Interpreting The Process Of
Configuration of Packaged Software Through the Lens of Social Construction of Technology

A Dissertation
Presented to
The Faculty of the C.T. Bauer College of Business of
University of Houston

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy

By
Elham Mousavidin
August 2010

Acknowledgements

This dissertation is dedicated to the loving memory of my grandmother Fatemeh Shamlou whose presence I still feel strongly every day…

The road has been enriching, joyful, fun, challenging and bumpy and I made it! Getting here would not have been possible without the support and love that I have received from many. The list is too long to recall every single name to put down on paper as I am writing these lines. I would like to take this chance though to thank certain individuals who have been influential specifically on the process of completion of this dissertation.

To the faculty…

Thank you Dr. Silva for being a great mentor, a patient advisor, a supportive colleague and a caring friend! Your constant encouragement to be critical and ethical, to have passion and joy, and to strive for high quality in conducting research has been a powerful force in the process of completion of my doctoral studies. I have learned so much from you.

Thank you Dr. Scamell for your constant support from the first pat on the shoulder you gave me when you hired me as a TA years ago, to encouraging me to pursue my doctoral studies, and to the last days of completion of my dissertation! Thanks for being critical of my work and providing me great feedback to revise my dissertation!

Thank you Dr. Kotarba and Dr. Porra for taking the time to examine my work and providing me invaluable feedback! It was an honor to have researchers of such high caliber on my dissertation committee.

Thank you Dr. Khumawala for all your encouragement to aim for high goals and to keep learning, thank you for believing in me!

I would like to also express my gratitude to the other MIS faculty members Dr. Adams, Dr. Chin, Dr. Cooper, Dr. Cossick, Dr. Ives, Dr. Johnson, Dr. Junglas, Dr. Parks, and Dr. Scott.

To the anonymous individuals…

I would like to thank all those anonymous individuals from the two organizations I collected my data from. Your acceptance, trust, and valuable insights made this dissertation possible.

To my friends…

Thank you my DISC colleagues and friends, Judy, Karen, Hesam, Rima, Vlad, Rick, Kevin, Cindy, Shane, Mengge, Minh, Doug, Manjari, Nellai, Siew Fan, Mike, Latoya, Mary, Ying, Barry, Tarun, and Sergei; whether we met briefly or shared an office during the doctoral program, I always felt I could count on your support.

Thank you Iris for always checking on me to make sure I'm on schedule to complete my dissertation and for providing me with such constructive (and at times tough!) comments… Thank you Kenny for supporting me when I needed it the most, for always encouraging me and genuinely being happy for my success… Thank you Reza for being a great officemate, study buddy, and friend; the writing process became more fun and less stressful because of you… Thank you Isabelle, Marjan, Neda, Ladanak and Simin for listening, encouraging, and reminding me to take breaths and breaks… Thank you Noor, Bahar, and Behrooz for the best tea, movie, and music breaks; all timely and very well needed…

I would like to thank five friends whose support and constant push have had a tremendous effect on the completion of this dissertation specifically… In alphabetical order those friends are: Jennifer, Lakshmi, Maushmi, Nazbanoo, and Tinoosh. Thanks doosteh man, Jennifer, for sharing so many great memories with me and loving me like a sister… Thank you Lakshmi joon for being an amazing pal and one of the best co-authors one could wish for; for reading my dissertation and constantly encouraging me to keep going… Thank you Maushmi for your kindness and big heart; you may never realize the depth of the impact your support has had on this process… Thank you Nazbanoo for always being there for me, believing in me, motivating me, and listening to my worries… and Thank you Tinooshi for your genuine friendship; for always looking after me, worrying about me, and encouraging me…

And last but not the least to my family…
To my parents… I would never be able to express the depth of love and gratitude I have for you… Thanks maman, thanks baba for sacrificing so much for us!

To my Amoo Hassan and Zanamoo Patricia… None of this would have been possible without your constant support… Thanks for always encouraging me and loving me!

To Omeed and Reza, the best brothers in the world… It sure feels great to be loved and supported by you… To Alana, Sara, and Mona, you made my all time wish for a sister come true; thanks for always loving me, believing in me, and supporting me… To my nieces and nephews Mina, Reza, Desiree, Ava, Chloe, Jack, Lauren, Madeline and the new additions Kaspar and Termeh, loving you is the best feeling in the world; the warmth of this love kept me going during the time of the Ph.D. program. To Glenn, James, and Liz, I am so fortunate to have you in my life; thanks for loving me and believing in me! To all my dear friends, aunts, uncles, and cousins back home, I miss you tremendously; the sweet memories we've shared are a treasure very dear to my heart.

Abstract

Abstract

Despite the predominance of packaged application software, information systems (IS) researchers have focused more on custom software development. Packaged software deserves to be investigated separately due to the uniqueness in terms of its development and implementation. One of the challenges for organizations is to fit packaged software into their organizational context in order to reap the benefits through adopting the best practices this type of software promises to offer. This dissertation concerns the implementation of packaged software, specifically, technical and organizational aspects of its post-purchase configuration. An interpretive study aimed at investigating this process through the lens of Social Construction of Technology (SCOT) was conducted. SCOT is a sociological approach to studying development of technological artifacts. One of the main premises of SCOT is that various relevant groups influence the course of development of a technological artifact based on their varied interpretations of the artifact. SCOT lent itself well to the context of this study, in which the influence of the understanding, perceptions and expectations of various groups on the configuration process of packaged software was studied.

The empirical part of this dissertation consists of two case studies: a pilot study at a non-profit organization and a main case study at a large public organization. The pilot study was conducted in 2005 with the aim of examining SCOT in the context of IS research and refining it as a theoretical lens for this dissertation. The data collection for the main case study began in Spring 2007 by contacting a public organization that was in the process of implementing a work management software package. The researcher participated as a neutral observer in the simulation sessions conducted during the configuration of this software. In addition, during the configuration process, the researcher reviewed organizational documents related to the project and conducted semi-structured interviews with the members of the configuration team.

The findings of this dissertation exhibited that the configuration of packaged software was an interpretive process through which various features were implemented. The final

implementation of each feature was the result of the interpretation and re-interpretation of various configuration options. Each process of interpretation and re-interpretation generally resulted in a dominant interpretation, the output of which took the form of a minimal, moderate, or elaborate configuration solution. For each feature, the choice of one of these solutions was influenced by the discourse forces (optimism, pessimism, and indifference) that were dominant at the time. The generalization of these findings was theoretically modeled in a mechanism for the process of configuring packaged software.

In addition, this study suggested a re-conceptualization of what constitutes an IS user. Traditionally, most IS studies have had an individualistic view of users and users are grouped based on their functional roles. The findings of this dissertation proposed treating users as social actors, the grouping of whom is based on their interpretations of the IS with which they interact. Based on such a view, grouping of users occurs after their interpretations and perceptions are sought. This is different than the prevalent approach of first grouping the users (mostly based on their functional roles) and then assigning certain beliefs and perceptions to each group.

The findings of this dissertation add to IS theory and practice. A theoretical contribution consists of adding to the literature on software configuration by providing a mechanism as well as a definition of the configuration process. In addition, this research makes a contribution to the SCOT approach in the context of its application in IS research since this theory has not been applied holistically in IS studies. It is believed that practitioners can benefit from the findings of this research by applying the configuration mechanism developed as an analytical tool to understand and manage the process of configuring packaged software.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1. Research Motivation

Although the origin of application packaged software dates back to the late 1960s, the first serious Information Systems (IS) studies on this topic began in late 1980s with Lucas (1988) seminal paper. Packaged software resulted from a 1968 initiative of the United States Department of Justice that required IBM to unbundle software from hardware (Carmel, 1997; Goetz, 2001a; Sawyer, 2000). This has changed the nature of software implementation by organizations dramatically over the past few decades. Today, organizations do not necessarily need to have highly skilled software developers in order to benefit from the latest software developments. Ability to purchase packaged software is believed to save organizations time, money and other resources, while providing functionality that would otherwise be economically not feasible (Janson & Subramanian, 1996).

Despite the domination of packaged application software, Information Systems (IS) researchers have concentrated mostly on custom developed software (Light & Sawyer, 2007). Research on custom development is important because many large (especially governmental) organizations, which need specialized software, still custom develop their software applications. However, a large number of organizations meet little of their software needs through custom development (Light & Sawyer, 2007; Yeow & Sia, 2008).

Packaged software deserves to be investigated separately (from custom-developed software) due to its uniqueness in terms of development and use (Light & Sawyer, 2007). From the development point of view issues such as not involving the users, building workable software for different organizational contexts, and offering products through third-party organizations to end users make packaged software significantly different from custom-built software (Sawyer, 2000). On the other hand, when it comes to using packaged software, organizations need to either adapt their organizational processes to the packaged software or configure the software to fit organizational processes. This is not necessarily the case in use of custom developed software because this kind of application can mirror existing organizational processes (Yeow & Sia, 2008).

The only form of packaged software that has received generous attention from the IS research community is Enterprise Resource Planning (ERP). Since the 1990s, there has been a tendency in the IS research community to exemplify packaged software by ERP applications (Light & Sawyer, 2007). While the findings of this research stream are valuable and insightful, understanding more 'generic' aspects of packaged software and challenges that organizations face when adopting third party software has been under-investigated. Therefore, IS literature lacks a clear understanding of the scale and scope of packaged software (Ibid).

Recently, some researchers in the IS community have begun to bring the importance and relevance of studying packaged software issues to the forefront (e.g. Davidson & Chiasson, 2005; Light & Sawyer, 2007). Nevertheless, the importance of packaged software in IS research appears to ebb and flow. Lucas (1988) strongly advocated the significance of this topic. In 1997, Carmel's article examined the factors that would sustain hegemony of the U.S. in packaged software market. In 2007, a special issue of

European Journal of Information Systems (EJIS) examined the topic of packaged software. Light and Sawyer, in the commentary that accompanied this issue, called for attention to theorizing about packaged software and its location in IS research. Other researchers have also shown interest in this topic around each of these timelines (e.g. Anderson, 1990; Iivari, 1990; Janson & Subramanian, 1996; Davidson & Chiasson, 2005). However, the body of literature on software packages remains scarce and fragmented.

Packaged software has received more attention from computer science and engineering communities, though both disciplines tend to focus on the technical rather than the social and organizational aspects. In these fields, for most part, the topics of selection and evaluation of software has been the primary focus (Mohamed et al., 2007; Soffer et al., 2001). One assumption of research centered on selection and evaluation is that once an organization selects the most fitting software for their organizational needs, they will be able to reap the benefits that the software is designed to offer (Montazemi et al., 1996). What these studies neglect to mention is that even though pre-purchase evaluations of different software options is useful for finding the best fit, the actual fit or misfit of the software will not be realized until it is being configured and implemented (Yeow & Sia, 2008). Even if an organization considers that they have found the best fitting software, the benefits of the software have the potential of not being realized due to organizational and social issues (Ibid). Unfortunately, discussion of the issues around post-purchase configuration of packaged software is rare even though extremely critical.

1.2. Research Question

A review of prior literature reveals that there is no common understanding as to what constitutes packaged software. Several attributes are commonly assigned to this type of software by different researchers. Some of these attributes include the software being already-built (Oberndorf, 1997; Vigder et al., 1996; Vigder & Dean, 1997) without any control from the customers over its development (Basili & Boehm, 2001; Vigder et al., 1996; Vigder & Dean, 1997), distributed in large number of copies, licensed, leased, or bought (Oberndorf, 1997; Brownsword et al., 2000).

Some of the less commonly mentioned attributes of packaged software in the literature include the software being offered by third-party vendors, which retain the intellectual property rights over the software (Brownsword et al., 2000), the software needing minimal change and no source code modifications, and customers not having access to the source code (Vigder et al., 1996; Vigder & Dean, 1997).

In this dissertation the focus is on the type of packaged software that is purpose-specific. In classifying different forms of packaged software, Morisio and Torchiano (2002) consider the attribute of role to distinguish various packaged software based on their intrinsic function. Role can have two dimensions: horizontal and vertical. A broad vertical role denotes that the software covers a large number of functional areas involved in the same business process. A broad horizontal role entails that the software covers large number of functions. For example, Enterprise Resource Planning (ERP) systems encompass broad horizontal roles. Therefore, horizontal role is what distinguishes ERP systems from the packaged software with smaller scope that is of the focus in this study.

It is argued that almost all software applications require some degree of configuration before they become workable (Sommerville, 2008). Configuration can vary from simply setting up the databases and the parameters to modifying organizational business processes (Lucas et al., 1988; Vigder & Dean, 2000; Soffer et al., 2001). Configuration of packaged software is seldom a simple ‘plug-and-play’ process. This process can become extremely complex and time-consuming (Feblowitz & Greenspan, 1998). The process of forming software that fits a specific organizational context can turn into a challenging process (Soh et al., 2000; Light, 2005a, b; Wagner et al., 2006; Yeow & Sia, 2008).

Accordingly most organizations need to either adapt their processes to the software or modify the software to fit their practices (Soh & Sia, 2004, Sia & Soh, 2007; Wagner et al., 2006). Therefore, organizations usually assign groups from different relevant functional areas to configure and implement the software to fit the organizational context. Each of these groups could have different perceptions and understanding of the software and its capabilities. In other words, the software and practices it supports could be interpreted in different ways. These interpretations are key in shaping the direction of the process of configuration: adapting the organizational process to the software, or configuring the software to fit the processes. Therefore, the configuration process of packaged software[1] can involve technical as well as social and organizational adaptations

[1]For the sake of brevity, throughout this manuscript, the terms “configuration process” and “configuration” have also been interchangeably used to refer to the term “configuration process of packaged software”.

(Davidson & Chiassen, 2005; Sommerville, 2008). Typical activities such as defining reporting format, changing the culture, and changing the business processes are, respectively, some of the examples of technical, social, and organizational adaptations in a configuration process (Light & Sawyer, 2007; Somerville, 2008; Yeow & Sia, 2008).

Delving into the dynamics of post-purchase activities, therefore, would be significant and timely. In order to gain an understanding of these dynamics, examining interactions of different groups who are involved in the process would be insightful. Studying how these interactions influence the way the software is configured and how in turn the options and limitations offered by the software influence the interactions will add to our understanding of software implementation and configuration.

One of the contributions of this dissertation is to offer a synthesized definition of packaged software based on the review of literature. The literature review is provided in chapter two. The type of packaged software that is of concern in this study is defined as:

> *Standalone dedicated application software that is already-built, sold, licensed, leased, borrowed, or given away free of charge in many copies in different forms (e.g. on CD ROM or on the Web), has been developed without involvement of the customers, and requires configuration (e.g. parameterization, setting up interfaces and databases) with involvement of different relevant groups. Configuration can vary from simple parameterization to tailoring and customization without change of the source code. This type of software has a narrow horizontal role (i.e low number of purpose-specific processes) but can have a narrow or broad vertical role (i.e. covering high number of functional areas in the same process).*

Another contribution of this dissertation is to shed some light on the understanding of the configuration process. This objective is summarized in the main two research questions that serve as the drivers of this dissertation. These research questions include:

1) Why is packaged software configured a specific way and not another?
2) How does this configuration occur?

In order to understand the process of configuration of packaged software and its technical and organizational dynamics, an interpretive case study was conducted at a large public organization, Pub Org. The Social Construction of Technology (SCOT) was the theoretical lens adopted to conduct this study. This lens was deemed appropriate for such a study since it is a theory about technological developments through the influence of relevant social groups that are involved in this process. SCOT is a theory of socio-technical change (Mumford, 2006), which takes a sociological approach to studying technological developments (Bijker, 1995). SCOT's emphasis on thick description (Geertz, 1973) of the *technological artifact* and its surrounding context makes it a well-rounded theoretical lens. One of the main premises of SCOT is that technology is interpreted differently by various social groups that are relevant to the technology. In other words, this process is considered to be an interpretive one. Therefore, one of the objectives of this dissertation was to understand interpretation of various groups that are involved in a process of packaged software configuration. The researcher intended to investigate how these interpretations influence the process of configuration and the final configured workable software.

SCOT is highly relevant in the context of configuration of packaged software. Various relevant groups are normally involved in the process of configuration. Perceptions and expectations of these different groups of the packaged software and its functionalities can be critical. Various understandings of the packaged software by the relevant groups are reflected in the suggestions they provide for how to configure the software to a workable form. The decisions that are made during the process can be highly influenced by different interpretations. Rich functionalities of software can make understanding of what the software potentially offers and how it can be configured challenging. This challenge can be amplified if different relevant groups understand the software differently.

The insights from this study were summarized as theoretical generalizations offered in the form a working definition for the configuration process as well as a mechanism that depicts this definition and the dynamics around the configuration process. These implications are discussed in the final chapter of this dissertation, chapter seven. The following sections provide a detailed overview of the contents of each chapter.

1.3. Dissertation Outline

This section provides a roadmap of the entire dissertation. The following sub-sections present an overview of the content of chapters two through seven.

1.3.1. Overview of Chapter Two

Literature Review-- Packaged Software & Technological Frames of Reference

Chapter two is organized into three different literature reviews. The first part looks into the literature on packaged software in computer science, computer engineering, and information systems. This section synthesizes a working definition of packaged software based on the literature. In this section advantages and disadvantages of packaged software are also discussed. Chapter two continues with a discussion of the configuration process of packaged software. This section elaborates on challenges that accompany the configuration of packaged software in organizations. This discussion also touches on some of the shortcomings of the literature on the phenomenon of configuration, which is eventually the basis of the main research questions of this dissertation.

The chapter then explains two possible theoretical lenses for conducting such a study. The first theoretical lens originating in the IS literature is Technological Frames of Reference (TFR) (Orlikowski & Gash, 1994). Orlikowski and Gash define technological frames as the

> "subset of members' organizational frames that concern the assumptions, expectations, and knowledge they use to understand technology in organizations. This includes not only the nature and role of the technology itself, but the specific conditions, applications, and consequences of that technology in particular contexts" (p. 187).

According to TFR individuals share a technological frame reference in a group. Moreover, usually there is frame incongruence among technological frames of various groups. TFR has been applied as a theoretical lens in several studies on use of information systems. This theoretical framework is discussed in depth in chapter two accompanied with a discussion of the origin of this theory. TFR was used in the pilot study of this dissertation. In this pilot study, conducted at a non-profit organization, Child Org, use of two information systems CDT and CC Solutions, and the need for a new system which would integrate functionalities of these two were investigated.

During the pilot study, the researcher faced some challenges in explaining within-group discrepancies that existed in the technological frames of reference. Initially, an emergent theme which was related to the expectations that various groups held from one another seemed to be explaining the discrepancies. However, upon further analysis, the researcher was convinced that TFR was not sufficient in explaining the case of Child Org. Further familiarity with the theory of social construction of technology led to the belief that SCOT might offer an explanation for the gaps that TFR could not fill. Therefore, Child Org's data was re-analyzed through the lens of SCOT. This case is explained in detail in chapter five. Therefore, chapter two offers a brief introduction of the theory of SCOT as well as a discussion of how and why SCOT would serve as an appropriate lens for this dissertation.

1.3.2. Chapter Three

Theoretical Lens of SCOT

Chapter three elaborates on the discussion of the theoretical lens of this dissertation, namely SCOT. One of the main premises of SCOT is that technology is interpreted differently by various social groups that are relevant to the technology. Each group consists of individuals who are, in a way, concerned with the technology. Members of each group share the same meanings about the technological artifact, which is different from the meaning(s) other groups assign to it (Bijker, 1992; Bijker, 1995).

Accordingly, the classification of different *relevant social groups* is based on their interpretations of the technology. In other words, individuals who interpret the technology similarly and thus approach its problems in a similar fashion are considered to be members of the same relevant group (Bijker et al., 1987; Bijker, 1995). This is dramatically different than the traditional way of classifying groups based on their functional areas (e.g. users, developers, and managers) that is prevalent in IS literature. An advantage of this new way of classification is that people are grouped based on their points of view rather than being attributed with some points of view based on their job descriptions. Due to different interpretations assigned to a technological artifact, the technology is said to have *interpretive flexibility* (Ibid). This means that different groups of people approach the technology differently. These groups of people contribute to gradual shaping of the technology until it takes a workable form. In the SCOT approach the focus is on *controversies and their resolutions*. By identifying the problems that various relevant groups have with technology, the researcher more closely examines how those controversies are approached, resolved and eventually the technology is interpreted as workable by various relevant groups (Bijker et al., 1987; Bijker, 1992; Bijker, 1995).

