investigations on terminologies appropriate to the application. Once the investigation is completed, a series of activities commences in which competency questions are described, and potentially reusable ontologies are studied for release in the initial draft of the ontology. Thereafter the terminologies are refined to produce an application-oriented ontology.

Fernandez-Lopez *et al.* (1997) proposed the **METHONTOLOGY**. The Ontological Engineering Group of the Technical University of Madrid created this methodology. The advantage of this methodology lies in its ability to build ontologies either from scratch, reusing other ontologies or reengineering them. The METHONTOLOGY framework enables the construction of ontologies at knowledge level. The main phase in the development process using the METHONTOLOGY approach is the conceptualisation phase.

2.5.4 Tools used in the Development of Ontologies

According to the literature documented by Gomez-Perez (2004b), ontology tools can be categorised into two groups namely, language-dependent ontology tools and the extensible language-independent ontology development tools and tool suites. The two groups of ontology represent the different stages of evolution.

Language-Dependent Ontology Development Tools

The first generation of ontology tools emerged in the mid 1990s. The similarity of the tools lies in the need to edit or browse the ontologies with their corresponding languages. One of the tools developed during this era was Ontolingua server consisting of web interface to ease collaborative development.

Extensible Language-Independent Ontology Development Tool Suites

The recent generation of ontology development tool suites developed are Protégé (Noy *et al.*, 2000), KAON tool suite (Meadche *et al.*, 2003), OntoEdit (Sure *et al.*, 2002) and WebODE (Arpirez *et al.*, 2003). These tools are robust in nature and can support most of the ontology life cycles.

2.6 Ontology Method, Methodology and Tool Suites applied in this Research

The methodology applied in this research is METHONTOLOGY. Its main advantage is the ability to build ontologies either from scratch, reusing other ontologies as they are, or by reengineering. It involves identifying the ontology development process, a life cycle, and techniques to carry out each activity.

In order to build the ontology, the Protégé editor has been made available. Protégé is an open source ontology development platform with a knowledge base framework. Protégé integrates the necessary and comprehensive tool suites for ontology development. It also has extensibility and platform independence. The development processes of Protégé are later described in section 6.3.

2.7 Summary of the chapter

The chapter provided a description on the background of ontologies. The chapter had a description of ontologies from a philosophical perspective as documented by the Greek philosopher Aristotle around the years 350 BC. The chapter also emphasised the philosophical differences between ontologies and from an ontology engineering perspective. Several authors have described different methods that can be employed when developing the ontology. The literature indicated that the method employed for the development was mainly dependent on the purpose of the ontology. Additionally, several elements can be adopted from other methods to aid the ontology's development process. METHONTOLOGY was the methodology that found most suited for the development in this research. Its main advantage is the ability to build ontologies either from scratch, reusing other ontologies as they are, or by reengineering. This literature on ontologies was collected in order to enhance the development processes. The documentation should serve as a guide for the novice ontology developer. The role of ontology in this research is to provide software developers with a program capable of mediating the meaning for Systems Development terminologies, which could be shared with the software developers to minimise the misunderstanding of SD terminologies. The next chapter provides a detailed description on Information Systems Development Methodologies (ISDMs).

CHAPTER THREE
INFORMATION SYSTEMS DEVELOPMENT METHODOLOGIES

3. 1 Introduction

In chapter two, literature relating to ontologies was discussed. It described the basic levels of the ontologies. It is this literature that provides the ontology developer with the development principles and steps to build the ontology. Once the steps to building an ontology are understood, elicitation of the relevant concepts for the ontology is inevitable. The development of an ontology of a specific domain requires extensive consultation of stakeholders in order to exhaustively elicit concepts used in a particular domain. This chapter is mainly dedicated to the elicitation of concepts in Information Systems Development Methodologies (ISDMs). The chapter contains the description of some Systems Development Methodologies with regard to themes that have evolved over the years. "The themes represent a particular foundation, school of thought or focus that has influenced Information Systems" (Avison & Fitzergerald, 2006:46). These themes are said to be part of a philosophical background of a particular technique, tool or methodology. The chapter also discusses the current direction in SDMs, and concludes with a summary of the chapter.

3.2 Background Literature

Systems Development Methodologies or Methods have shaped one of the central topics in Information Systems and Software Engineering. Currently the estimated number of systems development methods existing is over one thousand (Jayaranta, 1994). Despite the amount of effort dedicated to their usage, more understanding about them is required to minimise the misunderstandings on methodologies.

The first reason contributing to the misunderstanding is a lack of knowledge to support the topic (Wynekoop & Russo, 1993). The second reason is that the concepts relating to the subject matter are ambiguously described (Wynekoop & Russo, 1993). Wynekoop and Russo (1993) have identified numerous inconsistencies regarding the use of the term methodology.

It is vital to distinguish the differences between an approach, method and methodology, as the use of method may be at the level of approaches rather than specific methods. Hendler-Seller (1995) findings states that numerous stakeholders participate in different activities as users of the Systems Development Method. Software developers prefer applying methodologies during the software development as they serve as a guide in monitoring quality and progress of the software project (Westrup, 1993). To successfully ensure the implementation of software product, methodologies have to be applied. To date, there is no single definition of Systems Development Methodology accepted worldwide, but its influence on software development has highly been acknowledged (Westrup, 1993).

To provide meaningful scientific support for appreciating the development terminologies mentioned above, they have to be related with other branches of science such as organisational science and mechanical engineering (Brinkkemper, 1996). Some definitions by several authors, though not exhaustive, have been listed in table 3.1. These definitions reveal the inconsistencies in the definitions and also help to provide a better understanding of the terminologies approach, method and methodology.

Table 3.1: Definitions for Approach, Method and Methodology of Systems Development

Terms	Definitions with Authors	Components
Approach	"An approach is a set of goals, guiding principles, fundamental concepts, and principles for the system development process that drive interpretations and actions in system development" (Iivari *et al.*, 1998).	Goals, Principles, Concepts
Method	"A method is regarded as a path or a procedure by which the developer proceeds from a problem of certain class to a solution of a certain class. The steps-of a method imposes some ordering on the decisions to be taken during development" (Jackson, 1981).	Procedure, Method
Method	"A method is a systematic process, technique, or mode of inquiry used to aid in the creation of a satisfactory software product" (Blum, 1994).	Process, Technique
Method	"A method is an approach to perform a systems development project, based on specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with the corresponding development product" (Brinkkemper, 1996).	Rules, School of Thought
Method	"A method is the term used for an orderly, predictable and universal approach to Information Systems Development" (Truex *et al.*, 2000).	Orderly, Predictable, Approach
Method	"A method is always based on particular approach (functional, data, object-oriented). A method always covers just part(s) of the IS development process or some point of view such as data, function and hardware" (Repa, 2004).	Phases, Process, Approach
Methodology	"A methodology is a set of recommended steps, approaches, rules, processes, documents, control procedures,	Steps,

	methods, techniques, and tools for the developers, which covers a whole life cycle of an Information System" (Repa, 2004).	Approach, Rules, Process, Procedure, Method, Techniques, Lifecycle
Methodology	"A methodology is the aim of breaking down complex problems into manageable units in a disciplined approach" (Hardy *et al.*, 1995).	Approach
Methodology	"A methodology acts to prescribe developers' and users' behaviour while aiming to make the intermediate results of this process more visible to aid managerial control" (Westrup, 1993).	Management control stages, processes
Methodology	"A methodology is collection of procedures, techniques, tools, and documentation aids, which will help the systems developers in their efforts to implement a new information system. A methodology will consist of phases, themselves consisting of sub-phases, which will guide the systems developers in their choice of the techniques that might be appropriate at each stage of the project and also help them plan, manage, control and evaluate information systems projects" (Avison & Fitzergerald, 2006:24).	Goals, guiding principles, fundamental concepts principles of ISD process
Methodology	"A Methodology is a recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation, management and training for developers of Information Systems" (Maddison *et al.*, 1983).	Philosophy, Phases, Procedure, Rules,

		Technique, Tools, Management, Training
Methodology	"A methodology comprises an overall strategy for computer-based information systems development that includes a flexible framework of the sequence of development tasks along with the techniques used to accomplish each task" (Roberts *et al.*, 1998).	Framework, Tasks, Techniques,
Methodology	"A methodology is a systematic procedure for completing either a system or one of several stages of the systems development life cycle. It consists of goals, principles, specific methods and tools, which are selected on the basis of an underlying rationale or system development philosophy" (Iivari *et al.*, 2000).	Procedure, Stages, Lifecycle, Goals, Principles, Methods, Tools
Methodology	"A methodology is a systematic procedure for completing for either a system or one of the several major stages of the systems development life cycle. It consists of goals, principles, specific methods and tools, which are selected on the basis of an underlying rationale or system development philosophy" (Wynekoop & Russo, 1993).	Stages, Lifecycle, Goals, Principles, Methods, Tools

Methodology	"A methodology can be interpreted as an organised collection of concepts, methods, beliefs, values and normative principles supported by material resources" (Hirschheim *et al.*, 1995).	Concepts, Methods, Beliefs, Values, Principles, Resources
Methodology	"A methodology is a systematic description, explanation and evaluation of all aspects of methodical system development" (Brinkkemper, 1996).	Systematic
Methodology	"A methodology is defined as an explicit way of structuring one's own thinking and action. It contains models and reflects a particular perspective of reality based on different philosophical paradigms. A methodology implies a time dependent sequence of thinking and action stages" (Jayaratna, 1990).	School of Thought, Philosophy, Stages
Methodology	"A methodology informs the assumptions and conceptual structure of tools, so that they operate consistently, in an integrated fashion, and in accordance to some common logic" (Friesen & Orlikowski, 1989).	Tools, Logic
Methodology	"A methodology is an organised and systematic approach for handling the system lifecycle or its parts. It will specify the individual task and their sequence" (Palvia & Nosek, 1993).	Approach, Lifecycle, Task

Table 3.1 has outlined inconsistencies in meaning that arises by defining approach, method and methodology. The table showed that when the definition method is used, only the process of how the development process will be conducted is emphasised. As for the definition of methodology, components such as the approach, method, process model, tools and techniques have to be included in completing the definition. In conclusion a methodology can be described as having an approach, method, process model, tools and techniques whilst a method only describes a particular development process.

According to Iivari and Maansaari (1998), a number of problems relating to the use of the terminology Systems Development Method were identified. The classification highlighted scope and category as the problem leading to inconsistencies. Avison and Fitzgerald (2006) argued that the terminology methodology holds a wider concept than method, as it has certain characteristics that are not implied by method and the inclusion of a philosophical view. Therefore we define a Systems Development Methodology as a combination of the following components:

Approach (es)

The underlying principle of Systems Development Methodology is built on the philosophical approach applied (Iivari, 1986). It is set on goals, guiding principles and beliefs, fundamental concepts and principles of the system development process driving the interpretation and action (Iivari *et al.*, 1998). The approach could be concerned with cost, quality, development speed and adaptability. It is the approaches that dictate the software development process that should be undertaken.

Process Model(s)

Models are used to help the software developer in understanding the order and sequence in which a system will evolve (Wynekoop & Russo, 1993).

Method(s)

A method is a step by step approach to completing a single phase of a systems development. The method follows a particular approach, which covers some part of the systems development process (Repa, 2004). It consists of a set of guidelines, activities, techniques and tools that is associated with a particular philosophical view of systems development (Wynekoop & Russo, 1993).

Tool(s) and Technique(s)

Techniques are procedures that are utilised when conducting the development process. They usually define the exact process of particular activities, the use of tools and decision options in a typical situation (Repa, 2004). They include best ways to evaluate costs and benefits of different solutions and methods required to formulate a detailed design for the proposed software application (Wynekoop & Russo, 1993). According to Brinkkemper (1996), notations are commonly referred to as techniques.

3.3 Some examples of Systems Development Methodologies

There are six different Systems Development Methodologies (SDMs) described in the section below. For each a philosophical approach related to that methodology is explained coupled with their process models, methods, tools and techniques. The categorisation of the methodologies is prescribed by Avison and Fitzergerald (2006).

3.3.1 Process Oriented Systems Development Methodologies

The basic approach of processing modelling is the technique of using functional decomposition. It involves the breaking down of a complex problem into small manageable chunks, which are easier to trace and debug (Avison & Fitzergerald, 2006).

According to Avison and Fitzergerald (2006), process modelling describes the logical analysis of processes and not their physical level designs. In the process oriented approach, users view of the data that requires processing and the format it is converted to when the processing takes place is totally different.

Structured Analysis, Design and Implementation of Information Systems (STRADIS)

The development of Structured Analysis, Design and Implementation of Information Systems (STRADIS) gets its origin from an earlier development of a structured approach to the designing of software.

STRADIS emphasises that the selection of organisation of program modules and interfaces that will be applied to solve a problem as predefined. STRADIS is said to involve the definition of the problem, and so it is regarded as having a practical limitation as development of an Information System (IS) that requires both the analysis and design aspects not to be overlooked (Avison & Fitzergerald, 2006).

STRADIS is the choice in the development of IS irrespective of their size of the system, the developers wish to develop and whether the system requires automation as well. The methodology is known to perform very well in situations where there is a backlog of systems waiting to be developed and minimal resources are needed to be allocated to the development of the new system (Avison & Fitzergerald, 2006).

Approach

The approach that STRADIS adheres to is a top-down, process-oriented approach.

Process Model

The process model of STRADIS describes a series of phases conducted in a linear fashion. It also has activities conducted in parallel such as those in the physical design phase.

Method

STRADIS consists of four Phases

- Initial Study

This phase involves gathering vital information from the users and documents to help the software developers determine the costs benefit analysis of the proposed system.

- Detailed Study

This phase is used to determine the requirements and interest of the system from a user perspective. It is these requirements that will help the software developer to produce a logical design of the current system.

- Defining and designing alternative solutions

This phase involves defining and designing alternative solutions by emphasising system objectives initially indicated. They utilise these objectives to create a logical data flow diagram (DFD) of the system.

- Physical design

This phase involves the selection of one of the alternative solutions proposed. Once an alternative solution has been selected, a physical design of the system can be created from the logical DFD.

Tools and Techniques

The technique used to model the system is a data flow diagram (DFD). This model is then stored into a data dictionary, which indicates the meaning of all the entities that have been described in the system. According to Avison and Fitzergerald (2006), decision trees, decision tables, structured english, structure diagrams, action diagrams and entity life cycles can be used in conjunction with DFDs.

3.3.2 Data Oriented Systems Development Methodologies

The philosophical underpinning of this methodology considers data to be at the heart of developing an Information System (IS). Data is said to be more stable than processes or procedures that act upon that data. The philosophical view also believes that communication through the use of diagrams is the most effective way to document requirements to the stakeholders during the software development (Avison & Fitzergerald, 2006).

Information Engineering (IE)

Information Engineering (IE) is a software development approach covering all the aspects of the developmental life cycle in more detail. Other developers consider the methodology to be a framework that has several techniques that could aid the development of a quality IS in an efficient manner. The methodology follows a top-down approach and considers the development of the system with a management overview as a whole (Avison & Fitzergerald, 2006).

Approach

This systems development methodology follows a blended (data oriented) approach by placing emphasis on both process and data of the system.

Process Model

This systems development methodology follows the classical waterfall model when developing a system.

Method

IE consists of four stages (Avison & Fitzergerald, 2006).

- Information strategy planning

This phase is concerned with adhering to the corporate objectives. The organisational IS should be designed to meet the corporate plan and should consider IS to have a strategic importance to the organisation.

- Business area analysis

This phase involves identifying the information strategy plan. It is at this stage that detailed data and functional analysis is conducted. This stage also involves maximum participation of end users.

- System planning and design

This phase is divided into business system design and technical design. The activities conducted during this phase are preliminary data structure design, system structure, procedure design, confirmation and planning for technical design.

- Construction and cutover

This phase involves implementation, preparation, rolling out the system and training. The system is also maintained to correct the bugs and addresses any changes as requested by the users.

Tools and Techniques

The techniques mainly applicable to this methodology are entity modelling and normalisation. Other techniques that can support this development process are decision trees, data flow diagrams and dependency diagrams (Laurido-Santos, 1986).

3.3.3 Object Oriented Systems Development Methodologies

The philosophical approach regarding this development process is conducted using object-oriented concepts. It is different from other models such as process and data modelling. The difference lies in the fact that modelling of data and processes is considered to be fundamental to building the system. The objective is to treat data and processes together as one by encapsulating them into an object. This aspect has resulted in benefits such as the unification of many aspects of the Information Systems (IS) development process. It also facilitates in the realistic reuse of software codes, making application development quicker and more robust (Avison & Fitzergerald, 2006).

Rational Unified Process (RUP)

According to Jacobson *et al.* (1992), software industries are migrating from the traditional developmental methods to a single modelling language called Unified Modelling Language (UML). RUP is a use case driven, architecture–centric process as it permits people to see the blue print of the software prior to it being built. Additionally, the process is considered to iterate and increment, as the user requirements cannot be fully stated at the commencement of the project (Jacobson *et al.*, 1992). RUP consists of phases and workflows that produce a result of observable values (Avison & Fitzergerald, 2006).

Approach

The systems development methodology follows an object-oriented approach.

Process Model

This systems development methodology follows an iterative and incremental approach.

Method

RUP has developmental activities that are conducted in cycles throughout the development life cycle of the methodology. RUP is composed of four phases.

- Inception

The goal and objectives of the project are clearly stipulated.

- Elaboration

This phase involves understanding the architecture of the software to be developed. The developers require extensive insight of the software requirement to appreciate the architecture.

- Construction

The software is built by joining several component module programs.

- Transition

Once the software is constructed, it is deployed for use. The four phases undergo a series of activities that produce results of observable value.

These activities are categorised in the nine workflows listed below:

- Business modelling workflow
- Requirements workflow
- Analysis and design workflow
- Implementation workflow
- Test workflow
- Deployment workflow
- Configuration and change management workflow
- Project management workflow
- Environment workflow

Tools and Techniques

Object-orientation and use cases are the main techniques applied in this systems development methodology.

3.3.4 Rapid Application Development (RAD) Oriented Systems Development Methodologies

The objective of RAD is speeding up the development process of software. As a result of rapidly changing business requirements, systems developers are forced to develop information systems more quickly in order for organisation to remain competitive in the business environment (Avison & Fitzergerald, 2006).

Extreme Programming (XP)

Extreme programming is chosen when small to medium-sized applications and organisations require supporting quicker development of software. According to Avison and Fitzergerald (2006), XP is regarded as a series of principles for developing software rapidly rather than by a step-by-step methodology. It is a preferred choice of software development methodology because it has the least complex rules and amount of documentation during development.

Approach

In this type of systems development methodology, speed of development is regarded as the main focus of the approach.

Process Model

The systems development methodology follows an iterative process.

Method

The methodology is suitable to software that is constantly subjected to changing business requirements. XP consist of four phases.

- Planning

This phase in responsible for stipulating the scope of the project, prioritising the functions, to conduct the amount of human resource required, the content of each increment, estimating financial cost and adhering to high quality of development. User stories are used for obtaining user requirements.

- Designing

The principle of simplicity is followed during the design. The stakeholders of the project discuss the best possible solution collectively resulting in a participative approach to software development.

- Developing the code

The source code is produced through paired programming. This approach involves using programmer and user to test the code, which is continuously integrated with already implemented code.

- Testing

This phase involves unit, integration and acceptance testing of the software. As opposed to other methodologies, documentation is not a priority to delivering the software product.

Tools and Techniques

Some of the applicable techniques to this systems development methodology are test-driven development, pair programming and continuous integration. Refactoring is a technique applied during the design phase.

3.3.5 People Oriented Systems Development Methodologies

The philosophical approach is believed to address organisational issues, which are fundamentally concerned with the process of change. The approach is known for encouraging user participation during the development process. This requires the involvement of all the stakeholders of the system throughout the development process of the system (Avison & Fitzergerald, 2006).

Effective Technical and Human Implementation of Computer based Systems (ETHICS)

This is a methodology that is said to apply an ethical stance when it comes to software development. The methodology uses a participative approach during development of Information Systems (IS) (Mumford, 1995). User participation during software design enhances user satisfaction and commitment to the software once implemented (Avison & Fitzergerald, 2006).

Approach

The methodology follows a people-oriented approach, with emphasis on social issues and satisfaction of the end-users of the software.

Process Model

The systems development methodology follows a linear process, but also suggests that some activities are conducted in parallel.

Method

The methodology of ETHICS consists of fifteen steps

- Why Change?
- System boundaries
- Description of existing system
- Define the key objectives and tasks
- Diagnosis of efficient needs
- Diagnosis of job satisfaction needs (questionnaire used to measure job satisfaction)
- Future analysis
- Specifying and weighting efficiency and job satisfaction needs and objectives
- The organisational design of the new system
- Technical options
- The preparation of detailed work design
- Implementation
- Evaluation

The organisational design of the new systems and the technical options are conducted in parallel.

Tools and Techniques

Stakeholder analysis is the main technique employed in ETHICS.

3.3.6 Organisational Oriented Systems Development Methodologies

The philosophical approach states that information systems have to be developed for the widest possible context such as an organisation. And since organisations are regarded as open systems, their relationship with the environment is vital (Avison & Fitzergerald, 2006).

Soft System Methodology (SSM)

Checkland and Scholes (1999) consider SSM as both an approach and methodology. The main feature of this methodology is its focus on problem formulation which helps the users to identify relevant systems from the perceptions of possibly disagreeing with stakeholders. Additional aspects that require consideration when developing a system for an organisation as a whole is the acknowledgement of the importance of the people of the organisation. When applying soft system thinking, objectives of the system are considered to be more complex than simple goals that could easily be achieved and measured.

Approach

This methodology follows an organisational-oriented approach. The approach does not only consider the system that is being developed but also the organisational environment it will be affecting.

Process Model

In this methodology, some stages are conducted simultaneously. Iteration and backtracking are also important.

Method

SSM consists of seven stages.

- Unstructured problem situation
- Problem situation expressed
- Root definition expressed
- Conceptual model
- Comparison of conceptual model with reality
- Feasibility and desirable changes

- Action to improve the problem situation

Tools and Techniques

There are no specific tools and techniques that should be used during the development process.

3.4 Current directions in Systems Development Methodologies

From the mid to the late 1990s, organisations have questioned the level of effectiveness in the use of proposed methodologies as an aid in addressing their software development problems. As a result some organisations have turned to different methodologies and approaches while others have abandoned their use of methodologies completely (Avison & Fitzgerald, 2006).

Organisations introduced the use of methodologies as means of addressing the problems faced with applying the traditional development approaches. To maintain a competitive edge over its rival companies, organisations were eager to find better ways to control software projects finances, human resources and product quality. Their adoption of methodologies was seen as providing a disciplined approach to project management for organisations. For other organisations the use of the methodologies provided unsatisfactory results (Avison & Fitzgerald, 2006).

Some organisations have rejected the use of methodology altogether and are returning to less formal and flexible approaches. The developers adopt an approach to only use part of the methodology they can understand. While for others, it is not the concept of methodology that is the problem, but simply the inadequacy of current methodologies. It is believed that methodologies will continue to be developed and that existing ones will evolve. Additionally, it is unlikely that any single developmental approach will provide the solution to all the problems of information systems development. Then, most methodologies are developed for situations, which follow a stated or non-stated ideal type. We therefore see a contingency approach to Information Systems developed being adopted. Lastly, we see a movement towards external development in a variety of ways namely, through the use of packages and outsourcing. Some organisations are attempting to satisfy their systems needs by purchasing packages from the market place, as a quicker and more cost effective way of implementing systems for organisations that have a fair standard. Some of the advantages and disadvantages of methodologies can be found in the literature described by Fitzgerald *et al.* (2002).

