CHAPTER ONE

INTRODUCTION AND RESEARCH MOTIVATION

1.0 Introduction

"…So many books are produced to meet an identified market opportunity rather than written because an argument presses to be expounded."

Peter Checkland, 1999.

Much research has been done to address a market-related need quickly, to the detriment of the soundness and quality of the knowledge generated from such an endeavour. Although many market research projects are carried out within specific paradigms it is rare to find research projects that are intended to improve on or establish new paradigms in specific fields. Instead, the results from market-driven research usually create loopholes in their application areas, rather than eliminating these. This is because the grounding paradigms have been overtaken by new market requirements or societal changes. Market-driven results are often portrayed as comprehensive models, frameworks, architectures or as tangible artefacts but the most overarching problem is their lack of comprehensiveness. Comprehensive information system (IS) products are fundamentally based on the generally accepted design principles of systems engineering (Gerber, 2006).

In the present research, a balance is reached between addressing the needs of current and future information systems, that is, reacting to a market need as well as addressing a development approach need that is heavily grounded in the softer, neo-humanist paradigm. Checkland (1999) noted that, although many research projects claim to be introducing a new development paradigm in software development, they are merely new methodologies that are grounded in old paradigms. Checkland (1999) fell victim to this when, in an attempt to move the system development approach to a new paradigm, grounded in the soft systems approach, he was unable to outline a holistic, methodological approach that minimized the influence of the hard systems approaches after the analysis stage. In his case, after he employed a softer approach during analysis, the design and implementation stages were left to rely on the mechanistic dictates of the hard system paradigm. In his soft systems approach, the design and implementation stages lack the softer elements captured during the earlier analysis stage.

As will be noted in Chapters 4, 5 and 6, there are a plethora of software development approaches that are grounded in two development paradigms: the hard systems and soft systems development paradigms. The traditional development approaches, that is, the structured approach, the object-oriented approach and the agile approach, are heavily inclined to the hard systems paradigm. The only difference between these approaches is the methodologies they use.

Hence this research fills the knowledge gap in the fields of information systems and software development through the development of an ontology-driven approach to software development that is heavily grounded in the soft systems paradigm. The present research, therefore, explores and contributes to an approach to software development that has evolved in the research community but which is only sporadically adopted by software development practitioners. The major reason for its being sporadically tested and applied in practice is the lack of guiding principles in the use of this method. The proposed approach is based on an ontology-driven software development framework that lists a set of ontologies that should be developed and used in each phase of the development life cycle. This is an attempt to link the natural world, the social world together with the artificial world of human constructions during software development. As Gregor (2006) noted, information systems can only be understood if theory that links these three elements is found. Currently, many theoretical constructs that are used in the development of IS artefacts do not address this holistically. IS researchers are therefore encouraged to pull together the three disciplines of the natural sciences, social sciences and design sciences in order to address the requirements of IS development problems.

1.1 Research Motivation

"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.

The field of software development is characterized by mechanistic development practices that denote the software development field as rational and deterministic. Consequently, when the resultant software products are implemented in information systems, these systems exhibit a mechanistic character that eventually limits their usability. Mechanistic systems are based on the concept of explicit programming (Agentis, n.d.). Explicit programming produces software products that do not capture semantic or context-rich data, characteristics that are needed in most modern systems. However, these systems are very efficient in structuring data to enable and facilitate its interpretation.

Mechanistic practices also overlook the notion that, as representatives of human principals, information systems should forge some type of humanistic and non-deterministic behaviour. This behaviour can only be captured in the software product.

Some examples of mechanistic practices in systems development manifest themselves in products such as the conceptual schema concept (Sowa, 2000), requirements, design and implementation specifications. This conceptual schema is widely used in database management systems and knowledge bases.

Some mechanistic practices also manifest themselves as software development approaches, such as the structured approach (Pressman, 2005), object-oriented approach (Pressman, 2005, Dennis *et al.,* 2002), the software product lines approach (Carnegie Mellon Software Institute, n.d.) and the software kernels approach (Dittrich & Sestoft, 2005).

Software kernels (Dittrich & Sestoft, 2005) and software product lines (Carnegie Melon Software Engineering Institute, n.d.) have been used in industry, research and academia in an attempt to speed up the process of reproducing and developing software products. These rely heavily on the speed of program generation which is an example of a mechanistic explicit systems implementation strategy. Although they have gained widespread acceptance, they still leave a semantic gap in software products that needs to be filled. Many of these software development approaches have worked in the past but are starting to fail in the current organizational environment and could be worse in the future. This can be supported by Trim and Lee (2004:478-479) who noted that “the models, concepts and theories” that have been used in software development in the past cannot be used to solve peoples’ development problems in the future. This, therefore, necessitates research in new approaches to software development.

Several attempts have been made to reduce the mechanistic tendencies of software products. Harris *et al.* (2009) proposed controlled flexible approaches that allow flexibility in the development process but which are not too prescriptive, such as the waterfall approach, widely known as the plan-driven approaches. They argued that every software development approach requires some controls and that future research has to look at the controls that should be included in the development methodology and at how they should be introduced. The research should indicate whether they are outcome or emergent outcome controls (Harris *et al*., 2009). Wand and Weber (1990:63) strongly advocated for a system development process that would capture "structure (statics) and behaviour (dynamics) of the real world". As they argued, to achieve this, researchers must use ontologies to provide both "sufficient human-oriented and machine-oriented" descriptions of real world systems (Wand & Weber, 1990:63). Whitten *et al. (*2004) encouraged system developers to design their systems for growth and change. Pressman (2005), in discussing some software myths, raises issues such as the ability of software to evolve, quality measurement, throughput levels, reusability of software products, management of scope creep during development and documentation of software products, as some of the most misunderstood and difficult aspects that need to be handled carefully during a software development process. This misunderstanding stems from the fact that a socially constructed organizational system is forced onto a machine-like environment through the process of deconstruction, forcing it to lose some of its life states, hence losing its human elements.

'Towards an ontology-driven software development approach: An unended quest' is a quest for finding and defining an "effective problem representation" (Hevner *et al*., 2004:83). The approach discussed herein, as a construct can prove to be very crucial for the effective design of software solutions by software developers.

The Evolutionary Software Paradigm

Unlike software artefacts that are static, organizations are dynamic and always in a continuous state of change (Kawalek & Leonard, 1996). This posits software products at a difficult position to represent the true state of an organization. Legacy systems are, because of these reasons, difficult to adapt and maintain. Kawalek and Leonard (1996) and Meso and Jain (2006) argued that software products should be innovative, adaptive and replicable. Naturally change is endemic in organisations; therefore new methods of developing software products that exhibit these characteristics should be formulated. For this reason, Kawalek and Leonard (1996:189) advocated for development methods and practices that produce "instantly adaptable software that is able to support radically changing demands on a series of fast developing platforms and integrating with a series of end user developments".

The Software versus Organizational Paradox

Kawalek and Leonard (1996) discussed a paradox where software products hinder organizational change and progress. They reason that a holistic, real world system is forcibly embedded in a piece of software as a representation of an organization. If viewed in conjunction with Kant's Philosophy of Deconstruction, the piece of software is always a partially understood description of the status of an organization at any one particular point in time. Unlike software products, organizations are always in a state of flux (Dahlbom & Mathiassen, 1993; Kawalek & Leonard, 1996) which constantly change and duplicate in their existing environments. There is therefore a need for developers to match the static nature of software products, the dynamic nature of the organization together with the dynamic nature of the software model. Lehman (1991) noted that software products have to be evolutionary if they are required - and are able - to solve problems in the real world domain. This is however complicated by the fact that evolving systems rely on the organizational context, which itself is dynamic and a software product development process that is uncertain.

Current software products can be classified as the static type (S-type). These static type products stress that the correctness of a software design is the only criterion for the success of a software product, coupled with its fidelity, "in a strict mathematical sense" to the specification (Kawalek & Leonard, 1996:189). Because of this, software products cannot move from one steady state to another steady state if they are to reflect and implement what occurs in organizations. In other words, software products cannot be optimized against static goals since organizations are not static. All these arguments point to the fact that software development requires a different approach than the current hard approaches. The new approach should embrace the socio-technical nature of software products and recognize the duality of organizational context and that of developing software products. Besides the software and organizational paradox discussed herein, the software development field also faces several other problems, some of which can be heavily attributed to the communication methods used during development.

1.1.1 Communication Requirements in Software Development

Most of the problems encountered in software development can be attributed to deficiencies in the communication methods (*Section 4.3*), methodological processes, i.e.

the approaches (*Section 4.8*), techniques and tools used during the development of the software product. Stakeholders in software development, such as customers, analysts, designers and programmers, are faced with a serious problem of communication during software development. Modelling notations and languages such as data flow diagramming (DFD), entity relationship diagrams (ERD) and unified modelling language (UML) tends to be too technical and varied to be understood by system users. At times even analysts have problems of choosing a communication tool that can be understood by programmers. Contrary to most common beliefs, unified modelling language is not as unifying a communication language among all stakeholders as most developers and its inventors would have wanted (Dobing & Parsons, 2007). A real unifying language or communication tool is needed to lessen the burden of communication during software development.

Another problem is that, during system specification, many software development tools, modelling tools or case tools do not fully comply with the software development specifications. Examples of this are the UML specifications. Of the tools available, many have minimal application or use in big projects (Dobing & Parsons, 2007). Instead, Dobing and Parsons (2007:125) call for a development tool "that provides 15% of what is needed" in terms of integrating the different development tasks and providing a uniform, common language interface for all stakeholders. This implies that, of the available tools, none can provide a functionality that is more than 15% of the basic requirement of a software development environment (SDE). The nature and form of the development tool that should accompany any system development approach has therefore "a considerable influence on how the underlying methodology will be used" (Dobing & Parsons, 2007:125). Hence the choice of such a tool is very important in the development of pieces of software. The next section discusses the researcher's interest in this field.

1.2 The Research Interest

"The problem is that data is trapped in hierarchical silos, restricted by structure, location, systems and semantics. The situation has become a data graveyard."

Sheryl Torr-Brown (In-PharmaTechnologist.com, 2005.)

A global information society requires data, information and knowledge to be freely accessible and shareable. Despite being archived in disparate information silos, an exchange medium, or obligatory passage point should be introduced that allows unrestricted access to this data by different users in different organizations (Introna, 1997; Mavetera, 2004b). The medium could be a component of information systems. In a follow up to these requirements, research has focused on the development of the Semantic Web (W3C). This is a software development project that endeavours to provide documents and services that have meaning to the user. This meaning is captured as formal descriptions (meta-information) (Lemmens, 2006) lying on a platform of machine processable knowledge structure referred to as ontologies. Ontologies, its touted, as a medium in information systems, help to improve the capturing of the semantic content and of the pragmatic contexts of information systems.

However, the problem that continues to face software developers despite the introduction of ontologies is the lack of a method or tool that can augment current syntactic programming language technologies and software development methods through the addition of semantic-based tools to facilitate the development and construction of software products. Large data, information and knowledge silos exist in organizations but their storage structures, location and the systems that access them continue to be a limiting factor for their accessibility and usability (In-PharmaTechnologist.com, 2005). Organizational systems then become graveyards, as their resources cannot be accessed and used. Because of this a methodology for building information systems has to be developed that encourages sharing and adds meaning and context to the data and information.

1.2.1 Gap in the Field of Study

In industrial and academic research, as new problems arise, researchers have to face the challenge of finding solutions to these problems. As Basden (2001) explains, the problems that limit the usability of information systems and the lack of return on investment from these investments should be tracked back to the approaches, methods and processes that are used at the time the *"artefact"* is developed. This artefact is the software product.

It is the software product that is used to implement all the three basic (software based) technological components of an information system, that is, the database, user interface and applications (Sowa, 2000). At the same time, researchers also realized the importance of changing the syntactic nature of software products and, subsequently, of information systems. They thus focused their efforts on finding techniques and tools that could be used to change the syntactic nature of these products. However, they forgot to re-examine the way in which software products are developed.