The way a group approaches the technology and its related issues is called the *technological frame of reference (TFR)* of the group. TFR of a group guides interactions of the groups with one another and with the technology. TFRs are built when interactions around a technology begin. Different groups contribute to gradual shaping of the technology until it reaches stabilization and takes a working form. At this point the technology is said to have reached closure. There are different ways to reach closure. For example, power could be used as a mechanism to reach closure and to resolve controversies around the technology (Bijker et al., 1987; Bijker, 1992; Bijker, 1995). Closure does not mean that the technological artifact cannot change in form further. What it implies is that at the time of closure the controversies around the technology are resolved and its interpretive flexibility has diminished. Choice of SCOT to study the configuration process of packaged software seemed warranted since usually in a configuration process people from various groups are involved.

According to SCOT different people involved in the process of development of a technology can be grouped based on the shared meaning they assign to the software. This kind of approach to the process of configuration could allow us to examine different perceptions about the software, its functionalities, and configuration options. Taking a SCOT approach, a researcher would identify problems of various relevant groups regarding the software. Identification of these problems would allow the researcher to closely examine how they are approached and addressed. The researcher would be able to study the process of configuration as a series of technical, social and organizational activities. Furthermore, the emphasis of SCOT on the technological artifact (the packaged software in this case) and providing a thick description of its evolvement and the social and organizational issues around it make SCOT an insightful lens.

Chapter three concludes with a discussion of criticism of SCOT. This discussion was essential in order to be aware of the shortcomings of the theory when this study was conducted.

1.3.3. Chapter Four

Research Methodology

Chapter four elaborates on the research design of this dissertation. The empirical part of this dissertation consists of two case studies: the pilot study at Child Org and a main case study at a public organization (Pub Org). As mentioned before, chapter five elaborates on the pilot study. Data collection at Child Org began in Fall 2005. At the time, Child Org was in the process of searching for packaged software that would replace two of their existing software packages: CDT and CC Solutions.

Chapter six is allocated to the data analysis of the case of Pub Org. The data collection for the main case study began in Spring 2007 by contacting a public organization that was in the process of implementation of a work management system, 7i. This software was purchased as an off-the-shelf package and the organization was in the process of its configuration. The researcher participated as a neutral observer in twenty sessions conducted configuration of the software. These sessions were run by employees from different functional areas that are involved in the configuration process. The researcher also obtained and partially reviewed documentation material related to the project. Seven semi-structured interviews were also conducted; this was separate from the informal short interviews that the researcher conducted with the team members during configuration sessions. The process of data collection was guided by the theoretical lens of SCOT. This lens was also used in analyzing the data from different sources using a hermeneutic approach. Using various data sources - observation, interviews, and documentation – enriched the understanding of the phenomenon (i.e. the configuration process of packaged software), and helped in validating the findings.

1.3.4. Chapter Five

Pilot Study -- Child Org Case Study

The goal of this chapter was to provide an empirical content to the comparison of the two theoretical lenses of TFR and SCOT. In this chapter, first a discussion of the original analysis of Child Org's data through TFR lens is presented. Then, the re-analysis of the case through the lens of SCOT is provided. The chapter concludes by summarizing the lessons learned from the Child Org case study. These lessons not only supported the choice of SCOT as the theoretical lens throughout this dissertation but also shed light on how to apply SCOT in the context of information systems research.

As mentioned above, the initial analysis of the data proved to be challenging since the TFR framework did not seem to sufficiently explain the incongruity of technological frames of reference that was prevalent among members of the same group. The re-analysis of the same data through SCOT shed light on the challenge that was faced in the first round of data analysis.

1.3.5. Chapter Six

Data Analysis -- Pub Org Case Study

Chapter six elaborates on the analysis of Pub Org's data. The process of coding the data based on the theoretical lens of SCOT is first described in detail, providing a step by step description of the process through which data was read and the text that corresponded to SCOT theoretical lens and the configuration process were highlighted. The analysis of data is then presented in two sets. The first set of data analysis corresponds to technical features of the system. Through this analysis of data, controversies around various technical features of the system and various interpretations around them are identified. The analysis then elaborates on how these problems were addressed based on the diversity of interpretations. The second set of data analysis is related to organizational as well as individual dimensions of the configuration process. In this section, a discussion of organizational aspects as well as attributes of individuals which influenced the process of configuration is presented. The chapter concludes with a discussion of how various dimensions of the configuration process fit together. These conclusions set the stage for chapter seven, wherein a mechanism of process of configuration is developed and discussed.

1.3.6. Chapter Seven

Discussion and Implications

Chapter seven is divided into three sections. In the first section, development of a mechanism for the process of configuration and its technological, organizational, and individual elements are discussed. In the second section, contributions of this study to research, evaluation of these contributions, and contributions of this study to practice have been elaborated on. The chapter concludes with a discussion of the limitations of this research as well as some proposals for future relevant research areas.

Chapter 2

Literature Review: Packaged Software & Technological Frames of Reference

2.1. Introduction

The purpose of this chapter is to first define the type of packaged software that is of concern in this dissertation. In doing so, we locate packaged software within the classification of software in general. Second, based on various definitions and categorizations offered in the computer science and IS literature, we build our definition and classification of packaged software. Next, we discuss the issue of the configuration of packaged software. The social and organizational impacts of the configuration process are also discussed. We then argue how the SCOT approach could enrich our insights about the configuration of packaged software to workable forms in organizational contexts.

Accordingly this research was undertaken to study the process of packaged software configuration, in which several groups of people are involved. Organizational studies have shown that different individuals/groups interpret and understand a situation differently and they act according to their varied understandings (Walsh, 1995). Previous research, for the most part has focused on investigating perceptions of different parties (mostly individuals) about information systems based on some pre-defined attributes of the technology (e.g. ease of use) (Davis, 1986). Some studies treat understanding and perceptions of individuals/groups towards a technology as contextual and situation-specific. Even in these studies there has been a tendency to investigate the contextual difference of the individual/groups' based on some predefined domains (e.g. technology-in-use) (Orlikowski & Gash, 1994).

Although the aforementioned approaches would be valuable to studying packaged software configuration, this area of research could benefit from an interpretive study in which the actions and interactions of participating groups are scrutinized from a more holistic view. Social construction of technology (SCOT) could offer an interpretive lens to conduct such a study. This approach has been applied in sociological studies of hard technologies[1]. The importance and relevance of the application of SCOT in IS research has been repeatedly acknowledged (Orlikowski &Gash, 1994; Davidson, 1996, Lin, 2000) but it has not been extensively adopted. At the core of SCOT, the idea is that different groups that are involved in the development of a technology understand, interpret, and act towards it differently. Accordingly, the interactions within and across the groups and with the technology are guided by what are called their technological frames. The idea of technological frames with a socio-cognitive base, initiated by Orlikowski and Gash (1994), has been applied in IS research. Socio-cognitive technological frames is still a relatively under-researched topic and deals with unanswered questions. Since this approach is aimed at answering similar questions to those of a SCOT approach, a review of socio-cognitive view of frames is deemed necessary. Therefore, a comparison between the two clarifies how a SCOT-based view could add value to our study.

The second part of this chapter (section 2.3) is allocated to the topic of technological frames of reference, its roots, and applications and implications in the IS research. Finally, we introduce SCOT briefly and discuss its previous applications in IS literature.

---

[1] Technologies such as bicycle, which compared to information systems (called soft technologies in this research) are less malleable and flexible to change

We postpone a comprehensive discussion of SCOT as our theoretical lens to chapter three. Finally, at the end of chapter three we discuss how and why SCOT would be an appropriate lens to study our research questions related to the configuration process of packaged software.

2.2. Packaged Software

Depending on their purpose, it is possible to classify software in various ways. The Software and Information Industry Association (SIIA) categorizes software based on various industries to: ‘shrink-wrapped’ PC packaged applications; purpose-specific customized application software; enterprise software; software as a Service (SaaS); operating systems for networked and standalone systems; network management tools; and operating systems and software applications for mainframe computers (SIIA, 2004).

It is also possible to categorize software based on the purpose it serves. For example, Xu and Brinkkemper (2007) use a two dimensional matrix to classify software. These dimensions include “what is sold” and “number of copies sold”. The former refers to the software being sold either as a part of another appliance or the software itself being the product that is used. For example software could either be embedded in a machine such as a phone or it can be used as a tool such as Microsoft office. In each of these cases the software could be sold in large numbers or be custom-built for specific customers. When sold in a large number of copies, the software sold in an appliance is called embedded and the other type is called product software by Xu and Brinkkemper (2007). The authors call a specialized program such as one used in a spacecraft as micro-program. They refer to specialized one-time built software as tailor-made. Figure 2.1 shows Xu and Brinkkemper's (2007) classification.

What is sold?			
Appliance	**Micro-program**	**Embedded software**	
Software	**Tailor-made software**	**Product software**	
	One	**Many**	***Number of copies***

Figure 2.1: Software classification (Source: Xu and Brinkkemper (2007))

Different aforementioned types of software as classified by SIIA, could also be categorized according to Xu and Brinkkemper's matrix. This is dependent on our purpose and the view we take. For example, operating systems would be considered as embedded software from the view of typical non-IT users of an organization. This kind of software runs in the background and a typical user does not have to interact with it regularly. The type of software that is of concern in this dissertation would fall under the fourth category in Xu and Brinkkemper's matrix: Product software. Of course, Xu and Brinkkemper distinguish between the terms packaged software and product software. They define product software as "a packaged configuration of software components or a software-based service, with auxiliary materials, which is released for and traded in a specific market" (p. 534). Their definition of packaged software is "ready-made software products that can be readily obtained from software vendors and which generally require little modification or customization" (p. 534). As can be seen both product and packaged software would be qualified to be in the fourth category of their matrix (i.e. product software category). In addition, we also view the auxiliary material and the software itself all as part of a package.

Therefore, in order to have a broader use of Xu and Brinkkemper's matrix, instead of naming each category (e.g. embedded software, product software) we merely refer to them as categories 1 to 4. Moreover, we rename the values on the vertical axis ("what is sold") to Embedded and Standalone. This is how the authors themselves refer to these categories in the text of their article: "software can be either a standalone software product or software embedded in a system" (Xu &Brinkkemper, 2007, p. 532). For instance, they consider a micro-program to be an example of embedded software. In the matrix, however, they use embedded as one of the four types of software. The modified matrix would look like Figure 2.2.

	Number of copies	
Software product	**One**	**Many**
Embedded	***Category 1***	***Category 2***
Standalone	***Category 3***	***Category 4***

Examples of different categories:
***Category 1*: Embedded software in a spacecraft**
***Category 2*: Embedded software in a phone**
***Category 3*: Custom-built software for accounting**
***Category 4*: Application software such as MS Word**

Figure 2.2: Modified version of Xu and Brinkkemper's (2007) software classification

There is no common definition as to what constitutes packaged software. Various definitions of packaged software have been offered by researchers. At the same time, several terminologies have been used to refer to what many understand as packaged software. Packaged software has been variably referred to as terms such as off-the-shelf, commercial, commercial-off-the-shelf (COTS), shrink-wrapped, standard, common and product software (Keen et al., 1982; Visker & Bree, 1987; Carmel & Sawyer, 1998; Light & Sawyer, 2007; Xu & Brinkkemper, 2007). Some researchers distinguish these different terminologies and some use them interchangeably. Either way, definitions of this type of software are varied. In the following discussion, we examine different terminologies offered in the literature and gradually build the definition of the type of packaged software we intend to study in this dissertation.

2.2.1 Definition of Packaged Software

Software and information industry association (SIIA) defines packaged software that is "written for mass distribution, not for the specific needs of a particular user, and may be distributed in any format – electronic download, physical media, such as disk or CD, or web-based service" (SIIA, 2004). SIIA broadly categorizes packaged software as: operating systems, utilities, applications, and programming languages (SIIA, 2004).

Thus far, based on what we have explained from the literature, the type of software that is of concern in this dissertation (henceforth called packaged software), is what is commonly called application (vs. system) software. In addition, according to Xu and Brinkkemper's (2007) matrix, packaged software is usually sold in large number of copies as standalone software (vs. embedded). This translates to the following definition:

> *Packaged software is standalone application software, which is sold in a large number of copies.*

From a review of prior literature on packaged software some common attributes were identified which describe packaged software. These attributes imply that packaged software is already-built (pre-existing) (Oberndorf, 1997; Vigder et al., 1996; Vigder & Dean, 1997), is distributed in many copies (offered to the public) (Oberndorf, 1997; Vigder et al., 1996; Vigder & Dean, 1997), can be bought, leased, or licensed (Oberndorf, 1997; Brownsword et al., 2000), and that its customers have no control over its development (Basili & Boehm, 2001; Vigder et al., 1996; Vigder & Dean, 1997).

Some other attributes that were not shared by a majority of the studies we reviewed include that packaged software is offered by vendors (Brownsword et al., 2000), vendors retain intellectual property rights of packaged software (Brownsword et al., 2000), it does not require any source code modifications (Brownsword et al., 2000), customers do not have any access to the source code (Basili & Boehm, 2001; Vigder et al., 1996; Vigder & Dean, 1997), and it requires minimal change (Vigder et al., 1996; Vigder & Dean, 1997).

We modify our definition incorporating these new attributes. Some of these attributes do not make any difference to the phenomenon we are interested in investigating-namely configuration of packaged software. For example, retaining intellectual property rights of the software by vendor would not qualify it for or disqualify it from being considered by our study. In addition, whether the software is sold, licensed, leased, or given away free of charge, as long as it has to go through a configuration process involving different relevant groups we would consider it in our investigation. Therefore, our modified definition of packaged software would be:

> *Packaged software is standalone application software that is already-built, sold, licensed, leased, borrowed, or given away free of charge in many copies in different forms (e.g. on CD ROM or on the Web), has been developed without involvement of the customers, and requires configuration (e.g. parameterization, setting up interfaces and databases) with involvement of different relevant groups.*

Since our focus is the process of configuration in which different groups of people are involved, involvement of the original developers of the software or third party vendors/consultants in the process does not change the topic of our interest. As will be discussed later, we emphasize on grouping individuals based on their interpretations and not the functional areas they come from. Therefore, developers, consultants, and vendors would be treated the same as users as long this participation does not imply that the customers have option of customizing the software significantly (for example by changing the code). This is because limitations for customizing packaged software affect the social and organizational dynamics of the process of configuration. We are interested in seeing how this influences the process of interpretations and reinterpretation of the software till the software finds a workable form. Not having access to the source code is what separates the packaged software that we are interested in investigating from open-source software. In the latter case, the users have access to the source code and thus are able to make significant changes to the software, which has a different scope than configuring the software (e.g. setting the parameters and interfaces).

As can be seen from the literature and our attempt to come up with a definition of packaged software, packaged software itself can be classified in numerous ways. In an effort to clarify confusion about what to consider as commercial off-the-shelf software, Carney and Long (2000) suggest that we should avoid relying on and constantly inventing acronyms to refer to this kind of software. The main concern of Carney and Long is referring to this software as COTS and inventing terms such as GOTS ("off-the-shelf software owned by the government") and MOTS ("modifiable off-the-shelf software") based on the context of use.

The problem becomes even more critical and confusing when it comes to COTS-based development, which is in a nutshell development of more complex software products using several already-made available COTS (Kunda & Brooks, 2000; Ulkuniemi & Seppänen, 2004). The literature (academic and practitioner sources) show that in many cases COTS refers to COTS-based products. Therefore, there is no shared understanding of what COTS exactly refers to.

Carney and Long (2000) argue that since there is no agreed upon definition and classification of commercial packaged software, different acronyms are currently used interchangeably. This inconsistent use of various acronyms can be a source of great confusion. They state that, for example, many researchers or practitioners use the term COTS even when they refer to a modifiable packaged software. They conclude that the lines between different types of packaged software are blurry. The authors offer a framework, which distinguishes different types of packaged software based on their source of development and degree of modification. Carney and Long's (2000) framework is shown in Figure 2.3.

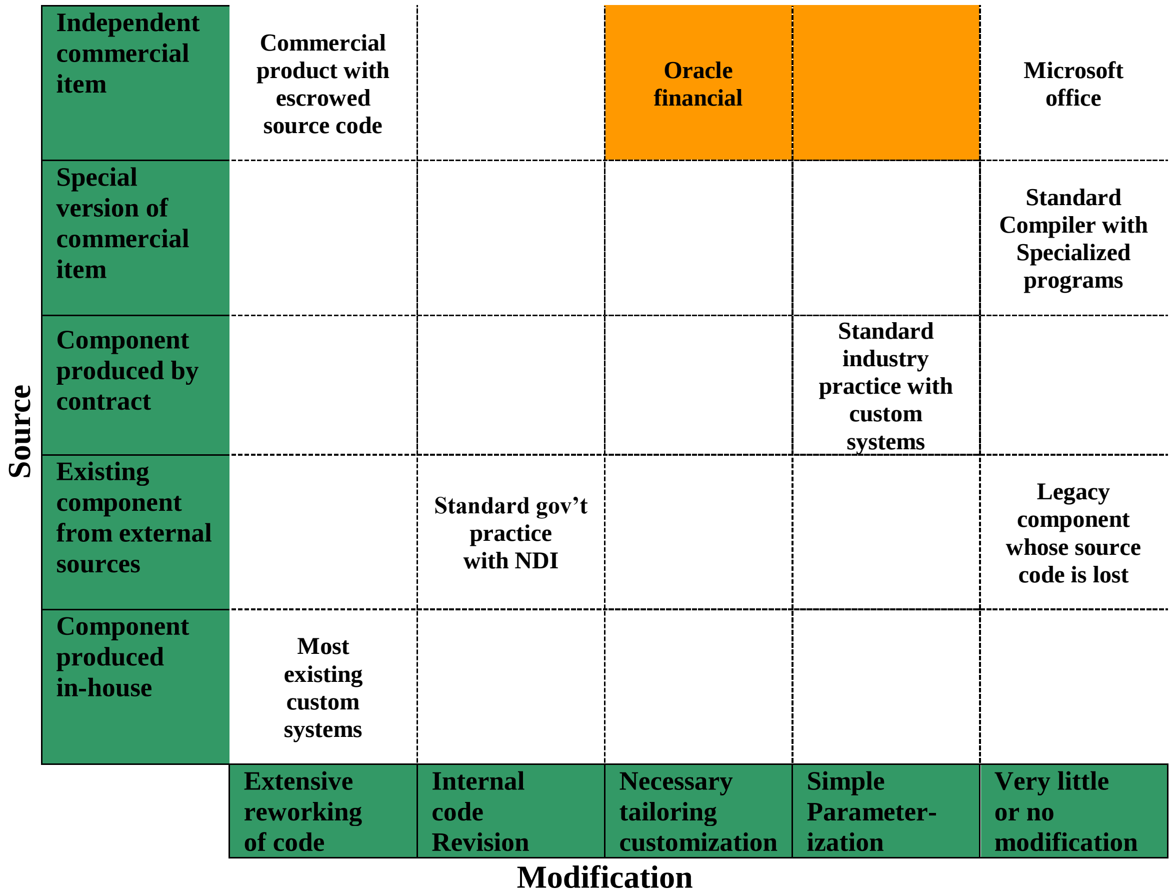

Figure 2.3: Carney and Long's (2000) framework for classifying packaged software

Incorporating Carney and Long's (2000) framework, the packaged software that is of concern in this dissertation would be considered an independent commercial item. As far as Modification dimension, this type of packaged software would fall somewhere between Necessary tailoring and customization, and Simple parameterization (highlighted in the graph). Therefore, instead of considering discrete values for the x-axis, we would consider the modification axis as a continuum.

We modify our definition of packaged software slightly based on this framework:

> *Packaged software is standalone application software that is already-built, sold, licensed, leased, borrowed, or given away free of charge in many copies in different forms (e.g. on CD ROM or on the Web), has been developed without involvement of the customers, and requires configuration (e.g. parameterization, setting up interfaces and databases) with involvement of different relevant groups. Configuration can vary from simple parameterization to tailoring and customization without change of the source code.*

Morisio and Torchiano (2002) also offer a way of classifying packaged software. Their framework includes two dimensions of source and customization which are similar to source and modification from Carney and Long's. In addition, Morisio and Torchiano consider two other dimensions of Bundle and Role.

Bundle attribute defines how the software is packaged and delivered, and what its size is. For packaging, Morisio and Torchiano (2002) consider values such as source code, statically and dynamically linkable library, stand-alone executable program, and binary component. The attribute of delivery is not very clearly defined by the authors. According to their framework, COTS can be either delivered completely or partly as an integrated part of another product, or be delivered separately (isolated) from any other product.

The last element of attribute of Bundle is the size of COTS, which can be small, medium, or large in terms of bytes. They also suggest a different measure based on the number of use cases that support the product. The authors use this method as a measure of computational size of software, which is not relevant to our study. The attribute of packaging has already been addressed in our definition of packaged software. Therefore, from their framework we adopt merely the dimension of Role to refine our definition of packaged software.