3.5 Summary of the chapter

The chapter has covered the background literature regarding the misunderstandings in the use of the terminologies such as approach, method and methodology. The systems development methodologies described in the chapter will act as a major source of the terminology or data for the development of the ontology for ISDMs. Other sources of data for the ontology will be gathered from interviews conducted with a software development practitioner and three academics. The next chapter discusses the research strategies employed in this research.

CHAPTER FOUR
RESEARCH METHODS AND DESIGN

4.1 Introduction

Research methodology in a dissertation helps the researcher to produce reasoning behind the study and place the emphasis on its selection. It is always ideal for a researcher to use rich literature on a methodology to inform the argument (Henning, 2007). In this chapter the concept of research, its importance, research approaches, the interpretive research paradigm, the narrative research design, narrative data collection strategies and methods, narrative data analysis methods, and the research design and ethical considerations are described. The chapter is concluded with a summary.

4.2 What is Research?

Smith and Dainty (1991) regards research as pertaining to problem solving, investigating relationships and building a body of knowledge. For a researcher to undertake a particular piece of research, there needs to be full comprehension of the assumption surrounding the research framework (Smith & Dainty 1991). The researcher must be aware of the merits and demerits surrounding the research activities in order to attain a successful outcome.

4.3 Why is Research Important?

Researchers must have an understanding and appreciation of the research studies being undertaken. Most people regard research as a topic only employed by colleges and universities. It is however also regarded to have importance in other academic settings (Creswell, 2008). The importance of research lies in its ability to improve societal issues, research practices and creating debates amoung academic topics (Creswell, 2008).

4.4 Research Approach

According to Smith and Dainty (1991), several research approaches have been noted to exist. Their existence is based on enquirer's values, assumptions and perceptions of the world. Academics in research have distinctively categorised quantitative and qualitative research as the possible forms of approaches that can be undertaken.

4.4.1 Quantitative Research

According to Creswell (2008), quantitative research stems its origin from the physical sciences. It is believed that people's behavioural pattern in the world can also be related to atom's laws and axioms. The subsequent theory led earlier researchers to begin identifying behavioural patterns, assess individual abilities and scoring and conducting extensive surveys.

Researcher's conducting studies in a natural setting are faced with the challenge of completely proving causes and effects. As a result, control and confounding variables play a vital role in when conducting a research. The former serves a purpose in quantitative researches, whilst the latter eliminates and destroys the experiments validity (Creswell, 2008). The essence of this type of research is to make a declaration on what results the researcher is expected to tabulated (Creswell, 2008).

Creswell (2008) also indicates that during this type of studies, researchers place their theory at the beginning of the proposal and using it to support the reasoning behind the research. The theory becomes the framework for the entire study which is tested from it. The researcher then applies an instrument to measure the behavioural pattern of the participants in the study. The researcher then uses the results to confirm or disconfirm the theory. Quantitative research is numerical in nature. It comprises numbers, questionnaires and structural interviews. This type of research can only be measured accurately by expressing it in numbers.

4.4.2 Qualitative Research

Qualitative research cuts across numerous disciplines, fields and subject matters (Denzin & Lincoln, 2005). Qualitative research is a situation activity that locates the observer in the world. This research comprises of interpretive material that is represented using interviews, notes, photographs and conversations. Qualitative researchers conduct their studies in a natural setting to allow them to make behavioural interpretation of the people under study (Denzin & Lincoln, 2005).

According to Jones (2004), qualitative source of material in a qualitative research setting ranges from description, observation and unstructured interviews to information from written sources. Taylor *et al.* (1995) also indicates that qualitative research provides a detailed picture of human social lives. The next section provides a description of the types of research paradigms a researcher can use.

4.5 Research Paradigms

There are three popular research paradigms that researchers can adopt when conducting research. The choice of the paradigm is usually dependent on the type of research to be undertaken. The paradigms comprise of positivistic, interpretive and critical research types. Research is regarded as a scientific means of creating true and objective knowledge. It is the researcher's responsibility to conduct data collection and form a comprehension of the study and subsequently make a conclusion. In the sections below, the three research paradigms are described.

4.5.1 Positivistic Research

To conduct a positivistic research, researcher's ability to observe and measure data is highly taken into account. This paradigm has its roots from the natural sciences which have now been transferred to social sciences. The general rule to applying this paradigm is to ensure that the universal laws governing social activities are always adhered to (Wardlow, 1989).

According to Orlikowski and Baroudi (1991:5), Information Systems (IS) research is regarded as being positivistic "if there is evidence of formal propositions, quantifiable measure of variables, hypothesis testing, and the drawing of influences to a fact from the sample to be a stated population".

This paradigm had its origin in the 19th century from natural sciences to social phenomena (Smith, 1983). For a research to qualify as being positivistic, knowledge and truth should be related to an external referent reality (Smith, 1983). In this approach, the use of empirical methods has an influence in dictating the logical structure of the investigation as well as testing it. The objective of this research is to exclude the involvement of the researcher during the study by ensuring the results can be replicated to the other researches (Walker & Evers, 1999).

The following description below provides the suitabilities and limitations of a positivistic research (Wardlow, 1989):

Suitabilities of Positivistic Research

- It is applicable to researchers that require valid and reliable knowledge as a set of universal principles that can explain, predict and control human behaviour across individuals and organisations.

- The physical world and social events are analogous in that researchers can study social phenomena as they do physical phenomena.

- In examining social events, researchers adhere to subject-object dualism in that they stand apart from their research subjects and treat them as having an independent existence.

- If there is a need to formalise knowledge using theories and variables that are operationally distinct from each other and defined accordingly.

- It is applicable when the hypotheses about principles of theories are tested by the quantification of observations and by the use of statistical analysis.

Limitations of Positivistic Research

- Influential contextual factors in organisations can be ignored by methods aiming to draw casual inferences through examining only phenomena that are readily observable.

- Another inherent limitation is that truth in the positivistic tradition is often stated probabilistically. To this extent, these researchers can seldom achieve their own goals of having specific truth but only probabilistic inferences of truth in which theory never becomes regarded as fact.

- The final limitation lies in the inherent constraints of the positivistic method in measuring phenomena that are by their very nature subjective.

4.5.2 Critical Research

The critical research paradigm is another means of conducting research. Popper (1963) states that critical research applies in situations where the researcher's objective is to identify imperfections in theories and their effects. The philosophy behind the paradigm is to make practical differences between old philosophies and their inabilities to address current situations (Marx, 1964). With respect to conducting Information Systems (IS) researches, a similar philosophy from the critical research is applicable (Klein & Myers, 1999).

According to Hill (2001), the following are the suitabilities and limitations of critical research:

Suitabilities of Critical Research

- It is suitable in providing a systematic way of accessing the validity results and usefulness of published research papers.

- Together with skills in finding research, it is the route to closing the gap between research and practice and as such make contributions to current situations.

- Critical approach skills are not difficult to develop, as it is the common sense approach to reading.

- It encourages objective assessment of the usefulness of information.

Limitations of Critical Research

- The critical approach can be time consuming initially although. with time becomes the automatic way to look at research papers.

- Critical researchers may be biased in providing a reasonable conclusion to the investigation.

- Critical approach can be disrupting if it highlights a lack of good evidence. It may take determination to persist with an area of interest when access to good research in the area is limited.

4.5.3 Interpretive Research

Another research paradigm applicable to the research arena is interpretive in nature. The interpretive research philosophy is based on hermeneutics and phenomenology (Boland, 1985). The focus of this research type lies in the creation of human sense as it emerges (Kaplan & Maxwell, 1994).

The findings by Klein and Myers (1999) indicate that the origin of interpretive research emanated from social constructions such as languages, consciousness, and shared meanings. The paradigm emphasises on the use of qualitative methods to ensure the researcher understands the data collected and analysed during the study (Klein & Myers, 1999).

Case studies and ethnographies are examples of interpretive researches. Klein and Myers (1999) indicate that it is because of this type of research that researchers continually seek better ways to conduct researchers. This method is fundamental in conducting and evaluating research in a hermeneutic nature. It also helps validate academic and practitioners sourced literature by using hierarchies of evidence. The interpretive research paradigm is applied in this research.

According to Klein and Myers (1999), the following are considered the suitabilities and limitations of interpretive research:

Suitabilities of Interpretive Research

- For the researcher to have better comprehension on the information that has been gathered, it is the social and historical context in which the study is being conducted that requires appreciation.

- For the research conducted to be considered valid there needs to be convincing evidence documented between the research and the final conclusion. When conducting this research it is imperative that research concepts are adhered to in order to obtain a positive feedback.

Limitations of Interpretive Research

- Such researches are usually biased so the legitimacy of the study is uncertain.

- It is stated that although behaviour patterns may be the result of the meanings that individuals attach to a situation, these individuals maybe falsely conscious. There may be an objective perspective which is different from that of the individuals themselves.

The next section provides a description of the types of interpretive research methods a researcher can apply.

4.6 Interpretive Research Methods

The research methods are mainly concerned with the design logic that the researcher will have to follow in order to document his inquiry in the research. It indicates the methodological requirements of the research question, which helps the researcher to decide on the appropriate type of data to extract and the means of processing it. The researcher should have a comprehensive understanding of methods, methodologies and the main theorists in that field of inquiry, and collect the most recent literature about it (Henning, 2007). The following methods e.g. case studies, ethnographic studies, grounded theory method, action research and narrative are some of the research methods that can be chosen when conducting an interpretive research.

4.6.1 Case Study

According to Creswell (2008), case studies are a useful means of conducting qualitative inquires. Most researchers performing case work provide their studies with different names. The name case study is important because it directs the attention to the question: 'what specifically can be learned about the single case'? Research communities regard case studies as experiential knowledge of case closely influenced by social or political context. Fals Borda (1998) regards a case as being a bounded system. In the social sciences and human services, most cases have working parts and purposes.

Ragin and Becker (1992) has emphasised the question: 'what is it a case of?' as if membership in, or representation of something else were the main consideration in a case study. Ragin and Becker (1992) were writing for the social scientist seeking theoretical generalisation, justifying the study of a particular only if it serves an understanding of grand issues or explanation.

4.6.2 Ethnography

According to Creswell (2008), ethnographic designs are described as qualitative research procedures analysing and interpreting a culture-sharing group's patterns of behaviour, beliefs and language developed over time. Culture is central to the definition of ethnography. A culture is "everything having to do with human behaviour and belief" (LeCompte *et al.*, 1993:5). It includes language, rituals, economic and political structures, life stages, interactions and communication styles. Ethnographers typically spend a considerable amount of time in the field interviewing, observing and gathering documents about the group sharing behaviours, beliefs, and language to understand the patterns of a culture-sharing group (Creswell, 2008). Ethnographic researches are conducted to provide an understanding of shared cultural issues relating to a group that lives together for a particular time period (Creswell, 2008).

Clifford and Marcus (1986) raised two issues that have raised much attention in ethnography in general. The first is the crisis of representation consisting of reassessment of how ethnographers interpret the group they study. And secondly, the crisis regarding its legitimacy. Researchers need to evaluate each ethnographic study in terms of flexible standards embedded within the participants' lives, historical and cultural influences and interactive forces of race, gender and class.

4.6.3 Grounded Theory Method

According to Creswell (2008), the most systematic manner to begin a theory from a wide conceptual level would be to apply the grounded theory method. Researchers applying this method have to follow s systematic procedure to relate various types of data to help the researcher come up with a theory following the research process.

Additionally, Creswell (2008) states that grounded theory method generates a theory when existing theories do not address your problem or the participants that you plan to study. The significance of this method lies in its greater explanation of the data.

Glaser and Strauss (1967) are regarded as the founders of grounded theory method. It is their publications that have created the backbone for the significant ideas applicable to grounded theory today.

The author Charmaz (2000) also founded another grounded theory method called the constructivist method. Charmaz (2000) was of the opinion that Glaser and Strauss (1967) approach to grounded theory was too systematic in their procedures. Her emphasis on the creation of grounded theory was to have participants that could ascribe to situations, acknowledge the roles of the researcher and the individuals being researched, and expand philosophically beyond a quantitative orientation to research.

4.6.4 Action Research

According to Mills (2000), action research has an applied focus. It uses data collected through the quantitative or qualitative method or both. This research design is suitable in situations involving researchers collecting information and subsequently improving, the way their setting operates. Action researches are useful when you have a specific problem to solve.

According to Creswell (2008), development stages of Action Research involves societal issues being addressed by involving practitioners in providing solutions to the problem and ensuring the participants assure responsibility for the changes made.

4.6.5 Narrative

The narrative research method involves collecting and narrating individual life experiences. This research method is distinct from other qualitative researches as its focus centres on collecting and reporting stories about individual experiences (Connelly & Clandinin, 1990).

According to Creswell (2008), narrative research is applicable in cases were individuals show willingness to narrate and researchers having the wish to report their stories. This kind of research creates a relationship between the researcher and the participants. Narrative researches are regarded to capture every day, normal form of stories relating to individuals. According to Casey (1995), narrative research can be divided into numerous forms. It is important for the researcher to identify which type needs to be applied.

According to Creswell (2008), the seven characteristics described below are central to narrative research:

Individual Experiences

The focus of narrative researchers usually involves the participation of one or more people. The researcher's intension is to explore the experiences of each individual.

Chronology of the Experiences

The researcher has to be aware of the participants past, present and future when conducting a narrative research. The researcher documents the participant's experiences in a chronological order.

Collecting Individual Stories

To construct a chronological account of the events, participants are requested to narrate their experiences.

Restorying

The researchers then retell the story according to their understanding. The stories are analysed for key elements and later re-arranged chronologically.

Coding for Themes

The researcher then segments the data into themes to add an insight regarding the individual's experiences.

Context and Setting

In the restorying of participant's story and the telling of the themes, the narrative researcher includes rich detail about the setting or context of the participant's experiences.

Collaborating with Participants

Throughout the research process, the researcher and the participants are actively involved in unfolding the inquiry. This is done in order to minimize the narrative told and narrative reported.

The following are steps to take when conducting narrative research (Creswell, 2008):

Identify a phenomenon to explore that address a problem

This step involves identifying the problems that need to be studied.

Purposefully select an individual from whom you can learn about the phenomenon

You then identify individuals capable of providing and understanding of the phenomenon particularly those with experience on specific issues.

Collect the story from that individual

You have to collect the participant's experiences through personal conversation.

Restory or retell the individual's story

You then analyse the data collected by retelling it in your own words. The story can then be arranged in a chronological order to highlight the individual's experience.

Collaborate with the participant-story-teller

You then actively cooperate with the participants to capture their experiences throughout the research process.

Write a story about the participant's experience

The researcher then writes and presents the story of the participant's experiences.

Validate the Accuracy of the Report

The author then validates the accuracy of the story by closely cooperating with the participants throughout the research process.

The next section provides a description on the types of narrative data collection strategies and methods that a researcher can use.

4.7 Narrative Data Collection Strategies and Methods

There are several data collection strategies that a researcher can utilise when conducting a research. The three forms of strategies are experiments, which allow strong casual inferences, surveys involving data collection from a large sample of respondents, and applying qualitative research design by using data collected through interviews, observation of focus groups (Hox & Boeije, 2005).

4.7.1 Narrative Data Collection Strategies

Primary Data Collection

Each time data is collected to address a specific research problem is an addition to the already existing storage of social knowledge (Hox & Boeije, 2005). The most commonly used data collection strategy is experiments. In such a case, the researcher applies a planned design to manipulate independent variables and observes the outcome of these variables (Hox & Boeije, 2005). The other primary data collection strategy available is through interview survey. In this survey, the researcher collects data by observing the experiences of the population (Hox & Boeije, 2005). Researchers can also use focus groups, documents, photographs, films and videos as a data collection strategy.

According to Hox and Boeije (2005), the following are the advantages and disadvantages of primary data collection:

Advantages

- The most important advantage of collecting one's own data is that the operation of the theoretical information constructs the research design.
- The data collected can be tailored to answering the research questions.

Disadvantage

- Data collection is costly and time consuming.

Secondary Data Collection

The data that is made available to the general research community for use is called secondary data. Secondary data can be used for conducting comparative research, re-analysis or methodological advancements. Researchers can access secondary information archived via the internet or on data discs (Hox & Boeije, 2005).

According to Hox and Boeije (2005), the following are the advantages and disadvantages of secondary data collection:

Advantages

- If relevant information on the research topic is accessible reusing its gains benefits, the data can serve to answer the newly formulated research questions.
- Secondary data is less costly and faster to obtain.

Disadvantages

- Interpreting data originally collected for a different purpose is difficult.
- Researchers must locate the data sources, which may be useful given their own research problem.
- The researcher must be articulate enough to obtain data that is important to the research.
- The researcher must analyse the data to meet the need of the current study.

4.7.2 Narrative Data Collection Methods

Observation

According to Creswell (2008), observation is a process of gathering open-ended, firsthand information by observing people and places at a research site. When observing, the researcher has to play the role of the observer. The role as an observer usually varies depending on the relationship with the participants.

According to Denzin and Lincoln (2005), the following are the advantages and disadvantages of observation:

Advantages

- Researchers get a personal interaction with the interviewees.
- Researchers can document the observations as and when they happen.
- The researcher makes extraordinary observations.
- Useful in exploring topics that may be uncomfortable for participants to discuss.

Disadvantages

- A researcher is perceived to be intruding in the participant's lives.
- The researcher may come across classified information that is not permitted in the investigation.

- Researchers may have poor listening and observation abilities.
- Some participants may highlight delicate issues in order to establish a better relationship.

Interviews

According to Creswell (2008), an interview involves one or more persons being asked questions and a researcher documenting their responses. The researcher may ask open-ended questions in-order to allow the participant to give unconstrained responses.

The following are the types of interviews that are applied in a qualitative research (Creswell, 2008):

One-on-One Interview

The interview conducted on a one-on-one basis involves the researcher posing questions and then recording responses. The role of the participant is respond to the researcher's questions.

Focus Group Interview

Researchers use focus group interviews to collect data from a group of people by asking them a small amount of general question. The researchers await responses from all the individuals in the group. Focus group interviews are useful when a researcher has limited amount of time to collect information from several individuals.

Telephone Interview

When it is practically impossible to interview participants due to geographical constraints, telephone interviews are conduct as a form of data collection. The major drawback with such a form of interviewing is in the loss of physical contact between the interview participants.

The following are the steps to conducting interviews (Creswell, 2008):

- Choose the people you will interview.
- Decide on the interview questions you will use to obtain the appropriate information from the interviewees.

- When conducting the interviews, ensure you record them to verify the conversations.
- You must also write notes during the interview in case the interview recordings are not audible.
- Conduct the interview in a quiet environment.
- Seek permission from the interview participants and notify them about the purpose of your study.
- Be flexible and concise when conducting the interview.
- Use probes to obtain additional information.
- Conduct your interview in a respectful and professional manner.

According to Denzin and Lincoln (2005), the following are the advantages and disadvantages of interviews:

Advantages

- Useful when participants cannot be directly observed.
- Past information can be obtained from the participants.
- The researcher is in charge of directing the questions to conduct the investigation.

Disadvantages

- Gives secondary information in the view of the interviewees.
- Information is obtained from selected places than ordinary environment.
- The presence of the researcher may lead the participants to provide one sided responses.
- Some people may not be clear and understandable.

Document Review

The use of documents is also considered to be a valuable source of information for qualitative research. The sources include newspapers, minutes of meetings, personal journals and letters. Document reviews has been applied as a data collection method for this research (Creswell, 2008).

The following are useful guidelines to collecting documents in a qualitative research (Creswell, 2008):

- Be aware of the documents that will contain the information to help address your research.
- Collect information from public and private documents.
- Ensure you have authorisation for the use of the documents once you have located them.
- If you ask participants to keep a journal, ensure you give them a guide line on how to write properly and its importance.
- Once you have authorisation to use the documents, analyse them to check for authenticity and relevance to the research.
- Document the collected information for future reference.

According to Denzin and Lincoln (2005), the following are the advantages and disadvantages of document review:

Advantages

- The researcher can use the documented verbal communication as data.
- Documents can easily be made available to the researcher.
- The data is well concentrated on topic under study.
- It liberates the researcher from transcribing.

Disadvantages

- Not every person can express themselves effectively.
- The information may be prohibited from access.
- Some information may be daunting to come across.
- Information may require translating before it can be captured electronically.
- Partial information may be provided.
- The material provided may not be genuine and correct.

4.8 Narrative Data Analysis Methods

Once the data gathering process of interviews and document review have been completed using the narrative data collection strategies and methods described in sections 4.7.1 and 4.7.2 respectively, the data analysis process may commence. The analysis of the data effectively commences with the research design, literature review and theory formation (McMicheal, n.d.). The analysis process plays an essential role in the researcher obtaining an in-depth understanding of the research under study.

4.8.1 Discourse Analysis

According to Truex (1996), discourse analysis involves exchanging meaning between the researcher and participant face to face. The face to face interaction allows for non verbal cues to exist thereby minimising ambiguity during the analysis.

In discourse analysis, the analysis of an object or phenomena is entrenched in a larger context, as subjective meaning of discourse would cease to exist isolated from context (Lennox, 2009). Discourse analysis believes that the context of the research is central to research methods and results (Lennox, 2009). Discourse analysis seeks to uncover social realities that are constructed, circulated and contested through interrelated bodies of text (Hardy *et al.*, 2004).

Discourse analysis as a form of analysis exhibits the following qualities (Truex, 1996):

- Having spontaneous ideas that are added or removed during the analysis.
- Has a life of its own, may be informal, inconclusive and semi-structured.
- Does not conform to usual grammatical structures of writing.
- Is a shared, social event.
- Can be ambiguous.

4.8.2 Narrative Analysis

According to Henning (2007), narrative analysis may be seen as a specialised form of discourse analysis because it searches for ways participants make sense of their lives by representing them in story form. Narratives have a structure known as a story grammar, and it is this natural form of expression and representation that intrigues the narrative analyst in social sciences. Postmodernism sometimes uses the terms narrative and text metaphorically. Text is not only written artefacts but also other kinds of human artefacts such as organisations, a stream of human discourse or social interaction (Truex, 1996).

4.8.3 Content Analysis

According to Cole (1988), content analysis involves analysing written, verbal visual communication. By using content analysis, the researcher is provided with a new insight of the knowledge provided by refining fewer related words. The aim is to attain a condensed and broad description of the phenomenon, and the outcome of the analysis its concepts or categories describing the phenomenon.

Content analysis can be used to develop an understanding of the meaning of communication (Cavanagh, 1997) and to identify critical processes (Lederman, 1991). Content analysis is applied as a data analysis method for this research.

The next section provides a description of the research design that is employed in this research.

4.9 Research Design

The research design describes manner in which this research has been conducted. The research involved conducting interviews with a software development practitioner, three academics and conducting an in-depth study of the literature on ontologies and Systems Development Methodologies. The four interview respondents were selected to provide useful concepts that are used in the development of the ontology for Information Systems Development Methodologies. The type of data that required collecting for the research was primary data from the interview and secondary data through document review. In the section below the research paradigm, research approach, research method, interpretive research method, narrative data collecting strategies and method, narrative data analysis method and ethical considerations are described.

4.9.1 Research Approach

The research approach involves the closely scrutinising of numerous amounts of non numeric data and rewriting it in order to address the problem of the research indicated.