This research has as its foundation, the requirement for reasoning to use concepts. Concepts are captured and manipulated by means of machines. At the same time they have a situatedness that is defined and bounded by the context of the situation. This approximates the way in which humans communicate and reason in organizations. It is for this reason that, Lemmens (2006) motivates for a paradigm shift in the development of information systems. He argues that current systems are too data-centric and that developers should move to functional centric methods of developing systems. At a functional level, since processes capture the behavioural aspects of systems, their focus will neglect the use of instances of data as system behavioural representations.

Data-centric systems are mechanistic. Their mechanistic nature can be attributed to the greater emphasis placed on the technical aspects of the systems development process than on the softer management side of this process (Checkland, 1999). In support of this, Hohmann (2007) and Beynon *et al.* (2008) advocated for the development of intuitive systems that are easily understood by humans at the same time increasing the productivity gains from their use. Beynon *et al*. (2008) argued further that marrying together intuition and software development has been made difficult because researchers and developers use frameworks that are inherited from the computer science discipline. These frameworks concentrate on "stable contexts of experience that can be engineered to exhibit law-like characteristics" (Beynon *et al*., 2008:4) and they do not allow some degree of freedom in cognition (Tarnas, 1991). The developers of mechanistic systems have concentrated on abstracting and representing organizational systems patterns and interactions and uses that can be automated.

In order to address the mechanistic nature of systems, industrial and academic researchers have focused their attention on the semantic properties of ontologies**.** This again is proving to be another futile exercise, because they are not considering how the ontology properties could be introduced methodologically in the development of software products. This is widely supported by Isabella (1990), who noted that many research studies focus on the design and development of concrete and observable aspects of ontologies in information systems, but that very few have paid attention to the identification and understanding, the interpretations and cognitions associated with the use of ontologies in information systems. Also, despite all the efforts to use ontologies in information systems, there is still no clearly defined purpose of IS ontologies. This is blamed on the lack of a clear distinction between philosophical ontologies and IS ontologies. Zúñiga (2001:188) believes that IS discipline is "either not equipped to advance general ontologies or (is) not employing the right methodology or theoretical approach" to the use of ontologies in this field. In short, despite these discussions, there is very little evidence of research on the methodological implications of introducing ontologies in software development. The present study takes this as its departure point and will try to address this.

1.2.1.1 The Software Development Process

Gonzalez-Perez and Henderson-Sellers (2006) characterize the software development process as a complex activity in which people, technologies and their organizations are participants. The effects of these participants on the software development process and the use of the software development products are discussed in *Section 4.6.3: Human Activity Systems*. In addition, there are various types of stakeholders in software development, whose diversity contributes greatly to the complexity and difficulty of developing software products. To add to the stakeholder diversity, there is no common vocabulary or world view that can be understood and used by all participants in the development process. The requirement of such a vocabulary is discussed in *Sections 4.8.4, 4.9 and 5.7* of this thesis.

On the software side, the development of software products and information systems has been bedevilled by the lack of a development approach that captures the human aspect of organizational systems. This problem is deeply embedded in the sociological development paradigms that are adopted in the analysis of the problem area that needs to be addressed by the subsequent software products. Depending on the paradigm adopted, whether functional, humanistic, radical structuralism or interpretive (Burrell & Morgan, 1979) (*see Section 2.5)*, the resultant analyses of the problem area will differ. At present, many software products are analyzed using the functionalist paradigm, a notion that has seen a plethora of mechanistic software products being developed and implemented. Strongly attached to this paradigm is the use of the reductionist dogma that also supports systematicity and system formation principles *(see Sections 4.5 and 4.5.1).*

This development approach, while addressing the data-processing requirements of most organizations, fails to address the dynamic nature of these organizational systems. Being themselves dynamic, organizations require dynamic systems to accomplish their tasks and achieve their information-processing goals. Dynamic systems are best modelled using predicative models, since these allow for “formal analysis of system behaviour” (Lemmens, 2006:144). In the context of this study, a task is an action performed in an attempt to accomplish “a goal in a particular context” (Lemmens, 2006: 144) and a goal is an intended achievement by an actor, such as a system or a person.

With the advent of the Internet and global networking infrastructure, ubiquitous and pervasive computing system environments have grown very large. This also extends the social domain that needs to be addressed by software systems. A socially grounded ubiquitous and pervasive environment such as this requires software products that can be adapted to the change in the processing and social requirements that are imposed by the demands of such a massive processing web.

At the same time the social aspects of computing environments require a shift in the paradigm that is consulted during the development of a software product. It is therefore essential that researchers and software developers come up with a development approach that is enshrined in the philosophy of interpretive or humanist sociological paradigms. This is supported by Kirlidog and Aytol (2010) who saw a pressing need for new methodological approaches to software development that could improve the quality of the software products.

1.2.1.2 Software Development and Ontologies

With the advent of ontologies and their use in software development, many researchers have quickly moved to incorporate these artefacts in the production and use of software products.

On the ontology research side, work has been done on ontology development, representation languages: OWL, DAML and Protégé. Tools for ontology building and mapping, semantic mark-up languages and mark-up of resources such as web services have also been developed. On the interoperability side, semantic interoperability frameworks (Mavetera, 2004b; Mavetera, 2007 and Lemmens, 2006) have also been suggested, some of which have been implemented. Several uses for ontologies in conceptual modelling have also been discussed (Wand &Weber, 1993; 2002; Sugumaran & Storey, 2002; Shanks et al., 2003).

1.3 Research Methodology

Every research project needs to be guided and informed by a specific paradigm, ontology and epistemology of the research area (*See Sections 2.5 and 2.6*). These three aspects will enable the researcher to understand the nature of the research problem and ultimately to plan one's methodological stance. The methodological process, like the three aspects of paradigm, ontology and epistemology, exhibits a dichotomous relationship (Fitzgerald & Howcroft, 1998) between qualitative and quantitative methodologies. At this point, the researcher needs to choose the research method that is best suited to the nature of the research problem at hand.

By definition, a research method is a strategy of inquiry which moves from the underlying philosophical assumptions of the research problem to research design and data collection. The choice of research method influences the way in which the researcher collects and analyses the data. Specific research methods also imply different skills, assumptions and research practices (Myers, 1997). The present study is of a qualitative nature. As such, Grounded Theory Method (GTM), the method used in this research, requires the researcher to start without any preconceived theories about the problem to be investigated. However, as will be explained in Chapter 3, preliminary research questions were used in this study. A brief description of this method is given in Section 1.3.1 and the method is fully discussed in Section 3.4.

1.3.1 GTM

Many people confuse grounded theory, i.e. the inductive theory generated using GTM with the methodology (GTM) itself. The GTM is an inductive type of research method that seeks to develop theory that is grounded in data, that is, grounded theory (GT) (Glaser & Strauss, 1967; Glaser, 1992). According to Olivier (2004), GTM starts by observing the field of interest and theory is allowed to emerge from (is grounded in) what is observed in the data. It is important that any bias in data collection be limited to the barest minimum, by the researcher desisting from reading literature in the substantive area of research, starting with a problem statement or hypothesis, hence the research questions as advocated by Glaser (1992).

Research data are then collected systematically and analyzed as and when they are gathered. According to Cornford and Smithson (1996), these data can be used to develop (induce) the final hypotheses, propositions, themes and classifications as the study progresses. Also, as an inductive, theory discovery methodology, GTM "allows the researcher to develop a theoretical account of the general features of a topic while simultaneously grounding the account in empirical observations or data" (Martin & Turner, 1986:141).

From the analysis of the first data samples, Glaser (1992) encourages the researcher to look for an emerging problem and theory. This becomes the tentative or preliminary theory, which will be adjusted continuously as new data are gathered and analyzed (Cornford & Smithson, 1996; Olivier, 2004). The basic tenets of GTM call for a continuous interplay between data collection and analysis.

GTM is extremely useful in developing context-based, process-oriented descriptions and explanations of the phenomenon (Orlikowski, 1993). A process of purposive sampling is the key to the success of this method, which can only be stopped if saturation level is reached. Saturation level is defined as the point at which the addition of new data from interviews or observations does not change the quality of the knowledge already gathered (Glaser & Strauss, 1967; Glaser, 1992; Olivier, 2004). In GTM, theory is something that shows itself after the collection and analysis of some or all of the data associated with or relevant to the research problem (Bryman, 2004). Lastly, the reader is reminded that GTM frees the researcher from the bondage of longstanding assumptions (Suddaby, 2006), such as starting with a problem statement and the like.

As a qualitative type of research method, research data are best gathered using data-gathering strategies such as interviews, questionnaires, analysis of Internet documents and literature surveys. In this study, literature surveys and interview strategies were employed. The data analysis was done using a text-analysis software product called Atlas.Ti. For a more comprehensive discussion of the data gathering and analysis processes, the reader is referred to Sections 6.1 and 6.4.

1.4 Generic Research Propositions

In support of Glaser and Strauss (1967) and Glaser (1992) that one must not start with hypotheses, this research started with preliminary propositions and, hence, preliminary research questions were also developed. These propositions can be supported from the background readings of Sections 1.1 and 1.2 above.

Proposition A:

The field of software development needs a framework that can be used in the development of romantic software products.

Proposition B:

The software development process can be improved by using an ontology-driven approach to software development.

These two propositions are grounded in the fact that the development of organizational information systems has long been declared to be "*a crisis*". Despite several attempts at improving the methodological process of developing these systems, little has yet been achieved. Researchers have directed their attention to the way in which software products, the aggregations of which make up systems, are being developed. Attention is being paid to the paradigm requirements of addressing socio-technical problems such as organizational systems. Ontologies are portrayed as facets that can address this paradigmatic problem. The framework requirements should, therefore, consist of ways of including ontology components in software development. These two propositions are further refined in Chapter 6, after the GTM process of open coding.

GTM does not encourage researchers to presumptuously propose aims and objectives of the research. However, based on Strauss and Corbin (1990)'s dicta, preliminary research questions are proposed herein. These questions emanate from the research interest highlighted in Section 1.2 and the above preliminary research propositions.

1.5 Preliminary Research Questions

Most research consists of what investigators refer to as the main and minor research questions. The answer to the main research question, if comprehensively dealt with in the study, will meet all the requirements of the investigation. The main research question can be broken down into several smaller questions whose sum total is equivalent to the single main question. In this study, these smaller questions are referred to as minor research questions. The success of any investigation depends on the ability of the researcher to formulate and answer the research questions that are derived from the problem statement, in this case, the research interest. This section is primarily concerned with the development of these preliminary research questions. The final research questions are discussed in Chapter 6 after initial data analysis.

In the present research the GTM was used as a methodology of study. In an attempt to align the research process with the soundness of the research method chosen and in compliance with its methodological dictum, the final research questions were formulated but only after the collection and analysis of three initial data samples. The preliminary research questions are also necessitated by the need to guide, bind and direct the research (Strauss & Corbin, 1990), as well as to prevent it from spiralling out of context and scope. As the proponents of grounded theory, Glaser & Strauss (1967) would have liked, and as was strongly emphasized by Glaser (1992) in Basics of Grounded Theory Analysis, the final research questions, were allowed to emerge from the data.

1.5.1 Main Research Question

From the research propositions, the main purpose of the present research was to find an approach, a framework of components that could be used to improve the development of software products. The research questions are meant to guide the scope of the investigation using GTM. Unlike many research studies that present a single main research question, this study has two main research questions, covering the software development field and ontology field as separate substantive areas respectively. The main questions to be addressed, therefore, are:

'What are the components of a software development approach that can be used to develop romantic software products?'

AND

'What is the role of ontologies in the development of romantic software products?'

In ubiquitous and pervasive environments such as the web, developers currently use intelligent software agents. These agents have gained much popularity in organizational information systems, in which they traverse different information silos in search of information that satisfies their human principals' request. Examples of these are the electronic market systems (Mavetera and Kadyamatimba, 2003). In these systems, application programs and databases are all linked together to form a large network that can be traversed and searched by software agents. Ontologies have found uses in the development of knowledge bases to complement these databases or application packages. This increases the communicative, semantic and contextual capabilities of these systems.