Role entails the intrinsic function the product can assume in its final form. Role could be in the form of horizontal or vertical functionality. Horizontal functionality “is not specific to a domain, but can be reused across many different application domains (e.g. DBMSs, GUIs, networking protocols, web browsers)” (Morisio and Torchiano, 2002, p. 7). Vertical functionality “is specific to a domain, and can be reused only in that domain (e.g. financial applications, [and] accounting)” (Morisio and Torchiano, 2002, p. 7).

A broad vertical role entails that the software covers involves a large number of functional areas in the same business process. According to this attribute, the packaged software that is of interest to this dissertation could cover many functional areas that are involved in the same related processes. For example, all the functional areas such as production and design that are affected by the process of equipment maintenance scheduling would be relevant to the same software (i.e. would be using the same software). A broad horizontal role denotes that software covers large number of functions. In other words, the software integrates various functional roles such as production and scheduling with sales and marketing. The type of packaged software we are concerned with is one with a narrow horizontal role (i.e. purpose-specific software).

Horizontal role is an attribute that distinguishes enterprise resource planning software (ERP) systems from a more domain specific packaged software which we are concerned with. This attribute gives ERP systems broader scope and scale in terms of functionalities, integration, and cost (Scott & Wagner, 2003; Wagner et al., 2006). Implementation and configuration of ERP systems deserve exclusive attention because of the complexity of the process. In this study we do not intend to delve into this topic. However, Implementation of one domain specific module of an ERP system (e.g. financial module) would also qualify as the type of implementation of packaged software that we are interested in investigating.

Incorporating the dimension of Role, our definition of packaged software becomes:

> *Packaged software is standalone application software that is already-built, sold, licensed, leased, borrowed, or given away free of charge in many copies in different forms (e.g. on CD ROM or on the Web), has been developed without involvement of the customers, and requires configuration (e.g. parameterization, setting up interfaces and databases) with involvement of different relevant groups. Configuration can vary from simple parameterization to tailoring and customization without change of the source code.* <u>*This type of software has a narrow horizontal role (i.e. low number of purpose-specific processes) but can have a narrow or broad vertical role (i.e. covering high number of functional areas in the same process).*</u>

In his seminal paper, Lucas (1988) distinguished between two types of packaged software. He broadly classified packaged software as dedicated and general purpose software. A dedicated package, he explained, “is dedicated to some particular function like accounts receivable, order entry, or production planning. Because the dedicated package is focused on a particular business function, rather than being general, an organization adopting the package may have to change its procedures or modify the package” (Lucas, 1988, p. 537). Even though some general purpose software (e.g. MS

Word) are configurable (Sommerville, 2008), we focus on *dedicated* software. This type of software involves various groups of people who are relevant to the business processes addressed by the software. Incorporating this attribute, the packaged software with which we are concerned is defined as follows:

> *Packaged software is standalone dedicated application software that is already-built, sold, licensed, leased, borrowed, or given away free of charge in many copies in different forms (e.g. on CD ROM or on the Web), has been developed without involvement of the customers, and requires configuration (e.g. parameterization, setting up interfaces and databases) with involvement of different relevant groups. Configuration can vary from simple parameterization to tailoring and customization without change of the source code. This type of software has a narrow horizontal role (i.e. low number of purpose-specific processes) but can have a narrow or broad vertical role (i.e. high number of functional areas in the same process).*

Accordingly, we offer the above definition for the type of software package, which is of the interest of this dissertation. This definition is graphically shown in Figure 2.4.

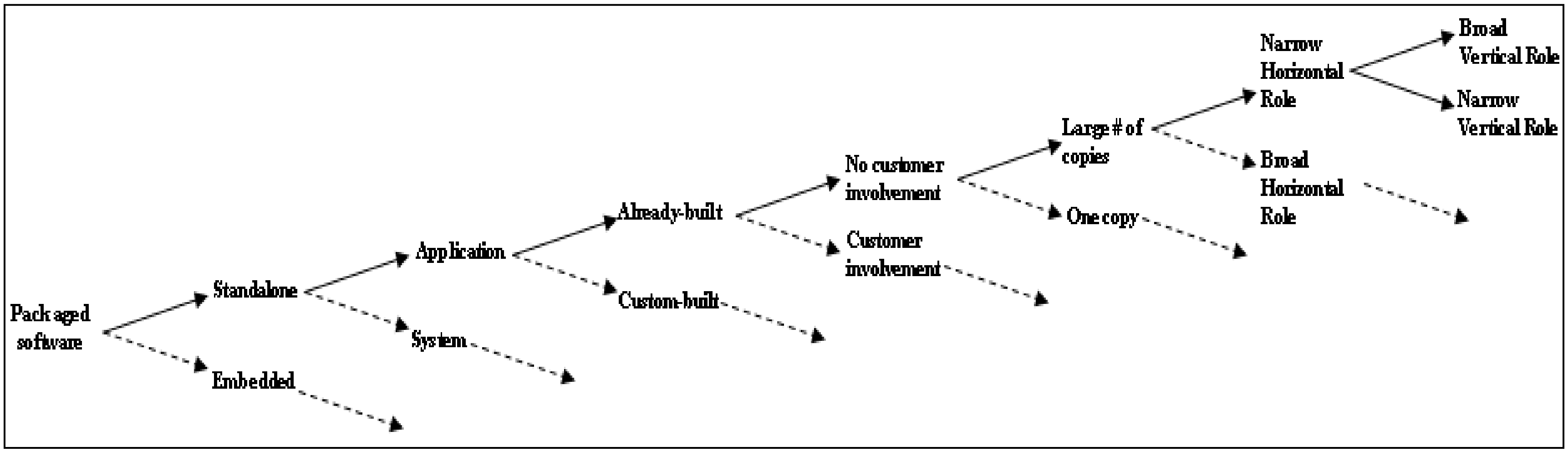

Figure 2.4: Definition of packaged software in this study

In next sub-section, we discuss some of the benefits and risks of using packaged software. We then turn to a discussion of the configuration of packaged software and its challenges.

2.2.2. Benefits and risks of packaged software

A discussion of the benefits and risks of packaged software helps us identify the value of studying packaged software. In doing so, we interpret the benefits and risks in terms of the configuration process, which is of concern in this dissertation.

Organizations choose to adopt packaged software for several reasons. When an organization does not have the required technical skills, time, and resources to develop the software they need, they purchase the equivalent commercially available package from industry. Packaged software is believed to offer: 1) rich functionalities, 2) broadly used, mature, and most up-to-date business practices of the industry, and 3) immediate availability and payback. These benefits are believed to be accompanied by the availability of constant support from expert organizations (Boehm & Abts, 1999; Light & Sawyer, 2007, Yeow & Sia, 2008).

However, packaged software is not devoid of disadvantages and challenges. Boehm and Abts (1999) identify some of the advantages and their corresponding disadvantages of packaged software. For example they believe that one of the advantages of using packaged software is that the organization has the option of adopting broadly used mature technologies. However, they argue that this advantage could result in the organization compromising in terms of functionalities and efficiency. The advantages and disadvantages identified by Boehm and Abts are summarized in Table 2.1.

Table 2.1: COTS advantages and disadvantages (source: Boehm & Abts (1999))

Advantages	Disadvantages
1) Immediately available; earlier payback	Licensing, Intellectual property procurement delays
2) Avoids expensive development	Up-front license fees
3) Avoids expensive maintenance	Recurring maintenance fees
4) Predictable, confirmable license fees and performance	Reliability often unknown or inadequate; scale difficult to change
5) Rich functionality	Too-rich functionality compromises usability, performance
6) Broadly used, mature technologies	Constraints on functionality, efficiency
7) Frequent upgrades often anticipate organization's needs	No control over upgrades and maintenance
8) Dedicated support organization	Dependence on vendor
9) Hardware/software independence	Integration not always trivial; incompatibilities among vendors
10)Track technology trends	Synchronizing multiple-vendor upgrades

After examining this list, we discuss these advantages and disadvantages of packaged software in the context of our study. Accordingly, we comment only on those items from Boehm and Abts's (1999) list that are applicable to various social and organizational aspects of post-purchase configuration.

The first advantage Boehm and Abts consider is immediate availability of the software. The disadvantage associated with this benefit is stated to be potential delays due to licensing and intellectual property procurements. We argue that immediate availability of this kind of software, in all cases, is questionable in the first place. The type of packaged software that we discuss in this study always requires configuration before taking a workable form. Further, the configuration process could potentially be extremely complex and thus result in delays. For example, the interaction of various relevant groups that are involved in the process might not go smoothly and delay the configuration.

Another advantage identified by Boehm and Abts relates to the rich functionality of packaged software. Boehm and Abts (1999) argue that, however, too-rich of functionality could compromise usability and efficiency. In line with Boehm and Abts, we believe that rich functionalities of packaged software could overwhelm the organization because of adding complexity to the software. This could result in the organizations not being able to adopt the capabilities of the software fully (Brooks, 1995; Davidson & Chiassen, 2005). We argue that the way software is interpreted by various relevant groups guides how they make decisions about it. Not understanding the functionalities that the software offers could add further complexity of the configuration process and even compromise on what software has the potential to deliver. The result could also be as extreme as not being able to configure the system to a workable form.

Because packaged software employs broadly used technologies, the adopting organization can take advantage of industry experiences and lessons learned (Boehm & Abts, 1999). However, as Boehm and Abts argue, this broad scope can also mean constraints on functionalities. These limitations can translate to more challenges for the relevant groups in configuring the software to work in the context of their organizational processes. Since the relevant groups face limited options regarding the functionalities, it can be more difficult to shape the software to a workable form for the organizational context.

Having discussed the benefits and challenges of packaged software in terms of its configuration, we delve next into the definition of configuration and its attributes.

2.2.3. Configuration of Packaged Software

The widely known term "configuration" associated with packaged software is related to configuration management (CM). CM is a technical term, which refers to a discipline for controlling the evolution of software systems (Dart, 1991). Configuration management is mainly for controlling the processes of building, modifying, and testing the code. In other words, software configuration management consists of controlling the evolution of complex systems (Estublier, 2000; Xu and Brinkkemper, 2007). Configuration management concerns mainly software developers. The goal of CM is to keep the evolvement of the software under control given time and quality constraints (Xu and Brinkkemper, 2007). A configuration manager's role is to ensure that these processes follow organizational policies and procedures. The focus of CM is keeping track of the versioning (Wingerd & Seiwald, 1997; Xu and Brinkkemper, 2007) of the software, and rebuilding it when necessary (Feldman, 1979; Dart, 1991; Xu and Brinkkemper, 2007) For example, issues of concern in CM include history of changes to the software, licensing issues of the product, and the level of support provided by the vendor.

Software configuration management activities are generally classified into: the *identification* of configuration option; *control* of the code; *status accounting*; *audit and review*. Through *identification*, design and data elements, and specifications are identified, described, and named. *Control* involves requesting, evaluating, approving or disapproving, and implementing modifications to the code. *Status accounting* is concerned with recording and reporting the status of project configuration items. *Audit and review* activities determine to what degree the actual configuration reflects the required functional and physical characteristics (Bamford & Deibler, 1995; Dart 1991).

The type of configuration addressed in this dissertation refers to the set of social, organizational, and technical activities that are taken by various relevant groups to bring the packaged software to a workable form applicable to the organization (or individuals/groups). Some researchers refer to the same activities as integration or implementation. We adopt the term configuration because this term conveys, more than others, that the final workable software can be different in various organizations. The discussion that follows aims at clarifying the definition of configuration in the context of this dissertation. We choose to discuss definition and challenges related to configuration of packaged software together because they intertwined topics. In other words, the definition of the configuration reflects the challenges associated with this process. By the same token, the challenges of configuration define the process by which packaged software is configured.

2.2.3.1 Definition

Configuration, the way we define it, has been mentioned but not extensively studied in either the IS or engineering fields. Therefore, there is no commonly agreed upon definition of what constitutes the configuration of packaged software. Some researchers use customization and configuration interchangeably (Soffer et al., 2001; Dreiling et al., 2006). These two terms have also been interchangeably used with the term tailoring (Vigder & Dean, 2000). On the other hand, some consider customization to involve modification of the source code but configuration to comprise of setting up the software to become workable (Vigder & Dean, 2000).

The only work that we were able to identify that focused on the issue of the configuration of the packaged software as a critical topic the way we define it in this study was Sommerville (2008). Sommerville's study is about "*configurable generic systems*," which are the basis of business-specific applications. His classification includes: a) Single PC-based applications with programming capabilities (e.g. MS Excel); b) Modular generic systems with a range of modules (e.g. manufacturing); and c) COTS assemblies, which are built by integrating various off-the-shelf software (e.g. integrating an invoice application with a web-based e-commerce system).

Sommerville (2008) argues that all software applications unavoidably require some degree of configuration. "This may simply involve providing some data about the operating environment or its users" (P. 5). Typical configuration activities include setting up the databases, scripting interfaces and setting up the parameters and their values

(Lucas et al., 1988; Vigder & Dean, 2000; Soffer et al., 2001). Sometimes configuration requires enhancing the functionalities of the software or even modifying organizational business processes (Soffer et al., 2001). Configuration can be as easy as switching on/off the functionalities or as complex as rethinking business processes (Dreiling et al., 2006). Therefore, the configuration process can be complicated and time-consuming; it is rarely a simple ‘plug-and-play’ solution (Feblowitz & Greenspan, 1998). This is because in practice “plain vanilla” implementation is not always feasible (e.g., Soh et al., 2000; Light, 2005a, b; Wagner et al., 2006; Yeow & Sia, 2008).

Plain vanilla implementation means adopting functionalities of the software As-Is and adapting the business processes as needed. This adaptation is not easy to put in practice. Moreover, the concern in configuration of the software is not merely the workability of the software in isolation. Organizations are also concerned with integration of the software with their other systems, its future maintenance, and upgrades. Therefore, the actions taken when the software is being configured are usually affected by these future concerns. The activities of the process of configuration are varied.

Sommerville (2008) lists the following activities in the process of configuration: a) selecting the required modules in the cases that the software is multi-modular (e.g. ERP systems); b) defining the data structures/models; c) defining the business processes/rules; defining the workflows; d) identifying external interactions; e) defining user interfaces; f) defining the reporting format; g) setting up the parameters; and h) re-defining the business processes if needed. Depending on the complexity, the process of configuration involves some or all of these activities.

These activities are needed in order to configure the packaged software in a way that addresses specific requirements of an organization and groups of users within the organization, characteristics of the platform to support the software, and essential interactions with other systems in the organization. As can be seen, the process of configuration involves constant adaptation of software to the business needs or modification of business processes to adjust to the software. In other words, the configuration process is highly dynamic, and involves constant social, organizational, and technical adaptations. Sommerville calls this constant adjustment co-design of software and business processes (Davenport, 1998; Davidson & Chiassen, 2005; Sommerville, 2008).

Sommerville (2008) emphasizes that business process configuration is critical since generic processes are embedded in packaged software. “It is essential that business processes are adapted to conform to this model. Otherwise, it is extremely difficult to make effective use of these systems” (p. 5). He acknowledges though that “existing business processes have evolved for good reasons to ensure that work is done effectively and efficiently” (p. 6). Therefore, at times, generic software must be configured to reflect the needs or specific organizations, groups, or even individuals. For example, in one of his studies on patient information systems, Somerville describes how doctors requested a specific configuration to make the software compatible to their non-standard ways of keeping the patients' records.

2.2.3.2. Challenges

Sommerville argues that if users are forced to conform to the standard processes of the software, sometimes they find workarounds outside the system. He gives an example of users of a system that generated PDF reports through an MS Word add-in. The users would convert the PDF report to a Word format and add further information to the report, which was needed but not provided by the system. Davidson and Chiassen (2005) also mention a similar approach that is sometimes taken to fit the software to the organizational context. One example they provide is that organizations often write interfaces that integrate the packaged software with their other applications. While these workarounds solve the limitations of the software temporarily, they are ad-hoc in nature and do not offer long term solutions. In addition, workarounds could also result in further problems (e.g. in consistencies in reports in the first example explained above).

Trauth and Cole's (1992) study, in which they introduce the concept of *organizational interface*, discusses a similar issue. The organizational interface, the authors argue, "provides IS and end-user managers with a means to consider alternative methods of providing support for systems based on purchased software with fixed user interfaces" (p. 35). By organizational interface, they mean a support other than in the form of technical hardware/software related user interface. The authors argue that, for example, if the user interface of a system is technically advanced and thus difficult for typical users, the organization can provide organizational support such as internal consultants or hot lines. Another example could be assigning a person who is familiar with the technical features of the software to generate reports for those who are not skilled enough to work with the reporting system of the software. From findings of Trauth and Cole (1992), and Sommerville (2008) it can be seen that organizations (or individuals/groups) address their needs related to software in various social, organizational, and technical ways.

To summarize, there are several ways to address the discrepancies between business processes and the functionalities of software. If the software is configurable and the policies and rules allow, it is usually configured to reflect the way the processes are handled in the organization. If the software requires the business processes to be standard, the processes are modified to fit the software. In this latter case, sometimes, the compromise that occurs as a result of the change in a process can be compensated by supplementary organizational/group/individual initiatives (similar to Trauth and Cole's (1992) concept of organizational interface). In this dissertation, we consider initiatives of this kind as a part of configuration because they are a part of the process by which the software takes a workable form.

Noticeably none of the abovementioned ways of addressing discrepancies between the software and the organizational processes is devoid of challenges. There is no systematic way to guide organizations how to configure their software to address their organizational needs (Finkelstein, 1996). The configuration process normally involves various relevant groups of people. These groups usually have different understandings, perceptions and thus expectations of the software and its configuration. A solution suggested by one group might not necessarily be recognized as the best answer by other groups. Presence of different understandings could offer some advantages as well as pose some challenges. It can be advantageous because each group's interpretation could offer an understanding of the system that other groups may not have been able to perceive based on their own interpretations. Differences in the understanding of the system could at the same time create resistance in accepting one another's points of view.

It is argued that understanding the meaning of configuration of software can be difficult because it is "defined by the underlying system". In the projects that he studied, Sommerville (2008) identified three main problems faced when configuring packaged software: a) understanding the configuration options; b) understanding the semantics of configuration; and c) understanding the ways to configure the system. He argued that "most configurable systems offer a range of different configuration options with, sometimes, subtle and difficult to understand interactions between the options. Sometimes, these options are obscure and poorly documented and there is rarely information available about how different options may interact" (p. 7). He argues that developers normally do not have access to extensive documentation and examples to be able to understand different configuration options. These options thus are usually inferred by the developers based on their limited understanding.

Sommerville (2008) adds that even when a developer identifies and understands different configuration options available, he faces the decision as to which configuration to choose.

In order to demonstrate the difficulty of the challenges faced during the process of configuration, Sommerville provides a simple example. He argues that even in the case of a taken-for-granted and thought-to-be-simple application such as MS Word, the user still might have difficulty when it comes to configuration. For example, the user has the option to configure functionalities such as preference screen, organizer screen, and macros, most of which might not be well understood by the user.

Even though Sommerville repeatedly emphasizes the importance organizational and social challenges of the configuration process, his study does not tap into these issues deeply. For example, he mainly focuses on developers (i.e. mostly programmers) in the organizations. In practice, however, various relevant groups from different functional backgrounds are involved in the process, which can add to the challenge. The purpose of Sommerville’s study is to bring the importance and criticality of the challenges of configuration to the fore. Sommerville calls for more attention from the research community to the topic of the configuration of packaged software.

In short, the research community has under investigated the issues and challenges related to configuration. A study that investigates these challenges and the ways that they are addressed is thus warranted. One way to go about conducting such a study would be an investigation of interactions of various relevant groups who are involved in the process of configuration. Examining how these interactions influence the way the software is configured and how, on the other hand, the options and limitations offered by the software influence the interactions would add to our understanding of software implementation and configuration. In this dissertation, we are interested in investigating the social, organizational, and technological dynamics of the configuration process through studying various interpretations of the software by different groups that are involved in the process. SCOT seems to offer an appropriate approach to delve into the dynamics of the configuration process focusing on the way the software is interpreted.

In the following sections we first look into the technological frames of reference (TFR) approach that has similar objectives to SCOT. Our goal by discussing TFR is to then compare it to SCOT and investigate how the latter would be more appropriate for our study. Chapter five, pilot case study of Child Org, then provides empirical content for this comparison. In the case of Child Org, TFR was initially adopted to analyze the data. The case then re-analyzed applying the theoretical lens of SCOT. The findings of Child Org case further strengthen the choice of SCOT, which is adopted in the analysis of the main case study of this dissertation, Pub Org.