4.9.2 Research Paradigm

The research paradigm for this research is interpretive in nature. This approach will be most appropriate as it allows the researcher to realise that what researchers in the past could have reported on are likely to have flaws and hence findings are subject to revision. This paradigm emphasises understanding of what has been reported as truth by encouraging obtaining a variety of data from various sources in order to prove its validity (Popper, 1963).

4.9.3 Interpretive Research Method

The interpretive research method used in this research is narrative. It will involve obtaining an account of stories relating to the subject matter. The literature from different sources will be gathered and interpreted in order to address the research question in this research. The literature for the study was gathered from textbooks and academic journals.

4.9.4 Narrative Data Collection Strategies and Methods

The data collection strategies used is both primary and secondary. In the primary data collection strategy, interviews were conducted with one software development practitioners and three academicians. The interview questions that were used in this research were designed by another researcher. Open ended questions were used to elicit information from the interview participants. The questions allowed the respondents to provide unrestricted perspective of their experiences. Then, the secondary data collection strategy involved conducting document review on the literature relating to Systems Development Methodologies. The detailed description relating to the data collection strategies have been documented in section 4.7.1. The interviews were conducted with four respondents. The interviews conducted lasted between forty and sixty minutes and twenty six questions were asked in order to aid the elicitation for SD terminologies used in the development of the ontology for ISDM. The questions asked in the interview are available in Appendix B. The other data gathered for the research came from the literature review on Systems Development Methodologies from textbooks and academic journals.

4.9.5 Data Analysis Method

The data analysis method applied in this research is content analysis. Content analysis is a method of analysing written, verbal or visual communication messages. It is also known as a method of analysing documents. To analyse the data, the concepts elicited from the interview were compared to the definition and categorisation process prescribed by Avison and Fitzegerald (2006) to identify the inconsistencies in the use of the terminologies by the interview respondents.

4.10 Ethical Considerations

Since this research has to use human beings to elicit information to help conduct the research, it is necessary to have fully have comprehension of the ethical responsibilities of conducting this research. The ethical codes ensure that the anonymity and confidentiality of the research participants are preserved. This ensures that names of the participants are not shown, but instead numbers are used to identify them. Since the interview questions and data were reused from a research done by another researcher, consent had to be obtained from the interview participants. The researcher convinced the participants that their confidentiality would also be respected and that their responses were going to be used purely for the purpose of this research.

4.11 Summary of the chapter

The chapter describes the research approach employed. Additionally the research method employed was described. The two data collection methods employed was interviews and document reviews. These methods both acted as major sources of the data for the development of the ontology for ISDMs. After the data was gathered, it was analysed in order to come up with agreed upon terminology that would be useful in the development of the ontology for ISDMs.

CHAPTER FIVE
DATA ANALYSIS AND INTERPRETATION

5.1 Introduction

The findings discovered in chapter three will act as a guide for the analysis of the primary data collected from the literature review. This chapter involves data analysis of interviews conducted with the four respondents. The chapter has a description on the interview responses, analysis and interpretation of data, summary of the data analysis.

5.2 Interview Responses for concepts of ISDMs

The analysis for this research has been conducted by collecting terminologies from interviews coupled with data gathering from the literature review. The analysed data was then used in developing the ontology for Information Systems Development Methodologies (ISDMs). Four respondents were selected to provide vital knowledge about terminologies that are used in Systems Development. The respondents were asked twenty six questions with the aim of eliciting the terminologies required to build the ontology. The interviews conducted were recorded and transcribed. The interviews identified the inconsistent use of the Systems Development terminology by the four respondents during their interviews.

5.3 Analysis and Interpretation of Data

To conduct the analysis, a terminology collected from the four respondents was tested for inconsistencies against the definition of ISDM as prescribed by Avison and Fitzgerald (2006). For a definition of methodology to be recognised as being consistent, the following components have to be present in the definition, namely the *philosophical approach, method, process model, tools* and *techniques*. These terminologies gathered through the interviews will be depicted as classes when developing the ontologies in Protégé. The detailed descriptions of the components of an ontology are described in section 2.4.1.

The interviews held with the four respondents indicated the number of times the SD terminologies was used, and whether the terminologies were used consistently or inconsistently with the definition as prescribed by Avison and Fitzgerald (2006).

The headings of the tables for the interviews conducted correspond to the following meanings:

Concepts: This is a particular word or expression to name or describe something.

Frequency: This means the number of times the words is being referred to during the interview.

Condition: This is looking whether the manner in which the term has been used is consistent or inconsistent with the classification.

Category: This means the terms being mentioned consistently or inconsistently in the interview belong to the stated group of concepts.

Table 5.1: A table showing the ISDM terminologies identified in interview one

Concepts	Frequency	Condition	Category
Waterfall Approach	1	Inconsistent use	Process Model
Agile Approach	1		Approach
Structured Methodology	4	Inconsistent use	Approach
Object-Oriented Paradigm	1		Approach
Model	15		Process Model
Software Development Process	5		Process Model
Sequence	1		Process Model
Mind Mapping	1		Tool
Joint Application Development	1		Technique
Collaborative Approach	1	Inconsistent use	Technique
Case Tools	6		Tool
Adaptive Software Development	5	Inconsistent use	Process Model
Evolving Software Products	3	Inconsistent	Process Model

		use	
Scope	3		Project Management
Configuration Management	2		Project Management
Stakeholders	1		Technique
Project	11		Project Management
Time box Approach	6	Inconsistent use	Technique

In interview one, several inconsistencies in the use of the SD terminologies were identified. The following inconsistencies in the use of terminologies are indicated below (for the terminologies gathered in interview one see table 5.1):

- Waterfall Approach is inconsistently used instead of Process Model.
- Structured Methodology is inconsistently used instead of Philosophical Approach.
- Collaborative Approach is inconsistently used instead of Technique.
- Adaptive Software Products is inconsistently used instead of Process Model.
- Evolving Software Products is inconsistently used instead of Process Model.
- Time box Approach is inconsistently used instead of Technique.

Table 5.2: A table showing the ISDM terminologies identified in interview two

Concepts	Frequency	Condition	Category
Software Development Process	2		Process Model
Object-Oriented Paradigm	7		Approach
Unified Modelling Language	3		Technique
Use Cases	5		Technique
Incremental Development	1	Inconsistent use	Process Model
Iteration	1		Process Model
Agile Methodologies	2		Approach
Extreme Programming	1		Method
Adaptive Software Products	2	Inconsistent use	Method

Evolving Software Products	1	Inconsistent use	Method
Scope	6		Project Management
Project	10		Project Management

In interview two, several inconsistencies in the use of the SD terminologies were also identified. The following inconsistencies in the use of terminologies are indicated below (for the terminologies gathered in interview two see table 5.2):

- Incremental Development is inconsistently used instead of Process Model.
- Adaptive Software Products is inconsistently used instead of Process Model.
- Evolving Software Products is inconsistently used instead of Process Model.

Table 5.3: A table showing the ISDM terminologies identified in interview three

Concepts	**Frequency**	**Condition**	**Category**
System Development Process	1		Process Model
Agile Methodologies	4		Approach
Extreme Programming	2		Method
Paired Programming	2		Technique
Brain Storming	1		Tool
Project	12		Project Management
White Boards	5		Tools
Metaphor	4		Tools
Data Modelling	1		Technique
Object-Oriented Paradigm	3		Approach
Unified Modelling Language	3		Technique
Use Cases	4		Technique
Data Flow Diagrams	4		Technique
Entity Relationship Diagram	1		Technique
Interactive Development	1	Inconsistent use	Technique
Iterations	6		Process Model
Evolving Software Products	1	Inconsistent use	Method

In interview three, several inconsistencies in the use of the SD terminologies were also identified. The following inconsistencies in the use of terminologies are indicated below (for the concepts collected in interview three see table 5.3):

- Incremental Development is inconsistently used instead of Process Model.
- Evolving Software Products is inconsistently used instead of Process Model.

Table 5.4: A table showing the ISDM terminologies identified in interview four

Concepts	Frequency	Condition	Category
Software Development Process	1		Process Model
Project	12		Project Management
Contingency Approach	1	Inconsistent use	Method
Iterative	1		Process Model
Rapid Application Development	3		Method
Agile Methodologies	5		Method
Use Cases	2		Technique
Unified Modelling Language	2		Technique
Entity Relationship Diagrams	3		Technique
Data Flow Diagrams	1		Technique
Object-Oriented Paradigm	4		Approach
Adaptive Software Products	8	Inconsistent use	Method
Evolutionary Development	2	Inconsistent use	Method
Incremental Development	1	Inconsistent use	Process Model
Extreme Programming	1		Method
Project	12		Project Management

In interview four, several inconsistencies in the use of the SD terminologies were also identified. The following inconsistencies in the use of terminologies are indicated below (for the concepts collected in interview four see table 5.4):

- Incremental Development is inconsistently used instead of Process Model.
- Adaptive Software Products is inconsistently used instead of Process Model.
- Evolving Software Products is inconsistently used instead of Process Model.
- Contingency Approach is inconsistently used instead of Method.

5.4 Summary of the chapter

The analysis indicated that there were a lot of inconsistencies in the use of the terminologies. The inconsistencies identified showed a clear need to develop an ontology. The inconsistencies in the use of SD terminologies during the interviews were identified by comparing them to the definition for Systems Development Methodologies as prescribed by Avison and Fitzgerald (2006). It is these terminologies that will be used as input data for the creation of the ontology for Information Systems Development Methodologies. Having elicited the terminologies required for the development of the ontology for ISDMs, the next chapter discusses how the ontology was designed and implemented using Protégé as the ontology editor.

CHAPTER SIX
ONTOLOGY DESIGN

6.1 Introduction

After the analysis of the data collected from the interview respondents, data was sufficient to develop the ontology. This chapter covers the design of the ontology, the steps to follow when developing an ontology, instruction on how to build a new ontology, and the created ontology for Information Systems Development Methodologies (ISDMs). The ontology for ISDMs that has been developed is a Prototype. The ontology developed focused mainly on the management side of ontology use, and hence has limited emphasis on Description Logics (DL) functionality. Furthermore, only the classes and instances for the Agile Methodologies are fully described in the ontology developed.

6.2 Design of Ontology

The terminologies for the development of the ontology for ISDMs were collected through the literature and coupled with interviews from the four interview respondents. The terminologies were analysed to come up with an agreed upon terminologies to develop the ontology for ISDM. The terminologies that have been used in the development of the ontology are shown in figure 6.1. Some competency questions were used to help to determine the scope of ontology development. The competency questions are discussed later in the chapter.

The terminologies that were collected contained classes, properties and individuals as components of the Web Ontology Language (OWL). In the figure 6.1 *Systems Development Methodologies* shows representation of the super classes for the ontology. Then *Approach, Method, Process Model, Tools and Techniques*, *Project Management* and *Training* represent the sub-classes of SDMs. The SDM super class is linked to the sub-classes through the relationship *has a* as indicated in the diagram. The sub-classes *Approach, Method, Process Model, Tools and Techniques*, *Project Management* and *Training* are linked to the SDM through the relation *is part of*. The subclasses have further been extended to have instances of the classes. The detailed description of the components of the ontology is described in section 2.4.1. The ontology for ISDM has only been fully described for classes and instances for the Agile Methodologies.

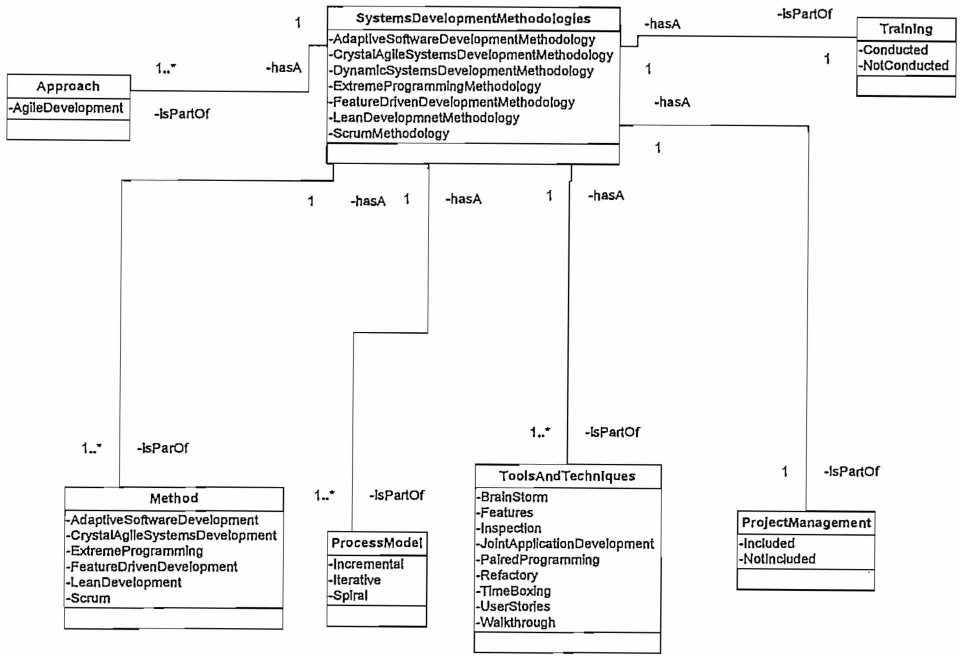

Figure 6.1: Class Hierarchy of the ontology for ISDMs

6.3 How to develop an ontology for ISDMs

The following steps indicated below help with the development of an ontology:

Determine the domain and scope of the ontology

The most ideal way of developing an ontology is by first defining its domain and scope. The following basic questions should be answered to obtain the domain and scope of the ontology:

- What is the domain that ontology will cover?

  The development involves representation of terminologies from Systems Development Methodologies such as *Approaches*, *Methods*, *Process Models*, *Project Management*, *Training*, *Tools* and *Techniques*.

- What use will the ontology provide?

  The ontology can be shared by people conducting software development projects. The ontology could help the development team to obtain meaning of terminologies from System Development Methodologies being used during the development process.

- What information will the ontology provide to the user?

  The ontology will be used in helping software developers in understanding the appropriate terminology to apply when conducting software development.

- Who will use and maintain the ontology?

  The ontology will be used and maintained by the software development team.

Create a list of important terms for the ontology

- Specify all the terms that will be described for the ontology to the user.

  The classes that will be used in the development of the ontology have been elicited from the interviews held with the interview respondents coupled with the literature gathered on SDM.

- Describe the properties that the terms will posses.

  The relationship type between the super class and the sub-classes has been described.

Define the classes and the class hierarchy

The following class hierarchy can be used when developing the ontology:

- In a **top-down approach**, the development process begins with defining the most general concepts and subsequent specialisation of the concepts in that domain. For example, *Systems Development Methodologies* as the general concepts. Then *Approach, Method, Process Model, Training, Project Management* and *Tools and Techniques* are specialised.

- The **bottom-up approach** commences by defining particular classes, and leaving other groups of classes as general concepts.

- The **middle-out** approach involves the mixture of the above two mention approaches simultaneously.

The top-down approach has been applied in the definition of classes and their hierarchy in this research.

Defining the class properties

Once the definition of the classes is completed, you must describe the internal structure of concepts. For example the *has a* and *is part of* are the two object properties used in the creation of the ontology. See figure 6.2.

- *has a* is linked to the *Systems Development Methodologies* domain. It has sub-properties called *has an approach*, *has a method*, *has a process model*, *has a project management*, *has a tools and techniques*, *has a training*.
- *is part of* has an inverse property of *has a*.

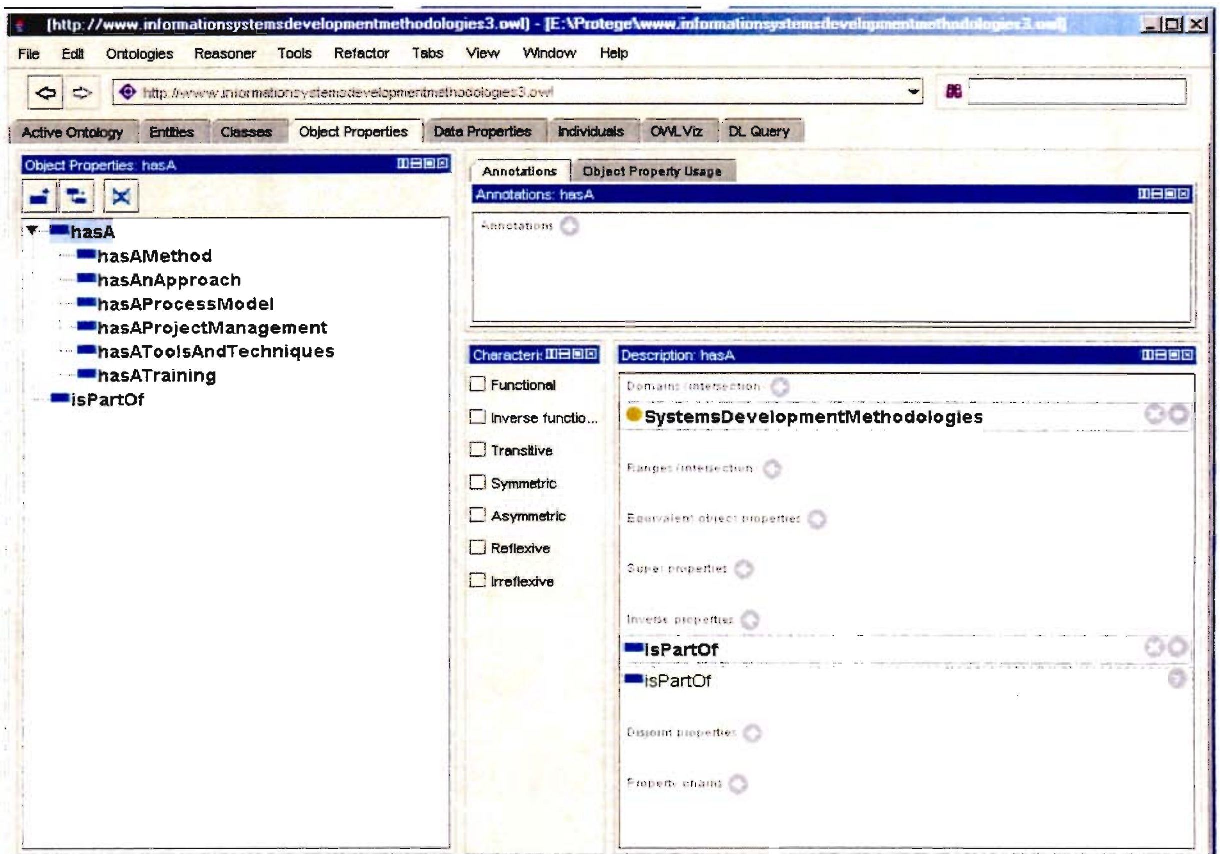

Figure 6.2 Object Properties

Create Instances

The creation of class instance is the final stage in defining individuals of the ontology. Usually the decision on which particular term will be a class or an instance for the ontology lies in the nature of the application to be created. Figure 6.3 shows instances by classes.

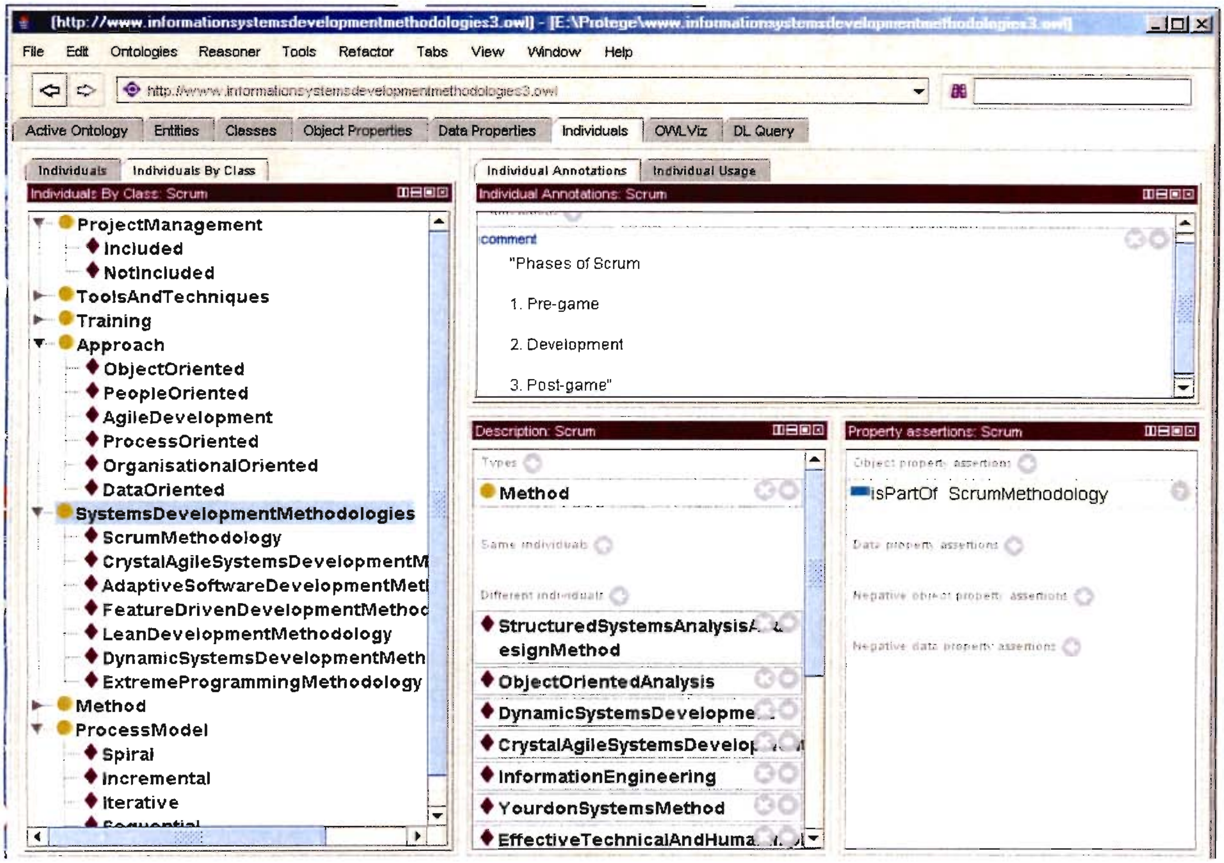

Figure 6.3 Instances by Classes

Invoking the Reasoner

One of the key features of ontologies is the reasoner. One of the main services offered by a reasoner is to test whether or not one class is a sub-class of another class. By performing such tests on the classes in an ontology, a reasoner can compute the inferred and asserted ontology class hierarchy.

Protégé allows different OWL reasoners to be plugged in. The reasoners task is to automatically compute the class hierarchy by checking the classes logical consistencies. The reasoner to use with Protégé is called *Pellet 1.5*. The manually constructed class hierarchy is called the asserted hierarchy whilst the automatically computed hierarchy is called the inferred. Figure 6.4 shows the asserted class hierarchy.