This research investigated the role that ontologies can play in improving the software development process and data awareness in systems, in adding meaning to information systems and in improving the pragmatic aspects of impact and knowledge in information systems. This objective was achieved by focusing on how the software product as an artefact is currently being developed.

1.5.2 Secondary Research Questions

In order to formulate the secondary preliminary research questions, the Process Based Research Framework (PBRF) developed by Roode (1993) was used in this study. The PBRF, as a tool for developing research questions, gains its validity in its ability to capture the ontological, phenomenological, epistemological and normative characteristics of a research problem. It is generally agreed that each problem can consist of four generic research questions. These are shown in Figure 1.1 as the 'what?', 'why?', 'how should?' and 'how does?' type of the problem statement.

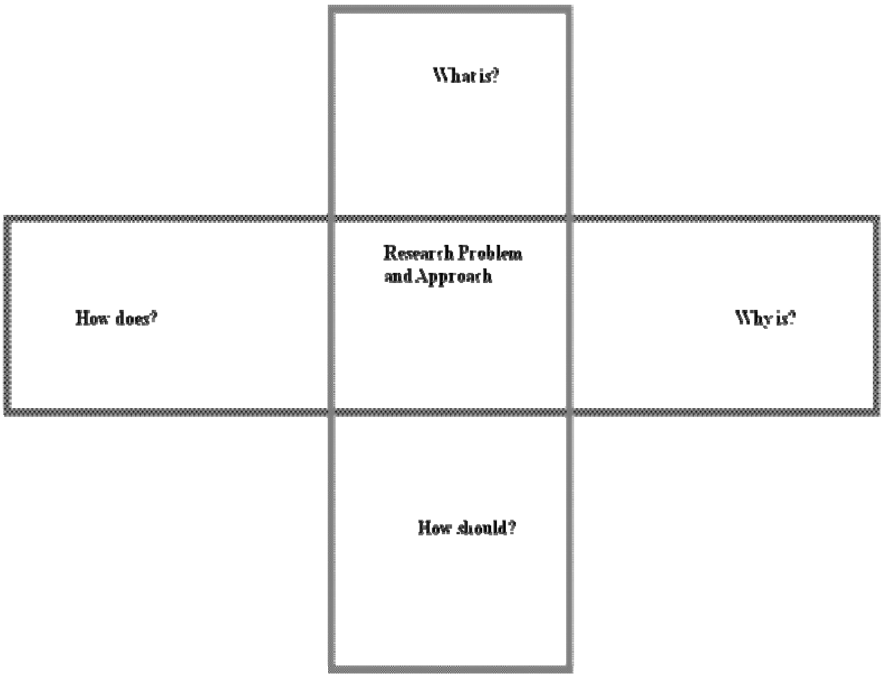

Figure 1.1: Preliminary Research Questions (*Adapted from Roode, 1993*)

1.5.2.1 The 'What is?' type of questions.

These types of questions explore the fundamental nature or essence of the research problem. When these questions are answered, the structure of the problem is made explicit and the ontological and epistemological aspects of the concepts used in the research are discussed. A critical and radical assessment of the problem domain and the underlying paradigms is the focus of such type of questions. The existential proof of the problem is also explored. In this study, the following questions were identified:

What is a mechanistic software product?

This research was driven by the need to improve the usability and adaptability of current information systems. It is, therefore, necessary to discuss the current methods of information system development that lead to mechanistic system development. Mechanistic information systems can be described as systems that rely on - and are based on - the ability of computer systems to represent and arrange the signs that are used in a language. These systems can relate one sign to another but they do not give the meaning of these signs to the user. The systems allow efficient and effective computation and structuring of data but still remain very syntactic. Good examples of these systems are the popular electronic data interchange systems (EDI) used in electronic business systems. As mentioned earlier, these do not provide meaning and context to the user of what has been processed. The user requires another cycle of interpretation in order to understand the output from these systems. In Tarnas' (1991:266) words, the material particles captured and represented in these systems possess "neither purpose nor intelligence".

In this thesis, the notions of explicit programming that have been in widespread use in the fields of software engineering are discussed. The researcher then discusses the more accommodating methods that use ontology and lead to romantic systems development.

What is a romantic software product?

In this study, romantic information systems (RIS) are not only limited to the dictates of syntactic machine representations. They are also based on the romantic world view that considers the "world as a unitary organism" in sharp contrast to the rational atomistic view of the mechanistic world (Tarnas, 1991:366-7).The idea for romanticism in information systems is supported by Hohmann (2007:18), who calls for a "pluralisation of our culture and the humanization of technology". In his vision, he sees a future that demands technologies that stimulate creativity and inspire thoughts, thereby reconciling the "contradictions between technology and art" (p.18) that characterize the modern era. The ontological approach discussed in this thesis captures the requirements of implicit programming that lead to reusable, context-aware software products. Along the way, the research justifies a methodological approach to software development needed in the development of romantic software products.

The notion of romantic software products as touted in this study requires some elucidation. These products should possess human and behavioural characteristics as those found in organizations where humans play a central role in task management. The research will, therefore, use the following working definition of romantic systems:

'Romantic information systems are socially-constructed systems that capture and maintain the softer elements of organizational systems such as culture, social context, and semantics and to an extent pragmatics.'

To borrow Beynon *et al.* (2008:5)'s diction, romantic information systems (RIS) should have a balance of the formal methods evident in classical computer science and the informal indeterminate pragmatic methods that "focuses on the complexity of the sociological processes that surround software practice and experience and invokes a more human centred approach". It must be noted that the current researcher is possibly the first to coin these systems 'romantic information systems'. However, a romantic information system can be viewed as one that has "gloried in the unbounded multiplicity of realities" (Tarnas, 1991:368) that are realised in organizations as a result of the subjectivity and the divergence in perspectives of the people found in them. Romantic systems should accept the notion that "reality is constructed by the mind, not simply perceived by it, and many such constructions are possible [...]" (Tarnas, 1991:396). These systems should instead capture and reflect all the possibilities in organizational life states (intensions) than to concretize a single life state only as a fact.

What would a Romantic Information System look like?

This section uses an agent-mediated electronic-market system (e-market) as discussed in Mavetera and Kadyamatimba (2003) as an example to illustrate the concept of a romantic information system. In an e-market system, there are seven stages (phases) as derived from the consumer buying behaviour and the business-to-business transactions models whose transactions need to be automated using software agents. These are need identification, partnership formation, supplier brokering, negotiation, contract formation, purchase and delivery and lastly, product service evaluation, what many organizations may refer to as 'after-sales service' (Mavetera and Kadyamatimba, 2003:160).

Central to the success of e-market systems is the database schema that is used to structure the information and data that is exchanged in these systems. However, if the data schema of one participant in the market system changes, most, if not all the participants' schema need to be changed to conform with these new changes. This becomes very critical in e-market systems where big players trade with small business. This process affects the software development stacks that make up the architecture of the e-market distributed web-based system. This process besides being laborious is also very expensive especially to the small players in the e-market system. Finally, this may limit the participation of small businesses in the market place.

As an example, during the negotiation process, software agents require a shared understanding of the negotiation process. Hence, a negotiation protocol is needed that is understood by all the agents involved. In a romantic e-market system, there must exist several negotiation protocols that can cater for most if not all the participating agents' requirements such as rules and parameters of negotiation. This can be realised as a negotiation template that captures all the possible intended negotiation environments (negotiation intensions) that can be anticipated in that specific industry. A single negotiation protocol may limit the participation of agents that are not designed for that specific protocol. Also, the negotiation parameters can either be constrained or open but like a human mind, should not remain static during the negotiation process.

Consistent with the definition or romantic systems given above, as representatives of human principals, software agents must therefore possess organizational culture, should have a shared understanding of the product space (a repository of all the products in the market) and should understand the requirements of other agents (shared meaning). Most importantly, they should understand the organizational social context and finally, they should be able to plan, do, check and act decisively as would their human principals. This is a measure of intuition and pragmatics. An agent, for example, can decide to quit a negotiation process at any time if other agents bring unfavourable terms. This can be done after balancing the terms of the negotiation process with its organizational culture and context. This example will be revisited again in chapter 5 to highlight the role of ontologies in building romantic information systems.

What is ontology?

Ontology has attracted many definitions but the one that seems to be widely adopted portrays ontology as a formal specification of a shared conceptualization (Guarino, 1998; Gruber, 1993; Studer, 1998). Guarino *et al*., (1994:2) also regard it as "knowledge about a priori structure of reality"; implying that, when one chooses a particular intended model for a logical theory, one has to make "implicit assumptions about other models that are compatible with the chosen one". In information systems, ontology can be regarded as a "formal language designed to represent a particular domain of knowledge" (Zúñiga, 2001:187). In this field, it plays a functional role, since it is almost always designed for specific purposes.

In the study, the philosophical and epistemological aspects of ontology were dealt with in answering this question *(Section 5.1)*. Despite its widespread use in philosophy as an essence of existence, there is no unanimous agreement in the information systems field on what ontology is. There are differing schools of thought according to whether ontology is portrayed as a data, information or knowledge artefact (Mavetera, 2004b). Furthermore, its structural components, semiotic requirements and representation in information systems also constitute a subject of debate in this research. All these issues will be addressed in Chapter 5 of this thesis.

1.5.2.2 Why ontology?

There have been several calls for new methods of developing software products with recently, Cretu (2010) proposing a business process oriented software methodology and Bensta (2010) advocating for a software development method that merges ontologies with currently "existing methods, techniques and tools" that are used during the analysis phase. Bensta (2010) further argues that the merger can improve significantly the process of software development at all stages of analysis through to maintenance, by facilitating the faithful translation of user requirements into object models that are used to develop the system specification. Mavetera (2007) proposed the Ontosoft framework that also, like Bensta (2010) framework, positions ontologies at the centre of an automated software development case tool. Hofferer (2007) discussed a semantic interoperability approach that combines meta-models and ontologies. In this approach, ontologies complement meta-models by adding semantic expressiveness to business models and this can be important when integrating business models for different organizations. In addition, ontologies have long been accepted as artefacts that can improve the software process (Falbo *et al.*, 2002) and Wand and Weber (1990:69) say that an "ontological approach to understanding and formalizing information systems concepts provides […] the rudiments of a theory of the deep structure of an IS". The lack of such a theory has deeply undermined research in information systems. The rationale for using ontologies in software development is further supported by Sugumaran and Storey (2002; 2006). They noted that an ontology-driven software development environment is quite effective especially when developing entity-relationship diagrams (Sugumaran & Storey, 2006) but they also lamented the lack of organized, systematic methods for developing and using ontologies in information systems (Sugumaran & Storey, 2002).

This research study justified why information systems need to incorporate the ontology artefact during their development *(Chapters 5 and 6)*. The architectural and structural aspects of the ontology artefact are discussed and used to justify an ontology-driven approach for software development *(Chapter 5)*. Of the two parallel streams, that is, semiotics and formal logic that can be used to describe ontology, the semiotic stance (Sowa, 2000; Stamper, 1992) has been used to build an argument that posits ontology as an artefact that can solve among others, the current problems of linguistic communication during software and systems development *(Section 5.10*) and the capturing and representation of human and softer characteristics of organizations. Since IS ontologies are confined to that which can be represented, it should be noted that subjective things such as feelings, even though they exist, cannot as yet be represented, especially in automated machines. This is not a limitation to the development of romantic software products touted in this research, but is a shortfall attributed to the technology used for representation.

A working definition for ontology that will be used in this study defines ontology as: *a linguistic model of the world that comprises of syntactics, semantics, pragmatics as well as the social context of that which is represented. Despite some unavoidable informal indeterminacy in the real world view, it should allow a shared, descriptive, both structural and behavioural modelling and representation of a real world view by a set of concepts, their relationships and constraints under the open world assumption.*

1.5.2.3 The "How should?" And "How does?" types of questions

This research arrived at a methodological solution to current problems in software development. In this solution set, which is a framework of ontology components, the ontology is positioned as an artefact that can be used during the development of information systems and, at run-time stage, to decrease the semantic gap currently existing in information systems.