2.3. Technological Frames of Reference

As briefly mentioned in the first chapter, studies in which interpretation of technology by different individuals/groups are studied have been initiated in the IS field in the past two decades. The origin of this concept however goes back to more than six decades. Gallivan (2001) attributes the origination of the notion of technological frames to studies of cultural anthropology in early 1950s. He specifically refers to Sharp's (1952) study, in which he identified the problems that were encountered when a steel axe was first introduced to an Australian aboriginal tribe. Sharp's study shows how the introduction of this instrument brought about changes in the relationships among family members, and patterns of economic exchange (Sharp, 1952) resulting from different frames of reference which affected tribe members' behaviors (Rogers, 1995).

The concept of frames of reference has long been of interest to organizational researchers. This concept was viewed as a cognitive and later as a socio-cognitive phenomenon. Frames of reference specifically related to information technologies became a wide-spread topic of interest in the information systems (IS) field in the seminal work of Orlikowski and Gash (1994). This work has a socio-cognitive basis. Meanwhile, the importance and relevance of the concept of technological frames of reference based on social construction of technology has also been acknowledged by IS researchers. This work originated in sociological studies of technologies. Bijker is one of the forerunners of this approach, which has been used to explain development of hard technologies (e.g. bicycle) and not information technologies per se.

In section (2.3.1) origination of the concept of frames of reference in organizational studies is first introduced. Then the notion of technological frames of reference (TFR) in IS research is discussed (section 2.3.2). This is followed by reviewing the studies in IS research that have adopted the concept of TFR (section 2.3.2.2). Next, the SCOT approach and its application in IS research is briefly introduced (section 2.4). This discussion will allow us to compare SCOT and previous similar approaches. The goal of this comparison would be to investigate whether SCOT has the potential to add to our understanding of the phenomenon of the configuration of packaged software. In next chapter, we delve more into details on SCOT and discuss potential contributions of SCOT as a theoretical lens for studying the configuration of packaged software.

2.3.1. Frames of Reference in Organizational Studies

In organizational studies, studying frames goes back to socio-cognitive perspectives of frame, which itself can be traced back to the concept of cognitive maps. Interest in managerial and organizational cognition started in early 1980s in response to a lack of theoretical perspective for understanding how managers added to or decreased value of an organization. These kinds of studies considered managers as "information workers," whose jobs are to retrieve, process, and disseminate information about problems and opportunities, and the issues organizations deal with. Since managers work in complex information worlds, they need to impose some sort of "knowledge structure"[2] to be able to make decisions. This view gave way to "problem-focused" view of cognition. Eden et al. (1979, 1983), Sims and Gioia (1986), and Huff (1990) were some of the early works on this research area.

Researchers have long been interested in socio-cognitive processing of information by individuals. While information processing of this kind is argued to happen in an individual's cognition, researchers believe that cognitive understanding is also shared with members of the groups that the individual belongs to. It is argued that cognitive processing is influenced by social contexts. One of the premises of social cognition is that

[2] This term, which refers to frames of reference, was coined by Walsh (1995) who conducted an extensive literature review of socio-cognitive research on this topic. It is also referred to as schemas, interpretive schemes, and scripts (Davidson and Pai, 2004).

interactions with others and negotiation of social meanings structure an individual's knowledge about a domain (Weick, 1979). Social cognitive research (Fiske & Taylor, 1984; Gioia, 1986; Kiesler & Sproull, 1982; Markus & Zajonc, 1985) suggests that frames of reference provide templates for problem solving and thus affect individuals' social information processing. Knowledge structures or frames of reference, in this process, "focus attention [of an individual] on information consistent with existing structures while masking inconsistent information, and fill information gaps with information that conforms with existing knowledge structures" (Davidson, 2002, p. 330).

'Knowledge structure' or 'schema' was a construct that entered into modern psychology through clinical neurology. Bartlett (1932), Woodworth (1938), and Oldfield and Zangwill (1942) pioneered this type of research. However, it was not till Neisser (1967) wrote *Cognitive Psychology* that researchers started showing more interest in cognitive rather than stimulus-response psychology (e.g. Skinner 1953).

> "A knowledge structure is a mental template that individuals impose on an information environment to give it form and meaning … An individual's knowledge structure orders an information environment in a way that enables subsequent interpretation and action … A key point to recognize here is that this mental template consists of *organized* knowledge about an information domain" (Walsh 1995, p. 281, 282).

The notion of knowledge structure is based on a theory-driven approach to information processing. Theory-driven approach to problem solving is a top-down as opposed to a bottom-up (or data-driven) approach. In a top-down approach,

> “past experiences in a similar circumstance guide present information processing, or they can let the current information context guide information processing … [in this approach,] the cognitive structures generated from experience affect individual’s abilities to attend to encode, and make inferences about new information” (Walsh, 1995, p. 281).

Frame content (domains of knowledge) and frame structure (integration and organization of knowledge) are commonly the topics of study in organizational studies on knowledge structures or frames of reference. Studies of content are far more in number than studies on structure (Davidson and Pai, 2004).

Knowledge structures are usually related to specific information domains (Walsh, 1995) and identifying the content is usually the first step in understanding managerial cognition. Different factors have been identified to determine the content of individuals’ schemas. Some of these factors include organization’s culture, individual’s status in the company, years of experience, person’s position in the organizational hierarchy, and overall success of the individual in the organization. Some researchers, however, have argued in favor of ‘content-free’ analyses and focusing on identifying concepts or structures (Weick and Bougon, 1986). This is because knowledge structures are context specific.

Walsh (1995) argues that content and structure are intertwined nevertheless they have been studied separately by organizational management researchers. Structure is related to the dimensions of a frame and content reflects those dimensions in each specific context. For example, if the dimensions of a frame of reference include years of experience, managerial skills, and technical skills, the content of these dimensions for a manager could be ten, strategy implementation, and database management respectively.

The two most common structural attributes that have been studied by socio-cognitive researchers are *differentiation* and *integration*. The former refers to the number of dimensions within a knowledge structure and the latter refers to the degree of interconnectedness of these dimensions. For example, between two managers who both oversee a software development project, one could have more differentiated and the other more integrated knowledge structures to which they would refer when making decisions. The knowledge structure of the manager who has programming knowledge, project management skills, and is knowledgeable about new technologies is an integrated one. The manager who understands politics and culture of the organization, understands technical language of the programmers, and knows the software development market enjoys a more differentiated knowledge structure.

Researchers have argued that the narrower the dimensions, the narrower the frame (Bartunek et al. 1983; Kiesler and Sproull 1982), which results in ineffective management behavior in the case of complex information environments (Ashby 1956; Weick 1979). In other words, "A 'narrow vision' in colloquial terms may translate to a 'less differentiated knowledge structure" (Walsh, 1995, p. 300). Therefore, the researchers' advice to the managers has usually been to have more complex but loosely coupled framework. Concepts of integration and differentiation underlie most of the organizational research on socio-cognitive theories.

Knowledge structures are held individually but they have also been looked into at the level of group, organization and the industry. Walsh (1995) warned, however, against viewing organizational cognition as merely aggregation of individual cognitive processes, which is how most researchers have approached it.

When individuals come together, some sort of collective knowledge structure emerges. The challenge in investigating knowledge structure at this level is “to account for the role of social processes in the acquisition, retention, and retrieval of information” (Walsh, 1995, p. 291). This kind of knowledge structure has been called different terms such as collective cognitive map, or collective mental model. In all the studies reviewed by Walsh, the collective knowledge structure was an aggregation of that of individuals. For example, Roberts (1976) combined experts’ judgments to build a collective mental map about commuter transportation policy. Langfield-Smith (1992), similarly attempted to assess a collective map of members of a fire department. However, she had to put a stop to her effort because of the vagueness of the language the members used to create a group map[3].

In sum, the main approach to studying frames of reference has been socio-cognitive. According to this view, frames are individually-held but can be shared by groups, which sometimes is called collective knowledge structure. A majority of the research (especially

---

[3] In the words of the author:
“In the experiment, difficulties arose in determining which beliefs were held in common by the members of the group, due to the unique nature of the elements in each map. A large proportion of the group discussion concerned the meaning of various words. Disagreements arose as to what was the most appropriate language with which to describe a particular idea” (Langfield-Smith, 1992, p. 364).

theoretical works) on frames has focused on the use of frames (e.g. Ashforth and Fried, 1988; Schneider & Angelmar, 1993). In these studies, the concern is how the already-formed knowledge structures are used in different situations to make sense and take action (e.g. managers employing their available knowledge structures when purchasing computer systems (Shrivastava and Mitroff, 1983). Within these studies the focus has, for most part, been on the content rather than structure of frames. Studies on technological frames in IS research have had a similar trend. Majority of the studies have applied technological frames as developed by Orlikowski and Gash (1994) to investigate frame content in different situation. Contributions to structure and development of frames and technological frames have been minimal. The next section elaborates on this by reviewing the IS studies on application of technological frames.

2.3.2. Technological Frames in IS Research

In this section, the discussion of technological frames of reference in the context of IS research is discussed in two parts. Section 2.3.2.1 examines the definition of TFR and section 2.3.2.2 reviews the applications of this theory in the IS literature.

2.3.2.1. Socio-Cognitive Technological Frames

Research on social cognition in organizations for most part has focused on topics such as innovation, strategy, or change. Orlikowski and Gash (1994) deemed it necessary to focus on social cognition around technology, the discussion of which also includes specific conditions, applications, and consequences of the technology in different contexts.

Orlikowski and Gash's (1994) study aims at understanding and examining the underlying assumptions, expectations, and knowledge of people towards an information technology. They call this interpretation of technology by people, Technological Frames. In line with socio-cognitive studies of frames, Orlikowski and Gash (1994) define technological frames of reference (TFR) as an individual level phenomenon, which is also characterized at the group level as shared assumptions of individuals. The authors posit that when in an organization, technological frames of key stakeholders (e.g. users, developer, managers) differ dramatically there may be difficulties and conflicts in development, use, or change of technology resulting in unexpected or undesired consequences.

Before Orlikowski and Gash (1994), a few researchers (Bostrom & Heinen, 1977a&b; Ginzberg, 1981; Goodman et al., 1990) had studied assumptions and expectations of individuals toward technologies. This stream of research mostly focused on designers' cognitive understanding in designing technical artifacts (Boland, 1978 & 1979; Bostrom & Heinen, 1977a&b; Dagwell & Weber 1983; Markus & Bjørn-Andersen, 1987). This approach was based on the belief that "because technologies are social artifacts, their material form and function will embody their sponsors' and developers' objectives, values, interests, and knowledge of that technology" (Orlikowski & Gash, 1994, p. 179). Building on these works and also considering sociological perspectives of the social construction of technology (Bijker, 1987; Bijker et al., 1987; Henderson, 1991; Sætnan, 1991), Orlikowski and Gash (O&G) aimed for a more systematic investigation of frames of reference towards information technologies (i.e. technological frames).

Orlikowski and Gash (1994) define technological frames as:

> "the subset of members' organizational frames that concern the assumptions, expectations, and knowledge they use to understand technology in organizations. This includes not only the nature and role of the technology itself, but the specific conditions, applications, and consequences of that technology in particular contexts" (p. 187).

These authors identify three dimensions that characterize the interpretation of subjects of their study toward the technology they were using. These three interdependent and interactive domains include:

1. Nature of Technology: refers to people's understanding of the capabilities and functionalities of the technology.

2. Technology Strategy: refers to people's view about the vision and motivation for adopting the technology (i.e. their understanding of the reasons why the technology was introduced to the organization).

3. Technology in Use: refers to people's understating of the way the technology would be used daily and the actual and likely consequences of this use (i.e. their understanding of how the technology would be used to create different changes in work).

As a part of their study, Orlikowski and Gash (1994) investigate technological frames of different groups (users, managers, and technologists) in an organization toward a groupware system. Each group interpreted each domain differently. For example, technologists viewed the technology to be extremely easy to learn on one's own but the users felt that they had received inadequate training.

Orlikowski and Gash had derived a set of seven categories in an earlier work in 1991 by examining the IS literature on IS development. Two of these categories (issues around use and criterion of success) appeared in their seminal (1994) study as part of the domains of technological frames. The domains identified in their (1994) study are much more context specific than those found through their (1991) literature review. This is consistent with their view that frames are time and context-bound and that "are always more valid when examined *in situ* rather than assumed ahead of time" (p. 184).

Orlikowski and Gash (1994) introduce the concept of congruence in technological frames as the alignment of frames of different groups based on key elements or categories. They explicitly clarify that congruence does not mean identical but related in content and structure, which is similar to cognitive consensus (Gioia and Sims, 1986; Isabella, 1990). According to the authors, structure refers to common categories (i.e. the domains) and content refers to the value of those common categories in specific situations. They believe that whenever there is incongruence between technological frames of key stakeholder groups, the organization might experience difficulties in implementation and use of information technologies. Orlikowski and Gash (1994) suggest, though, that more research is required to examine where/when *incongruence* would occur, tolerable degree of incongruence, and the process that would decrease/increase incongruence.

In their study of use of Lotus Notes (a groupware technology), in its initial stages of use, Orlikowski and Gash (1994) found that different groups of stakeholders (users, managers, and technologists) held different frames of reference towards this technology. In order to examine the congruence/incongruence of TFR, the researchers investigated each individual frame domain for each stakeholder group. They found that, for example, users viewed Notes as an individual productivity tool whereas technologists viewed it as a group productivity tool. The researchers interpreted the differences in the use of technology by these two groups to be influenced by the incongruence between frames of reference of the groups. In the same way, Orlikowski and Gash examined the other two domains and conducted comparisons between every two groups. They observed both congruence and incongruence of frame domains for different pairs of groups.

As the Orlikowski and Gash acknowledge, their framework for TFR is preliminary and requires more investigation. For example, the domains identified by these researchers were found to be relevant in the context of use of a groupware technology. The authors emphasize that the domains of TFR need to be studied in situ. Moreover, Orlikowski and Gash's framework does not explain how frames are formed and shared. Therefore, even though Orlikowski and Gash's framework is insightful, it does not explain all the relevant issues around technological frames of reference. This framework has been a stepping stone for other studies on technological frames.

The following sub-section is a review of IS literature which has adopted socio-cognitive technological frames of reference. This discussion is aimed at further clarifying the concept as introduced by Orlikowski and Gash, and to investigate the extent of importance, relevance, and the interest in this concept in the field of IS. This section is then followed by a concise discussion of SCOT approach and the ways in which this approach could contribute to our understanding of configuration of packaged software.

2.3.2.2. Applications of Socio-Cognitive Technological Frames

Almost all the studies on TFR in IS research, even those that have borrowed ideas from SCOT, have adopted a socio-cognitive approach. These studies are mainly based on Orlikowski and Gash's definition of TFR. Davidson is one of the researchers who has expanded Orlikowski and Gash's idea of technological frames significantly in the IS literature (Davidson, 1996, 1997, 2002, 2006; Davidson & Pai, 2004). Works of Davidson have specific importance because these works have extensively contributed to this theoretical concept of TFR. However, other researchers have for most part mainly applied TFR as a theoretical framework in the context of their studies. The literature on TFR in IS research is reviewed extensively in this section.

Davidson defines technological frames as "stocks of knowledge which organization members draw on to communicate meaning in their actions and interactions around IT development, implementation, and use … technological frames are both articulated (discursive) and tacit (practical)" (Davidson, 1996, p. 30). In other words, they are taken for granted and implicit knowledge, assumptions, and expectations, which are usually expressed symbolically through metaphors, language, stories, and visual images (Davidson, 1997).

Davidson's definition of technological frames is similar to socio-cognitive definition of schema. According to this definition schemas are considered as scripts for taking action, sensemaking devices, and filters of information (Abelson 1981; Gioia 1986; Poole, Gioia & Gray 1989; Isabella 1990; Schneider 1991). Davidson, similar to Orlikowski and Gash, defines technological frames at an individual level, shared by members of a group. In her (1997) study, she identifies two main groups of stakeholders: core team members (with major information systems development (ISD) responsibilities), and system constituents (potential users). She adopts a socio-cognitive perspective with a focus on an analytic concept of technological frames of reference to examine requirements definition processes in ISD.

Davidson (1997) found that participants used various sensemaking devices such as scenarios-of-use, organizational and/or personal stories, and project history narratives, to communicate and negotiate their frames with their other group members or other groups. This helped them share their understandings of the technology in order to collaborate. Other researchers have identified similar sensemaking devices such as "use of metaphors (Boland and Greenberg 1988, 1992; Mason 1991), stories (Boje 1991; Brown and Duguid 1991; Hirschheim and Newman 1991), narrative constructions (Tenkasi and Boland 1993); and scenarios-of-use (Walz et al., 1993)" (Davidson, 1996, p. 306).

Communicating understandings and interpretations is important in a process of IT requirements definition, which in Davidson's (1997) words, "can be understood as an ongoing social cognitive process, in which organization members negotiate, consider, reconsider, and perhaps change their interpretations of and decisions about the [requirements]" (p. 483).

Communication of individuals' technological frames with one another has been considered important by both Orlikowksi and Gash (1994) and Davidson (1997). However, there is no consensus between the two in considering the alignment of frames as a premise for success in implementation of a system.

According to Orlikowski and Gash (1994) misalignment (or as they called it, incongruence) of frames of different stakeholders would translate into conflict. However, even though Davidson (1996, 1997) found support for this argument in some cases, she also found that incongruence, in some situations, could be inconsequential or even have potential benefits. She added that congruence in frames has the potential to be dysfunctional because of preventing the problems and contradictions to surface. In her study, Davidson found that Core team members had similar frames of reference and they tended to reinforce each other's frames by drawing on their shared understandings and taking actions accordingly. This reinforcement resulted in some challenges and implicit assumptions not surfacing. Orlikowski and Gash suggest (but do not elaborate on) the idea of intervention in order to change frames and possibly reach alignment. This is based on the idea that frames are flexible and could change in different contexts.

Davidson's (2002) study showed little change in participants' technological frames during the time period of her research. This was despite the numerous changes that happened in the organizational context which triggered episodes of negotiation over IT requirements. This finding is in line with the claims of socio-cognitive literature according to which schemas do not change readily (Fiske & Taylor, 1984; Lord & Foti, 1986; Markus & Zajonc, 1985). According to this literature though schema can change if new information acquired by an individual is inconsistent with his existing frame of reference (Fiske & Taylor, 1984; Lord & Foti, 1986).

Contextual changes can make new knowledge salient to the sensemaking and result in shifts in an individual's frame (Bartunek, 1984; El Sawy & Pauchant, 1988; Gioia, 1986). They can influence sensemaking of environment by an individual and the decisions he makes at that time (El Sawy & Pauchant, 1988). However, the shifts can be sudden and temporary. The process model of framing provided by Davidson (2002) suggests that an individual's frame must be sensitive to changes that take place yet they must be stable enough to filter unrelated information so that the individual can make decision.

In her 1997 study, Davidson introduced a new concept which she calls *collective fund*. She describes collective fund as the collective account of different technological frames, on which individuals and groups draw to make decisions. In an ISD project different individuals and groups with a variety of backgrounds, skills, and knowledge participate. Frames of these individuals and groups form "the collective fund of technological frames”. In this process, frames of some individuals or groups dominate the collective fund.

Davidson (1997) found that during some episodes of ISD requirements planning, an influential individual could manipulate interpretation of other participants by bringing new ideas into the project. A dominant interpretation of IT application was the definer of each episode. New episodes emerged whenever critical assumptions or expectations of participants changed. This latter change was itself usually as a result of critical changes in the organizational context (Davidson, 1997). This is in line with social cognitive research findings that executives’ frames of reference are usually influential when it comes to interpreting organizational change (Daft & Weick, 1984; Gioia, 1986; Isabella, 1990; Kiesler & Sproull, 1982).

As can be seen, Davidson’s studies have contributed significantly to Orlikowski and Gash’s (1994) original work on technological frames. To summarize, Davidson and her colleagues added the discussion of frame shift and collective fund to TFR framework. These researchers also found that incongruence of frames does not necessarily lead to conflict and can actually be beneficial. They added that congruence of frames might actually be disadvantageous in some contexts. Davidson and colleagues conducted longitudinal studies, which were valuable in studying development and change in frames. Other IS researchers have also adopted and contributed Orlikowski and Gash’s (1994) technological frames.

Davidson and Pai's (2004) conducted an extensive literature review on technological frames of reference. Even though Davidson and Pai (2004) found hundreds of references to TFR (e.g. through search in Social Science Citation Index, Google) only eight were peer-reviewed articles that had focused on contributing to the concept of technological frames. These eight studies had all adopted a qualitative case study research method, using interviews as the major data collection method.