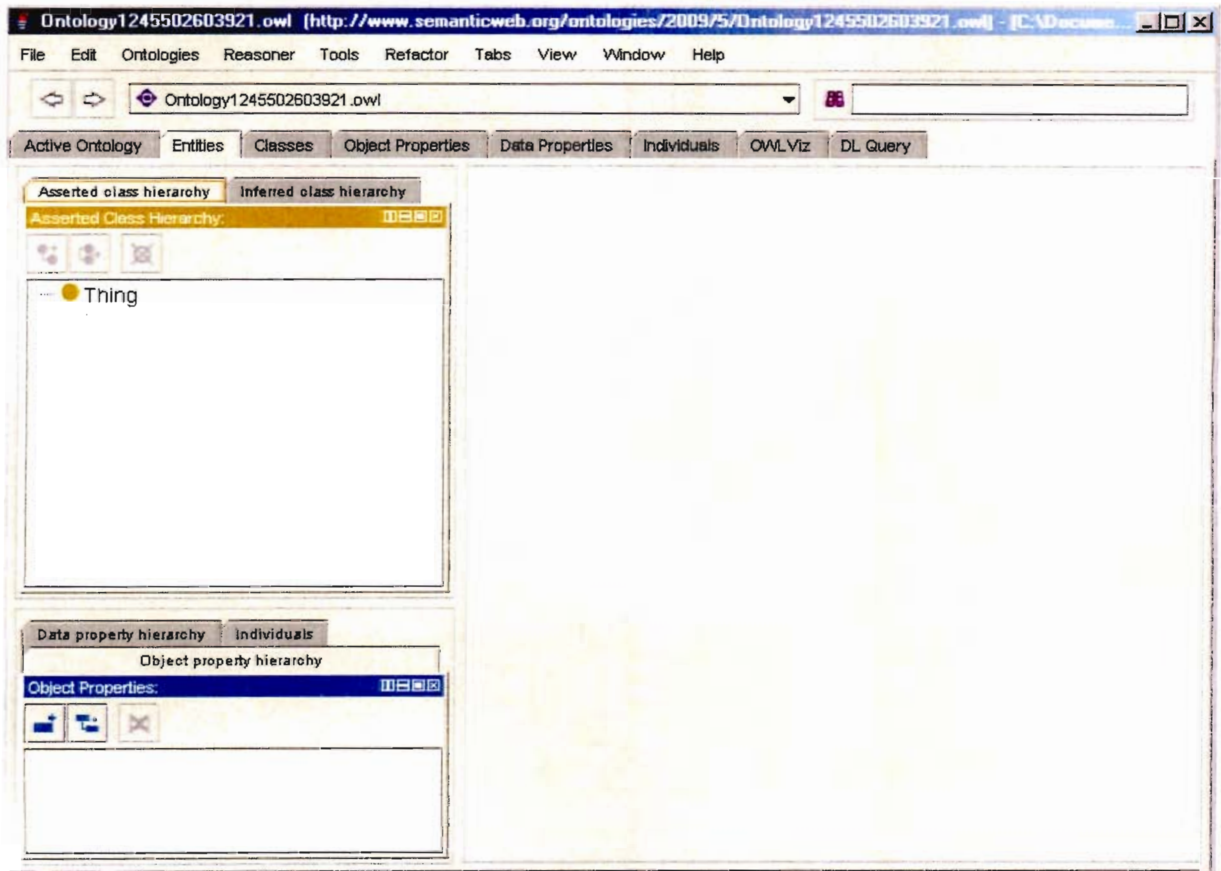

Figure 6.4: Asserted Class Hierarchy

6.4 Building an OWL Ontology

The steps below guide the user on how to create an ontology using Protégé.

6.4.1 Create a new OWL Ontology

- First start Protégé.
- When the Welcome To Protégé dialog box appears, press the Create New OWL Ontology.
- A Create Ontology URI Wizard will appear. Each ontology is named using a Unique Resource Identifier (URI). Replace the default URI with for example http://www.isdm.com/ontologies/informationsystemsdevelopmentmethdologies.owl and press Next.
- You will also want to save your Ontology to a file on your PC. You can browse your hard disk and save your ontology to a new file, you might want to name your file '*informationsystemsdevelopmentmethodologies.owl*'. Once you choose a file press Finish.

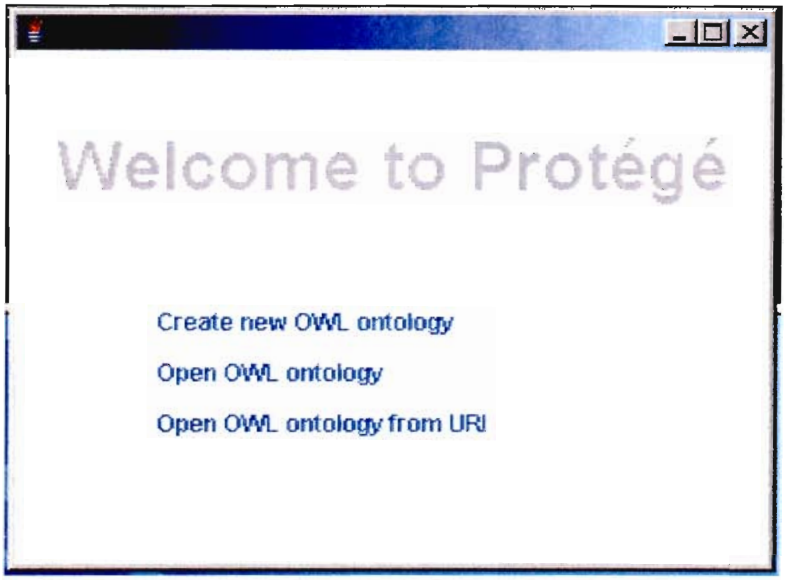

Figure 6.5: Starting Protégé

6.4.2 Active Ontology Tab

After a few moments an empty Protégé file will have been created and the **Active Ontology Tab** shown in figure 6.6 will appear. The Active Ontology Tab allows information about the ontology to be specified. For example, the ontology URI can be altered, annotations on the ontology such as comments may be added and edited via this tab.

- The Active Ontology Tab displays other useful tabs for developing the ontology such as the *Entities*, *Classes*, *Object Properties*, *Data Properties*, *Individuals*, *OWLViz* and the *DL Query*.
- Add a comment to the ontology.
- Ensure that the Active Ontology Tab is selected.
- In the Ontology Annotations view, double click to the right of the comment property name. An editing window will appear in the table.
- Enter a comment such as 'This is an ontology for Information Systems Development Methodologies'. Then press CTRL+ENTER to assign the comment.

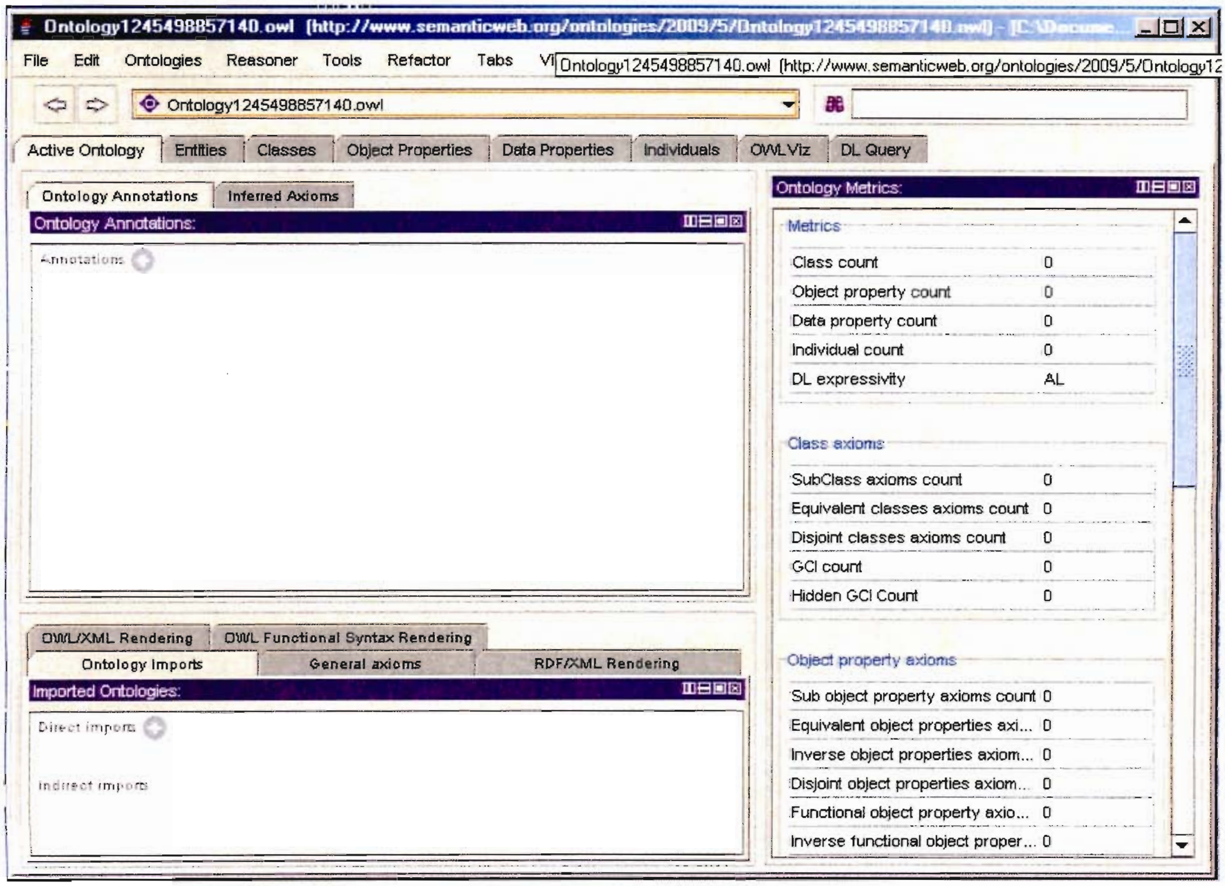

Figure 6.6: Active Ontology Tab

6.4.3 The Entities Tab

The entities created have the responsibility of displaying the internal communications and behaviour of the classes created. See figure 6.7 displaying the entities tab.

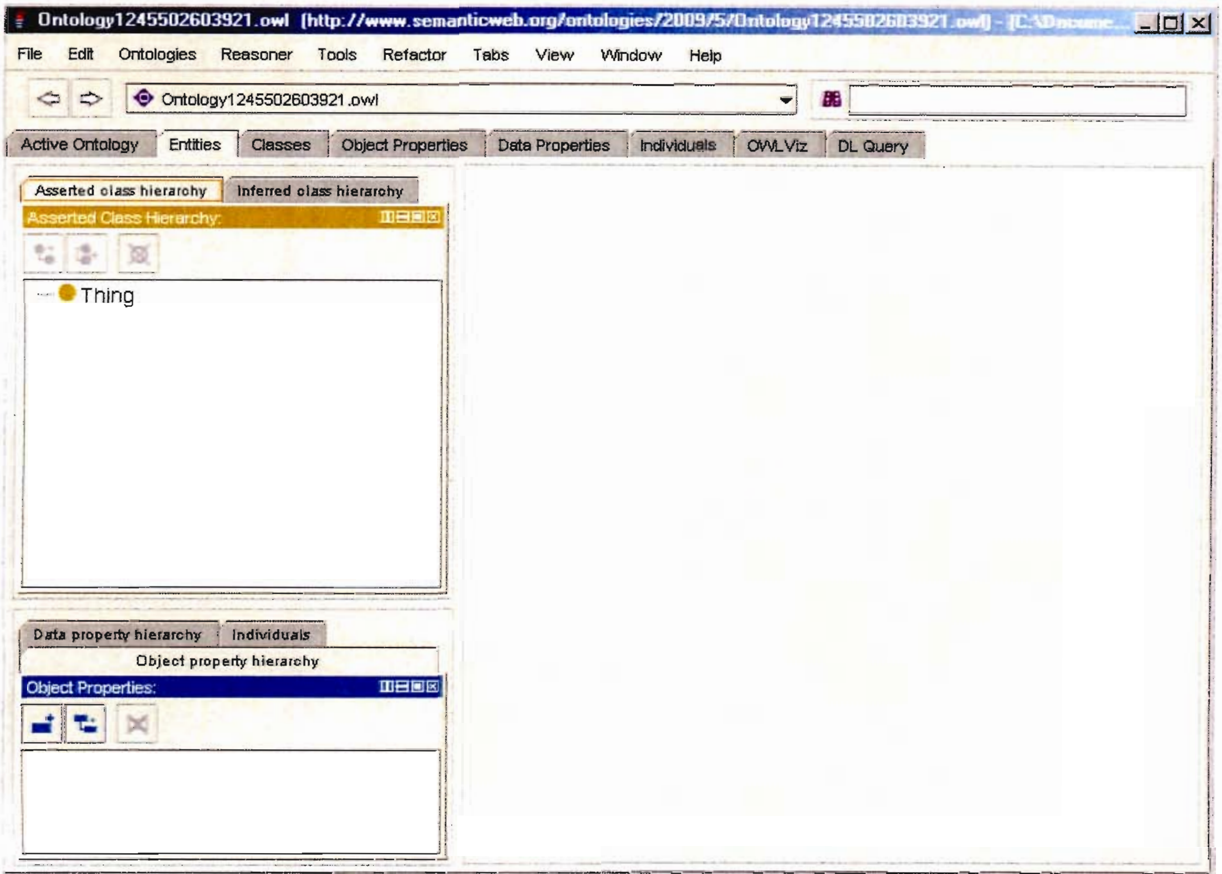

Figure 6.7: Entities Tab

6.4.4 Class Tab

The classes are interpreted as sets that contain individuals. They are described using formal descriptions that state precisely the requirements for membership of the class. These classes may be organised into a super class and sub-class hierarchy. The empty ontology contains one class called *Thing*. See figure 6.8.

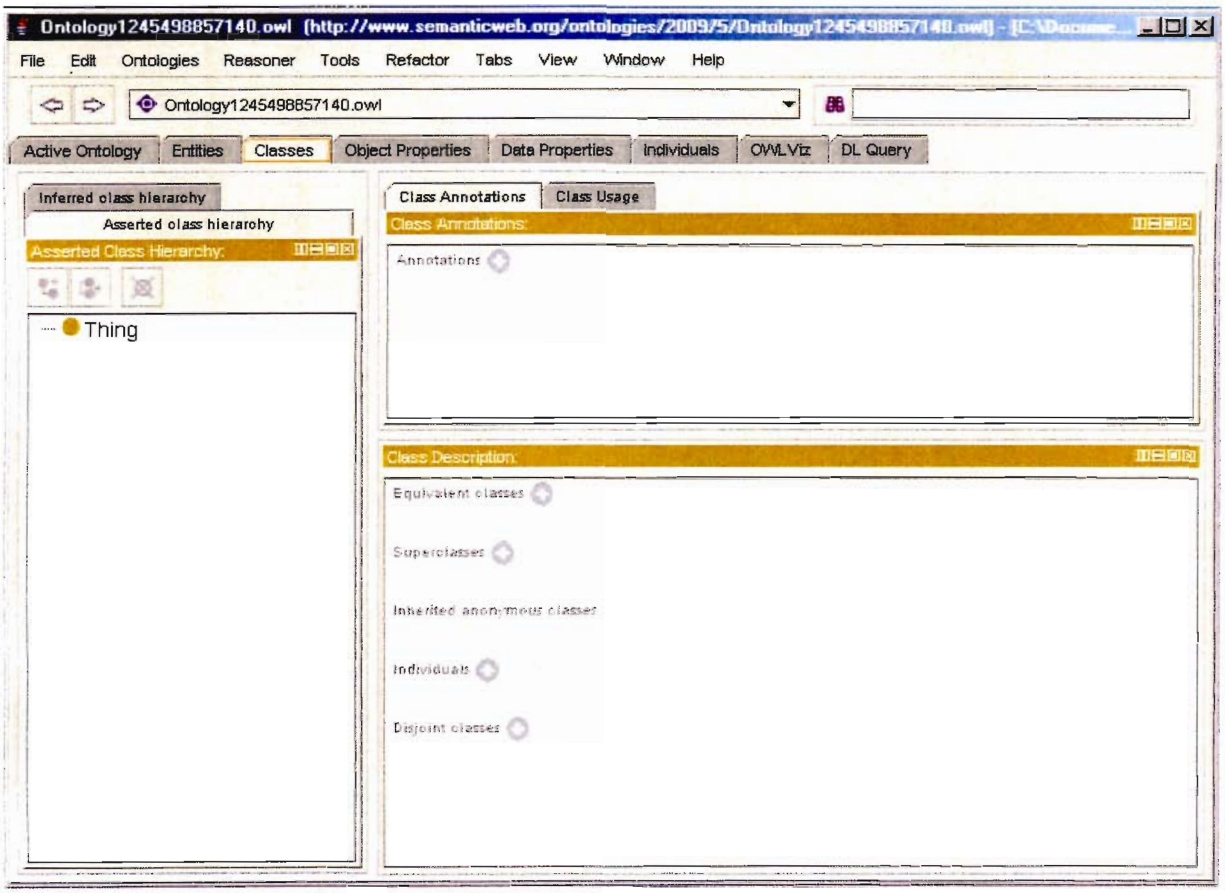

Figure 6.8: Class Tab

6.4.5 Creating a Class

The following steps below show the user in what way to create a class for an ontology. See figure 6.9.

- Highlight the class Thing.
- In the Asserted class hierarchy select the Add sub-class.
- A text box appears allowing you to enter the class name, enter the class name and press the OK button.
- To enter other classes for example *Approach*, *Method*, *Process Model*, *Project Management*, *Training* and *Tools and Techniques* simply follow the above instructions.

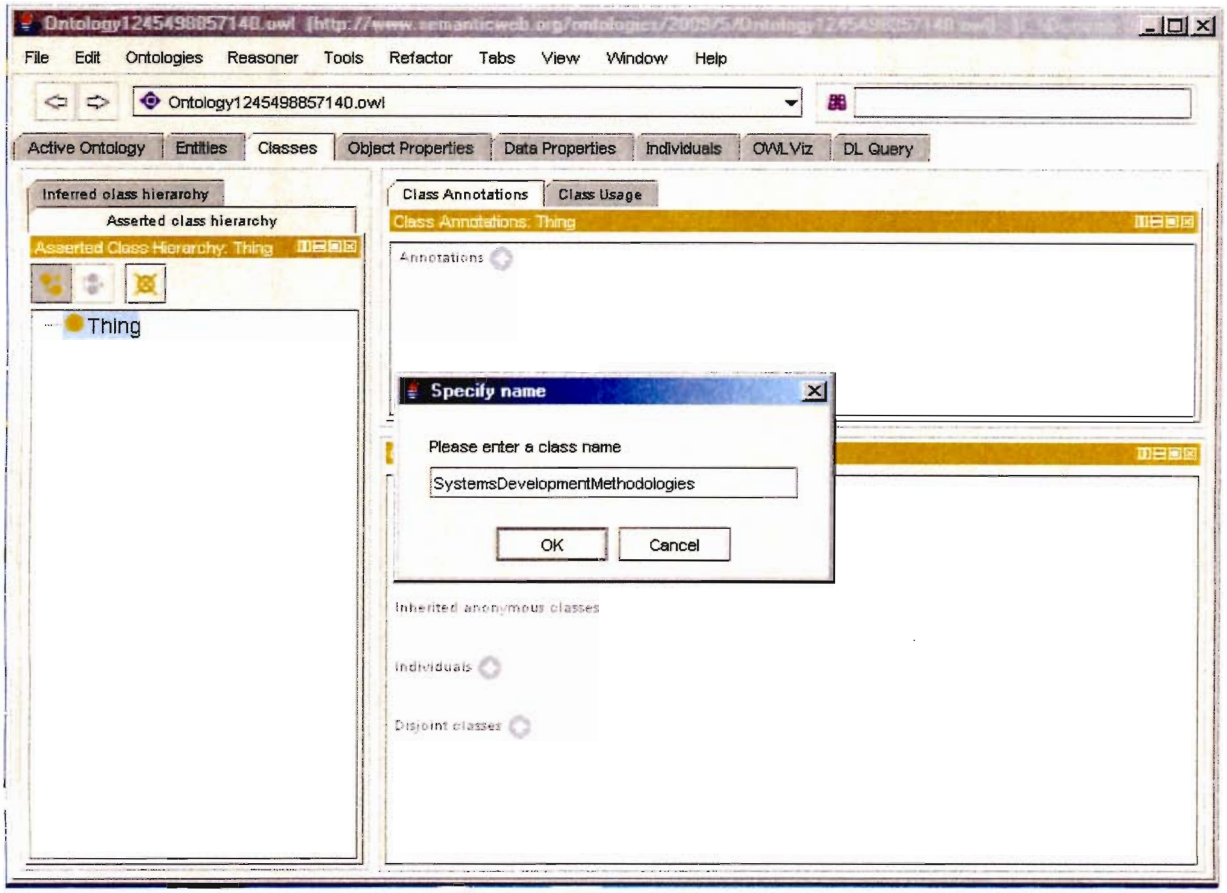

Figure 6.9: Creating Classes

6.4.6 Object Properties Tab

To create the **Object Properties**, switch to the Object Properties Tab. The OWL Properties represent relationships. The properties are used to link two individuals. Properties are binary relations to individuals. Properties can have inverses and they can be limited to having a single value such as a functional. They can also be either transitive or symmetric. See figure 6.10.

OWL Object Property Characteristics

The characteristic describes the association of the classes of the ontology developed. Described below are the characteristics that properties classes can posses:

Functional Properties: If a property is functional, for a given individual, there can be at most one individual that is related to the individual via the property.

Inverse Functional Properties: If a property is functional inverse then it means that the inverse property is functional.

Transitive Properties: If a property is transitive, and the property relates individual b and also individual c, then we can infer that individual a is related to individual c via property P.

Anti-symmetric Properties: If a property P is anti-symmetric and the property relates an individual a to individual b then individual b cannot be related to individual a via property P.

Reflexive properties: A property is said to be reflexive when the property must relate an individual a to itself.

Irreflexive properties: If a property P is irreflexive, it can be described as a property that relates an individual a to individual b, where individual a and individual b are not the same.

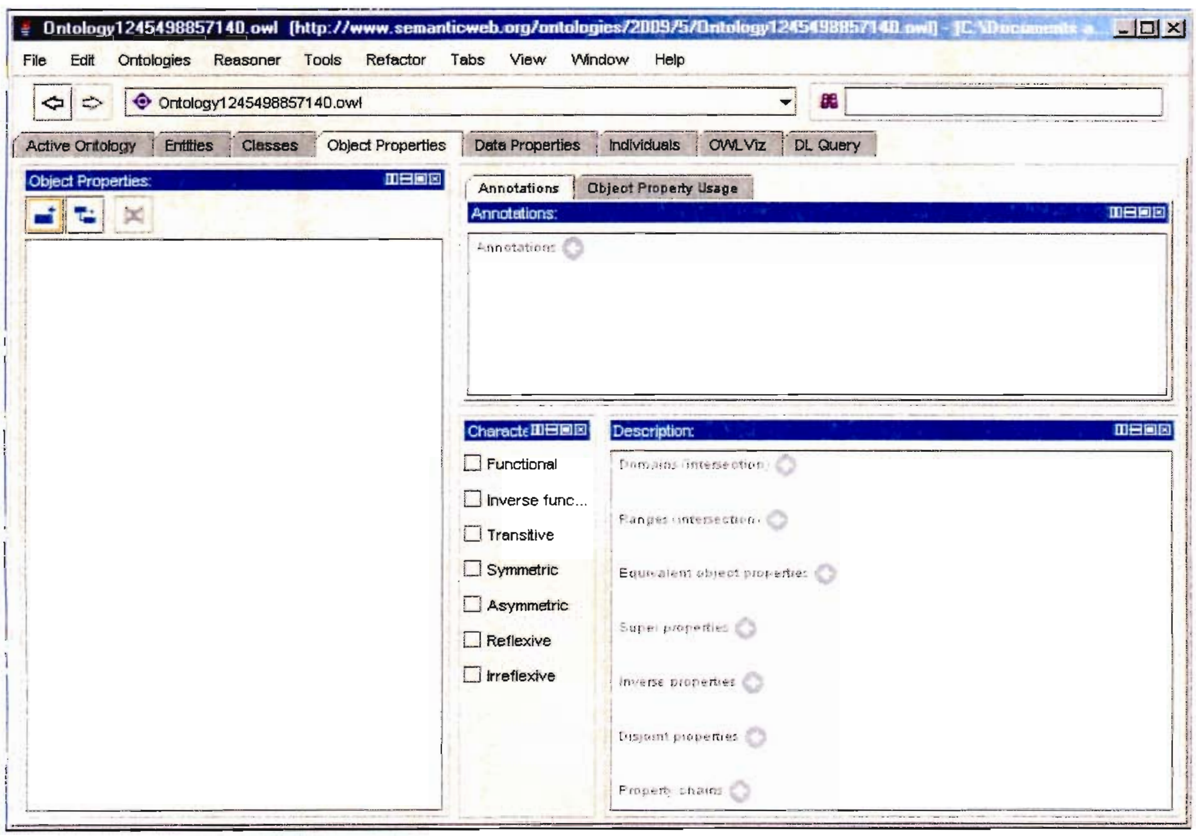

Figure 6.10: Object Property Tab

6.4.7 Creating the Object Properties

The following steps below show the user how to create an object property for an ontology. See figure 6.11.