This process was realized by taking the preliminary propositions and building them up through a literature survey (*Chapters 4 and 5)* and empirical data gathered during interviews (*Chapter 6*). This empirical study, using GTM, was done in order to develop the ontological framework accepted in the software development field. The framework depicts components of the ontology that can be looked for by any would-be system developer in a bid to make the resultant system romantic.

As the research problem does not focus on the implementation part of the research findings, it has very few technical requirements. It focuses more on the logic, the 'What?' of the problem. This emphasis on the philosophical justification of abstract concepts and ideas makes the 'How does?" type of questions less relevant. These types of questions are meant to explore issues directly observable or those that can be described as they manifest themselves in reality (Roode, 1993). At the minimum they look at the development of a design science artefact.

1.5.3 Aims and Objectives of the Study

"In this way, knowledge becomes reusable, visible and accessible and enables teams to share their knowledge and profit from experience."

Sheryl Torr-Brown (In-PharmaTechnologist.com, 2005).

1.5.3.1 Aims of the Study

This study aims to find a software development approach that can improve the software development process. This comes as a framework of components that should be considered during the software product-development process.

The approach, in addition to focusing on the unavoidable use of syntax, should also find ways of ensuring the capturing of semantics, pragmatics and social context of the organizational system. Context ensures that the fuzzily defined social requirements of every system are also included in the resultant system.

It is worthwhile to pre-empt the research outputs by advocating the use of the ontology artefact during software product development. The aim of the study is to position ontology as an artefact that can be used during the development of software applications, at run time or as part of an information system, in order to increase semantic and context-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, especially in agent-mediated environments such as those described by Mavetera and Kadyamatimba (2003). It is the intention of this research to investigate and bring to the fore the humanistic, interpretive and softer aspects of ontologies as used in software development and information systems.

1.5.3.2 Objectives of the Study

In line with the aims of the study, the objectives of the study can be summarized as the need to develop a general ontological framework that satisfies the aims of this study. The ontological approach to software development challenges traditional software engineering approaches as expounded in this research. 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 software products. The abstract ontology theory could be operationalized into a concrete form of knowledge that could be applied readily in software development (Agerfalk, 2004). Operationalization is the process of converting or moving from abstractions to some concrete artefacts that have practical applications in a social practice such as software product development.
- To position ontology as an artefact that is needed by software developers to experientially and intuitively develop the linguistic models needed for communication in information systems.
- To develop an ontology model-driven approach to software development (El Baze, 2005). This approach would ensure that the resultant model moves from a purely

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

This research, while it also specifically discusses implementation issues, however, placed greater emphasis on software development methods than on the technological tools needed in software development.

1.6 Research Scope and Delimitation

The research purpose was to use socio-technical behavioural theories to explain the social nature of organizations. These organizations need to be represented as information systems using software products. Although several software development approaches are discussed, the discussion is basically to bring to the fore the problems that are embedded in those approaches that need to be addressed. The research is constructivist, focusing on the development of a new software development approach and of a methodology that can guide the development of software products that capture the human aspects of organizations. This is done through a process of identifying persistent software development problems that have not yet been addressed fully by researchers.

The research developed a working definition of ontology as used in the field of information systems. The definition was arrived at by following philosophical and practical paths of arguments as discussed in Chapter 5. The study also investigated current software development issues and concerns through a literature survey *(Chapter 4)* and by means of interviews with software development practitioners (*Chapter 6).* During the literature survey, literature related to software development and engineering, information systems and information systems development methodologies was studied. This was used both as existing and unrelated literature to set a tone for the investigative process. The research then focused on the relationship between the requirements of romantic software products and the capabilities of the ontology artefact (*Chapter 6).* It is however an attempt to provide an approach or methodology, guiding principles or a "specification of the process followed plus the modeling approach" to be undertaken in order to develop romantic software products (Gonzalez-Perez & Henderson-Sellers, 2006:125).

1.6.1 What this Research Study is Not

According to Hevner *et al.,* (2004:84), the objective of IS research is to acquire and create the knowledge and the necessary know-how that enables the design, development and subsequent implementation of IT solutions. These solutions may address previously "unresolved as well as important business problems". There are two basic types of research strands that have since guided research in IS, namely behavioural and natural science research. Behavioural scientists are preoccupied with the understanding of the phenomena that affects human behaviour while natural scientists focus on natural phenomena in general (Purao *et al.,* 2008). If the research falls in the behavioural science category, it should focus on the "development and justification of theories explaining or predicting phenomena that occur" (Hevner *et al.,* 2004:84). Between the behavioural and natural science research groups we find design scientists. These use knowledge from both natural science and behavioural science to develop "means and prescriptions" to aid human endeavours (Hevner *et al.,* 2004; Purao *et al.,* 2008). The intention of design science research is to construct and build concrete artefacts in the form of a model. Purao *et al.,* (2008), while accepting that the boundaries and contours for design science research are still ill defined and fuzzy, they see it as being prescriptive and in contrast to behavioural science research, which is more descriptive. Among other things, the primary goal of design science research is to improve the working state of an artefact, with understanding of the phenomenon being of secondary importance. Design science research should follow six steps identified by Peffers *et al.,* (2008) as: problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation and communication. This research will not discuss these different strands of research further but will leave the interested reader to check with the sources cited herein.

However, the distinction between design science and other types of research will assist in clarifying the focus of this study. Several issues are discussed in this research that are not necessarily the focus of this study. Although several issues will be touched on and discussed, this research does not solely:

- Aim to identify problems that trouble software developers in the past, present or in the future;

- Intend to develop ontologies as artefacts, a product of design science research. It should be borne in mind that this research does not intend coming up with a piece of software, model, or a design artefact; or
- Aim at characterizing software development approaches, methodologies and methods that are in current use. Such a characterization can be found in several literature publications on software development, for example, Iivari *et al.* (1998).

The research does not deal with the technical aspects of programming and description logics. This technical part, which involves the actual implementation of components of the approach develop in this study, will form part of a future research project.

This research, however, aims at producing a set of guidelines, a conceptual framework that should inform developers on the artefacts to use when developing software products. In addition, these guidelines are packaged as an approach (at a very abstract level) and methodology (more prescriptive) that can be likened to a software development process model (Schach, 2005) such as the waterfall, spiral or iterative models. Although the ontology artefact proposed here requires a design science type of methodology (Hevner *et al.,* 2004), to develop and implement, the packaging of the approach discussed in this thesis is best addressed using a constructivist type of research method. Hence GTM is the method of choice. Furthermore, while the research presents an approach and a methodology, it does not prescribe a specific process for the construction of pieces of software.

1.7 Contribution of Research to the Body of Knowledge

The research study focused on ontology discipline, a field of science whose body of knowledge is continuously growing and is at times not well documented in books or journals. Consequently, it was very important for the researcher to document all the research activities that were carried out during this study. This documentation manifests itself as the conference and journal publications that are listed in Appendix B. In this list, Mavetera (2004a & 2004b) motivates for the use of ontologies in systems development, information systems and their philosophical understanding. In both these works Mavetera discusses the new paradigm requirements of software development approaches, that is, the move from mechanistic to romantic development approaches. These issues are covered in this chapter as well as in Chapters 4 and 5 of this study. Mavetera (2007) also discussed an ontology software development case tool architecture that is also included in Section 5.8 of this thesis.

Mavetera and Kroeze (2009a & 2009b) address the mechanistic nature of software products and issues that need to be considered in software development from the practitioners' perspective. These are covered in more detail in Chapters 4 and 6. Mavetera and Kroeze (2008 & 2009c) discuss the methodological facets of GTM that were used in this study. These are captured in Chapter 3 and later applied in Chapter 6 during data analysis.

Also, Mavetera and Kroeze (2010a; 2010b) and Mavetera (2011) together presented the ontology-driven software development framework, approach and methodology that can be used to introduce ontologies in the development of software products. Lastly, Kroeze *et al.* (2011) addresses the need for enriching information systems and in this paper, this author discusses the need for introducing ontologies in development of software products. These outputs could be used in the quest for developing more usable software products and they represent a modest attempt to show the importance of this research study.

1.8 Structure of the Thesis

Some researchers, especially those at PhD level, start with a pre-planned, preconceived and an almost standard way of presenting their research outputs. The presentation in this thesis differs considerably from many theses that the reader may have previously encountered. The structure of this thesis is dictated by (grounded in) the research method employed.

Apart from this chapter, all the chapters in this thesis were arranged and positioned in accordance with the steps that were followed during the research process. In the words of Roode (1993), the structure follows a "process based investigation framework", in which the sequential arrangement of activities in the research dictated the position of the chapter in which they are described. The arrangement of the activities is in turn dictated by the steps followed when GTM is used as a research tool. The layout of the rest of the thesis is shown in Figure 1.2 and is described below.

As shown in Figure 1.2, Chapters 4, 5 and 6 were dealt with simultaneously and spanned almost the entire research process. In other words, while working on the data collection and part of the analysis, the researcher also consulted literature on software development and ontologies. The reasons for this are given in Chapters 3 and 6.

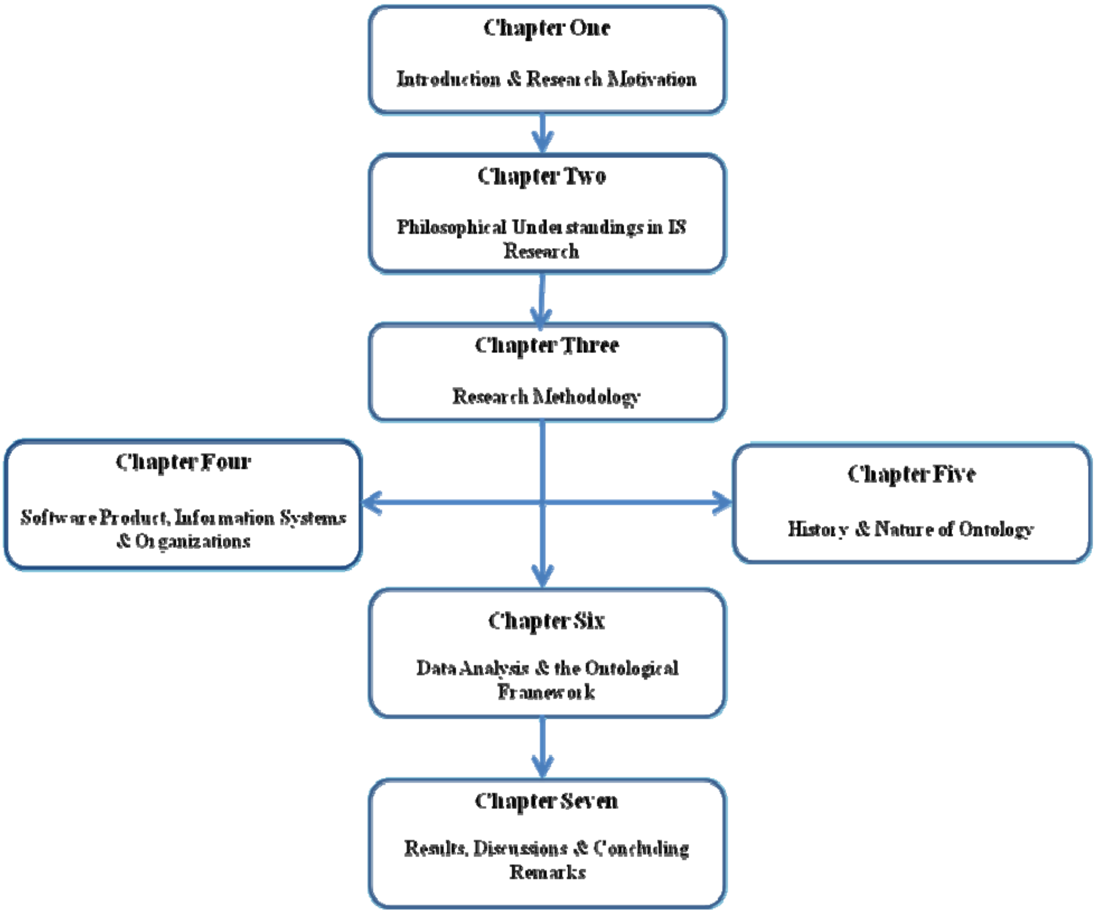

Figure 1.2: Structure of Thesis

Chapter 1, Introduction and Motivation, gives a brief background to the software development problem. The motivation for doing the research, the research interest, a brief description of the methodology used and of the scope of the research is also discussed.