Davidson and Pai (2004) categorized the contributions of these studies under three main categories: theoretical development; methodological refinement; and contributions to practice. They used these categories to investigate the contribution of the eight papers. Under theoretical developments, they examined contributions to five areas: defining additional frame domains and content; understanding of incongruence, frame convergence, and divergence; frame change over time and change triggers; frame tolerance and rigidity; and complementing political analysis (Davidson & Pai, 2004). As with Orlikowski and Gash's (1994) study, the main focus of the aforementioned eight papers is on identifying and defining the domains of technological frames.

Generally, the domains are similar to those in Orlikowski and Gash's study, with slightly different terminology. In some cases, depending on the context of the study, new domains were added to or eliminated from the original set of domains in Orlikowski and Gash's framework. As Davidson and Pai (2004) state "this variation in naming, number, and precise definition of frame domains is consistent with the idiographic nature of TFR analysis" (p. 480). For example, the domains that Davidson (2002) identifies in the context of her study are similar to those of Orlikowski and Gash's (1994) but reflect the contextual aspects of her specific study.

Davidson (2002) identifies four domains of technological frames. The domains identified by Davidson include: IT delivery strategy, IT design and capabilities, Business value of IT, and IT-enabled work practices. *IT delivery strategy* is "generalized knowledge and expectations about how IT functionality should be developed or acquired and provided" (Davidson, 2002, p. 337). This was not one of Orlikowski and Gash's domains because they looked at use of an already implemented system. *IT capabilities and design*, which is similar to Nature of technology from Orlikowski and Gash's domains, is "generalized knowledge of and expectations about relational databases, graphical user interfaces, report generation software, client server architectures" (p. 337). *Business value of IT*, which is similar to Technology strategy in Orlikowski and Gash's study, represents "generalized assumptions and expectations about how IT can be used to influence the organization's relationship with and control over its external environment, including customers or competitors" (p. 337). *IT-enabled work practice*, similar to Technology in use from Orlikowski and Gash's domains, refers to "assumptions and expectations about how an IT application will fit into day-to-day work practices" (p. 337).

In order to summarize the findings of the TFR literature, Table 1 from Davidson & Pai (2004) has been adopted. In this table, Davidson and Pai summarize their comparison of the papers they reviewed based on the study context, frame domains, and user groups. In the article, however, the authors look into other aspects such as congruence/incongruence, frame change, and analysis of power/politics as complementary to socio-cognitive processes. Orlikowski and Gash (1994), and Davidson's studies while suggesting that power and political interventions are sometimes used to manipulate frames do not delve into this issue deeply. We added the elements of power/politics as well as incongruence/congruence and frame change to Davidson and Pai's (2004) original table for the purpose of our literature review. The results of our review are summarized in Table A.1 in appendix A. This table includes all the articles from Davidson and Pai (2004), Davidson (2006) as well as other TFR papers, which were published since then or had not been included in the reviews by Davidson and Pai.

Similar to the findings of Davidson and Pai (2004), the TFR articles that we reviewed are mostly standalone studies, which provide rich insights about individual cases. These studies focus mainly on applying the domains of TFR in the context of their study. As Davidson and Pai (2004) point out, there is a need for cumulative theorizing. Therefore, the TFR framework is still in a state that might fall short in explaining the phenomenon of interpretation of technology. We acknowledge that the socio-cognitive approach of TFR as initiated by Orlikowski and Gash (1994) and expanded by Davidson and others is insightful in studying interpretations of information systems. However, we believe that a similar but more holistic approach such as SCOT could enrich our understanding of this topic even further. Therefore, in addition to contributing to the literature on packaged software, this dissertation aims at contributing to understanding and application of the SCOT approach in the context of IS research.

In the section 2.4, we discuss the theory of SCOT briefly, compare it to TFR, and discuss its applications in IS research. Finally, we conclude the chapter by explaining why SCOT would be an appropriate lens for our study. A more comprehensive discussion of SCOT is provided in chapter three.

2.4. Introduction of Social Construction of Technology Theory

Social construction of technology is an approach to studying development of technologies. This approach was initiated by Bijker, Pinch, and Hughes in 1987 and has been expanded ever since. In a nutshell, relevant social groups, interpretive flexibility, inclusion, technological frames, stabilization, and closure constitute elements of SCOT. A first step in such an approach is identifying different groups (i.e. *relevant social groups*) that are concerned with the technology in one way or another. This is done based on the meanings the relevant groups attach to the technology. A technological artifact is believed to have *interpretive flexibility*, meaning that it is interpreted differently by each group.

Different researchers might take different approaches in order to identify social groups relevant to a technology. Typically, different opinions form about a technology when there is a controversy related to the technology. Therefore, locating the problems and investigating different ways that the controversy is being approached could be a starting point to identify relevant social groups. As a researcher discovers different groups, he also examines the ways these groups interact with the technology and approach their problems related to the technology. The *technological frame* of each group guide interactions of members of the group with the technological artifact as well as within and across the groups. As the problems related to the technology are addressed, the artifact finds a more and more *stabilized* form. As the controversies around the technology are addressed, interpretive flexibility of the artifact reduces. This is when the technology is said to have reached *closure*. Closure does not imply that the artifact cannot change form any further. What it means is that at the time of closure the controversies around the problem are addressed and the interpretive flexibility has diminished.

Chapter three elaborates on each of the elements of SCOT in more detail. In the following sub-sections, first, a comparison of SCOT with socio-cognitive approach to technological frames has been provided. This discussion is followed by examining IS literature, in which SCOT ideas have been adopted. The chapter concludes with a discussion of the reasons for choosing SCOT as a theoretical lens for this study.

2.4.1. Why SCOT?

Social interaction is the essence of the phenomenon under investigation in this dissertation (i.e. configuration of packaged software). Thus, choosing an approach that is based on social interaction to study this phenomenon is deemed to be most appropriate. Social construction of technology has been previously used to study development of technologies (e.g. bicycle). The appropriateness and applicability of this approach has also been acknowledged in IS research (e.g. Orlikowski and Gash, 1994; Lin & Cornford, 2000; Davidson, 2004). According to this approach, interpretation of technology, and interaction with technology and with others is investigated at the level of group.

Therefore, the investigation enriches one's insight about how interpretations of different groups influence the process of configuration of packaged software.

As discussed in section 2.3.2, socio-cognitive approaches have been adopted to study development of information systems based on interpretations and expectations about the technology (e.g. Orlikowski & Gash, 1994; Davidson, 1997, 2002, 2004, 2006). These studies are an extension of cognitive studies, which investigate how individuals make sense of technologies based on their frames of reference (e.g. Bostrom & Heinen, 1977; Dagwell and Weber, 1983). Traditional cognitive approaches have been criticized for downplaying the contextual influence on human behavior and focusing on internal cognition (Norman, 1993; Lin, 2000). The assumption of socio-cognitive view is that frame resides within individuals. According to this view technological frame belongs to individuals who share their frame with other individuals in the same relevant social groups. It is believed that a technological frame provides an individual with a set of values, expectations, and assumptions about a technological artifact. Meanings of the artifact are then shaped in individual's cognition.

According to the socio-cognitive view, a frame already exists in an individual's mind. Therefore, it is sometimes called a mental map (or model). An individual then refers to this "already existing" mental map when facing a new situation to make sense of an unknown situation (Walsh, 1995). Technological frames, based on this view, are made up of various elements, which are brought forward at appropriate times. Accordingly, *schemas* are used to make sense of different situations individuals face. Schemas store knowledge, philosophy, and experiences of an individual about various categories (Walsh, 1995). *Scripts* then guide an individual's behavior based on what schema suggests (Walsh, 1995; Lin, 2000). The socio-cognitive approaches to technological frames claim that frames can be formalized and used to predict individuals' actions.

Cognitive process has been viewed as fundamentally a linear process. Based on this view, frames are considered self-contained, structural and procedural. However, socio-cognitive approach has not yet been successful in offering a formalization of the frames (Lin, 2000). In order to avoid criticism of traditional cognitivism, Orlikowski and Gash (1994) assert that their definition of technological frames is not merely cognitive and it has elements of SCOT approach. Orlikowski and Gash's framework is deemed useful in understanding motivations of individuals in acting towards technologies (e.g. Lin, 2000; Azad & Faraj 2007, 2008). However, essentially their technological frame is based on the idea that frames can be formalized and actions of the actors can be predicted, which is not in line with SCOT philosophy. In addition, social construction approach denies the claim that a frame preexists in individuals' minds. According to this approach, frames are formed when the interaction around the technology begin.

Social interaction is not clearly explained through a socio-cognitive approach such as that of Orlikowski and Gash's (1994). Orlikowski and Gash add ideas from social theories such as SCOT (Bijker, 1987) and structuration (Giddens, 1984) to their framework to consider a social context for technological frames. For example, Orlikowski and Gash borrowed the concepts of *Interpretive Flexibility* (Orlikowski, 1992), *Relevant Social*

Groups, and *Shared Frames* from SCOT theories (Davidson and Pai, 2004). Essentially though, Orlikowski and Gash's technological frame is individual-level socio-cognitive structure which might be shared with other individuals. Bijker (1995), on the other hand, defines technological frame as a social concept that "structures the interactions among the actors of a relevant social group. Thus it is not an individual's characteristic, nor a characteristic of systems or institutions; technological frames are located between actors, not in actors or above actors" (Bijker, 1995, p. 123). Therefore, socio-cognitive-based technological frames do not delve into explanation of interactions of individuals with one another as well as with the technology.

In order to address this issue, Lin (2000) adopts the idea of symbolic Interactionism from Blumer (1969). According to symbolic Interactionism, people act based on the meanings of things for them, which arise from social interactions with others (Blumer, 1969). Lin (2000) does not agree with Orlikowski and Gash's assertion that through technological frames one can explain and anticipate outcomes. She states that this view about technological frames "oversimplifies the process of formation of meanings to their being relevant predicates of human conduct" (p. 63). Moreover, this view ignores the significance of social interactions in the process of formation of meanings. In addition, it overstresses the importance of cognitive structures in an individual's behavior. Lin (2000) emphasizes that meaning is not merely a psychological but also a social product. According to symbolic interactionsim social interaction is a social process through which an individual interprets others' actions and determines his own accordingly based on the adjustments he makes in his understandings and intentions.

Socio-cognitive researchers acknowledge the role of social interactions in understanding an individual's behavior. However, to them social interaction is merely a process in which one expresses his psychological elements such as feelings, attitudes, and perceptions (Lin, 2000). According to symbolic interactionist approach, on the other hand, meanings are not merely conveyed through social interactions but they emerge from these interactions and, at the same time, influence these interactions.

Therefore, Lin (2000) builds a framework, according to which an individual's action towards a technological artifact is based on both self- and social- interaction. Accordingly, when interacting with a technology, an individual makes sense of others' behaviors toward the technology and adjust his own understanding of and consequently action towards the technology. The view of self- and social- interaction considers a dynamic nature for technological frames which is in contrast with socio-cognitive view of technological frames as Orlikowski and Gash explained it. Accordingly, Lin (2000) states that

> "Orlikowski and Gash see technological frames as a set of existing cognitive properties and individuals develop their understandings and interpretations of technologies on the basis of such a set. Because technological frames are treated as given in their analysis Orlikowski and Gash's discussion focuses mainly on the properties and dimensions of frames, and the process of formation, reformation, and change is overlooked" (p. 65).

To look back and summarize, the main criticism of socio-cognitive view of technological frames is that social interaction which is important in the process of meaning formation is missing. This gives technological frames a static nature, which attributes properties and structures to the frames. Since self- and social- interactions change and are unpredictable, technological frames cannot be used as a tool to anticipate actions and meanings. As such, technological frames are constructed in the situation and through interactions. In this approach, actors' backgrounds, experiences, and existing knowledge are not ignored. However, the frame within which actors make sense of the technology and solve the problems related to the technology is not viewed as pre-existing mental maps. The frame is built when the interactions around the technology begin.

Moreover, in socio-cognitive studies of technological frames the level of analysis is individual. Individuals are interviewed, surveyed, or observed and conclusions are made about groups by aggregating the responses from the individuals. Since one of the premises of this approach is that technological frames (are mostly implicit but) can be articulated, individuals are asked explicit questions about structure and domains of their frames of reference. However, in a SCOT approach to investigating technological frames different relevant social groups are identified and observed. Technological frames of different social groups emerge as the researcher views interactions within the groups, among the groups, and with the technological artifact.

In addition, current socio-cognitive approaches have not focused on the concept of technological artifact itself when it comes to studying technological frames of reference in the process of development of an artifact (i.e. they are strictly socio-cognitive). SCOT, on the other hand follows the changes in the artifact itself as it investigates how problems and controversies around the technology are addressed.

In sum, social construction of technology is in line with the interest and the scope of this study. It is believed that this approach would be enriching in gaining an understanding of a technological artifact as well as the dynamics of the groups that are involved in the process of its development. Therefore, SCOT is deemed to be an appropriate theoretical lens for this study. At the end of this chapter, we will discuss further how SCOT would be more specifically appropriate in investigating the configuration process of packaged software.

2.4.2. Application of SCOT in IS Research

Since the inception, SCOT has been applied in the studies on history and development of technologies. Its main application has been in what researchers call hard technologies such as mountain bike (Rosen, 1993), automobile (Kline & Pinch, 1996), and missile (MacKenzie, 1989). The relevance and importance of SCOT in IS research has been acknowledged. However, it has not been extensively applied in this field. In most cases the theoretical framework has been partially adopted. For example, the main elements from SCOT that have been applied in the studies include the notions of relevant social groups, interpretive flexibility and closure. Almost all SCOT related IS studies have mentioned these elements, specially the first two. However, application of these notions seems to be merely at a definition level rather than the theoretical view being applied as a whole. In studying hard technologies, researchers seem to have remained more faithful to the SCOT approach.

Kline and Pinch (1996) take a SCOT approach in studying early adaptation of cars in rural areas of the United States. The main goal of the authors is to shift the focus of the studies on history of technology from the “producers” of technology to the “users” of technology. In order to achieve this goal, they consider farmers as active participants in the development of automobiles. As the authors acknowledge themselves, they adopt some of the ideas of SCOT such as social groups, interpretive flexibility of the artifact, and closure. They attempt to extend SCOT approach by investigating a case of a well-stabilized artifact, for which its users adopted new ways of usage. The authors investigated different interpretive flexibilities focusing on gender role relationship. They found that as long as people used cars for different purposes than what it was originally meant for, this technology had a high degree of interpretive flexibility. The technology started to stabilize once car manufacturers took those uses into account and manufactured artifacts that would address them. In the words of the authors,

> “the interpretative flexibility [they had] described for the car disappeared by the early 1950s. Closure had occurred (once again) and farm people had stopped using their autos for grinding their grain, plowing their fields, or carrying their produce to town. Instead, they had begun to buy tractors and pickup trucks in large numbers new artifacts that manufacturers developed partly in response to these novel interpretations of the car. The users, so easily overlooked in writing the story of technology, had made their mark” (Kline & Pinch, 1996, p. 794).

SCOT-based studies such as Kline and Pinch (1996) empirically support the theoretical concept of SCOT. No other study than the ones conducted by the original researchers (Bijker, Pinch, and Hughes) have expanded this approach theoretically. The application of SCOT in IS literature has also been an empirical one.

In the case of IS research, the number of studies that have adopted SCOT theoretical view is few. Some of researchers who have applied technological frames to their studies, have adopted a combination of both socio-cognitive (more specifically Orlikowki and Gash's) and SCOT approaches. For example, Azad and Faraj's (2007, 2008) article looks into a ten-year e-Government implementation project. The authors describe and analyze *frame evolution* processes, which include *frame differentiation, frame adaptation*, and *frame stabilization*. Azad and Faraj's (2008) focus is on investigating what they call *truce frame*. The authors propose mechanisms of frame alignment. These mechanisms include two of the actor-network theory elements: black-boxing and obligatory passage points. The authors argue that through these mechanisms competing frames are translated into a *truce* frame (i.e. a converged frame) (Azad & Faraj, 2007).

Azad and Faraj acknowledge the usefulness of both SCOT and socio-cognitive approaches and "juxtapose Orlikowski and Gash's approach with Bijker and Pinch's (2002) conceptual framework vis-à-vis aligning technology frames" (Azad & Faraj, 2007, p. 2). Their findings show that achieving a relative stability of frame involves a process of negotiation towards reaching a balance of goals of different groups involved. These negotiations involve adjustments to the design of the system. Stabilization of frame refers to the process by which *truce* frame emerges (Azad & Faraj, 2008). Azad and Faraj contribute to technological frames theoretically by investigating frame alignment. This adds to Orlikowski and Gash's findings since they introduced the notion of frame alignment and called for more attention to this concept. Azad and Faraj's contribution is also in line with SCOT approach since it could fall under the discussion of *closure mechanisms*[4] through which interpretive flexibility of a technological artifact is reduced and the technology reaches closure.

Kilker and Gay's (1998) article is another article in IS research among the more faithful ones to SCOT approach. In their case study, the authors utilize SCOT for studying both development and evaluation of a specific type of digital library (DL). The authors state their goal as highlighting different perceptions of the technology's performance in order to anticipate challenges of design and use of the technology. Kilker and Gay (1998) touch on the concepts of relevant social groups, interpretive flexibility, and closure. However, they do not include a discussion of inclusion and technological frames.

The authors identify relevant social groups in terms of goals, experience, and technical, as well as the ability to influence the final project. They acknowledge the challenging nature of identifying the boundaries of relevant social groups. However, they find these boundaries to be clear in the case of the project under their study. These groups include funders, evaluators, librarians, faculty, students, and software developers. However, this categorization appears to be similar to a traditional way of classification of stakeholders

[4] This concept is discussed in length in chapter three.

of a project based on their functional area. In SCOT, classification of relevant individuals is based on their interpretations and the meanings they attach to the technological artifact.

Kiler and Gay (1998) have a more elaborate discussion on interpretive flexibility than in most other works. Their contribution is to argue that the concept of closure and relevant social groups need modification for the case of DL. They argue that because DL comprises of multiple systems and has a malleable nature, its closure is more complex than a hard technology like bicycle. An interesting and noteworthy conclusion the authors make is the notion of softness or hardness of a technology: softer technologies being more flexible to change than the harder technologies.

One of the contributions of this paper is the modifications made to the concepts of social groups and closure. They argue that closure is not needed in the case of DL because having multiple interfaces is preferable to one dominant one. Their modification to the concept of relevant social group is more complex. They propose adding dimensions of *relevancy* and *mediation* to this concept. The concept of relevancy which examines the reasons and consequences of some groups being more influential than others emphasizes the challenges of balancing the needs of various groups. The concept of mediation is concerned with mediating effects of one group influencing another.

On a closer look, we can argue that both of these contributions have been at least implicitly acknowledged in Bijker and Pinch’s original SCOT. For example, Pinch and Bijker also argue that closure is a provisional concept. Moreover, the example given by

Kilker and Gay emphasizing that different interfaces are needed for DL does not imply that the artifact has not and never will reach closure. In the case of a DL with multiple interfaces, the artifact is really one with different features, each of which is of use to a specific group. The concept of *relevancy* and *mediation* also cannot be counted as a shortcoming of SCOT. Bijker and Pinch repeatedly emphasize that there is no one way of identifying relevant social groups. In the examples they provide also some relevant social groups are more influential than others. SCOT also does not reject the idea of a relevant social group playing the role of a mediator. In short, the findings suggested by Kilker and Gay could be explained by SCOT as well depending on the case of the technology under the study and its context.

Other IS studies have not fully applied ideas of SCOT by Bijker and his colleagues but have cited their work when referring to social studies of technology (Rosenkopf & Tushman, 1998), contextual studies of technology (Lin & Cornford, 2000; Lin & Silva, 2005), mutual shaping of technology (Boczkowski, 1999; Rosenkopf & Tushman, 1998), relevant social groups (Lin & Cornford, 2000), interpretive flexibility (Boczkowski, 1999; Lin and Cornford, 2000; Lin & Silva, 2005), stabilization (Allen, 2000), closure (Allen, 2000; Orlikowski & Iacono, 2000; Rosenkopf & Tushman, 1998), technological frames as defined in SCOT (Allen, 2000), symbolic meaning of information technology (Lin, 2000), and social construction of technology (Rosenkopf & Tushman, 1998).

In sum, ideas of SCOT have been partially applied in IS research. Different studies have shown that application of SCOT is relevant and valuable. Review of this literature also shows that adoption of SCOT in IS research is far from complete. In the next chapter we, first, elaborate on the concepts of SCOT. We then provide a discussion of potential contributions of SCOT to configuration of packaged software.

Chapter 3

Theoretical Lens of SCOT

3.1. Introduction

Social construction of technology (SCOT) is an approach to investigating design and development of technological artifacts. This approach was developed in early 1980s due to dissatisfaction, within the community of sociologists of technology, with the available linear models of technological development. Bijker, a sociologist of technology, and Pinch, a sociologist of scientific knowledge, are the forerunners of this approach. While studying the development of several technological artifacts such as bicycle, Bakelite, and fluorescent lamp, Bijker (with collaboration of others such as Pinch and Hughes) developed SCOT descriptive approach.