- Switch to the Object Properties tab. Use the Add Object Property button to create a new Object property.
- Name the property *is a* using the Property Name Dialog that pops up as shown in figure 6.11, then press OK.
- The Object Property has to be linked to a Domain in the Description View window.

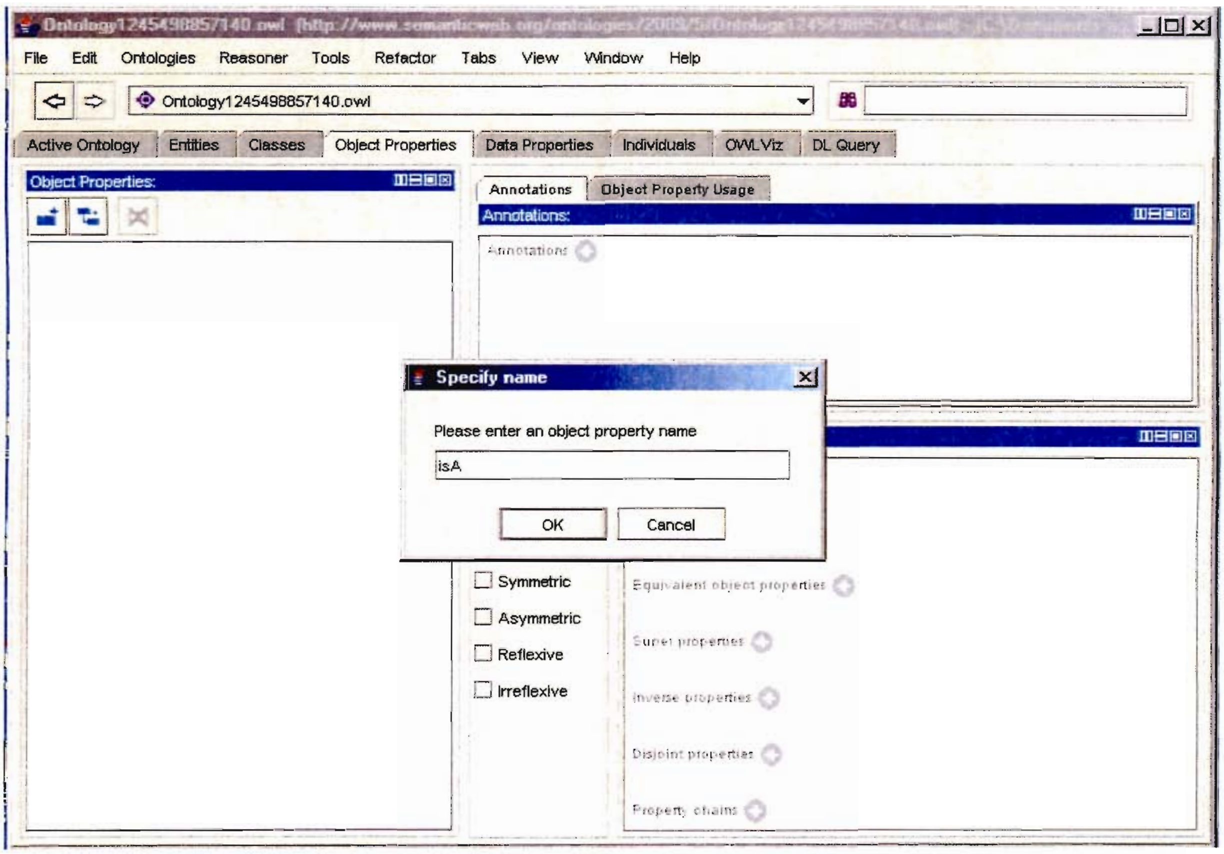

Figure 6.11: Creating Object Properties

6.4.8 Individual Tab

To create **Individuals**, switch to the Individual Tab. In OWL it must be explicitly stated whether individuals are the same than the other, or different to each other. The individuals are also known as instances of classes. Below are the steps to follow when creating individuals

- Create Method as a sub-class of *Thing*.
- Switch to the Individual Tab as shown in figure 6.12.
- Press the Add individual button and name the Individual *Extreme Programming*.
- Select the 'Select and add class' button from the Individual Types View located in the centre of the individual tab. Choose the *Method* from the class hierarchy that will make *Extreme Programming* an individual of the class *Method*.

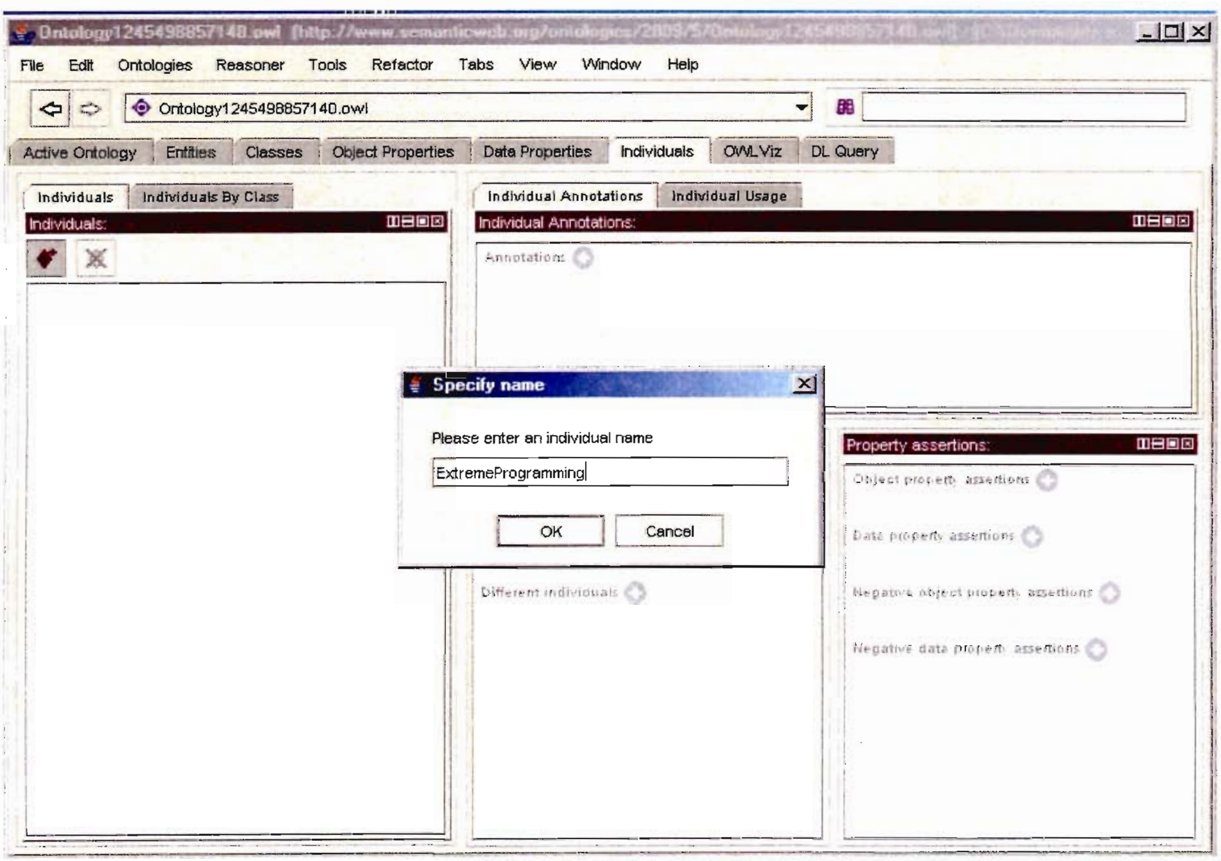

Figure 6.12: Individual Tab

6.4.9 DL Query Tab

The **Query** function is used to query the classes that have been created. Note that the classes can only be queried after the ontology reasoner has been computed. To compute the reasoner, go to the tool bar and select the *Pellet 1.5*. This reasoner will compute the inferred and asserted class hierarchy of the created ontology. See figure 6.13.

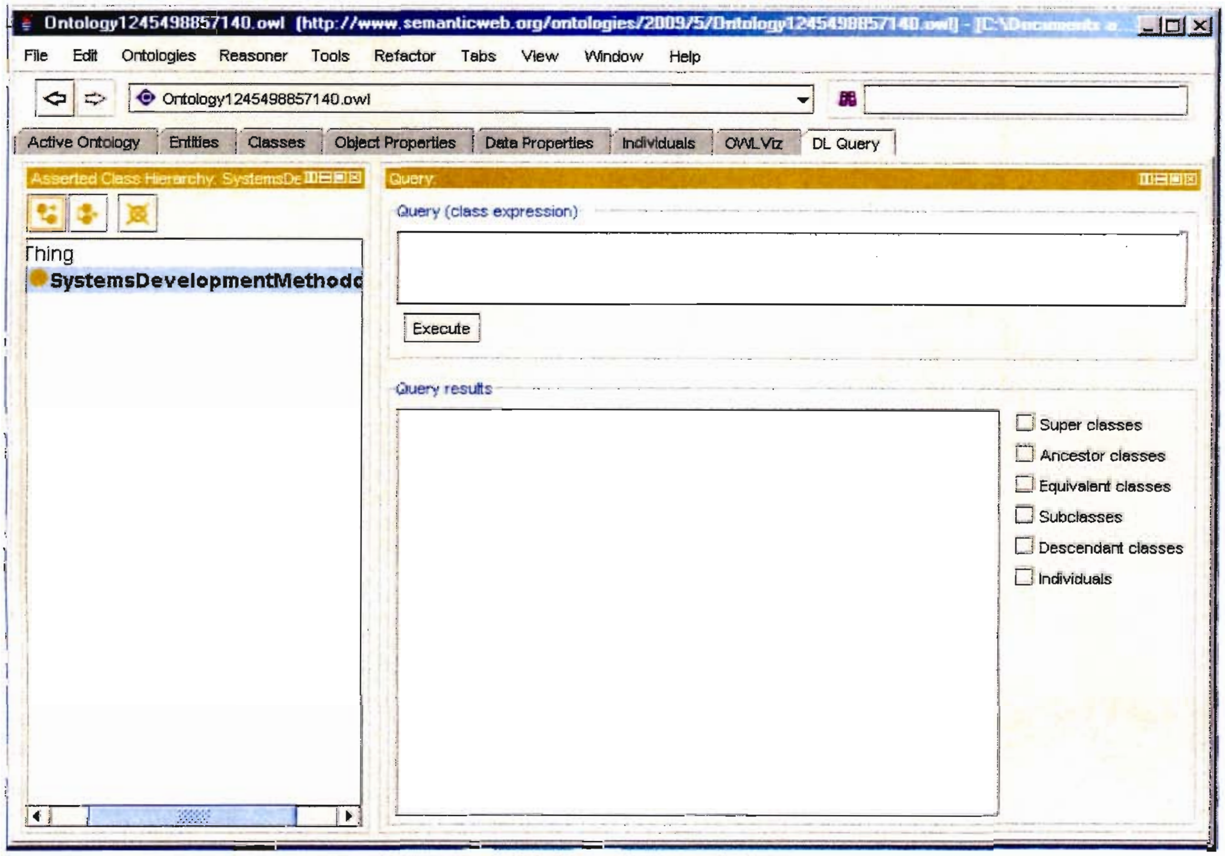

Figure 6.13: DL Query Tab

Querying created Classes

The following steps indicate how to query a class that has been created in the ontology:

- Enter the class name in the query class expression as shown in figure 6.14.
- Select the Reasoner and click *Pellet 1.5* to run the ontology.

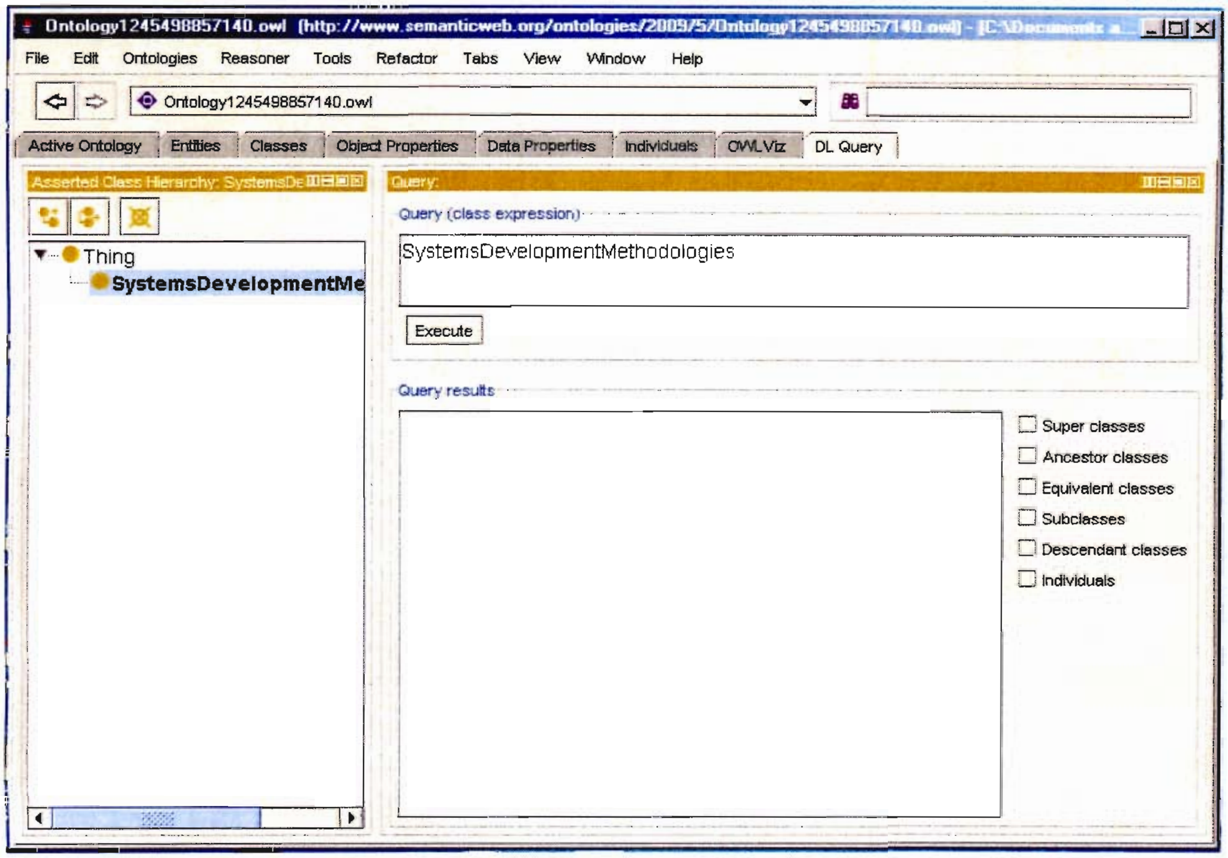

Figure 6.14: DL Query

Query after running Reasoner

The steps below show the user how to query a class after the reasoner has been invoked:

- After running Reasoner, Select the Execute button to display the query results.
- Then select the check boxes namely super class, ancestor class, equivalent, descendent and individual options that you would like to be displayed as shown in figure 6.15.

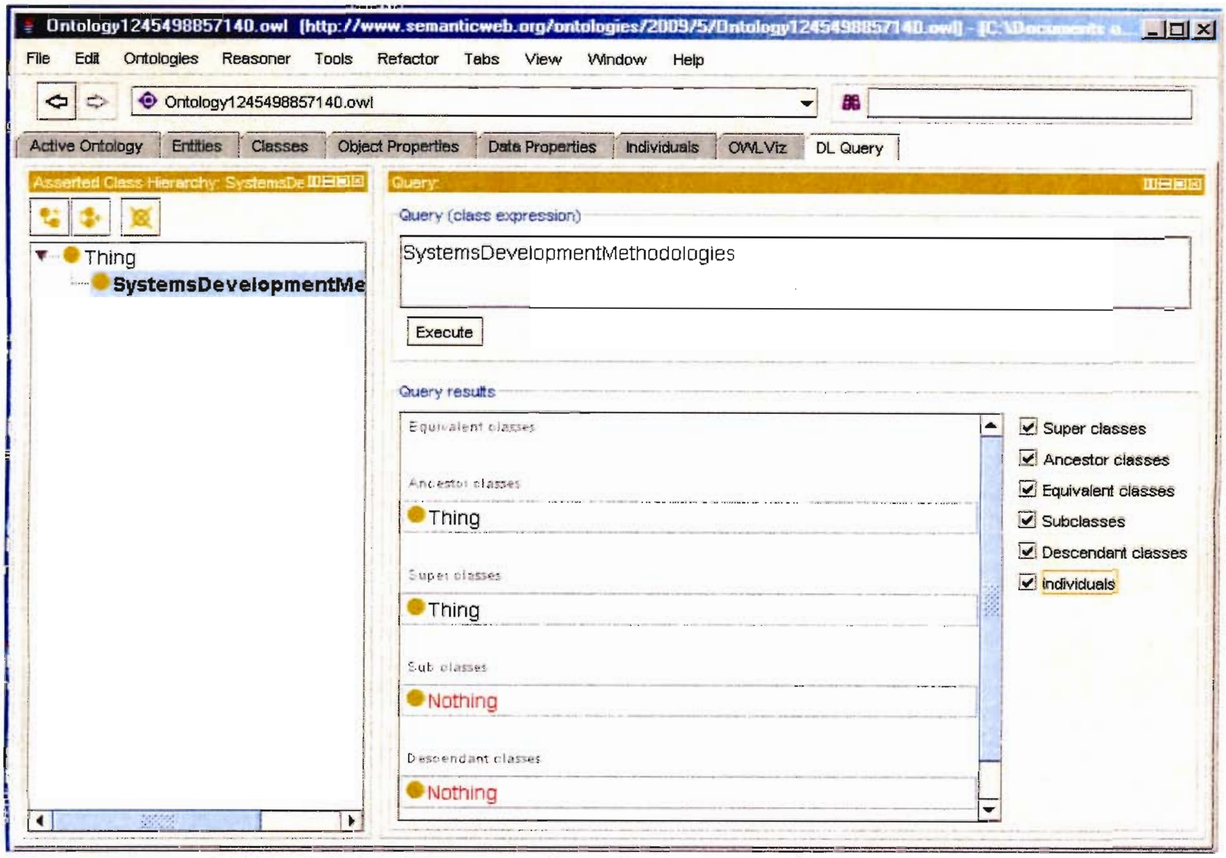

Figure 6.15: Query display after running Reasoner

6.5 Opening the created OWL ontology for ISDMs

The following steps guide a user in opening and using the ontology for ISDM that has been created. The ontology for ISDM developed has only fully been described for the Agile Methodologies:

- Start Protégé and Insert the Disc containing the ontology into the CD ROM drive.
- When the Welcome To Protégé dialog box appears, click on Open OWL ontology.
- You will select to open the OWL file from the storage location. In this case the ontology is stored on the disc.
- Click on the OWL file (informationsystemsdevelopmentmethodologies3.owl) to open the ontology in Protégé.
- After a short amount of time, the Protégé file will be opened and the 'Active Ontology' will appear. The active ontology tab contains the author of the ontology, the university attended and the title of the ontology. See figure 6.16.

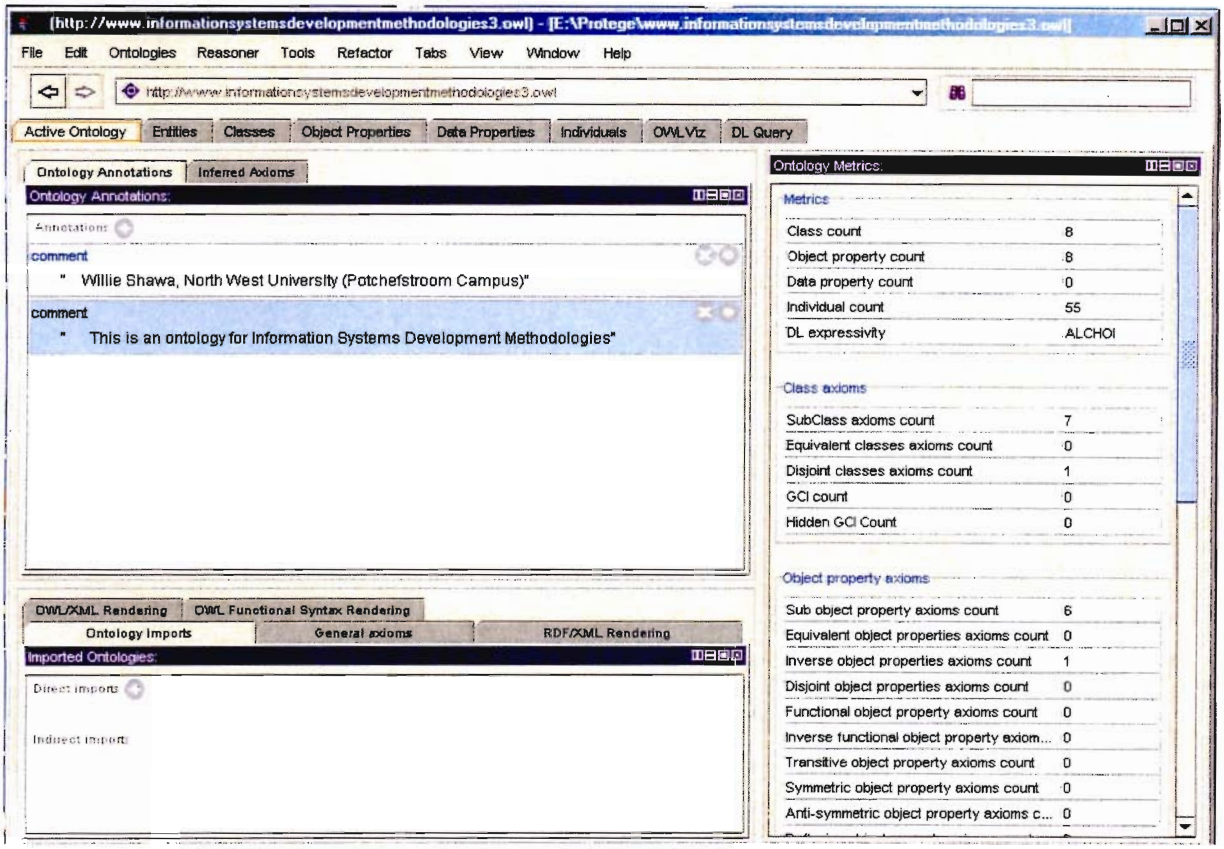

Figure 6.16: Opening the created Ontology

6.5.1 Viewing the Entity Tab

The steps indicated below allow a user to view the entities created in the ontology. The entities tab permits a user to have an overall view of the entities created. Here super classes, sub-classes, their object property relationships, instances of the classes and annotations containing the description for the classes and instances can be viewed.