As sound scientific research has to be based on some philosophical grounding, **Chapter 2,** Philosophical Understandings in IS Research, discusses some of the philosophy of information systems research, issues such as the ontological and the epistemological aspects of research and the paradigms of information systems research. The philosophical grounding, that is, the IS Research Paradigms, Approaches and Methods are also included. These are used to guide the researcher in choosing the right research methodology relative to the nature of the research problem.

Chapter 3, Research Methodology and Design, discusses the research method used, data acquisition methods, data analysis techniques and the justification for their selection and use. It contains a literary discussion of the steps used during the research and the adjustments of the method made, where this deviated from the GTM principles as proposed by Glaser and Strauss (1967). The discipline of information systems development and, hence, of software development is very complex. To support this, **Chapter 4,** Software Development Practices, Information Systems and Organizations, deals with aspects of software development, complex and adaptive information systems, the context of information systems, mechanistic versus romantic systems and the role of actors in modern information systems. It is specifically dedicated to the study of existing literature. The theoretical frameworks that guide the interpretation and analysis of information systems components in organizations are also discussed. These frameworks include - but are not limited to - actor network theory, human activity systems, theory of complexity, deconstruction and philosophy and the software development problem.

Chapter 5, the History and Nature of Ontology, amongst other aspects, discusses the philosophical grounding and the foundation of ontology as an artefact for use in software development. A structural definition of ontology, based on the semiotic levels of Stamper (1992), is also given.

Chapter 6, Data Analysis and the Ontological Framework, discusses data analysis and the ontological approach, which is the output of the study. Its validity in respect of the software development problem is also discussed. **Chapter 7**, Concluding Statements and Remarks, concludes the study and makes some recommendations, proposes quality requirements of the research and suggests the way forward in the field of software development. Also included in this chapter are the research findings, evaluation, limitations and contribution of this research to the field of software product development.

1.9 How to Read this Thesis

Several issues need to be explained concerning the presentation of work in this thesis. These include:

- The use of italics in the thesis body and the proposition sections of the appendices.
  In the body of the thesis, some words are italicized to indicate the emphasis the researcher put on those points to express the points under discussion. This emphasis is for the researcher alone and not the original source of the quotations. In the propositions Section (*Chapter* 6); italics were used to indicate where a set

of words or phrases as collected from the respondents support the proposition listed.

- The referencing style used in this thesis.
  The researcher used the University of Pretoria's Harvard referencing style which can be found on the Faculty of Engineering, Built Environment and Information Technology (EBIT) website. For ideas that span the whole text of a chapter, a book or journal paper, the researcher only cited the authors' surnames and the year of publication. In the case of specific quotations, the surname, year and page number(s) were included. In the case of a quotation that followed immediately after another and which was on the same page of an article as the previous quotation, the page number only is included.
- There are many cross-references in this thesis. This represents an attempt to link information together as well as to make it easier for the reader to find the thread in the discussions that fall in different chapters or sections. They also group the thesis discussion as a single body of knowledge. More importantly, the double-mapping principle discussed in Section 3.5 necessitated the mapping of concepts from the software development study discipline to ontology study discipline. This mapping can clearly be shown by using cross-references.
- Usually, most research studies completely discuss their research design on the research methodology chapter only. In this study, the research design straddles two chapters: C*hapter 3: Research Methodology and Design* and *Chapter 6: Data Analysis*. This is not unusual and was necessitated by the requirements and dictates of the research method used, in this case GTM. GTM requires that, as one analyses the data, one concurrently needs to motivate for the choice of the data samples and their coding.

1.10 Summary

This chapter gives an introductory tone to the thesis and its structure. It gives an overview of the aspects covered during the research. More importantly, readers are directed to the issues and problems that have bothered software developers since the software development process was termed a *"crisis"* in 1968 (Randell, n.d.). In Chapter 2 the philosophical underpinnings to information systems (IS) research are discussed. As knowledge is regarded as a social creation, the discussion of these philosophical concepts about reality will help the reader to gain some understanding of the researcher’s world view.

CHAPTER TWO

UNDERSTANDINGS IN INFORMATION SYSTEMS RESEARCH

2.0 Introduction

Chapter 1 gave an introduction to the research study. It gave a road map of the issues that will be covered in subsequent chapters of this thesis. Chapter 2 will give a brief background of some philosophical tenets that are used to guide research in information systems (IS). Research in information systems, as in any scientific discipline, is guided by different philosophical paradigms, research approaches and a myriad of research methodologies. Information systems research is a form of inquiry, an investigation to discover knowledge and facts. Researchers in this field should be convinced (and should be able to convince others) that the knowledge gained from the research process can indeed be relied upon (Olivier, 2004). Most research in information systems should establish actionable knowledge (Agerfalk, 2004) as the research output. Actionable knowledge means the theories, strategies and methods that govern or control the actions performed in social practices by people (Goldkuhl, 1999).

On this note, a research framework is, therefore, needed to guide researchers for them to get actionable knowledge as the research output. The research framework provides answers to questions about the nature of the problem under investigation. This usually forms part of the methodological grounding of the research. The guiding research framework should have as its grounding the philosophical understandings of nature (social paradigms), the guiding research paradigms and the research methodologies. This framework dictates the research approach, the methodology to be used for the research and the subsequent quality of the research process.

Most research tasks usually start with some theoretical considerations. The research is then designed to answer questions posed by those theoretical considerations. Most importantly, the research should be soundly based in theory and undertaken in a systematic way using appropriate research methods. As an output, each research task should either generate or test theories and also provide material for the development of general laws in a discipline.

In addition to this, every research project should build a body of scholarly work in the particular field of study. This scholarly contribution is needed to add new knowledge in a specific study discipline. In this research, the field of study is information systems.

According to Kuhn (1922-1996), scholarly work is judged by sound theoretical underpinnings that exist in a stable paradigm. The paradigm should be clear to people who share the same thinking in a field of study. These theoretical underpinnings exist also as theoretical frameworks upon which all actions performed in social practices can be soundly based. Continuous investigation or research in a scholarly field may lead to changes in accepted theories; new ideas may gain authority and result in the rejection of old ideas. This may lead to a paradigm shift in the practices of people in the discipline.

This chapter is dedicated to the discussion of philosophical tenets that apply to information systems research. The discussion informs the researcher about the nature of the research problem, the world in which the research is situated and the most appropriate methodological process to follow when solving the problem. As a starting point to this chapter, the general types of theory found in research and the research strategies that are used to generate the theory are briefly discussed.

2.1 What is theory?

> *"A theory is not some hunch, or half baked idea that you come up with while taking a shower, or being under the influence of something or other."*
>
> *Tucker, W., 2009:1*

Theory is a term employed in both the science field and day-to-day common usage. It has generated varied definitions from the scientific community and the world at large. In common usage, theory may mean an opinion, conjecture or even speculation. The electronic Dictionary.com regards theory as a coherent group of general propositions used as principles of explanation for a class of phenomena. In a scientific discipline, theory is defined as a logical, self-consistent model or framework that explains the behaviour as a result of the interaction of some related natural or social phenomena. Berg (2007) conceives theory as a set of statements or propositions that are used to describe some aspects of phenomena. Supporting this notion, Sutton and Staw (1995:372-373) see theory as consisting of conceptual arguments, together with logical explanations of these arguments. They further add that, "theory is about connections among phenomena, a story about why acts, events, structure, and thoughts occur." From the natural sciences perspective, Tucker (2009:9) adds that theory must be a "scientifically tested principle or body of principles that incorporates and explains a significant body of evidence". In short, theory must express some truth about the relationship between elements of the reality space. Supporting theory as evidence of some truth, Suddaby (2006:633) added that scientific truth "is a product of (empirical) observations and the agreed revelations from the data that are observed by a group of similar minded observers". This is supported by Bryman (2004) who also describes theory as an explanation of observed regularities. In many practices, for one to generate or prove theory, data have to be collected to build or test these theories. As Berg (2007:20-21) noted, a theory should, therefore, include "patterns, concepts, process, relationships or events" that are interrelated and are able to explain the being of some occurrences. Furthermore, concepts can either be "symbolic or abstract elements" that are used to signify some special occurrences of phenomena. There is also a need to indicate the "concepts and causal arguments" borrowed from citations and their link or association to the theory under development (Sutton and Staw, 1995:373). These concepts in a theory are generally used as communicative symbols.

It is widely accepted in the scientific domain that any theory generated must be falsifiable, but Sutton and Staw (1995:371) note that the research fraternity lacks consensus on whether "falsifiability is a prerequisite for the very existence of a theory". This is coupled by the diverse uses of the word in different scenarios, contexts and in different fields of study. Sutton and Staw (1995:371) urge editors of journals to consider research papers that may "test part rather than all of a theory". This is an acceptance that theory can also be generated by considering only part of a sample size of data or evidence needed to verify or refute a hypothesis from the theory. In addition, Sutton and Staw (1995:375) advise that the research papers can also "use illustrative rather than definitive data"

The fact that theory is able to explain the relationships between natural or social phenomena is not a licence to its acceptance in a study discipline. Quite in agreement with what led to Aristotle's death when he refuted the geo-centricity of the galaxy, Sutton and Staw (1995:372) note that even well grounded theories, those that fit the empirical data gathered can be rejected if they happen to clash with some "particular conceptual tastes" of the gurus in the field such as editors of journals. The presence and existence of a theory and its subsequent acceptance are not necessarily linked but, it must be noted that, good theory is one that is "representational and verbal" (Sutton and Staw, 1995:376). The next section will discuss some different types of theory found in research.

2.1.1 General Types of Theory

In research practice, there exist generally two types of theory: grand theories and middle-range theories. Grand theories deal with more abstract and general explanations of things (Merton, 1967). Grand theories do not clearly indicate to the researcher a way of collecting empirical evidence, nor do they provide a clear guide to the researcher on the methods of data collection. This type of theory may not be very useful in a researcher's quest to determine practical relevance of the research.

Merton (1967) also describes middle-range theories as those that are in-between general theories of social systems which are too remote from particular classes of social behaviour, organization and change to account for what is observed and to those in depth orderly descriptions of particulars that are not generalized at all. Mid-range theories are required to guide empirical enquiry and these operate in a limited domain that falls between grand theories and empirical findings. Researchers most often use middle-range theories to understand and explain a limited aspect of social life (Bryman, 2004). However, after noting the ambiguity in the meaning of mid range theories, Gregor (2006:613) warned researchers always to bear in mind that the field or discipline of study influences the "nature of theory" generated by a specific type of research.

In her description of the five types of IS theory, Gregor (2006:612) noted that the "structural nature or the ontological character" of what is termed theory is not well defined, especially in the IS discipline. This ontological nature refers to the "language for talking about the nature and components of theory". She, however, lamented the continual lack of a clear definition of the term theory in many published articles when the term is used more than once in the same article. In her view, whenever an author uses the word theory, the context in which it is used should also be given.

Gregor (2006) just like Hevner *et al.* (2004) and Purao *et al.* (2008), went on to describe the five types of theory in IS as theories for: analyzing, explaining, predicting, explaining and predicting and, lastly, for design and action. Of importance in this study are the theories of analyzing and explaining, the categories on which this research study is based. In brief, the theory for analyzing classifies characteristics of phenomena by summarizing common attributes that are found in discrete observations. These theories state that the "what is?" of a population sample and the most evident examples of this type of theory are frameworks. Although they have a descriptive element in them, they also analyse and summarize the most important salient attributes of phenomena and their relationships thereof. The most important thing is that the relationships are may not necessarily be causal relationships (Gregor, 2006).