Using SCOT as a lens, a researcher studies the development of a technology by following the changes in its features, identifying the groups that relate to the technology, and investigating the ways these different groups approach the technology and issues around its development and use. The researcher continues this quest until the technology reaches a rather stable form, which is interpreted (or understood) relatively similarly by various groups. One of the main elements of SCOT is technological frames, which is utilized to explain the ways controversies around a technological artifact are approached and addressed. A technological frame guides development of the artifact and at the same time is transformed as a result of this development.

In the following sections, we first provide a background on SCOT theory. We then discuss this theory comprehensively. We conclude the chapter by taking a critical stance towards SCOT and elaborating on its limitations.

3.2. Background of SCOT Theory

Application of descriptive approach of SCOT to understanding technological developments was a result of the response to inadequacy of linear models of development of technical artifacts. According to linear models, there is an implicit assumption that the development of a technological artifact follows an orderly or rational path (Ferguson, 1974). Figure 3.1 demonstrated a six-stage linear model of the innovation process, presented in Bijker (1995). Linear models are most popular when there is a tendency to focus on successful technological developments. According to this view, new forms of a technology are logical descendent of the previous forms (van Nierop, 1997).

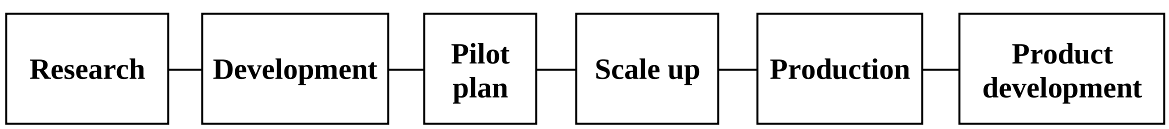

Figure 3.1: A six-stage model of the innovation process (source: Bijker, 1995)

In SCOT on the other hand there is equal emphasis on successful and unsuccessful technological developments since, as it will be discussed, its focus is the meaning of technology for different groups of people. Unlike the linear models, SCOT does not distinguish distinct stages for the development process. In this approach relevant social groups are the key starting point in the analysis (Bijker, 1995). Technical artifacts cannot exist without social interaction among members of a group or among different groups that are in any way concerned with the technology (i.e. relevant social groups, as called in SCOT) (Ibid.). In SCOT these groups are classified based on the way they interpret the technology and the meanings that they assign to the technology (Clayton, 2002). This is because various groups interpret the technology differently. This flexibility in interpretation of the artifact is explained by demonstrating that the artifact presents itself in various forms to different relevant social groups (Bijker, 1995).

Each relevant group to the technology has some specific interests in the technological artifact. At any time, each of these groups might face problems regarding the technology. This is the case specifically at the time that the artifact is still young (Bijker, 1995). Based on how various relevant groups interpret the artifact, they approach their problems regarding the technology differently. Therefore, the meanings that they attach to the technology are embodied in their approach and solutions to their problems.

The series of problems and solutions regarding the technology that different relevant groups face, define the design details of the artifact. This approach is different than a view based on which the improvements in a technological artifact are attributed to specific individuals. Bodewitz et al. (1987) argue that even though specific ideas may come from individuals, in order for a technical solution to a problem to become part of the artifact, it has to become socially constructed and embedded in existing social networks. Therefore, unlike linear models of development, in the SCOT approach there is no orderly pattern for the stages of development of an artifact.

The seeds of SCOT research program were planted when Pinch and Bijker met for the first time at the meeting of European Association for the Study of Science and

Technology (EASST) in September 1982. The purpose was to consolidate Pinch's in-depth studies of development of science with studies of technology by Bijker. This objective was fulfilled to some extent in their 1984 article *The Social Construction of Facts and Artefacts: Or How the Sociology of Science and the Sociology of Technology Might Benefit Each Other* (Pinch & Bijker, 1984). In 1984, Bijker, Pinch, and Hughes organized a workshop at Twente University. Around thirty participants from six different nationalities and with diverse backgrounds in history, sociology, and philosophy attended in the workshop. The discussions of this workshop led to further formalization of SCOT. The articles presented in the workshop were published in a book, *The Social Construction of Technological Systems* in 1987 edited by Bijker, Hughes, and Pinch (Bijker et al., 1987; Clayton, 2002).

Before Pinch and Bijker, other researchers had attempted to develop new approaches different from old-fashioned linear approaches to studying technology. For example, Johnston (1972) and Dosi (1982) promoted an approach to the description of technological knowledge in terms of Kuhnian paradigms (Kuhn, 1962). The approaches offered by these authors, however, do not consider an equivalent need for looking at both successful and failed technological artifacts, something that is highly advocated in SCOT approach (Kline & Pinch, 1996).

In addition, other approaches more similar to SCOT, which also took a social constructivist stance towards studying technological developments of technology have also been taken. For example, Fischer (1992) and Martin's (1991) studies demonstrated that telephone callers use of the technology convinced the industry that telephone could be used as a social as well as a business tool. Douglas (1987) showed that radio amateurs influenced the change in dominant interpretation of radio from a point-to-point communication medium to a broadcasting one.

It is argued that although these studies have been conducted under the rubric of social construction, SCOT is more advantageous in explaining the role of actors in technological development. SCOT studies investigate how different social groups' actions and interactions change the interpretation and development of a technology until it is relatively a stable artifact (Kline & Pinch, 1996).

The focus on various interpretations that different relevant groups to a technology assign to it is what differentiates SCOT from other social constructivist approaches. SCOT "goes beyond saying that technology is merely embedded in human affairs. SCOT focuses attention upon what counts as a viable working artifact, and what indeed counts as a satisfactory test of that artifact" (p. 766). SCOT approach warns against assuming a taken-for-granted deterministic meaning of a technology across times, spaces, and communities (Kline & Pinch, 1996). In other words, according to SCOT the definition and meaning of technology constantly changes for different groups of people and across different contexts. For example, in studying historical development of bicycle, Bijker (1995) finds that bicycle that is interpreted as a means for sports in modern times was interpreted as a symbol of being macho sometime in late 1800s.

Pinch and Bijker’s SCOT view was built based on the empirical programme of relativisim (EPOR). EPOR is an ongoing effort by sociologists in order to understand the natural sciences from a social constructivist approach (Pinch & Bijker, 1984). Understanding scientific knowledge according to EPOR can be explained in three stages.

In the first stage, scientific findings exhibit a flexibility of interpretations. In other words, the findings of a scientific endeavor, at this stage, are interpreted differently by various groups that are relevant to the scientific endeavor (Collins, 1983). However, diversity of interpretation of findings fades as the scientific community reaches a consensus as to what constitutes the ‘truth.’ In the second stage, the social mechanisms which limit flexibility of the interpretation of the scientific findings and permit a closure in the scientific controversies are described (Ibid). In the third stage, these ‘closure mechanisms’ are tied to wider social and cultural environment. EPOR is a well-established approach within scientific community and it has been supported by much empirical research (Ibid). Bijker and Pinch identified some similarities between construction scientific knowledge and development of technology before they formulated SCOT. Therefore, they adopted ideas from EPOR and take a similar approach in sociological studying of technological developments.

In their 1984 article, Pinch and Bijker set forth a contention that a unified social constructivist approach to science and technology could benefit both fields. At the time, the social constructivist view was an established approach in science. Bijker and Pinch’s purpose from proposing such unification was to adopt an approach to studying technology similar to the social constructivist approach of EPOR towards knowledge in the science community. In doing so, Pinch and Bijker first explained the similarities of scientific endeavors towards knowledge and sociological studies of technology. Then they argued that adopting a similar approach as EPOR would be justified to study technological developments.

Pinch and Bijker (1984) argued that with the growth of studies of science during late 1970s and early 1980s most researchers became more inclined to consider a separation between science and technology. Later on, one of the major developments of the field of science was application of sociology of knowledge to studying hard sciences. Bloor (1973) outlined such an approach to hard sciences, according to which searching for scientific knowledge is done in the domain of social world rather than natural world (Pinch & Bijker, 1984). This led to generation of a rigorous program for empirical research. By 1984, when Pinch and Bijker wrote their article, there had been wide-spread agreement within the science community that scientific knowledge was socially constituted. The authors believed that social constructivist approach not only was gaining wide-spread acceptance but also had demonstrated potentials for a broader application. The authors argue that the idea that "science is about discovery of truth whilst technology is about the application of truth" (p. 402) is something of the past and "will no longer suffice" (p. 403). In this regard, they quote Layton (1977):

> "Science and technology have become intermixed. Modern technology involves scientists who 'do' technology and technologists who function as scientists … The old view that basic sciences generate all the knowledge

> which technologists then apply will simply not help in understanding contemporary technology" (p. 210).

In short, Bijker and Pinch (1984) considered science and technology both as socially constructed cultures. Therefore, taking a similar approach to EPOR and making parallel comparisons between EPOR and SCOT, Pinch and Bijker (1984) outline SCOT by explaining it using the case of development of bicycle.

> "In the social constructivist approach, the key point is not that the social is given any special status *behind* the natural; rather, it is claimed that there is nothing but the social: socially constructed natural phenomenon, socially constructed social interests, socially constructed artifacts, and so on" (Bijker et al., 1987, 109).

As a result, Bijker and his colleagues defined a technological artifact to be socially shaped as an essential part of stable socio-technical networks to reflect interests and activities of different relevant groups (Pinch & Bijker, 1984; Bijker et al., 1987). This implies that an artifact might go through several processes of design, testing, selection and redesign of different options. In this process, new social groups might emerge which might shape further direction of the artifact based on their preferences. This might result in a dramatic change in an artifact from start to a more stabilized form (Bijker, 1995). "Through an examination of the competing technological options, the changing designs and applications, and the role of the various groups involved, it thus becomes possible to analyse the physical development of a new technology in sociological terms" (Hedgecoe & Martin, 2003, p. 330) The belief is that new knowledge is constantly created along with new technological artifacts and socio-technical relations in a process of mutual shaping (Hedgecoe & Martin, 2003, p. 329).

Having described the background of SCOT and explained it briefly, next we delve into each element of this approach in detail.

3.3. SCOT Theoretical Framework

According to SCOT, a technological development is a result of a social process rather than an isolated process pursued by an individual. According to this theory, different relevant groups are involved in development of a technology. In addition, ‘working’ and ‘nonworking’ attributes of a technology are also socially constructed rather than being inherent to a technology (Bijker, 1995). “In this way, the ‘working’ and ‘nonworking’ are now being treated as *explanandum*, rather than used as *explanas* for the development of technical artifacts” (Ibid, p. 75).

SCOT’s emphasis on ‘thick description’ is a fundamental difference between this and other views (Geertz, 1973). In this type of studies, a detailed description of the technological artifact itself and its surrounding context is provided. By providing a thick description of the technology, the researcher follows the design of the artifact as it takes form and evolves (Bijker et al., 1987).

SCOT was briefly introduced in the beginning of previous section. In this section we further elaborate on different elements of this theory. As is discussed, these different elements are all intertwined since technological development is an ongoing process. In order to explain SCOT in further details we have classified its elements in interrelated pairs. These pairs include, Relevant social groups and Interpretive flexibility, Technological frames and Inclusion, and Stabilization and Closure.

3.3.1. Elements of SCOT

In this section different elements of the SCOT theory which were introduced in chapter two (section 2.4) are discussed. These elements include: relevant social groups and interpretive flexibility, technological frames and inclusion, and stabilization and closure.

3.3.1.1. Relevant Social Groups and Interpretive Flexibility

A main concept of SCOT approach to investigating technological development is identifying different relevant social groups. According to SCOT different relevant groups interpret the technology in different ways. Studying the history of bicycle, Bijker identified different meanings that various relevant groups attached to the bicycle. One of the early forms of bicycle was called high-wheeled ordinary (ordinary for short). Figure 3.2 shows an example of this type of bicycle. Two of the interpretations that were identified by Bijker (1995) classified the Ordinary into categories of “Unsafe” and “Macho”. The former was due to interpretation of a group of people (mostly seniors and women) who believed riding the bicycle was not safe. The latter meaning was given to the ordinary by a group of people (mostly young men) who viewed the ordinary as a symbol of masculinity.

Figure 3.2: A high-wheeled ordinary (Source: Smithsonian[1])

Therefore, based on different meanings that were given to the bicycle throughout its history of development, Bijker identified various relevant social groups. In SCOT shared meaning is the key element in identification of a relevant social group (Bijker, 1995). Initial identification of relevant social groups is provisional. The analyst divides heterogeneous social groups to different more homogeneous groups according to the meanings they assign to the artifact (Pinch & Bijker, 1984). Members of a social group "may of course share other properties of family resemblance, which also give them their group characteristic" (Kline & Pinch, 1996, p. 765). However, the shared meaning of the technological artifact is the key in explaining its further developmental paths (Kline & Pinch, 1996).

[1] http://www.si.edu/
http://historywired.si.edu/object.cfm?ID=221

Bijker (1995) argues that actors provide an effective starting point for identifying relevant social groups. In other words “relevant social group is an actor category” (p. 77). Bijker follows different relevant social groups by following the actors because he believes that “they are quite explicit about it” (p. 76) in the sense that they usually refer to their relevant group when discussing issues around the artifact.

Bijker (1995) acknowledges the potential shortcomings of this method. For example, he believes that “it may be difficult to decide whom to treat as spokespersons for a specific relevant social group” (p. 77). However, he believes that this will become clear if the analyst allows the actors to speak for themselves. Bijker also admits that only those attributions of meaning that are vocalized by the actors are analyzed. This ethnographic approach purposely focuses on meanings that are attributed to the artifact and it does not attribute hidden interests to different parties (Bijker, 1992).

Bijker recommends “historical snowballing” by reviewing the historical documents, identifying all the actors and social groups mentioned in them, and following the identified actors (Bijker, 1992). Other groups can also be identified by observing the interactions and negotiations of the first social relevant groups discovered (Ibid). “Of course this is an ideal sketch, because the researcher will have an intuitive idea about what set of relevant social groups is adequate for the analysis of a specific artifact and, consequently will not follow this road to its very end” (Ibid, p. 77).

Bijker (1995) suggests that as the researcher learns further about different interpretations, he might change the structure of relevant groups he has identified. For example, he might break down larger groups to smaller one to reflect specific meanings assigned to the artifact more clearly. Therefore, "'relevant social group is both an actor and [the researcher's] category. When following the actors in their identifications definitions, and delineations, it is the actors' relevant social groups that we get" (p. 78). At the same time the choice of defining social groups, and arranging and re-arranging them is dependent on the researcher and his intuition. Thus, relevant social groups is also the researcher's category.

According to SCOT, since different relevant social groups interpret a technological artifact differently by assigning different meanings to it, a technological artifact is said to have interpretive flexibility. In other words, a technological artifact which is still in its development process and has not reached a relatively stable form means various things to various people.

In order to explain interpretive flexibility empirically, Bijker explains how the use of air tire was received by various relevant groups when it was first introduced. When air tires were mounted on bicycles, they were received differently by different relevant social groups. A group of people who received this solution enthusiastically believed that air tire was a solution to the long-term problem of vibration of "safety"[2] bicycles or low-

[2] Safety bicycles were descendents of ordinaries. They were developed to address the safety issue which prevented some relevant groups (such as women and elderly) from riding the bicycles.

wheelers (Figure 3.3). Another group saw it as a way of improving the speed[3] of the bicycle. For yet another group "it was an ugly looking way of making the low-wheeler yet more unsafe (because of side-slipping)" (Pinch & Bijker, 1984, p. 422). By constantly identifying various relevant groups, Bijker followed interpretive flexibility of bicycle and thus development of this artifact.

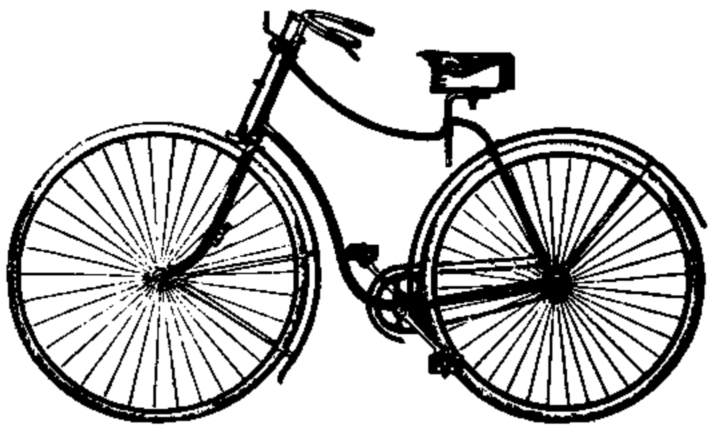

Figure 3.3: Rover Safety bicycle (source: University Rhode Island website[4])

Interpretive flexibility implies that there is flexibility in how to interpret or think of artifacts. Moreover, it means that "there is not just one possible way, or one best way, of designing an artifact" (Pinch & Bijker, 1984, p. 421). As Bijker found new relevant social groups he deconstructed bicycle artifact into different artifacts based on various meanings different groups attached to it. Bijker (1995) argues that "Once an artifact has been

[3] Eventually, the interpretation that considered air tire as a means to improve speed became dominant. This will be explained in the section on "stabilization and closure".
[4] www.phys.uri.edu/~tony/bicycle/bikehist.html

deconstructed into different artifacts, it is clear what has to be explained: how these different artifacts develop; whether, for example, one of them peters out while the other becomes dominant" (Bijker 1995, p. 77).

Various interpretations of a technology lead to different designs of the artifact through different "chains of problems and solutions," (Pinch & Bijker, 1984, p. 423) which emphasizes on the socially constructed nature of it. Hughes state this clearly:

> "Technological systems solve problems or fulfill goals using whatever means are available and appropriate; the problems have to do mostly with reordering the physical world in ways considered useful or desirable, at least by those designing or employing a technological system" (Hughes, 1987, p. 53).

It has to be noted that only those problems are considered relevant for which there is at least one relevant social group (Pinch & Bijker, 1984). In other words, when there is no relevant social group, there is no problem associated with the artifact. When development of an artifact is based on identifying problems and solutions, different kinds of conflicts could be identified. These conflicts could be based on a range of issues from technical requirements to moral issues. The solutions to these conflicts are not always technical. For example, designing comfortable yet conservative clothing for female riders was a solution to the 'moral' problem of women riding bicycles wearing 'inappropriate' clothing.

In sum, identifying relevant social groups is key when it comes to studying development of a technological artifact through SCOT approach. Each groups attributes different meanings to the technology (i.e. interpret it differently). There is no one way of identifying social groups and to a high degree it is dependent on the analyst. However, there are some steps that if taken could point the analyst to discovering new relevant social groups. For example, normally different opinions shape when there are problems and controversies around the technological artifact. Therefore, locating the controversies and finding out the ways different groups approach it could be one way to identify different relevant social groups. In addition, reviewing historical (and any relevant document) could also be beneficial in identifying various groups and their interpretations of the technological artifact.

3.3.1.2. Technological Frames and Inclusion

The descriptive theoretical model of technological frame is a guide through which a group interacts with the technology, interprets it, approaches the controversies around it, and finds solutions to their problems related to the technology. This guide (i.e. technological frame) is built when a specific relevant group is formed around a technology and the interactions of its members begin. The frame is built based on the background of the group such as the beliefs of the group members, and their problem solving strategies. "Existing practice does guide future practice, though without logical determination. If existing interactions move members of an emerging relevant social group in the same direction, a technological frame will build up; if not, there will be no frame, no relevant social group, [and] no future interaction" (Bijker, 1995, p. 123). Therefore, technological frame of each group distinguishes it from other groups. Since, technological frames of various groups guides how they approach the technology, frames have an impact of the course of development the technological artifact takes. At the same time, further development of the technology, modifies the interactions of the groups and thus further shapes the technological frame. Therefore, the process of development of a technology is a dynamic one during which the elements of SCOT including technological frames are constantly changing. In the discussion below we further explain the nature of technological frames.

The concept of technological frames was formulated to refer to the ways that different relevant social groups assign meanings to a technological artifact. Members of a relevant social group share the same technological frame. Technological frames influence the way their members view the technology and act towards it and interact with one another. Technological frames encompass the goals, ideas, and the tools they would need to take actions and interact with the technology, within the group, and with other groups. In addition, frames are used as guides to tackle controversies. Technological frame of a group is based on their views and philosophies of approaching the problems (Bijker, 1995). Bijker explains technological frame as follows.