- To view the Entities that have been created, select the Entities Tab.
- The Entities Tab displays the Classes, Object Properties and the Individuals that have been created as shown in figure 6.17.

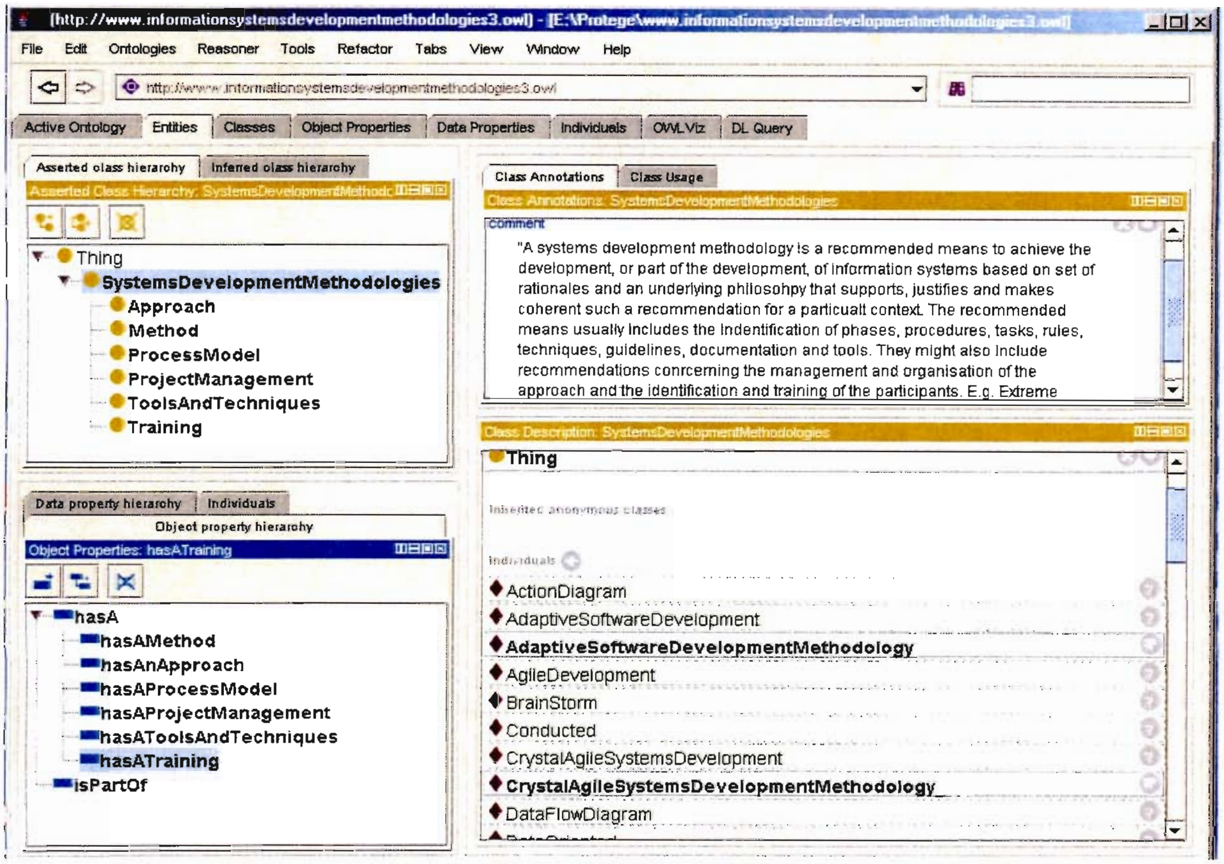

Figure 6.17: Entities Tab

6.5.2 Viewing the Named Classes

The steps indicated below allow the user to view the classes created in the ontology:

- In Protégé viewing of classes is done by switching to the Classes Tab to view the classes created.
- To view the descriptions of other sub-classes namely *Systems Development Methodologies*, *Approach*, *Methods*, *Process Model*, *Project Management*, *Tools and Techniques* and *Training*, you will ensure that *Thing's* drop down arrow is selected in order to view the other sub-classes linked to Thing in the Asserted class hierarchy as shown in figure 6.18.

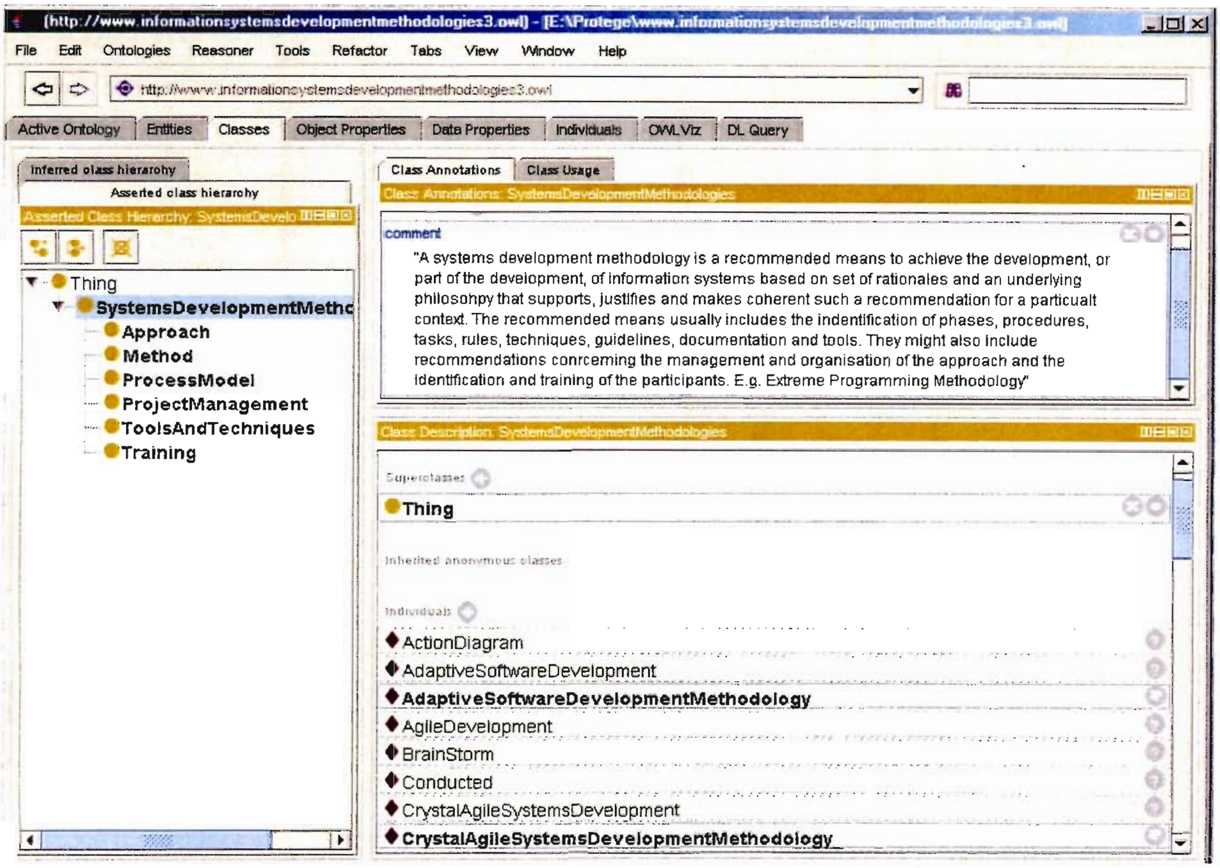

Figure 6.18: Class Hierarchy

6.5.3 Viewing the created Object Properties

The following steps guides the user on how to view the Object Properties created in the ontology:

- To view the Object Properties of the created ontology, switch to the Object Properties tab. See figure 6.19 showing the created Object Properties.
- There are two object properties that have been created, the one is *has a* with sub-properties of *has an approach*, *has a method*, *has a process model*, *has a project management*, *has a tools and techniques* and *has a training*. The other object property created *is part of* and has an inverse relationship with *has a* object properties.

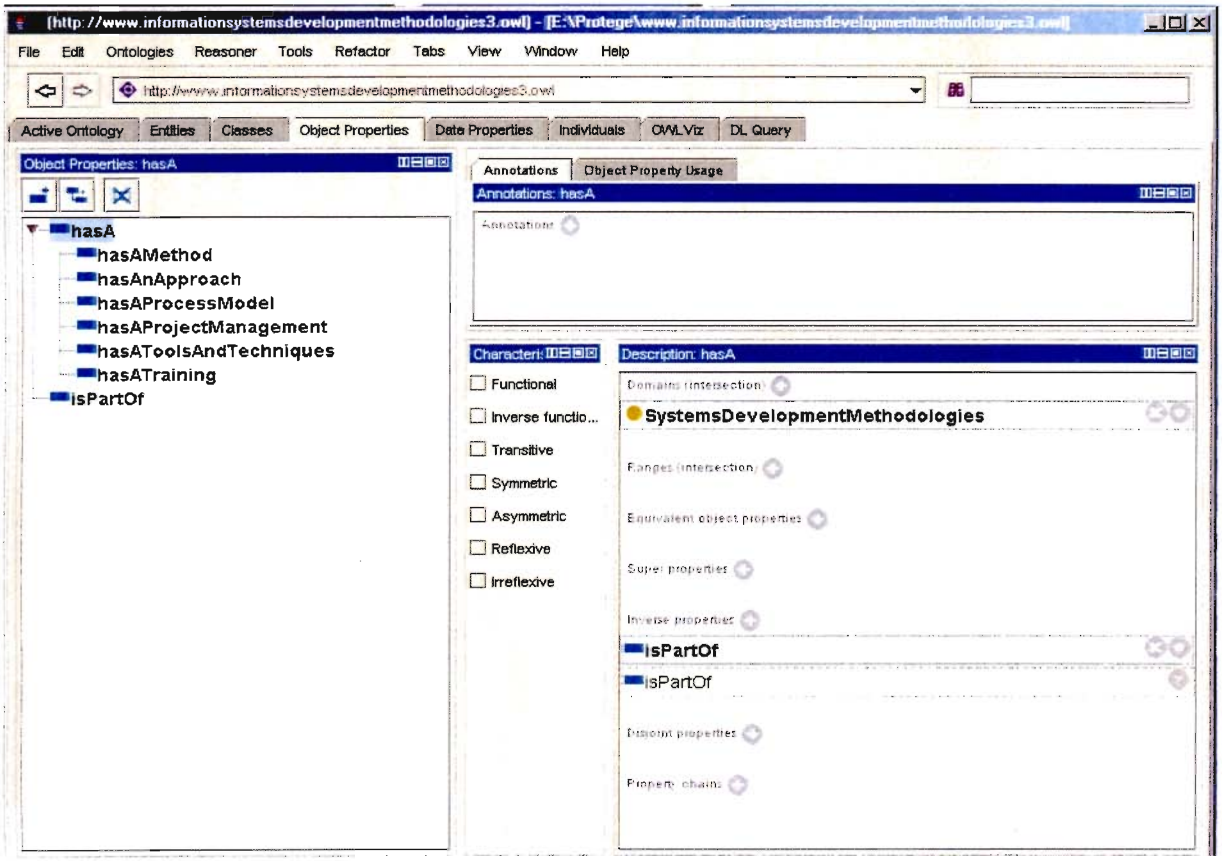

Figure 6.19: Object Properties

6.5.4 Displaying Individuals by Classes

The following steps guide the user on how to view the individual (instances) created for the classes in the ontology:

- To view the Individuals in the created ontology, switch to the Individuals tab. See figure 6.20 showing the created Individuals.
- Select the Individuals by Class tab to view the Individuals according to the Classes they relate to.

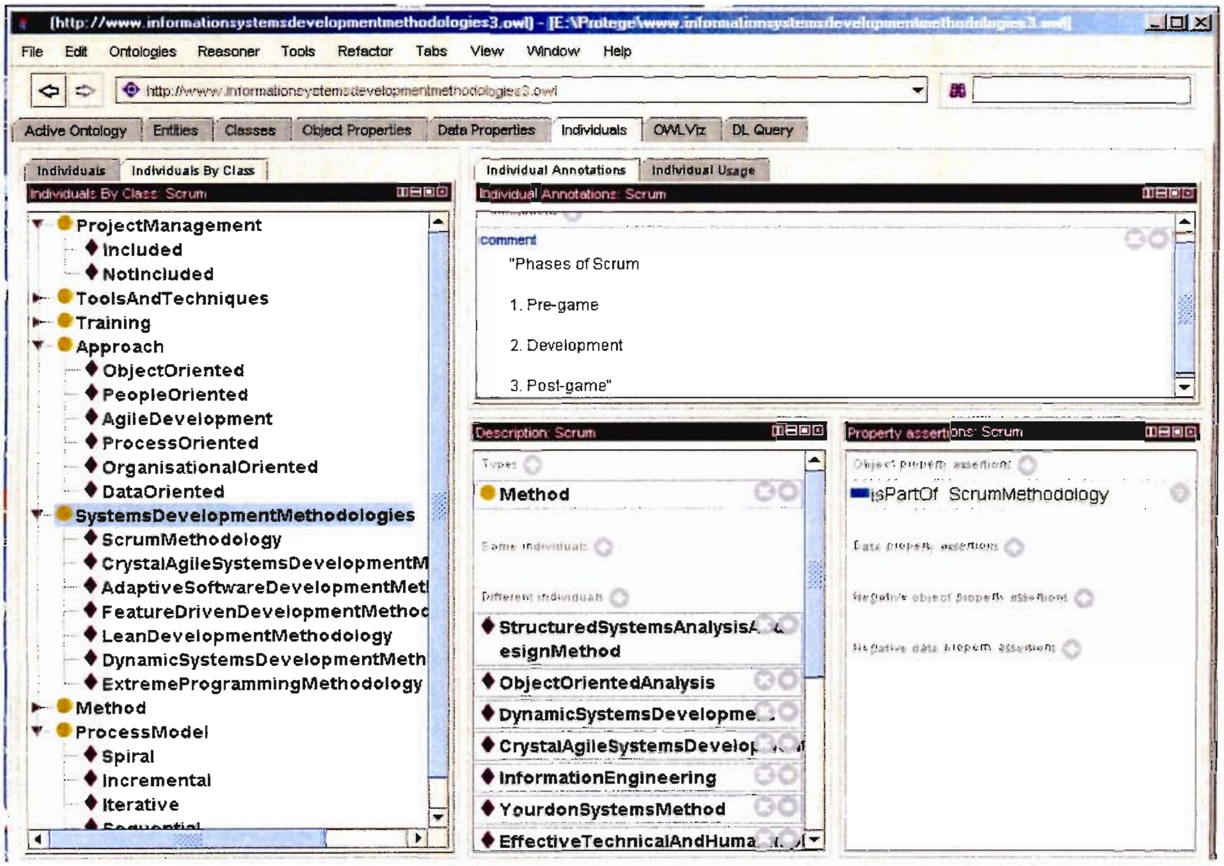

Figure 6.20: Individuals by Class

6.5.5 Viewing DL Queries

The following steps guides the user on how to query the classes created in the ontology:

- To view the Queries of the created ontology, switch to the Query tab.
- Select the Reasoner *Pellet 1.5* to compute the ontology.
- Enter the Class name you wish to query in the Query (class expression) and click the Execute button to run the query.
- Select the check boxes such as super-classes or sub-classes to view the relevant classes queries. See figure 6.21 showing the subclass displayed for the classes *Systems Development Methodologies* entered in the Query (class expression).

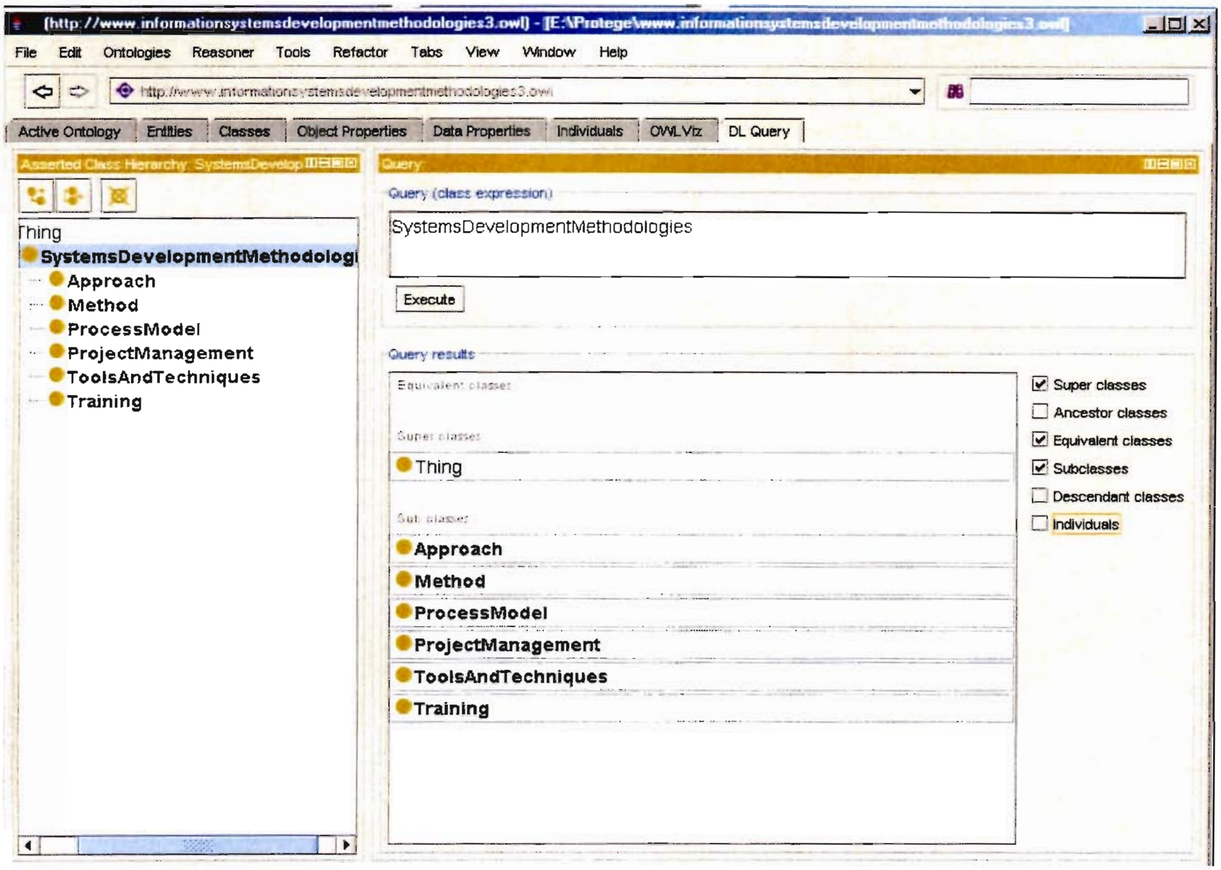

Figure 6.21: Query Sub Classes

6.5.6 Viewing Query for Individuals

The following steps guide the user to view the individual that have been created in the ontology:

- To view other queries for other classes enter the classes name such as *Systems Development Methodologies* in the Query (class expression) window as shown in figure 6.22.
- Ensure the Reasoner, *Pellet 1.5* has been computed before selecting the Execute button to run the query.
- The query displays all individuals created for the *Systems Development Methodologies*.

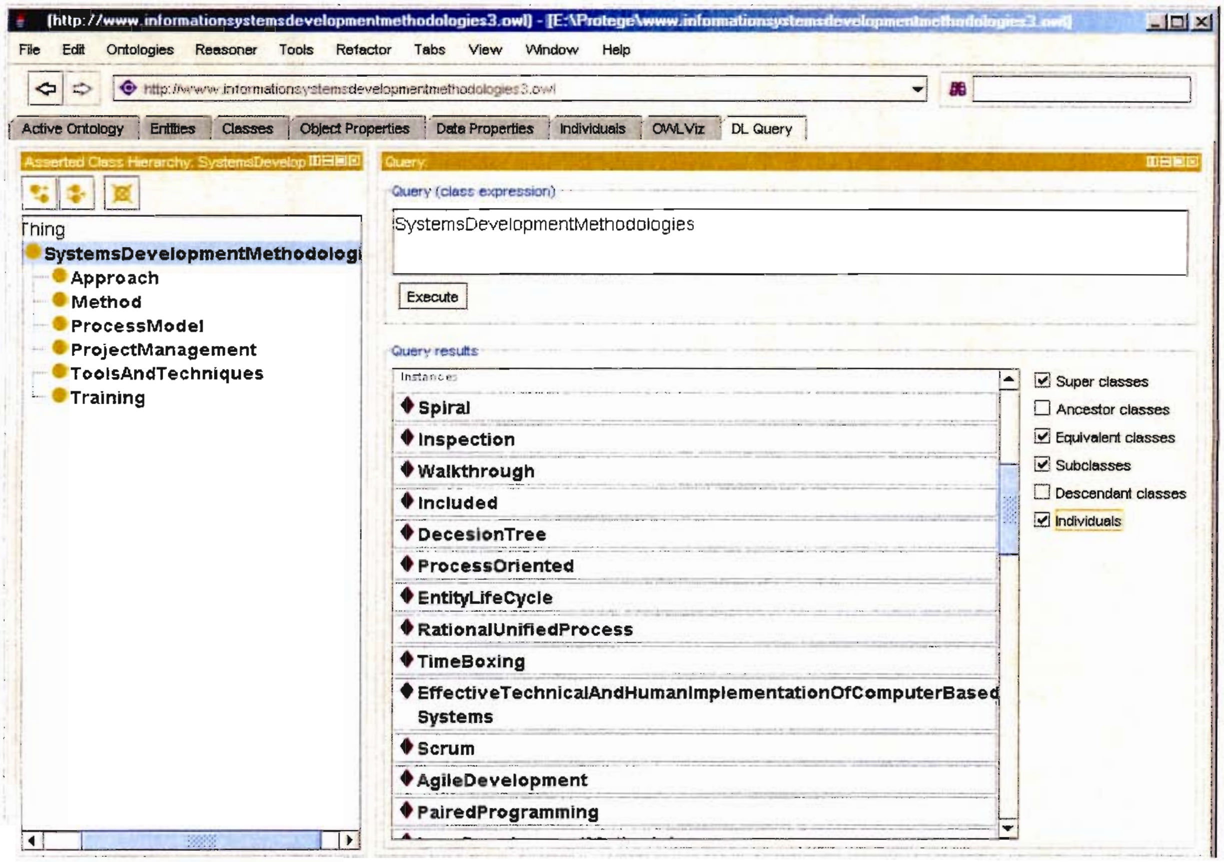

Figure 6.22: Query Instances

6.5.7 OWLViz

The following steps guide the user in view the results of the OWLViz:

- Switch to the OWLViz tab.
- Select the class *Systems Development Methodologies* in the asserted class hierarchy.
- Select the Show Children in the OWLViz window as shown in figure 6.23
- The results display the class hierarchy for the asserted classes for the created ontology for ISDMs.