The theory for explaining, also known as theory of understanding (Gregor, 2006), focuses more on the "how?" and "why?" of the existence of some phenomena and their relationships.

In this type of theory, the focus is not on developing testable predictions but simple explanations of "how, when, where and why events" (Gregor, 2006:624) happened are sufficient. These explanations usually give rise to process-type theory. The issue here is to provide new insights as to how nature is, that is, insights that are different from those previously known.

Any research that generates or tests a theory needs to be grounded in a research strategy. The next section will look at the strategies that are usually used in conducting information systems (IS) research.

2.2 Strategies for Conducting Research

The general purpose of a strategy is to facilitate easier understanding and resolution of problems. Bryman (2004) describes a strategy as a "general orientation to the conduct of social research". According to Olivier (2004), there are three principal strategies that can be used to conduct research. These are:

a) Compiling information on a topic, in which the bits and pieces have already been discovered, (often by other researchers) but where the bits and pieces have not yet been integrated into a single coherent body of knowledge;

b) Looking with new eyes at existing knowledge (standard ways of doing things) and trying to find a better solution for a problem that has previously been solved and, lastly,

c) Solving a problem for which there is no known (or apparent) solution.

From the research interest and objectives of the research described in Chapter 1, this research is concerned with all three strategies. The purpose of the research is to integrate bits and pieces of knowledge that can be added to the software book of knowledge and the ontology research body of knowledge. It also searches for a solution to the problem of mechanistic development of software. The research strategies usually inform the researchers on the type of research they must undertake.

2.3 Types of Research in Information Systems

There are two major types of research: theoretical research and empirical research. These two types of research are governed by differing research problems and by the nature of these problems. The type of research implicitly or explicitly determines the type of theory generated and is also guided by the research strategy adopted. The two types of research are discussed below.

2.3.1 Theoretical Research

Theoretical research is concerned with developing and refining a body of abstract understanding of phenomena and issues. Dahlbom and Mathiassen (1992) regard theory as something very fluid and possessing some romantic aspects. They regard theory development as a way of bringing order and sense to a complex real world. It is a process of structuring or finding the rules that govern the relationship between objects or artefacts that make up particular social phenomena. Bringing order and sense to a complex world is supported by Dahlbom (1996) as going away from the discernible phenomena to deeper, unseen layers of reality in an attempt to define concepts and general laws in terms of which the disorganized flux of visible facts can be systematized and explained. This notion deromanticizes the meaning of theory. Theoretical research may lead to empirical research.

2.3.2 Empirical Research

Empiricism relies on the notion that only knowledge gained through experience and senses is acceptable. It claims that, if ideas are revealed through grand- or middle-range theories, they should be subjected to the rigours of testing before the research fraternity can accept them as knowledge (Bryman, 2004). Empirical research requires the collection of data or facts relevant to the research problem. Empirical research is usually used to provide evidence to drive the process of theory development. In some cases, empirical data may emanate from a theory. The collection of facts is considered a legitimate goal in research and can justify the existence of knowledge. Some researchers describe this second process as naïve empiricism.

In the second stage, when data collection is complete, the data are critically analyzed to try and make sense of them (Cornford & Smithson, 1996). The major task in empirical research is to find inter-subjective observable knowledge among actors that justifies the existence of an occurrence. Processes such as the interview data-collection process used in this project are examples of the empirical issues referred to here. Regardless of whatever type of research is followed or one can follow, the researcher has to contend with three major types of research tasks.

2.4 Research Tasks in Information Systems

Both the theoretical and empirical types of research are associated with a research task, that is, the activities needed to accomplish a specific type of research. Both theoretical and empirical researches can be conducted through constructive, nomothetic or ideographic research tasks. These three research tasks are discussed below.

2.4.1 Constructive Research Task

Constructive research is also known as inductive research. This research task is concerned with the development of frameworks, refinement of concepts or pursuance of technical developments (Cornford and Smithson, 1996). The task of the enquiry is to build theory as the main outcome (theory building). Observations are made and are subjected to scrutiny to check for trends and patterns in the data. The general relationships among data gathered are used to formulate a hypothesis or a proposition and, subsequently, a theory.

It may be necessary to validate the inductively generated theory by subjecting it to more tests using the deductive process. This process is commonly used in GTM. According to Iivari (1991), constructive research may develop models and frameworks that do not describe any existing reality. The purpose of these frameworks, therefore, is to create new forms of reality. If constructive research is used for theory building, researchers should then have a task that can be used to prove this theory.

2.4.2 Nomothetic Research Task

Also known as deductive research, nomothetic research is concerned with the exploration of empirical data. The idea is to use the data to test a theory or a hypothesis. Nomothetic research strives to find evidence to support general laws or theories that cover a whole class of cases (Cornford & Smithson, 1996). Loosely said, its task is to validate proposed theories empirically. The output of this task is deductive theory.

2.4.2.1 Deductive Theory

In all research, there has to be a link between theory and research. This link is established through the theoretical grounding provided using middle-range theories such as deductive theory. The deductive process starts by generating a hypothesis. This hypothesis is generated using existing knowledge about phenomena in a specific domain and the theoretical considerations in that domain (Bryman, 2004). The study of existing literature in the study discipline can be used to generate these hypotheses. This hypothesis is then subjected to rigorous empirical verification.

Literally, a hypothesis can be likened to a provisional idea that needs to be evaluated to find its merit. After evaluation a hypothesis may be accepted or rejected, depending on the conformance of the data to the proposition. Bryman (2004) depicts a hypothesis as a carrier of concepts that need translation into researchable entities. The task of translating these concepts into researchable entities is termed operationalization.

During operationalization, the researcher specifies how data can be collected in relation to the concepts that make up the hypothesis (Bryman, 2004). The deductive process, therefore, starts with both theory and hypothesis. These later drive the process of data collection. The final reflections on the implications of the data gathered on the theory and hypothesis introduce some element of inductive process to the research. The third research task is called ideographic research.

2.4.3 Ideographic Research Task

Unlike constructive and nomothetic research tasks, an ideographic research task is concerned with exploring particular cases or events and providing the richest picture of what transpires. A phenomenon is taken individually and studied in its own right in order to understand its particular context. It emphasizes the analyses of subjective accounts based on particular or close association with everyday events. Ideographic research may lead to the generation of inductive theory. All these research tasks have to be guided by the way researchers conceive their world. What makes up the researcher’s reality space? This can be explained using the sociological paradigms.

2.5 Sociological Paradigms Applied to Information Systems

A paradigm is defined by the Merriam-Webster online dictionary as a *philosophical or theoretical framework of any kind*. The framework encompasses assumptions, concepts values and practices that a community of people have when looking at reality. It is a way of seeing, thinking and interacting with phenomena in a reality space. In the software development field, Schach (2005:25) refers to a paradigm as a “model or a pattern” and not as a style of software development.

In line with this view, Burrell and Morgan (1979) developed four sociological paradigms that are now in widespread use in the research fraternity: the functionalist, interpretive, radical structuralist and the neo-humanist paradigms. These four paradigms, as used in information systems research, are depicted in Figure 2.1. Burrell and Morgan developed these paradigms as answers to four critical questions that always confront researchers when they look at the nature of the world.

The first question deals with the nature of reality. It questions whether researchers should accept reality as a given or not. In other words, does reality exist outside the observer's mind? On the other hand, should people conceive reality as a figment of their mind? There is a strong motivation for people to consider reality as a creation of the mind.

The second question deals with human understanding. Some researchers contend that understanding things require people to experience them. It is, therefore, important to know the determinants of human behaviour. The third dimension considers behaviour as a voluntary action. However, some argue that the environment can shape and determine the outcome of a person's behaviour. Lastly, the fourth question arises based on the factors that dictate human understanding: Should people use scientific methods in an attempt to gain understanding or should they directly experience the situation to understand it?

These four questions led the sociologists Burrell and Morgan (1979), to look at the nature of the world, using the ontological and epistemological axes shown in Figure 2.1 as a lens. Ontologically, the world can be considered as a continuum, ranging from a collection of ordered artefacts to a set of artefacts that are always in a state of conflict. Epistemologically, the world again can be considered as a continuum, viewed as consisting of objective artefacts through to subjective artefacts. The word epistemology, derived from the Greek word *episteme*, which means knowledge, looks at the forms of knowledge that can be obtained about the world. It is a concept that focuses on the truthfulness of knowledge relative to the observer of the reality. The four paradigms as used in information systems research are briefly discussed below.

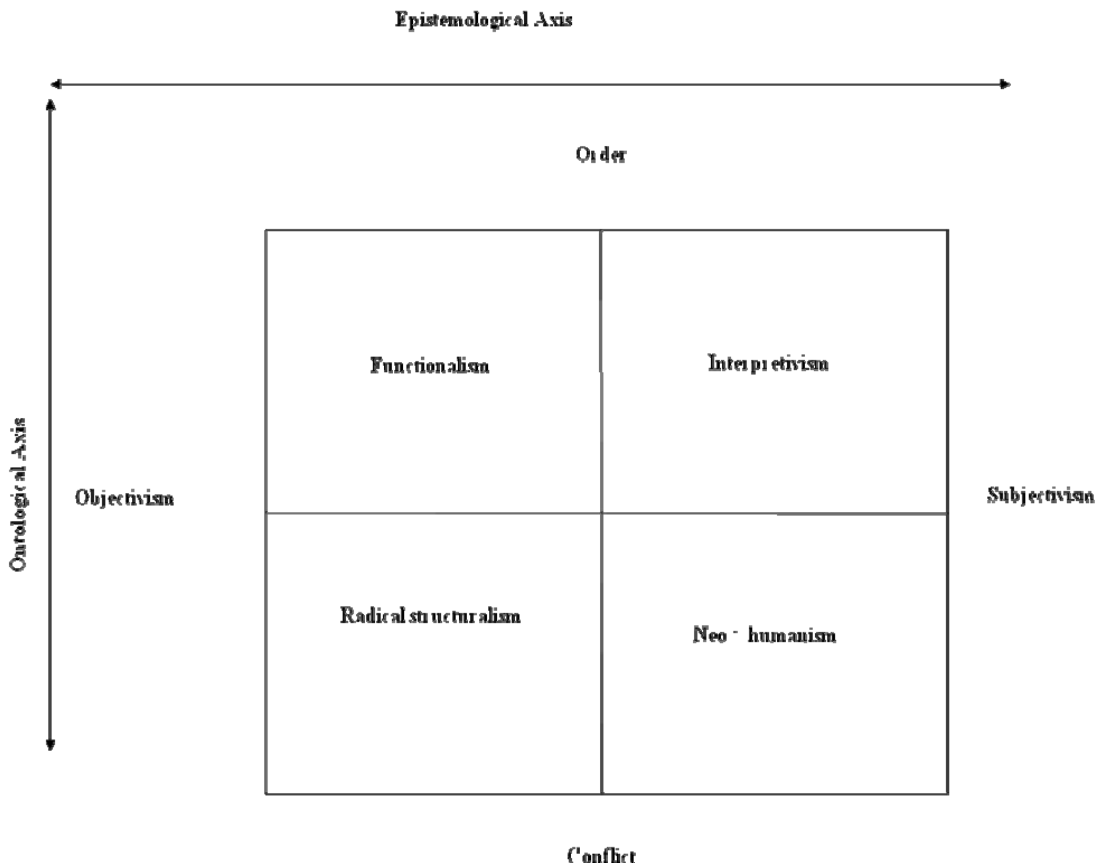

Figure 2.1: Sociological Paradigms (*Adapted from Burrell and Morgan, 1979*)

2.5.1 The Functionalist Paradigm

The functionalist paradigm perceives society as being made up of structures that can be aggregated to form a system. This system can be decomposed into its functional constituencies. The paradigm assumes that the social world is ordered and composed of relatively concrete empirical artefacts whose relationships can be identified, studied and measured through approaches derived from the natural sciences. Most information system problems have always been viewed and solved using this functionalist perspective. For more detail the reader is referred to the discussion on systematicity and system formation in Section 4.5.1.