> "A technological frame comprises all the elements that influence the interactions within relevant social groups and lead to the attribution of meanings to technical artifacts-and thus to constituting technology … These element include (to begin with, at least): goals, key problems, problem-solving strategies (heuristics), requirements to be met by problem solutions, current theories, tacit knowledge, testing procedures, and design methods and criteria" (Bijker, 1995, p. 123).

Technological frames of different groups influence the development of the technological artifact. At the same time, further development of the artifact modifies the frames. Bijker developed the concept of "technological frames" to explain the 'double-sided' (i.e. social aspects impacting development of technology and technological artifact having social impact) character of technological development.

> "The technological frame of a social group is shaped while an artifact, functioning as exemplar, further develops and stabilizes within that social group … But a technological frame in turn also determines (albeit to different degrees, depending on the degree of inclusion different actors have in that frame) the design process within that social group" (Bijker, 1992, p. 98).

Therefore technological development has two sides: social impact; and social shaping (Hughes, 1987; Bijker, 1992). As shown in Bijker's (1992) case study of development of fluorescent lamp, high-intensity lamp was socially shaped as a result of interactions and negotiations of different relevant social groups to resolve a controversy. "On the other hand, this artifact also influenced society by giving rise to new lighting standards which in the end became universally accepted – so this artifact also had quite a social impact" (Bijker, 1992, p. 98).

Even though a technological frame assists its group members with identifying problems and finding solutions, it also results in constraining the freedom of members. This is because of the structure that is created by actions and interactions within a relevant social group. "Within a technological frame not everything is possible anymore (the structure and tradition aspect), but the remaining possibilities are relatively clearly and readily available to all members of the relevant social group (the actor and innovation aspect)" (Bijker, 1995, p. 192). Technological frames thus can be both enabling or constraining.

Not all the members of a specific relevant social group are equally involved in the technological frame of that group. The notion of inclusion takes into account the degree to which individuals are involved in a specific technological frame(s) (Bijker et al., 1987). Bijker (1995) introduces ‘technological frame’ and ‘inclusion’ as two related topics.

The extent to which an actor’s interactions are structured by technological frames of a group is influenced by the degree of inclusion of that individual in the frame. Different actors have different degrees of “inclusion” in a technological frame. At the same time, every actor can at any time be a member of different social groups. “Often one individual can partake in a number of different technological frames and can be weakly included in some frames and strongly included in others” (Kline & Pinch, 1996, p. 765, 766).

High degree of inclusion in a technological frame means that he thinks, acts, and interacts to a large extent in accordance with the frame. Degree of inclusion is adopted by the analyst to explain his interpretation of the patterns of interactions of actors. It also helps the analyst to understand how individuals diverge from the shared meaning of a relevant social group (Bijker, 1995).

Similar concepts to technological frames such as ‘technological paradigm’ (Van den Belt and Rip, 1987) exist. Technological frame shares some similarities with these concepts. For example, the notion of ‘exemplar’ by Van den Belt and Rip indicates a collection of successful heuristics around a technological artifact. Bijker uses ‘exemplar’ as a label for an artifact that, in stabilizing, structures the technological frames of different relevant groups (Bijker et al., 1987). Technological frames concept is different from ‘paradigmlike’ concepts in two important aspects. Firstly, technological frames applies to all kinds of relevant social groups (for example not just to scientists). Secondly, technological frame is an interactionist concept (Bijker, 1987, p. 185).

In sum, technological frame of every relevant social group refers to the way the group interprets a technological artifact. The frame guides the way the individuals interact within the group or with other groups. A technological frame is based on the approaches the group takes to solve the problems around a technological artifact. In order to understand technological frame of a group, the analyst would pay close attention to ways the group tackles the controversies. In short, he would investigate about the goals, views, and problem-solving strategies of each relevant social group. People differ in their degree of inclusion in a specific technological frame. Each individual might belong to more than one technological frame at any point in time. Looking into degree of inclusion of actors, the analyst would gain an understanding of the patterns of the interactions of actors. In addition, it would assist him to understand why and how actors might diverge from the shared meaning of the group towards a technological artifact.

3.3.1.3. Stabilization and Closure

Two other elements of SCOT are “stabilization” and “closure”. According to these notions, interpretive flexibility of a technological artifact is not everlasting. Eventually, the artifact reaches "stabilization" and "closure" in a way that some forms of the artifact appear to be less controversial and increasingly become the dominant form of the technology (Kline & Pinch, 1996). The concepts of stabilization and closure show that development of a technical artifact is a continuous process, which does not occur at equal rates at every point in time (Bijker, 1995). Here we discuss this two concepts further in detail.

The focus of stabilization (which is a matter of degree) is on the artifact itself and the closure focuses on its interpretive flexibility. Stabilization has been introduced as an intra-group notion of development of an artifact. Degree of stabilization is different within each relevant social group. Closure is a similar concept but refers to inter-group analysis of development of an artifact. Degree of interpretive flexibility decreases with closure of an artifact. Meanwhile, one artifact becomes dominant and other artifacts diminish. The dominant artifact then develops with an increasing degree of stabilization within one or more relevant social groups. In short, "a simplified way to distinguish between closure and stabilization is that closure is about people and stabilization is about the artifact" (Humphreys, 2005, p. 249).

In "closure" the focus is on different meanings given to the artifact by different relevant social groups. In "stabilization," on the other hand, the focus is the artifact itself. The former refers to "acceptance of dominant technology interpretation across relevant social groups," (Davidson & Pai, 2004, p. 476) which is almost irreversible (Bijker, 1995). The latter refers to the "strengthening of a TFR within a social group" (Davidson & Pai, 2004,

p. 476). According to Bijker, “if the closure concept has a primarily social interactionist origin, the stabilization concept is colored more by semiotics” (Bijker, 1995, p. 85).

Different interpretations of an artifact by various social groups are often accompanied with controversies around the technology and it achieves its final stabilized form when “debate and controversy about the form of an artifact is effectively terminated” (Law, 1987, p. 111). For example, when the ‘safety bicycle’ became stabilized, this term indicated a low-wheeled bicycle with diamond frame, rear chain drive, and air tires. These features were taken for granted after ‘safety bicycle’ was stabilized and one did not have to provide specifications when referring to it. In social constructivist approach this process is called closure.

Closure in a technological development involves ‘disappearance’ of problems related to the artifact. However, “to close a technological ‘controversy’ the problems need not be *solved* in the common sense of that word” (Pinch & Bijker, 1984, p. 427). What is important is that the problem is seen as solved by the relevant social groups. In other words, there is not one objective solution to a problem. For example, a problem to be solved “may postdate the emergence of the system as a solution” (Hughes, 1987, p. 53).

Bijker (1995) offers two mechanisms for closure, though does not restrict the possibility to these two. For example, Bijker explained how advertising can play a key role in shaping the meaning given to an artifact by a relevant social group. That is why an attempt was made to ‘close’ the ‘safety’ controversy of the high-wheeler by advertising that the artifact was perfectly safe. This claim was a rhetorical move, which worked in convincing the public about the safety of the high-wheeler. Otherwise, it was perfectly known to the engineers at the time that the height of the bicycle and the forward position of the rider could pose safety problems (Pinch & Bijker, 1984).

As another example, in Bijker's case study of fluorescent lamps, when the utilities company feared a decrease in sales of electricity they decided to solve this problem by advertising. In order to address the problem of decrease in electricity sales, "electrical utilities through advertising and other marketing tactics stimulated the need for home appliances that would use electricity during hours when demand was low" (Hughes, 1987, p. 53). Bijker (1995) suggests that closure can happen through different mechanisms. In the aforementioned examples, the closure mechanism was a rhetorical one. Bijker (1995) also proposes another closure mechanism, which he calls *closure by redefinition of problem*.

In the case of bicycle, eventually redefinition of the problem air tire was expected to solve brought about closure for some relevant social groups. In the beginning, it was assumed that air tire would solve the problem of vibration. As mentioned before, opinions about the potential of the air tire for safety bicycle varied. As explained by Pinch and Bijker, air tire,

> "for most of the engineers was a theoretical and practical monstrosity. For the general public, in the beginning it meant an aesthetically awful accessory … For Dunlop and the other protagonists of the air tyre, originally it meant a solution to the vibration problem. However, the group

> of sporting cyclists riding their high-wheelers did not accept that to be a problem at all" (Pinch & Bijker, 1984, p. 428).

However, after air tire was installed on a racing bicycle, it demonstrated significant improvement in the speed. Therefore, it was the promise of addressing the problem of speed that closed the air tire controversy by convincing more relevant groups about the benefits of air tire.

Pinch and Bijker (1984) argue that "one can say, we think, that the meaning of air tyre was translated to constitute a solution to quite another problem: the problem of 'how to go as fast as possible'" (Pinch & Bijker, 1984, p. 428). This shows that closure does not necessarily happen the same way and at the same time for all the relevant social groups.

In addition, closure and stabilization of a technological artifact might not be permanent. New controversies and problems may arise, which can lead to reappearance of interpretive flexibility (Kline & Pinch, 1996). Based on this view, stability of the artifact is merely provisional "even after a technological artifact appears to have solidified, with the discourse around its functions and features apparently having reached 'closure'" (Orlikowski & Iacono, 2000, p. 358). This provisional nature could be due to different reasons, for example, reappearance of the competing artifacts, invention of new materials, new standards being set, and unanticipated/unintended uses of technology (Mackay 1988; Orlikowski & Iacono, 2000; von Hippel 1988)

In sum, stabilization of a technology, which has its roots in semiotics is an intra-group concept and is related to the artifact itself. As a technological artifact stabilizes within a relevant group, the problems of the group regarding the technology diminish. Stabilization also helps with strengthening a technological frame within a group. The degree of stabilization can vary from one group to another. The concept of closure, which originates from social interactionism is an inter-group phenomenon, relates to acceptance of a dominant technology across groups. As a technological artifact reaches closure, its interpretive flexibility diminishes. At the closure stage, the controversies around the technology disappear, even if temporarily. At this stage, the technology becomes taken-for-granted. Closure of a technological artifact does not imply that the artifact will not change. Closure happens in the meaning and interpretation of the artifact at the time being. Stability of the artifact could be temporary and for different reasons it could be subjected to change. In the next section we look into some of the criticisms of SCOT approach and the ways these issues have been addressed.

3.4. Limitations of SCOT

The social constructivist approach to explaining scientific knowledge and technological artifacts by Pinch and Bijker (1984) has been criticized by some. One of the main critiques of their work is Clayton's (2002) article, issues of which Bijker and Pinch address in their (2002) article.

Clayton (2002) criticizes SCOT by Bijker and co-authors mainly based on the 'factual errors' in the narrative of the development of bicycle. He argues that due to these errors, the developed SCOT approach is flawed because it is not supported by empirical data. For example, Clayton argues that the choice of the social groups of women is based on two empirical accounts: a picture of a girl[5] riding a bicycle which was used out of context; an un-referenced magazine clipping[6]. Clayton asserts that Bijker et al.'s justification for considering women's social group is weak because women "never actually rode the ordinary except on the stage or in the circus" (Clayton, 2002, p. 356).

Bijker and Pinch (2002) responded to this criticism arguing that Clayton's conclusion about the inadequacy of SCOT is based on his misconception of the relationship between theory and empirical verification. They argue that Clayton has misread their text in that they never claimed women's group were also riders of the ordinary bicycle. "On the contrary, [they] highlighted the problems that existed for women" (Bijker & Pinch, 2002, p. 363). They further explain that Clayton has misunderstood the concept of relevant social groups because he interprets social groups as users of the technology. They add that "the whole point of introducing the concept of relevant social groups was to get away from such narrow definitions of who and what are relevant in the development of technology" (Bijker & Pinch, 2002, p. 363). In their view, relevant social groups need to be defined broader. For example, women were considered a relevant social group because they played a role in the development of bicycle exactly because they were not using the ordinary but they desired to. In addition, the researchers did not insist on calling this

[5] "James Starley's niece sitting on a Ladies Ariel … The studio publicity photograph of Starley and his niece is well-known to bicycle historians, as is the fact that the machine was unrideable by either sex and never marketed" (Clayton, 2002, p. 356)

[6] "supposedly in reply to a letter from a young lady in 1885 … [which read] The mere fact of riding a bicycle is not in itself sinful, and if it is the only means of reaching the church on a Sunday, it may be excusable" (Clayton, 2002, p. 356).

social group “women’s group”. They basically intended to include those (e.g. women or elderly) who desired to ride the ordinary but were prohibited because of safety or cultural issues. That is why the authors repeatedly try to clarify that the notion of social groups is not based on pre-defined classifications but is based on the meanings that the artifact has for a specific group. They can change from technology to technology and over time. Individuals can also belong to multiple relevant social groups.

Moreover, Clayton (2002) criticizes the nature of the concept of relevant social groups as being an analyst’s subjective choice. Moon (1997) expresses a similar concern. As Bijker and Pinch (2002) themselves acknowledge, this approach to relevant social groups was intentional. From this and the other shortcomings that Clayton attributes to SCOT, it can be seen that he expects a more deterministic definition of the elements of SCOT, which is in contrast with the nature of this approach. Even Moon (1997) recognizes the complexity of the nature of technological development and regards SCOT as an appropriate heuristic framework or as Bijker (1995) calls it a set of “sensitizing concepts”. Therefore, Bijker and Pinch (2002) welcome this criticism as further strengthening rather weakening SCOT.

Clayton’s other criticism is directed toward the concept of interpretative flexibility. Clayton specifically criticizes the validity of the empirical evidence of SCOT. Once again, Clayton argues that Bijker and colleagues have supported SCOT with incorrect factual data. He specifically disapproves of the example of air tire. As mentioned before, Bijker (1995) introduces air tire as an artifact that was introduced to address the problem of anti-vibration of bicycle. An air-tire-mounted bicycle was not received well by neither but one relevant social group. In order to convince the other groups of the benefits of air tire was redefined to address the problem of speed rather than vibration. This redefinition of problem resulted in the closure of the bicycle for the rest of the relevant social groups.

Clayton argues that the inventor of pneumatic tire, Dunlop, was aware of its capability for increasing speed since its inception. He criticizes Bijker and his colleagues for relying merely on Woodforde (1970) in considering Dunlop's initial view of pneumatic tire as an antivibration rather than a speed-increasing device. Again, as Bijker and Pinch (2002) acknowledge themselves, Clayton's criticism does not fundamentally question the concept of interpretive flexibility. Firstly, as they mention they provided this example as an illustration of the case they were making and they did not use it as "empirical evidence" to confirm the concept of interpretive flexibility. Pinch and Bijker defend this point when they say "we chose to explain certain aspects of the SCOT approach by using the bicycle as an illustration (as Bijker later used Bakelite and the fluorescent lamp to explain other aspects), but the theoretical approach itself does not rest upon this one example" (Bijker & Pinch, 2002, p. 365). Secondly, the context in which they used this example would not dramatically affect the conclusions they drew. They recognized that some social groups were not convinced of the value of air tire as an antivibration device. However, the same group became enthusiastic about the technology once they saw its benefit in addressing another problem. This conclusion would not be undermined whether or not Dunlop was aware of this attribute of the tire. Moreover, the points that Bijker and his colleagues were making were that different groups had different interpretations of air tire and that it took redefinition of the problem in order to convince other relevant social groups who were not interested in this technology. Criticism of Clayton regarding interpretive flexibility, therefore, is not relevant.

Clayton also criticizes the notions of closure and stabilization because he believes that the authors' choice of the start and end dates (1879–1898) for development of bicycle is not factual. Bijker and Pinch (2002) argue that Clayton misses the point by expecting an exact timeline for the development of bicycle. The authors add that "this is exactly what we claimed when criticizing the linear model," since it is a mistake to consider a start and end point for development of an artifact. Clayton argues that like the concept of relevant social groups, closure and stabilization depend on analyst's point of view. Bijker and Pinch (2002) state that:

> "The difference between Clayton and us is not that we deny this; the difference is that we *positively value* this characteristic of theoretical concepts and argue that exactly because of this characteristic they provide the much needed antidote against naïve empiricist ideas about doing history of technology" (Bijker & Pinch, 2002, p. 366, 367).

As a matter of fact, Bijker and his colleagues mention numerous dates for different versions of the artifacts that bicycle could be attributed to date back to. For example, they mention "running machine," which dates back to 1817 or Célerifère, which goes back as far as 1791. From reading the cases about development of bicycle (or any of the other technologies that Bijker and his colleagues study) in SCOT studies, one does not get the impression that the researchers attempt to force a timeline over the evolution of this technology. In addition, Clayton (2002) seems to be missing the point of the concept of closure. Even if Bijker and his colleagues had truly used an incorrect timeline, Clayton's criticism would not undermine the concept of closure per se.

Other criticisms of SCOT are mostly directed to the influence of broader societal structural relationships on technological developments. For example, Winner (1993) argues that issues such as gender, race, class, and ethnicity might affect the course of development of a technology. Douglas (1990) criticizes SCOT for disregarding political economy regarding technological developments. Both Winner (1993) and Russell (1986) critique relativism of SCOT. Winner (1993) argues that relativist and subjective approach of SCOT might lead to an oversimplified view of technology and the society. Some of the issues raised by these researchers have been addressed to Bijker's (1995) work that was published after these studies. In his (1995) book, Bijker explains the concepts of SCOT in more detail; therefore, some of the misunderstandings of the theory are resolved.

As far as considering structural issues such as politics and economy, Bijker does not deny their influence. These issues could be taken into account when identifying social relevant groups and their interpretations of the technology. What needs to be taken into consideration is that SCOT was meant to be a theoretical lens through which technological developments can be studied- studying technological development through interpretations and meaning. For example, studying the influence of people who are relevant to the technology by classifying them based on their race is something that Bijker and colleagues refrain from though they do not deny their influence on interpretation and meaning.

All in all, as the developers of this approach have acknowledged, SCOT is a not a deterministic theory of technological development. These researchers do not claim that the phenomenon of technological development cannot be studied with other approaches. SCOT is a means to be used as a sensitizing lens to study development of a technological artifact paying attention to both its social and technological aspects. Process of development of a technology can become complex due to study due to various relevant groups involved in it. SCOT lends itself well as a theoretical lens to approach investigation of technological developments. This is regardless of its potential limitations and uncertainties, which are by definition a part of every theory.

Having elaborated on the concepts of SCOT and discussing its limitations, we explain how SCOT could contribute to our understanding of configuration of packaged software.

3.5. Potential Contribution of SCOT to the Packaged Software Configuration Literature

SCOT is highly relevant in the context of configuration of packaged software. Various relevant groups in an organizational are normally involved in the process of configuration. Perceptions and expectations of these different groups of the packaged software and its functionalities can be critical. Various understandings of the software by the relevant groups could be reflected in the suggestions they provide for how to configure the software to a workable form. The decisions that are made during the process can be highly influenced by different interpretations. Rich functionalities of software can make understanding of what it can offer and how it can be configured challenging. This challenge could be amplified if different relevant groups understand the software differently. Therefore, the nature of the process of configuration of packaged software is in line with the SCOT approach. Accordingly, based on SCOT different people involved in this process can be grouped based on the shared meaning they assign to the software. This kind of approach to the process of configuration could allow us to examine different perceptions about the software, its functionalities, and configuration options.

Taking a SCOT approach, a researcher would identify problems of various relevant groups regarding the software. Identification of these problems would allow the researcher to closely examine how they are approached and addressed. The researcher would be able to study the process of configuration as a series of technical, social and organizational activities. Furthermore, the emphasis of SCOT on the technological artifact (the packaged software in this case) and providing a thick description of its evolvement from a generic package to a workable system and the social and organizational issues around it make SCOT an insightful lens. Figure 3.4, schematically demonstrates the relationship of different elements of SCOT in the context of the configuration process of packaged software.