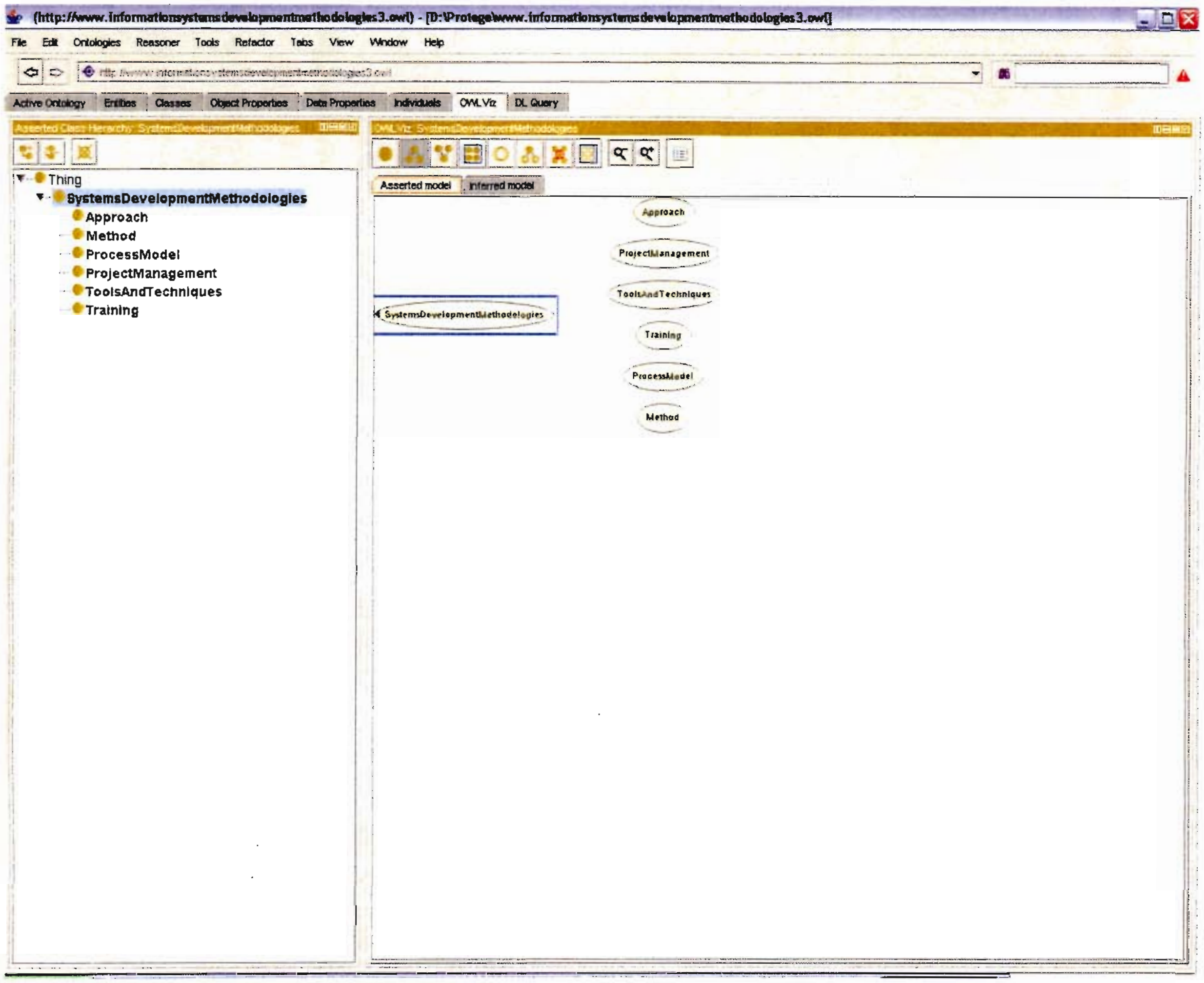

Figure 6.23: OWLViz

6.6 Testing of the Ontology

The ontology for Information Systems Development Methodologies (ISDMs) was tested to see its usefulness. It was tested in the academic set-up and it proved to be useful to the users. The users were able to obtain meaning of terminologies found in Systems Development. The users were also able to perform queries on the created classes in order to view the classes and instances of the ontology. To test the ontology for ISDMs functionality, the following tests were conducted.

Entities

To test the functionality of the sourced entities, you firstly have to switch to the Entities tab. The entities give a global view of the classes created. It shows the class hierarchy for the class created namely, the *Systems Development Methodologies* and the sub-classes *Approach*, *Method*, *Process Model*, *Project Management* and so on. The

Entities tab, also shows the description of all the classes that have been created. For this prototype of the ontology, only the full description for the Agile Methodologies has been included. The Entities tab also shows the object properties, namely the *has a* and is *part of* which represent the type of relationship that exists amoung the classes that have been declared. Then, the Entities tab also highlights the instances that have been created. For example for the instances for SDM, *Adaptive Software Development Methodology*, *Crystal Agile Systems Development Methodology* are highlighted in the class description window.

Classes

To test the functionalities of the classes and sub-classes created for the ontology, you first click onto the classes tab. The Systems Development Methodologies is linked to the *Things*, which acts as the default link for all other classes that are created. Once a user clicks on the class Systems Development Methodologies, The class provides detailed description regarding that class in the class annotation window. The description provides a definition of what a *Systems Development Methodology* , and also indicates the components that constitute it. To view a list of all the SDMs that have been created in the ontology, you have to look in the class description window. The class description window highlights all the instances of the SDMs that have been created. Only the SDMs for the Agile Methodologies have been described for this Prototype of the ontology. The SDMs are *Adaptive Software Development Methodology*, *Crystal Agile Systems Development Methodology* and many more.

Object Properties

To test the functionality of the object properties, switch to the Object Properties tab. The object properties indicate the type of relationship that exists amoung the classes created. The two object properties created in the ontology are *has a* and *is part of*. The *has a* contains sub-properties namely has an approach, *has a method*, *has a process model* and so on. The *has a* property and its sub-properties all belong to the SDM domain which is indicated in the description window of the Object Property tab. The other object property that has been created is the *is part of* property. This property has an inverse relationship to the *has a* property.

Individuals

To test the functionality of the individuals that have been created in the ontology, you have to switch to the Individuals tab. The Individuals tab lists all the instances that have been created for the ontology. The instances have a definition describing in the individual annotation window. And the description window shows which classes are linked to which instance. For the purpose of this prototype created, only the instance related to the Agile Methodologies have been full described.

OWLViz

To test the functionality of the OWLViz, you have to switch to the OWLViz tab. The purpose of OwlViz is to show graphically the hierarchical structure of the classes and sub-classes that have been created in ontologies. To use this functionality, you firstly need to ensure you compile the ontology with the reasoner *Pellet 1.5* provided in the menu bar. The OWLViz shows the asserted model, which is the classes hierarchy before the ontology is computed. The asserted model shows that *Systems Development Methodologies* as the parent class and *Approach*, *Project Management*, *Tools and Techniques*, *Training*, *Process Model* and *Method* are its children classes. And the inferred model automatically computes a new class hierarchy after the compilation of the ontology has been completed.

DL Query

To test the functionality of the DL Query, you have to switch to the DL Query tab. To query the classes that have been created, you have to enter the class name in the query window. Note that the Protégé is case sensitive and will only display a query if the reasoner has been run. After entering the class for example, *Approach* in the query window, you click the Execute button to get the desired results. The check box to the right of the Results window namely, Super classes displayed as *Systems Development Methodologies*. Ancestor classes displayed *Systems Development Methodologies* and *Thing*. Individuals displayed *Object Oriented*, *People Oriented*, *Agile Development*, *Organisational Oriented* and *Data Oriented*. The check boxes allow the user to display the desired results.

6.7 Summary of the chapter

This section described the steps to undertake when a user wishes to develop an ontology. The ontology for ISDMs developed is able to provide its user with the meanings of the SD concepts that have been created. The ontology also showed the relations between the classes and instances that are created. As a major benefit, the ontology would provide a lot of help in reducing the misunderstanding of SD concepts when conducting software development processes. The created ontology could also be shared and reused by other software developers to provide meaning to SD concepts especially in cases where software development is being conducted through outsourcing, and software developers are located in different geographical locations. An example of an already created ontology was described for the user to see and use. The full ontology has been saved in Appendix C on the Compact Disc Read Only Memory (CDROM).

CHAPTER SEVEN
RESEARCH FINDINGS, CONTRIBUTIONS, LIMITATIONS AND FUTURE WORK

7.1 Introduction

The ontology for Information Systems Development Methodologies (ISDMs) that has been developed can be useful to the software developer in minimising the misunderstanding of terminologies used in Systems Development (SD). The ontology can be used to mediate the meaning of the SD terminologies among the software developers especially in cases where software development is conducted through outsourcing, where software developers have to contribute to the development of the same piece of software whilst located in different geographical positions. This chapter briefly outlines the findings of the research conducted. It also describes the research contributions, limitations, conclusion and future work to be conducted on the research.

7.2 Research Findings

To successfully conduct this research, a qualitative research approach was applied. The literature documented in chapter four by Denzin and Lincoln (2005) indicates qualitative research as the appropriate approach to apply in this research because of its suitability to involve empirical material such as interviews and artefacts to describe meaning in individual's lives. An extensive literature review on ontology in chapter two, and Systems Development Methodologies in chapter three was collected to help the researcher in answering the main and minor research questions. The literature on ontology, provided understanding of the subject matter from a philosophical perspective and engineer's perspective as reported by Aristotle and Gomez-Perez *et al.* (2004a) respectively. The chapter also reported on numerous but almost similar definitions of ontology by the authors Gruber (1993b), Guarino (1995), Studer *et al.* (1998), Swartout and Tate (1999) and Hendler (2001) to provide a better understanding of literature. To aid the developed of the ontology, literature on the components of the ontologies (Corcho *et al.*, 2006), METHONTOLOGY methodology (Fernandez-Lopez *et al.*, 1997) and Protégé (Noy *et al.*, 2000) was documented.

Then, chapter three on Systems Development Methodologies (SDMs) also provided additional understanding to the research undertaken. The literature acted as a source of terminologies for the development for the ontology for ISDMs. The chapter contained literature documenting the definition of an ISDM and some examples of SDMs based on their philosophical view as prescribed by Avison and Fitzergerald (2006).

To develop the ontology for ISDMs, interviews with a software development practitioner and three academics were conducted. The data collection principles documented in chapter four by Creswell (2008) was applied to aid this process. This involved conducting one-on-one interviews and asking the interviewees open-ended questions related to software development. After the interviews were conducted, the responses collected were analysed using content analysis as described by Cole (1988) in chapter four. The actual analysis of the interview responses is tabulated in chapter five of this research. Once the data was analysed, the ontology development principles documented in chapter two of this research were applied. The steps followed when developing the ontology are also documented in chapter six.

In this research a prototype of the ontology for ISDMs was developed. This ontology contained agreed upon terminologies from SDM that could be used by software developers to minimize misunderstandings of SD terminologies during the software development process. The ontology contained classes, sub-classes and these were linked together by the object properties. Additionally, instances for the classes were created, and only descriptions of the Agile Methodologies have been provided in the ontology developed. The ontology for ISDMs is available on Appendix C.

The findings of this study are based on the following main and minor research questions:

Main Research Question

- What role can ontologies play in Information Systems Development (ISD)?

The main research question was answered by addressing the minor research questions.

Minor Research Questions

What is ontology?

Several definitions by different authors exist describing what an ontology is. They have different elements namely conceptualisation, explicit, formal, shared, semantic network and inference considered to constitute the definitions. The following definitions have been applied in this research.

Studer *et al.* (1998:162) and Hendler (2001:54) define ontology as *"a formal explicit specification of a shared conceptualisation of knowledge terminologies such as vocabulary exhibiting semantic interconnection and rules of inference in a particular domain area". Conceptualisation* refers to an abstract model of some phenomenon. *Explicit* implies the concepts and constraints applied should be clearly defined. *Formal* implies the data requiring processing in the ontology should be converted in a form readable by a computer. *Shared* implies that the concepts used to create the ontology must be collectively accepted by stakeholders. The *semantic network* states that the association between the concepts of the ontology must be described. For *inference* an ontology must enable some form of reasoning applied to the categorisation of concepts.

The ontology's aim is to capture consensual knowledge that is shareable and reusable across software applications by different developers. The detailed description on this literature can found in section 2.4 of the research.

What is ISDM?

The research discovered that a lot of inconsistencies existed when giving an appropriate definition of a methodology. Other authors have inconsistently used the terminology method or approach when they intended to mention methodology (Jackson, 1981; Blum, 1994; Iivari *et al.*, 1998).

The definition provided by Avison and Fitzergerald (2006:24) closely captures the elements of what can constitute the definition of ISDM. They state that, "a methodology is a collection of procedures, techniques, tools and documentation aids, which will help the systems developers in their efforts to implement a new information system. A methodology consists of phases, themselves consisting of sub phases, which will guide the systems developers in their choice of the techniques that might be appropriate at each stage of the project and also help them plan, manage, control and evaluate information systems projects".

Additionally, Avison and Fitzergerald (2006) has indicated the term methodology to contain a more comprehensive meaning than method by possessing the characteristic philosophical view in its definition. The definition of ISDM should have the combination of the following components namely *Approach, Process Model, Systems Development Method, Tools* and *Techniques.* The detailed description on this literature can be found in section 3.2 of the research.

What is an ontology for ISDMs?

An ontology for ISDMs is a knowledge-base aimed at capturing consensual knowledge from the ISDM domain. The knowledge was captured by collecting terminologies of Systems Development through interviews conducted on one software development practitioner, three academics and an in-depth survey of the literature review on SD. The ontology can be shared by software developers to enhance their understanding of the terminologies applied in Systems Development (SD). The ontology for ISDMs can also be used in developing other types of ontologies that can be reused across software application by different developers. The detailed description of the ontology for ISDMs can be found in Appendix C.

Why is there a need for an ontology in ISDMs?

After conducting the study through interviews and literature reviews on SD, it was found that numerous inconsistencies existed when using SD terminologies. Different people misused terminology such as method to refer to a methodology. So in order to minimise the misunderstanding of SD terminologies, an ontology for ISDMs was developed. The purpose of the ontology would be to mediate the meaning of terminologies in SD. The ontology can then be shared across the network to minimise the misunderstanding of SD terminology, especially in cases where software development has to be conducted by stakeholders located in different geographical locations.

How do you develop an ontology for ISDMs?

To successfully develop an ontology for ISDMs, it will be required that you collect the agreed upon terminologies from the interview respondents and relevant SD literature. There after the appropriate methodology, method and tools have to be available and used to guide the developer during the ontology development. The methodology used in the development of the ontology was METHONTOLOGY (Fernandez-Lopez *et al.*, 1997). According to Fernandez-Lopez *et al.* (1997), METHONTOLOGY methodology consists of three developmental activities namely management, developmental and supportive activities. The detailed description on METHONTOLOGY can be found in section 2.5.2. The tool used in the development of the ontology is called Protégé. Protégé is an open source software package that is free and available for download on the Internet. It has extensible architecture and is platform independent. The developmental process of the ontology is described in section 6.3.

How will software developers use the ontology for ISDMs?

The software developers will use a share drive, where the ontology for ISDMs can be accessed to view the captured SD terminologies during the software development process.

Then in cases involving software development through outsourcing, software developers located in different geographical locations can use the same ontology in order to minimise the misunderstanding of SD terminologies.

7.3 Research Contributions

- The research contribution is the development of the ontology for ISDMs. This will be useful in helping software developers to have a better understanding of the terminologies applied in SD. Using the ontology to mediate the meaning of the SD terminologies will help to reduce the inconsistent use of SD terminologies during the software development process.
- The ontology has been described in chapter five and a running ontology for ISDMs is saved as an Appendix C on the CDROM.

7.4 Research Limitations

- The first limitation encountered when conducting the research was evident during the use of a functionality in the ontology development editor Protégé. The software plug-in called OWLViz was unable to function properly. The OWLViz functionality automatically creates class hierarchy diagrams of inferred and asserted classes of the ontology. The diagram was supposed to indicate the relationships existing among the created classes.

- The second limitation was that the data gathering process was only limited to only four respondents. The process should have been extended to a wider range of software development practitioners and academics, increasing the amount of the data gathered, and providing a greater understanding of the SD terminologies that can be added to the ontology for ISDMs.

- The ontology developed is a prototype only describing the Agile Methodologies. Most of the advanced functionalities of the Description Logic (DL) have not been implemented as only the management side of the ontology development was considered.

7.5 Future Work

- In the future, a software development organisation must develop an ontology for Software Development (SD) projects and use the ontology for ISDMs to match the methodologies developed to a particular type of project being conducted. There is a parallel project that is discussing a software development approach that may use the ontology for ISDMs as a base platform for improving the software development practices.

7.6 Summary of the chapter

In this research, an ontology for ISDM has been developed to help the software development practitioners in minimising the inconsistent use of SD terminologies during the software development process. The full ontology for ISDMs including the software used in the development of the ontology (Protégé) is included in the Appendix C. The instructions on how to create and use the ontology have been described in section 6.3 of the research.

REFERENCES

AGERFALK, P.J. 2004. Grounding through operationalisation. Constructing tangible theory in IS Research. (*In* proceedings of the 12th European Conference on IS (ECIS), Turku, Finland.)

ARMSTRONG, D.M. 1978. A World of States of Affairs. Cambridge: Cambridge University Press.

ARNOLD, W.R. & BOWIE, J.S. 1985. Artificial Intelligence: A Personal Common-sense Journey. Englewood Cliffs, N.J: Prentice Hall.

ARPIREZ, J.C., CORCHO, O., FERNNDEZ-LOPEZ, M. & GOMEZ-PEREZ, A. 2003. Web ODE in a nutshell. *AI Magazine,* 24(3):37-48.

ARTALE, A., FRANCONI, E., GUARINO, N. & PAZZI, L. 1996. Part-Whole Relations in Object-Centered Systems: an Overview. *Data and Knowledge Engineering,* 20(3):347-383.

AVISON, D.E. & FITZGERALD, G. 2006. Information Systems Development: Methodologies, Techniques and Tools. Berkshire: McGraw-Hill.

BATEMAN, J.A. 1995. On the Relationship Between Ontology Construction and Natural Language: A Socio-Semiotic View. *International Journal of Human-Computer Studies,* 43:929-944.

BENJAMINS, V.R. & FENSEL, D. 1998. The Ontological Engineering Initiative. (*In* Guarino, N. (ed.), Formal Ontology in Information Systems. *Proceedings of FOIS,* Trento, Italy, Amsterdam, IOS Press. p. 3-15.)

BERNARAS, A., LARRESGOITI, I. & CORERA, J. 1996. Building and reusing ontologies for electrical networks applications. *In* Proceedings of the 12th ECAI, Budapest, Hungary, 36(9): 298-302.

BLUM, B.I. 1994. A Taxonomy of Software Development Method. *Communication of the ACM*, 37(11):82-94.

BOLAND, R. 1985. Phenomenology. A preferred Approach to Research in Information Systems. (*In* Mumford, E., Hirschheim, R.A., Fitzgerald, G., Wood-Harper, T. (eds), *Research Methods in Information Systems*, North-Holland, Amsterdam. p. 193-202.)

BORGO, S., GUARINO, N. & MASOLO, C. 1997. An Ontological Theory of Physical Objects. (*In* Proceedings of Qualitative Reasoning 11th International Workshop, Cortona, Italy. p. 223-231.)

BRINKKEMPER, S. 1996. Method engineering: engineering of information systems development methods and tools. *Journal of Information and Software Technology,* 38(1):275-280.

CASATI, R. & VARZI, A. 1997. Spatial Entities. (*In* Stock, O., (ed). *Spatial and Temporal Reasoning*. Dordrecht: Kluwer. p.143-160.)

CASEY, K. 1995. The new narrative research in education. *Review of research in education*, 21(3):211-253.

CAVANAGH, S. 1997. Content analysis: concepts, methods and applications. *Nurse Researcher*, 4(1):5–16.

CHANDRASEKARAN, B., JOSEPHSON, J.R. & BENJAMINS, V.R. 1999. What are the ontologies, and why do we need them? *IEEE Intelligent Systems*, 14(1):20-26.

CHARMAZ, K. 2000. Grounded theory: objectivist and constructivist methods. http://www.oise.utoronto.ca/CASAE/cnf2007/Proceedings-2007/AERC%20CASAE%20RT%20Storberg-Walker%20Wiessner-2007.pdf. Date of access: 20 March 2009.

CHECKLAND, P. & SCHOLES, J. 1999. Soft systems methodology in action. Chichester: Wiley.

CLIFFORD, J. & MARCUS, G.E. 1986. Writing culture. Berkeley: University of California Press.

COCCHIARELLA, N. 1996. Formal Ontology. (*In* Nijhoff International Philosophy Series, 53. Dordrecht, Boston, London: Kluwer Academic Publishers. p. 1-21.

COLE, F.L. 1988. Content analysis: process and application. *Clinical Nurse Specialist*, 2(1):53–57.

CONNELLY, F.M. & CLANDININ, D.J. 1990. Stories of experience and narrative inquiry. *Educational Researcher*, 19(5):2–14.

CORCHO, O., GOMEZ-PEREZ, A., & VICENTE, O. 2006 Web ODE: An Integrated Workbench for Ontology Representation, Reasoning and Exchange. http://www.dia.fi.upm.es/~ocorcho/documents/EON2002_CorchoEtAl.pdf. Date of access: 29 July 2008.

CRESWELL, J.W. 2008. Qualitative inquiry and research design: Choosing among five approaches. 3rd edition. Thousand Oaks, CA: Sage.

DENZIN, N.K. & LINCOLN, Y.S. 2005. Qualitative Research.3rd edition. CA: Sage.

DITTRICH, Y & SESTOFT, P. 2005. Designing Evolvable Software products: Coordinating the evolution of different layers in kernel Based Software Products. http://www.itu.dk/research/sdg/doku.php. Date of access: 15 March 2008.

DOBING, B. & PARSONS, J. 2007. What practitioners are saying about the Unified Modelling Language, (*In* Proceedings of International Resources Management Association (IRMA). p. 123-126.)

EL BAZE, N. 2005. Cracking Software Development Complexity: In order to reap the benefits of progress in processor technology, we should fundamentally rethink how to write software, Line 56-The Business Executive daily. http://www.line56.com/articles/. Date of access: 30 March 2008.

FALS BORDA, O. 1998. People's participation: Challenges ahead. New York: Apex.

FERNANDEZ-LOPEZ, M, GOMEZ-PEREZ, A. & JURISTO, N. 1997. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. (*In* Spring Symposium on Ontological Engineering of AAAI. Stanford University, California. p. 33-40.)

FITZGERALD, B., RUSSO, N. & STOLTERMAN, E.2002. Information Systems Development: Methods in action. UK: McGraw.

FRAASSEN, B.C. 1989. Laws and Symmetry. Oxford: Oxford University Press.

FRIESEN, M.E. & ORLIKOWSKI, W.J. 1989. Assimilating case tools in organisations: An empirical study of the process and context of case tools. *Centre for Information Systems Research*, 199:1-31.

GAINES, B. 1997. Editorial: Using Explicit Ontologies in Knowledge-based System Development. *International Journal of Human-Computer Systems*, 46(1):181.

GLASER, B. & STRAUSS, A. 1967. The discovery of grounded theory. Chicago: Aldine.

GOMEZ-PEREZ, A. 1994. Some Ideas and Examples to Evaluate Ontologies. California: Knowledge Systems Laboratory, Stanford University.

GÓMEZ-PÉREZ, A. 1997. Knowledge Sharing and Reuse: The Handbook on Expert Systems. UK: CRC Press.