Information systems professionals who work in this paradigm believe that the world, organizations and information systems are aggregations of functional units that are objective and ordered. This view allows system developers and researchers alike to use reductionist principles when solving information system problems. Using this approach, researchers assume a realist, positivist, determinist and nomothetic view to the world around them.

Functionalists accept the notion that general laws and theories applicable to the study of natural sciences can also be used to study human behaviour. In their quest to find truth, functionalists take the position that, by using the correct instruments and methods, the right procedures and value free investigation, one can objectively find the truth. They contend that the truth is "out there" waiting to be discovered. By the same norm, the researcher assumes the role of an objective and independent observer detached from the problem being researched. The researcher's position as an independent observer is challenged by researchers working from the interpretive paradigm.

2.5.2 The Interpretive Paradigm

This paradigm is characterized by a need to understand the world as it is, to understand the fundamental nature of the social world at the level of subjective experience. It seeks explanations within the participant's frame of reference, rather than from an objective observer of the action. Although it assumes an orderly world, it recognizes the existence of a subjective world, in which truth can only be found as a by-product of the observer. There is a greater need for researchers in this field to get to the level at which one can experience the phenomenon under study.

Interpretivists assume a nominalist, anti-positivist and voluntarist view of the world. At the methodological level, an ideographic stance is taken. Researchers perceive the world as an emergent social process, which is continuously created and recreated by people. The researcher becomes part of the researched and, only then, can one understand the world from within.

2.5.3 The Radical Structuralist Paradigm

It is important to note that radical structuralists share an objective departure point with the functionalists but, in addition assume that radical change is built into the very nature and structure of contemporary society. Radical structuralism focuses primarily on the structure and analysis of economic power relationships. It is assumed that society has inherent structural conflicts that result from political, economic and social forces, all of which aim to change the world.

In addition to sharing the same ontological view with the functionalist paradigm, the radical structuralist paradigm believes in society's radical change, its emancipation and potentiality. Information systems researchers are more interested in the power relations that exist amongst actors in an organization and in how these power forces combine to effect radical change in an organization. The world is viewed as being composed of artefacts that are always in a state of conflict.

2.5.4 The Neo-Humanist Paradigm

Unlike the radical structuralist paradigm that accepts the world as objective, the neo-humanist paradigm assumes a subjective interpretation of the social world, while accepting its conflicting and disorderly state. It differs from the interpretivist paradigm only in the sense that it is committed to overthrowing or transcending the limitations of the existing social world.

Information systems researchers working in this quadrant also share a romantic view of the world. This view of the world, while approximating the real world status quo, is challenged by the fact that not all the softer issues of society or world can be systematized, coded and subsequently implemented using technology.

Study of the four paradigms is of the utmost importance to information systems researchers so that they can understand the philosophical groundings of information systems and how people perceive the world. Once it is understood how people view the information system world, the ontological, epistemological and humanist nature of the information system problem to be solved has to be discussed.

2.6 Philosophical Groundings in Information Systems Research

When undertaking information system research, one should ground the research on some underlying philosophical assumptions. Questions on what constitutes valid research and what is the most appropriate research method should be answered (Myers, 1997). The underlying assumptions should be spelt out and identified both before and during the research process.

Also, in order to conduct and evaluate the research, it is important that the researcher identifies these sometimes hidden assumptions. Three philosophical assumptions: the ontological, epistemological and human nature, which are frequently used in the study of information systems, are discussed below.

2.6.1 Ontological Grounding to Information Systems Research

At the start of an investigation, the researcher should have an appreciation of what makes up the world. The researcher should find out what exists in the world and what can be known about this world. Also included in the search are people's conceptions of reality. The study of these underlying assumptions made about the phenomena under investigation is termed ontology. As a field of study, Ontology (capital letter "O") deals with the nature of social entities (Bryman, 2004).

The underlying assumptions in ontology exist as a continuum, ranging from nominalism through to realism. In between these two, materialism is found. Nominalism, usually referred to as idealism, is the belief that things are what we, as individuals, think they are. This nominalism positions the social structure of the world and its interpretation at the core of reality construction. The law of causality, cause and effect does not apply in the idealistic ontological position.

In the nominalist view, Roode (1993) advocates the inclusion of structures that are created by individuals through the processes of naming, labelling, and defining concepts. He argues that there are no invariant structures in the world that are waiting to be discovered "out there". In creating reality, everything needs the consciousness of individual human beings. Meaning given to artefacts is created and recreated through interactions between people, technology and the environment in which they all reside.

In contrast to the nominalist assumption, realism assumes that the world is made up of objective facets that possess concrete characteristics. It holds that there is an external reality, detached from the observer's beliefs and perceptions (Reetley, 2003).

Lying between nominalism and realism, materialism is found as a notion that argues that reality exists but that this reality is only confined to material features such as physical objects in the world (Reetley, 2003). Values, beliefs and experiences are not considered as contributing to the existence of reality.

2.6.1.1 Objectivism

Objectivism deals with the question of whether social entities have an objective reality that is external to social actors or not. In fact, it addresses the issue of whether reality, human conceptions and their interpretations exist independently (Reetley, 2003).

Objectivism views social phenomena and their meaning as external facts whose structure and existence cannot be influenced by the observer. This notion accepts that there exists a reality that is external to people who act or interact with social phenomena. Reality is viewed as being independent of the social actors that work or act on the phenomena and that there is a reality somewhere "out there" that is always waiting to be discovered.

2.6.1.2 Constructionism (Constructivism)

Constructivism on the other hand, discusses whether social activities are the result of some social construction that evolves in an actor-network environment as a result of the perceptions and actions of social actors (Bryman, 2004). This views social phenomena as having emergent properties. Reality is always in a continuous state of construction and reconstruction. Social interaction creates social phenomena and the classes that are thus produced are in a continuous state of revision. The way meaning is assigned a contextual attribute is a typical example of this. In short, social objects and categories are socially constructed. Data can be processed into information. People can deduce the meaning of social artefacts within a context. The sharing or existence of a commonly accepted inter-subjective context leads to general meaning being given to social artefacts. In addition, context is time- and place-dependent. As contextual meaning leads to knowledge, knowledge is a social construction. Constructionism can be used to reflect the inadequacy of our knowledge of the social world. There is thus a continuous creation and re-creation of knowledge in the world.

2.6.2 Epistemological Grounding to Information Systems Research

Cornford and Smithson (1996) describe epistemology as a type of (valid) knowledge that can be obtained about a phenomenon being studied. Olivier (2004) refers to this as the knowledge that people can and cannot have. Olivier's argument is that some people may possess knowledge unknowingly since it is the expertise and skills acquired by a person through experience or education. Epistemology can also be regarded as the study of how organisms get to think, decide and come to know their social world (Reetley, 2003). In other words, it is the study of what can be regarded as acceptable knowledge in a specific discipline. However, this study is not a static discipline, as Hacking (2002:8) explains:

"…epistemological concepts are not constants, free standing ideas that are just there, timelessly."

These concepts, as knowledge, are mutable, as several successive generations of researchers have used them to organize knowledge and to conduct scientific inquiries.

As depicted in Figure 2.1, the epistemological axis runs as a continuum from objectivism (positivism) through to subjectivism (anti-positivism). This continuum covers the positivist notions of the natural sciences and the realist notion (which is usually called critical realism) through to interpretivism. These epistemological notions are discussed below.

2.6.2.1 Positivism versus Anti-Positivism

The positivist strand believes in realism, that is, that there exists a single view of the real world that we all share. Any observer of this real world will get the same view and results, which will be independent of the observer (observer-free perspective).

In the objective world, the social world can be subjected to the same form of enquiry as the natural world, which consists of models and methods. As human action and institutional aspects inherent in the social systems are totally independent of each other, we can thus have an observer-independent result from such an investigation (Cornford & Smithson, 1996; Roode, 2004).

On the other hand, the anti-positivist notion believes in relativism. Relativism accepts that there is no real world "out there" waiting to be discovered. Everything occurs as a result of constructions in individual human minds. Culture, society, language and the interaction of these will help to shape what is observed by researchers during their investigations. Issues such as time and place begin to make impact in the results of observations in a relativistic environment. Ideologically, anti-positivists believe that all knowledge, both scientific and non-scientific, is a result of social construction. Knowledge is formed and shaped by the values of a society of interacting actors. This school of thought holds that there is no clear distinction between facts and values (Cornford & Smithson, 1996).

With regard to information systems, there is no hard, real and already tangible information or knowledge that can be found. The knowledge is of a softer, more subjective, spiritual or even transcendental form that is based on the experiences and insights of the unique and essentially personal nature of the actors (Roode, 1993). This results in many problems in this field requiring a subjectivist approach to their solutions.

To summarize, positivists regard all these explanations as being based on regularity and causal relationships, whereas anti-positivists only give validity of knowledge to the viewpoint of the participant within a given activity. Anti-positivists contend that knowledge progresses from an objective to a subjective reality.

2.6.3 The Humanistic Grounding in Information Systems Research

This philosophical assumption considers the relationship between human beings (actors) and their environment. The experiences of actors in information systems are not necessarily prescribed products of the environment in which they exist. Their actions are not deterministic, although many developers using the functionalist paradigm approach regard them as deterministic. Determinism leads to the bias of many developers who assume immutable functional models instead of mutable action models. Information system actors, however, create their own experiences through interactions with the environment. Roode (1993) described this notion as the voluntarist view.

This humanistic-culturalist tradition takes as its epistemology the fact that, in a socio-technical environment, knowledge is constantly being created through the interaction of actors. According to Checkland (1999:277), actors are always "negotiating their interpretations of reality, those multiple interpretations at the same time constituting the reality itself". This is supported by Giddens (1976) who argues that knowledge is not a pre-given entity but one that is continuously created by the actors in a social setting and manifests itself as a shared cognitive base of the participating actors. The inclusion or existence of people as part of information systems makes these systems non-deterministic, unpredictable and, therefore, socially constructed systems.

These three philosophical groundings are used to inform the researcher about the correct research paradigm to use in this study. In addition to the sociological paradigms, they guide the researcher in his choice of the correct pair of world views, i.e. objective-order, objective-conflict, subjective-order or subjective-conflict pairs to position the nature of the research problem.

2.7 Information Systems Research Paradigms

There are basically two strands that a research project can follow. These strands (referred to here as research paradigms) are the qualitative and quantitative paradigms. The research paradigms can be likened to a strategy, as a way of thinking or as a way of looking at things. The research paradigm to be followed is dependent on the ontological and epistemological positions taken by the researcher about the problem. These two paradigms are described in Sections 2.7.1 and 2.7.2 below.

2.7.1 Quantitative Research Paradigm

This paradigm is usually used by researchers operating in the functionalist paradigm. Quantitative research is based on the assumption that natural phenomena can be quantified in numeric terms. The development of metrics (numbers) that can be used to describe phenomena (objects and relationships) under study are the focus of this type of research

(Cornford & Smithson, 1996). In addition to quantification in the collection and analysis of data, quantitative approaches entail the following (Bryman, 2004):

- They follow a deductive approach to the relationship between theory and research putting emphasis on the testing of theories.
- They incorporate the practices and norms of the natural scientific model. Hence they follow the positivist epistemological stance.
- Ontologically, they consider social reality as an external objective reality.

2.7.2 Qualitative Research Paradigm

Qualitative research targets the production of theories. Cornford and Smithson (1996) describe qualitative research as one that avoids metrication and search for other means of capturing and analyzing (understanding) data. Qualitative research involves the use of qualitative data, such as interviews, documents, and participant observation data, to enable social phenomena to be explained and understood (Myers, 1997). In qualitative research, as Reetley (2003) explains, verification and extension of theory are not the primary purposes of data collection. The intention is not to predict causes of human behaviour, but to understand how humans derive sense and meaning from their day-to-day actions. According to Olivier (2004), in qualitative research, the data to be used in a research, their appropriateness and their processing are issues that are all decided by the researcher.

Myers (1997) identifies peoples' ability to talk as the distinction between people and the natural world. He used this distinction to motivate researchers to embrace qualitative research methods. Qualitative research methods are designed to help researchers to understand not only people, but the social and cultural contexts in which they live. Research in this paradigm has been associated with the intepretivist and relativist positions as expounded from the epistemological anti-positivist view of the real world.

The qualitative paradigm negates the generalizability which is inherent in models and theories of the natural sciences. However this qualitative paradigm does not imply the absence of numbers in the research. It focuses on how the analyses which led to the conclusion or findings of the research were executed. Cornford and Smithson (1996) contend that qualitative research supports the interpretation of the results and thus supports the nature of reality. Nature has to be interpreted and understood.

When conducting qualitative research, it is very difficult to use a highly systematic research approach. A systematic approach may lead to the loss of information that is potentially available in qualitative research data. Bryman (2004) summarizes the basic tenets of qualitative approach as follows:

- It relies on descriptions of phenomena during the collection and analysis of data.
- It generally uses an inductive approach to the relationship between theory and research and emphasizes the generation of theories.
- It relies solely on the way in which people interpret their social world (intepretivism).
- It views social reality as a constantly shifting emergent property of individuals' creation.

Qualitative data analysis is, therefore, a procedure that seeks to produce findings, concepts or hypotheses without the use of statistical methods (Glaser, 1992). The researcher believes that, after choosing the correct research paradigm, one should focus on the choice of the right research approach.

2.8 Approaches to Information Systems Research

There are two basic approaches to information systems research that are governed by whether one is a positivist or an anti-positivist. The research problem space, like the reality space, cannot be treated as discrete but as a continuum. Some research problems, therefore, lie along the line joining the positivist and the anti-positivist poles. Problems found along this line should, therefore, blend the positivist approaches and the interpretive approaches that are usually used by the anti-positivists. These research problems are, therefore, dealt with using the critical realist approach. The three research approaches: positivism, interpretivism and critical realism, may be regarded as the theoretical perspectives or frameworks to research (Sarantakos, 1997; Henning, 2004). More often than not, these are referred to as methodological groundings of the research.

2.8.1 The Positivist Stance

Bryman (2004) describes the positivist approach as a philosophical position that relies on crude and superficial data collection during the research process. In addition, as Reetley (2003) puts it, positivism calls for a study that is separated from the subjective elements of experience. Positivistic research treats the subject of research as a subject capable of developing general theories of universal applicability (Cornford & Smithson, 1996). In some cases the theories need to be proposed and tested.

The primary goal is to limit the impact of the researcher on the results. Cornford and Smithson (1996:27) added that positivist research regards true knowledge from research as a result of an "observation or experience of real phenomena in an objective and real world". The end product is regarded as undisputed facts that are timeless and without any social values attached. The researcher does not influence the results of an investigation and, assuming that all variables are kept constant, other researchers may reach the same conclusion as the first researcher, regardless of differences in personal traits (Olivier, 2004).

Bryman (2004) noted the following as the most accepted notions in positivism and the author states definitely that:

- Only phenomena and, hence, knowledge confirmed by the senses can genuinely be warranted as knowledge. This is called the principle of phenomenalism.
- The purpose of theory is to generate hypotheses that can be tested. These hypotheses will thereby provide explanation of laws to be assessed. This is called the principle of deductivism.
- Knowledge can only be found through a fact-gathering process. This provides the basis for generating laws and is called the principle of inductivism.
- Science can only be conducted in a value-free way. This is called the principle of objectivism.
- Finally, scientific statements differ from normative statements. The true scientist belongs in the scientific domain. This supports the notion of real knowledge being gathered and confirmed by the senses. Normative statements cannot, however, be confirmed by the senses.

Positivists believe that the scientist's conceptualization of reality directly reflects that reality and that the concern should be to find explanations of human behaviour.

2.8.2 The Interpretive Stance

Interpretivism is a school of thought that emphasizes the importance of interpretation and observation in attempts to understand the social world (Reetley, 2003). It shares the view that the subject of enquiry in the social sciences, that is, people and their institutions, is fundamentally different from that in the natural sciences and is in fact an integral part of the qualitative research paradigm. The epistemological grounding of interpretivism lies in the understanding of human behaviour that is arrived at through a series of cyclic processes of interpretation that are referred to as the hermeneutic cycle. The emphasis is on understanding human action and not on the forces acting on it.

Interpretive qualitative research uses the understanding of what is being researched as its primary goal. Avison *et al.,* (2008:11-12) classify it as research that is used in predicting and explaining "the status quo". The prime consideration is the fact that understanding is subjective and depends on the culture, language, history and background of the researcher (Olivier, 2004). Since language is socially constituted, the interaction between the researcher and participants will impact on the results. Furthermore, since language is subjective, all the interpretations arrived at, therefore, depict the researcher's or observer's point of view.

Interpretive researchers start with the assumption that access to reality (whether given or socially constructed) is only obtained through social constructions such as language, consciousness and shared meanings (Myers, 1997). Interpretive studies generally attempt to understand phenomena through the meanings that people assign to them. In addition to the hermeneutic–phenomenological tradition usually associated with intepretivism, Max Weber's (1864-1920) notion of *Verstehen* (understanding) and Mead's symbolic interactionism (Bryman, 2004) are included in the epistemological set of interpretivism.

Walsham (1993) regards interpretive methods, when applied to information systems research, as a task of understanding the dual and reciprocal effect of context on information systems and vice versa. Interpretive research does not predefine dependent and independent variables, but focuses on the full complexity of human sense-making as the situation emerges (Kaplan & Maxwell, 1994). The intepretivist's world is concerned with the examination of the "life world": the world of consciousness and of humanly created meanings. As Avison *et al.,* (2008) put it, interpretive research is used in predicting and explaining the status quo.

Interpretive research is thus regarded as being very fluid. Its fluidity and the grounding upon which it is based support the use in this research of the romanticized definition of theory.

2.8.3 The Critical Realism Stance

This lies between the positivist and interpretive stances. This realism holds that the natural world and the social sciences can and should apply the same kinds of approach to data collection and to its explanation. It can be split into empirical realism and critical realism.

Empirical realism, sometimes referred to as naïve realism, assumes a perfect correspondence between reality and the terms used to describe it (Bryman, 2004). This type of realism also holds that, with the use of appropriate methods, reality can be clearly discerned or understood.

On the other hand, critical realism believes in an external reality that exists and is separate from the observer's description of it. It contends that the social world can be understood and so changed if we manage to identify the structures at play that generate events and discourses in the social world. The structures are, however, a result of practical and theoretical work in the social sciences (Bhaskar, 1989).

Therefore, critical research, which is based on the grounding of critical realism, strikes a balance between positivist qualitative research and interpretive qualitative research. It is grounded on the fact that any artefact observed by a researcher in society is socially constructed. According to Mingers (2002), the critical research approach assumes the existence of a domain of structures and mechanisms, events and experiences (the Real). These structures may be physical, social, or conceptual and may well be unobservable except through their effects. It also recognizes that our knowledge is always provisional and is historically and culturally relative and that people do not have observer-independent access to the world. Critical research is about evaluation and transformation of the social reality that is the subject of investigation (Avison *et al.*, 2008).

Critical realism argues that scientists' conceptualizations of reality are just one way of trying to understand or know that reality. There is a distinction between the objects that are the focus of any enquiry and the terms or concepts that are used to describe those objects. Although generative mechanisms may not be observable, they are acceptable as long as their effects are observable (Bryman, 2004). These discussions on research approaches led the researcher to focus on the nature of the research problem in this study.

2.9 The Nature of the Research Problem

The discussion on the philosophical underpinnings of information systems research provides a lens through which the nature of the research problem can be identified. In research, three basic concerns about the problem have to be addressed: the ontological stance, the epistemological stance and the humanist stance.

A clear diagnosis of the nature of the information systems problem along these three basic assumptions is important, since this will be used to inform the methodological design of the research.

2.9.1 Ontological Diagnosis

Organizations and their information systems are considered as vast actor-networks. Although much of the technological components of these networks can be structurally separated, unfortunately their effects in the whole system cannot. In real terms, the socio-technical nature of the problem requires the neo-humanist stance to be adopted but there are also limitations on what can be observed, interpreted and implemented in software development. What really exists should be regarded holistically from the interpretive paradigm, anti-positivist stance because of the limitations caused by the formalization requirements in all system implementations.

2.9.2 Epistemological Diagnosis

The knowledge that people have about the world, especially about organizations and information systems is always provisional and in a continuous state of creation and recreation. Using the actor/network theory and structuration theory, coupled with the fact that meaning and knowledge have a situated practice that is heavily dependent on the context, the research should adopt a relativistic and voluntarist stance.

2.9.3 Humanist Diagnosis

Many of the problems facing software development and system development emanate from the fact that developers consider organizational problems as deterministic and rational. In view of this, hard software development methodologies have been used, an issue that has created many problems for the software developer. When solving software development problems, researchers are urged to take the voluntarist stance. This supports the fact that humans create their own environment and are neither shaped nor modelled by it. Although this stance has limitations when it comes to implementation of software products and systems, it will be considered continuously throughout this study.

However, in the implementation of any system, the world has to be considered as being structurally ordered to some extent. Hence we subject the world to some formalisms and conceptualizations which have to fit some defined intention.

In conclusion, the research rejects the objectivist position and calls for the subjectivist acceptance of how people view and interpret reality. This research is concerned with the creation, modification and interpretation of the world. This posits the research as relativistic throughout the project.

2.9.4 Methodological Stance

In any research, understanding the nature of that which is to be studied is of paramount importance. From the above it can be seen that research can be either quantitative or qualitative. The major goal of this research is to find ways of improving the development of software products, which are later incorporated into information systems. By its nature, the problem under investigation cannot truly yield purely objective results, irrespective of the research approach used in the study. This process makes the present research interpretive in nature. The research method to be adopted for this study "must be appropriate for the research question" (Avison *et al.,* 2008:11). In addition, it should be applied appropriately relative to the research problem under investigation. Suddaby (2006) concurs, saying that there should always be consistency between the research questions, the assumptions made about the world view and how people perceive it, and the methods used to address these research questions. This chapter has attempted to find a link between these tenets.

2.10 Summary

The contents of this chapter can be summarized as the *'Philosophy of Information Systems Research'*. Of importance to its inclusion in this thesis is the fact that it discussed most of the basic philosophical groundings in information systems. This chapter is a blueprint on thinking about research in the information systems discipline. It provides a road map to issues that should be discussed and embraced by researchers whenever they decide to conduct research into information systems. Researchers should understand that information systems problems vary in structure, form and context. This alone requires people to understand the philosophical nature of these problems before selecting a research methodology.

Any type of information system research should, however, answer the following questions:

- What type of research is being undertaken? This looks at the dichotomy between theoretical and empirical types of research.
- What is the purpose of the research: generation of theory or testing of theory? The research task should then be constructive, nomothetic or ideographic.

- With regard to the research problem, how should nature be conceived? The four sociological paradigms, together with the ontological, epistemological and humanist philosophical underpinnings, should be used as lenses to understand the research problem.
- What research paradigm should an information systems research project follow? This question is answered by the dichotomy between qualitative and quantitative research paradigms in conjunction with the three approaches to information system research: positivist, interpretive and critical realism.
- The last question then addresses the elicitation or positioning of the nature of the research problem. This part should guide the researcher in the selection of the methodological process for the research.

In conclusion, this research is equally theoretical and partly empirical, constructive and partly ideographic. It assumes a relativist, anti-positivist, interpretive stance and will use GTM as the research method. In Chapter 3, *Research Methodology and Design,* the methodological processes followed in this study are discussed.