GOMEZ-PEREZ, A., FERNANDEZ-LOPEZ, M. & CORCHO, O. 2003. Ontological Engineering. London: Springer-Verlage.

GOMEZ-PEREZ, A., LOPEZ, M.F. & CORCHO, O. 2004a. Theoretical Foundations of Ontologies. (*In* Welty, C. & Smith, B. (eds.), Ontology Engineering. London: Springer-Verlage, London Limited. p. 1-44.)

GOMEZ-PEREZ, A., LOPEZ, M.F. & CORCHO, O. 2004b. Ontology Tools. (*In* Grosof, B. N. & Poon, T. C. (eds.), Ontology Engineering. London: Springer-Verlage, London Limited. p. 293-361.)

GOTTS, N.M., GOODAY, J.M. & COHN, A.G. 1996. A Connection Based Approach to Commonsense Topological Description and Reasoning. *The Monist: An International Journal of General Philosophical Inquiry*, 79(1):51-75.

GRUBER, T.R. 1993a. A translation approach to portable ontology specification. *Knowledge Acquisition*, 5(2):199-220.

GRUBER, T.R. 1993b. Toward principles for the design of ontologies used for knowledge sharing. (*In* Substantial revision of paper presented at the International Workshop on Formal Ontology, Knowledge Acquisition, Padova, Italy.p. 199-220.)

GRUNINGER, M. & FOX, M.S. 1995. Methodology for the design and evaluation of ontologies. (*In* Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence, Montreal, Canada. p. 37-51.)

GUARINO, N. 1995. Formal ontology, conceptual analysis and knowledge representation. *International journal of Human-Computer Studies*, 43(5/6):625–640.

GUARINO, N. 1997. Semantic Matching: Formal Ontological Distinctions for Information Organisation, Extraction, and Integration. (*In* Pazienza, M.T. (ed.) Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology. London: Springer Verlage. p. 139-170.)

GUARINO, N. 1998. Formal Ontology and Information Systems. (*In* Guarino, N. Formal Ontology in Information Systems. Amsterdam, Netherlands: IOS Press. p. 3-15.)

HARDY, C.J., THOMPSON, B.J. & EDWARDS, H.M.1995. The use, limitations and customisation of structured systems development methods in the United Kingdom. Information and Systems Technology, 37(9):467-477.

HARDY, C., HARLEY, B. & PHILLIPS, N. 2004. Discourse Analysis and Content Analysis: Two Solitudes? http://www.victorialennox.ca/content/Incompatible_Worldviews.pdf. Date of access: 20 March 2009.

HENDLER, J. 2001. Agents and Semantic Web in action. *IEEE Intelligent Systems*, 16(2):54-59.

HENNING, E. 2007. Finding your way in qualitative research. Pretoria: Van Schaik.

HILL, A. 2001. What is critical appraisal? http://www.jr2.ox.ac.uk/bandolier/painres/download/whatis/What_is_critical_appraisal.pdf. Date of access: 12 March 2008.

HIRSCHHEIM, R., KLIEN, H.K. & LYYTINEN, K. 1995. Information Systems Development and Data Modelling: Conceptual and Philosophical Foundations. UK: Cambridge University Press.

HOX, J.J. & BOEIJE, H.R. 2005. Data Collection, Primary vs. Secondary. *Encyclopaedia of Social Measurement*, 1:593-598.

IIVARI, J.1986. An innovation Research Perspective on Information System Implementation. *International Journal of Management*, 21(1):1-10.

IIVARI, J., HIRSCHHEIM, R. & KLIEN, H.K. 1998. A paradigmatic analysis contrasting information systems development approaches and methodologies. *Journal of Information Systems Research,* 9(2):164- 193.

IIVARI, J. & MAANSAARI, J. 1998. The usage of systems development methods: are we stuck to old practices. *Journal of Information Software Technology*, 40:501- 510.

IIVARI, J., HIRSCHHEIM, R. & KLEIN, H. 2000. A dynamic framework for classifying information systems development methodologies and approaches. *Journal of Management Information Systems,* 17:179-218.

JACKSON, M.A. 1981. A Systems Development Method: Tools and notions for program construction. UK: Cambridge University Press.

JACOBSON, I., CHRISTERSON, M. JONSSON, P. & OVERGAARD, G. 1992. Object-oriented software engineering: a use case driven approach. Workingham: Addison-Wesley.

JAYARANTA, N. 1994. Understanding and Evaluating Methodologies, NISAD: A Systematic Framework. Maidenhead: McGraw-Hill.

JAYARATNA, N. 1990. Systems Analysis: The need for a better Understanding. *International Journal of Information Management*, 10:228-234.

JONES, C. 2004. Quantitative and Qualitative Research Conflicting Paradigms or Perfect Partners. *(In* proceedings of the 4th International Conference on Networked Learning. University of Lancaster, Lancaster, p. 106-112.)

KAPLAN, B. & MAXWELL, J.A. 1994. Qualitative research methods for evaluating computer information systems. (*In* Evaluating Health Care Information Systems: Methods and Applications. CA: Sage. p. 45-68.)

KLEIN, H. & MYERS, M. 1999. A Set of Principals for Conducting and Evaluating Interpretive Field Studies in Information Systems. *MIS Quarterly* 23(1):67-94.

LANG, E. 1991. The LILOG Ontology from a Linguistic Point of View: Text Understanding in LILOG. Berlin: Springer-Verlage.

LAURIDO-SANTOS, O. 1986. A course on information engineering. *ACM SIGCSE bulletin,* 18:194-202.

LECOMPTE, M.D., PREISSLE, J. & TESCH, R. 1993. Ethnography and qualitative design in educational research. 2nd edition. San Diego: Academic Press.

LEDERMAN, R.P. 1991. Content analysis of word texts. MCN. *The American Journal of Maternal Child Nursing*, 16:169.

LENNOX, V. 2009. Incompatible Word Views. Discourse and Content Analysis. Oxford: Oxford University Press.

MADDISON, R., BAKER, G.J., BHABUTA, L., FITZGERALD, G., HINDLE, K., SONG, J.H.T., STOKES, N. & WOOD, J.R.G. 1983. Information system methodologies. Chichester: Wiley Heyden.

MARIAS, J. 2001. Historia de la filosofia, Filosofia y Pensamiento, Alianza Editorial. 4th edition Madrid, Spain: Morelia.

MARX, K. 1964. Karl Marx / Friedrich Engels Studienausgabe. (*In* Fetscher, I. (ed.) Frankfurt a. M: Fischer Philosophie.)

MCGUINNESS, D. 1998. Ontological Issues for Knowledge-Enhanced Search. (*In* Guarino N. (ed.), Formal Ontology in Information Systems. *Proceedings of FOIS,* Trento, Italy, Amsterdam, IOS Press. p. 3-15.)

MCGUINESS, D.L. 2002. Ontologies come of age. (*In* Fensel, D, Hendler, J., Lieberman, H., & Wahlster, W.Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential, MI: MIT Press.)

MCMICHAEL, C. n.d. Qualitative Data Analysis. http://www.latrobe.edu.au/publichealth/Units/phe6/phe6hrb/Week%206%202007%20Analysis.pdf. Date of access: 29 April 2009.

MEADCHE, A., MOTIK, B., STOJANOVIC, L., STUDER, R. & VOLZ, R. 2003. Ontologies for Enterprise Knowledge Management. *IEEE Intelligent Systems,* 18(2):26-33.

MILLS, G.E. 2000. Action Research: A guide for the teacher researcher. NJ: Merrill/ Prentice Hall.

MINSKY, M. 1975. A Framework for Representing Knowledge. New York: McGraw-Hill.

MUMFORD, E. 1995. Effective Requirements Analysis and Systems Design: The ETHICS Method. Basingstoke, UK: Macmillan

NECHES, R., FIKES, R.E., FININ, T. GRUBER T.R., SENATOR, T. & SWARTOUT, W.R. 1991. Enabling technology for knowledge sharing. *AI magazine*, 12 (3):36-56.

NOY, N.F., FERGERSON, R. & MUSEN, M. 2000. The knowledge model of Protege-2000: Combining interoperability and flexibility. http://www.aifb.uni-karlsruhe.de/WBS/jvo/publications/aeon_jao_2008.pdf. Date of access: 30 October 2008.

ORLIKOWSKI, W.J. & BAROUDI, J.J. 1991. Studying Information Technology in Organisations: Research Approaches and Assumptions. *Information Systems Research,* 2:1-28.

PALVIA, P. & NOSEK, J.T.1993. A field examination of system life cycles techniques and methodologies. *Information and Management*, 9(2):164-193.

POPPER, K.R. 1963. Conjectures and refutations: the growth of scientific knowledge. London: Routledge.

RAGIN, C.C. & BECKER, H.S. 1992. What is a case? Exploring the foundations of social inquiry. Cambridge, UK: Cambridge Press.

REPA, V. 2004. Methodology Framework for Information Systems Development. (*In* CITSA Conference, Orlando, Florida, p. 2-5.)

ROBERTS, T., GIBSON, M., FIELDS. K. & RATNER, K. 1998. Factors that impact implementing a system development methodology. *IEEE Transactions on software engineering,* 24: 640-649.

RODRIGUEZ-PEREYRA, G. 2000.What is the problem of universals? *Mind,* Cambridge: Churchill College, 109:255-273.

RODRIGUEZ-PEREYRA, G. 2002. Resemblance nominalism: A solution to the problem of universals. Oxford: Clareondon Press.

ROSENKRANZ, C. & HOLTEN, R. 2007. Towards measuring the complexity of Information Systems: A language Critique Approach. (*In* Proceedings of International Resources Management Association (IRMA).p. 57-60.)

SCHREIBER, G., WIELINGA, B., DE HOOG, R., AKKERMANS, H. & VAN DE VELDE, W. 1994. CommonKADS: A comprehensive methodology for KBS development. *IEEE Expert*, 9(6):28-37.

SMITH, J.K. 1983. Quantitative versus qualitative research: An attempt to clarify the issue. *Educational Researcher*, 12 (3):6-13.

SMITH, N.C. & DAINTY, P. 1991. The Management Research Handbook. London: Routledge.

SOWA, F.J., n.d. Building, Sharing and Merging Ontologies. http://www.JFsowa.com/ontology/ontoshar.htm. Date of access: 18 March 2008.

SOWA, J.F. 1999. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Pacific Grove, California: Brooks Cole Publishing Co.
STAAB, S., SCHNURR, H.P., STUDER, R. & SURE, Y. 2001. Knowledge processes and ontologies. *IEEE Intelligent Systems*, 16(1):26-34.

STUDER, R., BENJAMINS, V.R. & FENSEL, D. 1998. Knowledge Engineering: Principles and Methods. *IEEE Transactions on Data and Knowledge Engineering*, 25(1/2):161-197.

SURE, Y., ERDMANN, M., ANGELE, J., STAAB, S., STUDER, R. & WENKE, D. 2002. OntoEdit: collaborative ontology engineering for the semantic web. (*In* Proceedings of the International Semantic Web Conference, Springer-Verlag, Sardinia, Italy. p. 221-235.)

SWARTOUT, B., RAMESH, P., KNIGHT, K. & RUSS, T. 1997. Toward Distributed Use of Large Scale Ontologies. (*In* AAAI, Spring Symposium on Ontological Engineering, Stanford University, California. p. 138-148.)

SWARTOUT, W.R. & TATE, A. 1999. Guest editors' introduction: Ontologies. *IEEE Intelligent Systems*, 14(1):18-19.

TAYLOR, P., RICHARDSON, J., YEO, A., MARSH, I., TROBE, K. & PILKINGTON, A. 1995. Sociology in Focus. Ormskirk: Causeway Press.

TRUEX, D. 1996. The Merits of Three Qualitative Research Methods. Discourse Analysis. http://www.qual.auckland.ac.nz/ Date of access: 15 March 2009.

TRUEX, D., BASKERVILLE, R. & TRAVIS, J.2000. Amethodical systems development: deferred meaning of systems development methods. *Accounting Management and Information Technologies*, 10:53-79.

USCHOLD, M. & KING, M. 1995. Towards a Methodology for Building Ontologies. (*In* Proceedings of the IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing. Montreal, Canada. p. 6.1-6.10.)

USCHOLD, M. & GRUNINGER, M. 1996. Ontologies: Principles, Methods and Application. *Knowledge Engineering Review*, 11(2):93-136.

WALKER, J.C. & EVERS, C.W. 1999. Research in education: Epistemological issues. (*In* Keeves, J.P., & Lakomski, G. (eds.), Issues in Education Research. New York: Pergamon. p. 40-56.)

WANG, Y. 2007. Ontology Engineering: The Brain Gene Ontology. Aukland University of Technology. (Thesis – MA. p. 1-84.)

WARDLOW, G. 1989. Alternative modes of inquiry for agricultural education. *Journal of Agricultural Education*, 30 (4):2-7.

WELTY, C. 1998. The Ontological Nature of Subject Taxonomies. (*In* Guarino, N. (ed.), *Proceedings of the First Conference on Formal Ontology and Information Systems,* Amsterdam, IOS Press.p. 3-15.)

WESTRUP, C. 1993. Information systems methodologies in use. *Journal of Information Technology,* 8:267–275.

WYNEKOOP, J.L. & RUSSO, N. 1993. System development methodologies: Unanswered questions. *Journal of Information Technology,* 10:65-73.

ZHANG, J. 2007. Ontology and the Semantic Web. *Proceedings of the North American Symposium on Knowledge Organisation,* 1. http://dlist.sir.arizona.edu/1897: Date of access: 10 July 2008.

APPENDICES

Appendix A: List of abbreviations.

A

AI	Artificial Intelligence

C

CDROM	Compact Disc Read Only Memory

D

DARPA	Defence Advanced Research Projects Agency
DFD	Data Flow Diagram
DL	Description Logic

E

ETHICS	Effective Technical and Human Implementation of Computer based Systems

I

IE	Information Engineering
IS	Information Systems
ISD	Information Systems Development
ISDM	Information Systems Development Methodologies

K

KB	Knowledge Base
KR	Knowledge Representation

O

OWL	Web Ontology Language

R

RAD	Rapid Application Development
RUP	Rational Unified Process

S

SD	Systems Development
SDM	Systems Development Methodologies
SSM	Soft System Methodology
STRADIS	Structured Analysis, Design and Implementation of Information Systems

U

UML	Unified Modelling Language

X

XP	Extreme Programming

Appendix B: Interview Questions.

AN ONTOLOGICAL APPROACH TO SOFTWARE DEVELOPMENT: Masters INTERVIEW RESEARCH QUESTIONS

Brief Introduction

All information systems that are developed consist of three basic components, database, user interface and applications. Linking all these three is a common schema as shown in fig 1 below (adapted from Sowa, n.d.)

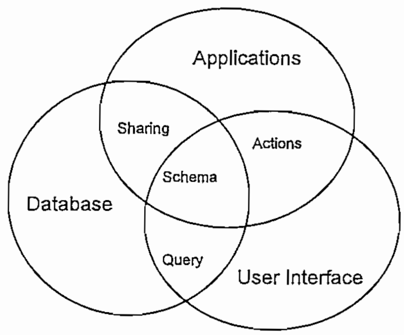

Fig 1: An Integrated Information System

The purpose of this research is to position ontology as an artefact that integrates all three, is part of each of the components and at the same time is a tool that can be used to develop evolvable software products with emergent properties. Such properties are the ones needed in fast growing, interactive and evolving systems.

The research approach is a qualitative empirical investigation into the domain of software development. The researcher investigates practices around the development and use of software in information systems. An analysis of the technical design of the software and information systems is also part of the investigation.

Statement of the problem

The problem software developers continue to face is the lack of a method or tool that can augment current programming language technology and methodologies with semantics based tools to enable construction of evolvable software products. Software kernels (Dittrich & Sestoft, 2005) and software product lines (Carnergie Melon Software Engineering Institute, n.d.) have been used in industry, research, and academia to try and address the speed of reproducing and evolve software products. They have gained widespread acceptance but they still leave a semantic gap that needs to be filled and incorporated in software products. An ontological approach to software development challenges most of the traditional software development approaches and borrows most of its characteristics from the agile methodologies family.

Aim of the Study

The aim of the study is to position ontology as an artefact that can be incorporated in software applications either at running time or as a part of an information system to increase semantic and contextual awareness in information systems. This development will result in information systems that are more dynamic and usable with components that are reusable, shareable, visible and accessible to other systems and users as well.

Objectives of the Study

The objectives of the study can be summarised as the need to develop a general ontology framework that satisfies the aims of the study. The importance and validity of the framework will be justified using data gathered from developers of information systems. More specifically the objectives of the study are:

- To develop or construct a tangible ontological theory within the field of software development that can be used to design and develop romantic information systems. The abstract ontology concept will be operationalised into a concrete form of knowledge that can be applied readily in software development (Agerfalk, 2004). Operationalisation is the process of converting or moving from abstractions to some concrete artefacts that have practical applications in a social practice such as inter-organisational systems.
- To develop a linguistic model that can be used by software agents to communicate in information systems.
- To position ontology as an artefact that is needed by software agents to experientially and intuitively develop their linguistic models needed for communication in information systems.
- To develop an ontology model-driven-approach to software development (El Baze, 2005). This approach will ensure that the resultant model moves from a

purely abstract conceptualisation of an ontology model to a truly operational model. This context-aware, purpose-specific actionable model can be deployed as a value-added service layer in software development processes.

Purpose

The aim is to understand and explain the software development practices and designs from a practitioners' viewpoint. This is explained out of the historical and situational context of software development. The research, as a secondary aim looks at driving for a new paradigm in systems development.

The questions will be open and unstructured. Where the situations are permitting, observation will be used to compliment the data gathering process.

Activities

The researcher investigates software development approaches, methods, techniques and tools that are used currently in software development. Considering the pervasiveness and ubiquity needed in current day information systems, the activities will unravel the discrepancies between current development methods with the one proposed and strongly motivated for by the literature study, that is, an ontology driven development approach.

SECTION A: These few questions is on demographic data

1. For identification purposes only, can you tell me your name or the name of the company you represent?
2. How long have you been involved in software /system development?
3. What type of software products or information systems does your company specialise in?
4. What is your role in the company?

SECTION B: These questions look at the different software/system development practices from the participants' viewpoint.

The questions generally look at the paradigm aspects of software development.

Question 1:

There are several software development paradigms in widespread use today.

a) Which paradigm do you advocate when developing software products in your organisation?

b) What is the main reason why you use this paradigm?

Question 2:

Some people say the difference between the structured methodology (use of process and data models and use of DFDs and ERDs among other techniques) and the object-oriented paradigm (use of use cases, classes and activity diagrams with UML as a technique) is that the latter captures semantics of a system. What is your opinion of that?

The researcher needs to find how context and meaning in systems is captured during development time and running time of software products.

'There is a reason why computers have not yet become fervent natural language speakers. (It's not a matter of processing power and never will be): we simply are not programming them correctly.'

(El Baze, 2005)

Question 3:

What is your comment on this?

Question 4a:

It is common knowledge that information systems do not behave the same way as humans do. What do you suggest should be done to add this human-like behaviour in information systems artefacts?

Question 4b:

Information systems comprise three parts, the formal, informal and the technical part. Can you explain how the formal and informal part is captured during your software development processes?

Question 5:

Organisations as systems exhibit a lot of "unstated" assumptions that are reflected through the shared (common sense) knowledge of people familiar with the social businesses and technical contexts within which the proposed system will operate (Rosenkranz & Holten, 2007:58) - the pragmatics. What can be done in software development to capture these assumptions?

Question 6:

Every organisation runs information systems within an organisational culture and context. When developing software products, how can developers ensure that this organisational culture and context are captured and implemented in the software products?

The next group of questions looks at communication problems during software development.

Question 7:

a) When communicating with clients, what communication tools and channels do you use in your organisation?

b) "It is easier for clients to understand the functionality of the software through the user interface sketches" (Dobing & Parsons, 2007:123). What is your view?

Question 8:

During software development, how can we improve the clients' understanding of the functionality of the proposed software?

Question 9:

In which stages of software development would you involve the client?

Question 10:

How do you ensure that the understanding of the client of what the system should do is correct and maintained throughout the development process from analysis to maintenance stages?

Question 11:

Communication is a very important facet of the software development process. Many people decry the lack of a communication language /tool that is understandable by both users (clients) and developers. What tools can be used to communicate between client-analyst and between analyst-programmer during software development?

Question 12: *(Check question 7 for similarity in answers)*

What type of clients do you have? Do they have knowledge of system development or software development tools?

Hint: What do you use to communicate with these clients?

The next group of questions looks at the ability of ontology driven case tool to retain the system context gathered during analysis through to maintenance.

Question 13:

What do you think can a software development tool that integrates the software development tasks and data or information gathered from analysis through to implementation improve on the software development process?

Question 14:

It is recommended that when you develop a software development approach or methodology, one must also develop an accompanying development tool. The tool has considerable influence on how the approach or methodology will subsequently be used. What functions or tasks would you want to see in a development tool that supports a development approach/methodology that allows adaptability and reusability in software products?

Question 15:

What is your understanding of adaptive software products?

Question 16:

How can you ensure the development of adaptive software products?

Question 17:

What is your understanding of evolving software products or systems?

How can you ensure the development of evolvable products?

This group of questions, among other things investigates issues affecting software development productivity. Ontology has the capability to improve software productivity as will be discussed in the thesis.

Question 18:

How do you ensure requirements gathered during analysis and design specifications are reused?

Question 19:

As a software developer, you have developed software products for different organisations that are in the same industry. How do you reuse knowledge gained from one development process to the next project?

Techniques that mediate between a dialect conflict of context and meaning

Question 20:

The success of any software development process depends on the ability to create a language community among stakeholders. This leads to the creation of shared knowledge that will later be captured in the resultant software product or system. The shared knowledge reflects itself through the concepts that will be used in a specific domain. How would you as a developer, mediate when working in the same industry, if the same concept is given different meanings? That is, there exists a dialect conflict of context and meaning (context versus meaning).

Software success metrics looking at quality of software products.

Question 21:

How do you ensure that the software development process is:

a. Within budget?
b. On time?
c. Easy to modify?
d. Faulty free?
e. User satisfied?

Question 22:

How do you manage scope creep (handle) but at the same time ensuring all of the above?

The following questions look at design problems that can be handled using ontology. The technical nature of the ontology artefact requires that questions become very general and the research extrapolates the technical responses at the data analysis stage.

Question 23:

How do you capture semantics (meaning) of designs, specifications in your systems?

Question 24:

How do you ensure that systems are developed within a specific application (domain) context?

Question 25:

How can we improve the changing and communication of a software product design change during coding time?

Question 26:

Discuss some of the challenges that you have encountered pertaining to developing software products that are reusable.

Hint for Interviewer:

- *Design for reuse.*
- *Cost of reusing the components*
- *Archiving and retrieval of reusable components*
- *Accessibility of the source code- if not home grown.*
- *Adaptability and evolvability of source code components*
- *Any other / maintainability*
- *Progress measure during software maintainability / development*

END OF INTERVIEW

Appendix C: Compact Disc Read Only Memory (CD-ROM).

The Appendix C contains the following items listed below:

- The Software Package Protégé 4.
- The ontology for Information Systems Development Methodologies.
- The interview transcripts.

Appendix D: Language editing.

Elma de Kock
Box 19184
Noordbrug
2252
083 302 5282

To whom it may concern:

Dear Sir/Madam

This letter is a confirmation of the fact that the language in Willie Shawa's Masters Dissertation was edited by me.

Thank you

Elma de Kock