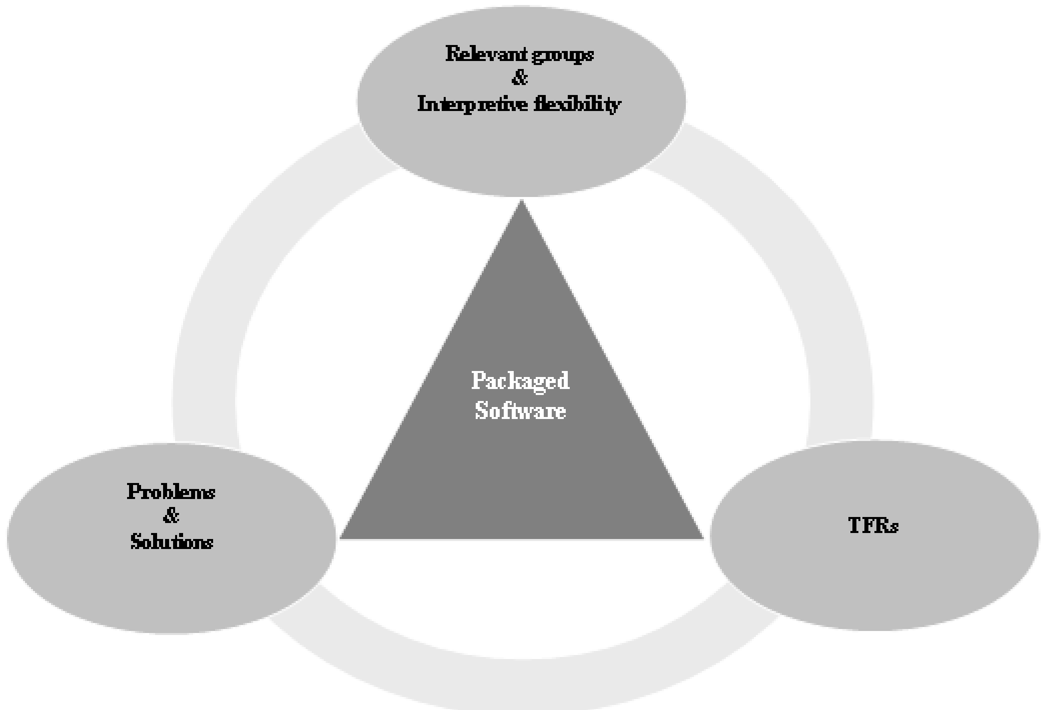

Figure 3.4: SCOT and configuration of packaged software

As a preliminary step to substantiate the application of SCOT in the context of studying packaged software, we searched for studies with similar objectives. Our goal was to look at the studies that took different approaches than ours. We then used SCOT approach to investigate how SCOT would have explained the findings of (or contradictions found in) the studies.

Yeow and Sia (2008), for example, examined the incongruence of socio-cognitive technological frames of different groups in the process of implementation of packaged software. The goal of the study was to investigate how the ‘best practices’ of the packaged software were negotiated by different groups and how these groups reached agreements. Yeow and Sia classified the participants of the project based on their functional areas (e.g. end-users, Financial policy department (FPD)).

Their findings from a pair-wise comparison of various technological frames demonstrated that several frames overlapped on some elements. For example, they found that both end-users and FPD groups' frames emphasized on audit and control issues. This finding is similar to the premise of SCOT that various groups take different approaches towards the technology. However, based on a SCOT approach the classification of the participants of the project would have occurred after identifying their perceptions about the software and its issues. Accordingly, the classification of different relevant social groups is based on their interpretations of the software. In other words, individuals who interpret the software similarly and thus approach its problems in a similar fashion are considered to be members of the same relevant group. An advantage of this new way of classification is that people are grouped based on their points of view rather than being attributed with some points of view based on their job descriptions. This is dramatically different than the traditional way of classifying groups based on their functional areas (e.g. users, developers, and managers) that is prevalent in IS literature. Approaching the process of configuration in this manner would allow us to investigate diverse understandings of the software, its functionalities and configuration options. The insights gained from such an investigation would allow us to further understand how the process of configuration is perceived and approached in an organization.

Examining Yeow and Sia's (2008) work assured our study in two ways. First, it showed that implementation and configuration of packaged software can become complex, studying of which is warranted. Second, SCOT approach demonstrated to be relevant to such studies. Therefore, the insights gained from an SCOT approach to the process of configuration of packaged software would allow us to further understand this process. These insights could also be incorporated by packaged software developers in order to build software that is more understandable by end-users. This contribution is particularly important since the nature of packaged software development does not allow for direct involvement of end users.

In the SCOT approach the focus is on problems around the technology and their resolutions. By identifying the problems that arise in the process of configuration, the researcher can more closely examine how those controversies are approached, and resolved. As discussed earlier, the process of configuration involves a series of technical, social, and organizational activities. These activities revolve around constant understanding of the software, deciding which functionalities of the software to use, and which business processes to change to fit the software. These decisions could be classified as a series of problems and solutions. Therefore, SCOT could offer a good lens through which these processes can be understood and discussed.

Chapter 4

Research Methodology

4.1. Introduction

In this chapter the empirical aspect of this research is discussed. An interpretive case study approach was taken to gather qualitative data for the empirical content of this dissertation. The data was then analyzed through hermeneutics. The empirical piece of this research consists of two case studies. The first case study was conducted as a pilot study with the goal of refinement of the theoretical framework of SCOT in order to be used in the context of the main case study of this dissertation. The second case study was larger in scope, through which the configuration process of packaged software at a large organization was closely examined. This case study was an in-depth embedded case study. In this chapter a discussion of the ontology and epistemology of our research, the data collection method, a description of the cases, and the method of data analysis is offered.

4.2. Ontological and Epistemological Stances of the Study

4.2.1. Ontology

Quine (1948; 1953) emphasizes that a proper way of conducting research involves clearly defining all the objects under the study. The scientist then has to remain committed to those objects all throughout his theorizing (Smith, 2003). Moreover, during the course of the research, the scientist's job also includes clarifying what kind of objects these ontologies approve or disapprove of (Munitz, 1981); this is also called criteria of identity (Chateaubriand, 2003). Accordingly, the concept of software configuration and the theoretical objects of SCOT theory are considered as the ontology of this study.

The main goal of this study is to contribute to the theoretical definition of the configuration process of packaged software. Therefore, the explanandum of study was defined as software configuration. Accordingly, the study remained committed to this ontology when designing the research questions, collecting the data, analyzing the data, and discussing the contribution of this research. In addition, other objects of the study which served to examine the explanandum of this research (i.e. configuration) comprised the elements of SCOT. These objects assisted the researcher as a theoretical lens in identifying the themes related to the explanandum. Clarifying the objects of the study helped the researcher to define the scope of the research from the onset. This was important since the amount of data (because of qualitative nature) would have proven overwhelming. Moreover, the research would not have followed a coherent direction.

4.2.2. Epistemology

Epistemology refers to the approach taken to acquire knowledge about the social world. Zuboff, in her 1988 seminal work, In the Age of the Smart Machine, defines epistemology as follows:

> "Behind every method lies a belief. Researchers must have a theory of reality and of how reality might surrender itself to their knowledge-seeking efforts. These epistemological fundamentals are subject to debate but not to ultimate proof. Each epistemology implies a set of methods uniquely suited to it" (p. 423).

As Quine (1969) suggested knowledge cannot be justified solely based on sensory data. Knowledge instead is constantly explored based on relating the sensory data to the theory; in other words, constantly making sense of received sensory data in the light of the theory. In this study, we define epistemology to be the empirical content of the theory and our epistemological stance is that of interpretivism. Geertz (1973), famously, summarized this view in a sentence: “What we call our data are really our own constructions of other people’s constructions of what they and their compatriots are up to” (p. 9).

In interpretive studies the assumption is that social reality is constructed through interaction and sensemaking (Orlikowski & Baroudi, 1991). Walsham (2006) argues that “interpretive methods start from the position that our knowledge of reality, including the domain of human action, is a social construction by human actors” (p. 320). Accordingly, knowledge is acquired by understanding those processes (Morgan & Smircich (1980). In such process of acquiring knowledge about the social world the focus is on text (Ricoeur, 1971; 1988) as tools for developing insights as to how people make sense of their situations. Walsham (1993) states “the interpretation of texts is an important part of the search for meaning and the essence of experience” (p. 9).

Noticeably, this epistemic endeavor is contextual. In other words, all the processes under the investigation are examined in the contexts they take place. Studying these concepts within their natural setting, the researcher gains an understanding of complex social interactions and sensemaking processes (Orlikowski & Baroudi, 1991). Through this process of understanding, the researcher delves into the meanings from the subjects’ perspective (Orlikowski & Baroudi, 1991). The role of the researcher is then to grasp the meanings that people attach to their situations and contexts, make sense of them, and reiterate them in the form of scientific theories. Generally, the researcher starts the process with a priori theory as a theoretical lens through which he understands the process he is investigating (Walsham, 1995; 2006).

The drivers of the empirical part of this dissertation were the main research questions. Here these questions are reiterated:

1) Why is packaged software configured a specific way and not another?
2) How does this configuration occur?

We chose SCOT as the theoretical lens through which we could understand the process of configuration. Our epistemology of interpretivism fits appropriately with the nature of these research questions. This philosophy is in line with the approach of the SCOT theory. In SCOT, the focus is to gain an understanding of the interactions and the interpretations. The researcher is concerned with understanding the way various relevant groups assign meanings to the technological artifact. According to an interpretive approach, people create the social and physical artifacts socially. In other words, “the same physical artifact, the same institution, or the same human action, can have different meanings for different human subjects, as well as for the observing social scientist” (Lee, 1994, p. 347).

This is the essence of a social construction of technology approach to studying technological developments (Bijker, 1995). The researcher interprets the constructed social reality; i.e. the way it is seen by various actors (Lee, 1994). In so doing, the researcher investigates the way human subjects interpret themselves and their surroundings. This kind of interpretive investigating can be done through various methods: anthropological ethnography, participant-observation, and hermeneutics (Ibid). For the investigation in this dissertation a hermeneutic approach was chosen, which will be discussed later (section 4.3.2). Therefore, an interpretive study lends itself well to the objective of our study, namely, finding and understanding the meanings in the process of configuration of packaged software.

The configuration process of software is a socio-technical one. The research questions of this dissertation were meant to tackle a process in which various social groups are involved. The goal of the study was to investigate how participants in the process of configuration make sense of this process and how they approach it. More specifically, the objective was to understand how the configuration process is influenced by the way these social groups make sense of it. The choice of the theoretical lens of SCOT demonstrated to be in line with the objectives of the research and the epistemological view of this dissertation.

SCOT studies a social process as it unfolds. Furthermore, one of the main driving elements of SCOT is interpretive flexibility, which mainly looks into various interpretations that are formed by relevant social groups involved in the process of development of a technology. Thus, when designing this research, SCOT seemed to be a promising theory through which the phenomenon under the study could be investigated.

The appropriateness of this theory was substantiated by examining another theory, namely TFR, which has been applied as a theoretical lens to study similar phenomena. SCOT theory as well as a comparison of it with TFR was discussed in chapters two and three. The pilot study, discussed in chapter five, provided empirical validation for more suitability of SCOT over TFR for our study.

4.3. Research Methodology

This section first elaborates on the appropriateness of case study research as the research methodology of this dissertation (section 4.3.1). This discussion is followed by discussing the two case studies that were conducted in this dissertation (sections 4.3.1.1 and 4.3.1.2). The chapter concludes with an explanation of the data analysis approach (section 4.3.2).

4.3.1. Case Studies

Case studies are appropriate when the researcher is investigating the how and the why of a phenomenon (Yin, 1986; 1993; 1994; 2002). The researcher seeks explanation of the processes when he is interested in the how of the phenomenon. He looks for reasons when he investigates the why of the phenomenon. In addition, when the phenomenon is a set of contemporary events and is being investigated in the real-life context, case study approach would be aligned with this goal (Lee, 1989). Through the case study approach, the meaningful and holistic attributes of real-life events, over which the researcher has little control, are retained. Moreover, case study offers the opportunity to collect data from various sources (Ibid). This helps with validating the theoretical generalizations drawn from the insights gained through the study (Ibid).

In this dissertation, the main concern was the why and how of the process of configuration: 1) why is packaged software configured a specific way and not another?; and 2) how does this configuration occur?. Corresponding to the elements of our theoretical framework, more detailed questions were developed to be used as a guide in data collection and data analysis. These questions investigated the interactions of various groups that are involved in the configuration process with the software and with one another. The questions also aimed at studying interpretations of those groups of the software, its configuration options, the process of configuration, and relevant organizational processes. These questions correspond to the elements of our theoretical framework, SCOT. The questions along with the elements of the theory to which they correspond to are shown in Table 4.1. These questions were initially developed by the researcher based on the process of configuration and the theory of SCOT discussed in chapter three (reflected in Figure 3.4). The questions were then reviewed and validated by a second researcher. The questions were then used as a guide in the second analysis of data of the pilot case study (chapter five, section 5.3). During this process the questions were further refined which are presented in Table 4.1. The questions then were used as a guide[1] during the data collection and analysis for the main case study.

These questions needed to be investigated in their real-life context so that the natural course of the configuration process could be followed over time without external interference (Pettigrew, 1985; Gummesson, 1991; Miles & Huberman, 1994). Moreover,

[1] It has to be noted that because of the semi-structured nature of the interviews not all of these questions were necessarily asked from the interviewees. By using the questions as a guide, we mean utilizing them as a framework to have a clearer focus when approaching the data collection and analysis.

multiple sources of data were needed to fulfill the objectives of this study. The interest was to investigate the technical details of the software as well as the social dynamics of the interactions of various relevant groups. For these reasons, there was a need review technical documents as well as observe the interactions of various groups. Conducting a case study would provide this opportunity. Since the goal from this study was to contribute to theory, the insights from the case study would allow for drawing theoretical generalizations (Walsham, 1995). Therefore, the case study approach was chosen to empirically investigate the process of configuration of software. Different methods of data collection were adopted a discussion of which follows, separately in the context of each case study.

Table 4.1: The researcher's guide for data collection and analysis

Elements of SCOT	**The researcher’s guiding questions**
Technological artifact	- What are the features of the software? - What is the software used for? - What are some of the controversial features (in terms of configuration) of the software? - What features have changed from a version to the next?
Relevant social groups & Interpretive flexibility	- What are different perceptions about the software? - Why and how each relevant social group interacts with the software? - Why and how each relevant group interacts with other groups? - Why and how members of each relevant group interact within their group? - What are the software-related problems/issues that each group deals with? - What problems does each group believe the software is solving for them? - What problems does each group believe the software has created for them?
Technological frames & Inclusion	- How do different groups approach their software-related problems? - What are some of the distinguishing beliefs of each group that might influence how they approach the software, the problems related to it, and its configuration? - How has each group tackled the problems related to the software? - Which individuals belong to more than one group? Do those groups share any beliefs or ideas? What are the roles of these individuals in inter-group interactions?
Stabilization & Closure	- What are the characteristics/features of the stabilized software? - How does the software stabilize to a workable form? - How are various closure mechanisms formed? - How and why does interpretive flexibility of the software diminish? - How do the controversies around the software reach closure? - How are the controversies/problems around the software addressed? - Which groups are more influential in bringing about the closure? What are their approaches?

4.3.1.1. Case One: Pilot Case Study -- Child Org

The case of Child Org looks into the perceptions of employees of two information systems and the need for one that would integrate the two. This case was an introductory case that became the basis for the choice of the theoretical framework of this dissertation. In this preliminary study which took place in 2005, the researcher adopted Orlikowski and Gash's (1994) technological frames of reference (TFR) as the theoretical lens to study IS use at Child Org. As it is discussed in the analysis of this case in chapter five (section 5.2), TFR proved to be insufficient in explaining the discrepancies that were observed in interpretations of employees regarding the two information systems. Therefore, the researcher sought another theoretical lens to analyze the data from Child Org. SCOT which has similar premises to TFR seemed to be appropriate. This theory was used in the re-analysis of the data from Child Org. In chapter five, a discussion of the analyses of data based on TFR and SCOT is offered.

4.3.1.1.1. Child Org Case Background

An interpretive case study was conducted at Child Org[2] between October 2005 and March 2006. Child Org is a nonprofit organization that offers a variety of social services to children. These services include Foster Family Care, Adoption, Counseling, Prevention, and Residential services. At the time of this research, the organization was using two main information systems (CDT and CC Solutions, or Solutions for short) that were directly related to the main functions of the organization: Child Welfare and

[2] In this dissertation, fictitious names are used for people and organizations.

Counseling[3]. CDT is a window-based system, which was used for traditional counseling. The system keeps a record of patients and their treatment processes. The system is mostly used for scheduling and billing. Since its initial implementation, CDT had gone through a major update from a DOS-based to a window-based program. Solutions is an internet-based clinical database system, which stores the information on foster children and their treatment plans. This system had also gone through a major update through which the system became an internet-based program.

In an initial meeting with Child Org, general issues about IT in the organization were discussed. One of the issues that repeatedly came up during this meeting was a major need for an integrated system that could replace CDT and Solutions. The reason for this need was explained to be that CDT and CC Solutions did not communicate with each other; this integration was essential to the operations of Child Org. At the time of this study, Child Org was in the process of finding a system that would integrate the functionalities of both CDT and CC solutions. In the existing form, the functions that required information from both systems were performed manually.

The concern over an immediate need for acquiring an integrated new system was also later on repeatedly heard from interviewees during the course of the study. Since one of the biggest challenges at Child Org was that CDT and CC Solutions were not integrated, the researcher decided to focus the study on these two systems. The goal was to investigate the perceptions of employees from different roles and areas within

[3] Programs such as foster care and residential care are part of child welfare program. Counseling programs include different types of clinical and behavioral treatments. These treatments are offered to children under foster, residential and non-residential care.

organization about CDT, CC Solutions and the need for an integrated system. This case is explained in detail in chapter five. Data collection techniques at Child Org are elaborated in the following sub-section.

4.3.1.1.2. Data Collection at Child Org

When the study was initiated, an initial meeting was scheduled with the chief operating officer (COO)[4], IS director, and the research and associate research supervisors. The data collection started with a semi-structured interview with the IS director, as a follow-up for the first meeting. In this meeting, the IS director showed CDT and its features to the researcher. She also discussed several issues around the use of CDT and Solutions. This interview was followed by thirteen other semi-structured interviews with different users, and IT and non-IT managers. The average length of each interview was one hour. The collected data from each interview was transcribed to the form of word documents and then transferred to NVivo[5], a qualitative analysis tool, which was used as a database and a means of data analysis. This helped with strengthening the validity of the study (Yin, 2002). During some of the interviews, the researcher was also able to observe the systems while users were operating them. This observation was limited because the data stored in the databases is highly sensitive. Some organizational documents were also available for review.

---

[4] COO was the main contact of the researcher with the organization.
[5] A short description of NVivo in the context of the main case study (Pub Org) is provided in Appendix B.

4.3.1.2. Case Two: Main Case Study -- Pub Org

The case of Pub Org constitutes the main empirical content of this dissertation. In this case, the configuration process of a software package was investigated using the theoretical lens of SCOT. In this section a background of the case as well as the techniques that were used for data collection are discussed. A discussion of the analysis of the data is provided in chapter six. Implications of this study are discussed in chapter seven.

4.3.1.2.1. Pub Org Case Background

This case study was conducted at a large public organization with over 20,000 employees. The real name of the organization is kept anonymous and it is referred to as Pub Org in this study. The project that was studied affected two of the four sub-divisions of PubDiv, one of the divisions within the organization. We call the two different sub-divisions DIVA and DIVB, with respectively 650 and 250 employees. Even though there had been discussions about "going live" or upgrading since 1999, it was not until June 2004 when Pub Org took some initiations to actually upgrade its existing work management system. Work management system controls work order processes for reactive as well as periodic preventive maintenance. This initiative was in response to internal concerns over the reactive nature of maintenance efforts at the organization. The goal of the project was to improve efficiency by creating a balance between planned and unplanned work. Management felt the need for changing the maintenance work to more proactive and preventive rather than reactive efforts. The ultimate goal was to improve efficiency and equipment availability, reduce cost, reduce asset failure risk, and achieve performance goals.

The goal of reversing the constant trend of increasing in reactive maintenance gave rise to the idea of upgrading the existing work management system MP5 to a newer version called 7i by the same vendor. Both DIVA and DIVB had seen a decline of preventive maintenance work from approximately 70% in 1996-1997 to 35-38% in 2003-2004. Besides the goal of controlling costs, improving efficiency, improving equipment availability, reducing asset failure risk, and achieving performance goals, PubDiv was believed to also benefit from consistently measuring and monitoring maintenance efforts and impact of those efforts on equipment availability. In addition, PubDiv did not have a systematic way of tracking their assets. Therefore, they saw the need for a system that would streamline this process. In line with these objectives the management decided to implement a software package that performed as both work management and asset management systems.

When the idea of upgrading was initiated, the existing work management system (MP5) was functional in fulfilling the core maintenance and scheduling functions of both subdivisions. However, it was decided that MP5 needed to be upgraded to not only address maintenance strategies but also be able to respond to target business processes, and organizational requirements. Therefore, a few leading work management systems such as Datastream 7i, MRO Maximo, and OracleAM were evaluated. “It was determined that Datastream 7i substantially provided the required functionality in an “out-of-the-box” state”. It was agreed that 7i would be selected for the upgrade if it significantly provided the functionalities needed without any required customization and that other applications did not offer major advantages over 7i.

In addition, PubDiv had a balance of money as credit from Datastream and they would be using that credit towards the fees of upgrade if they decided to upgrade to 7i. This would ease the process of attracting funding for the project from Pub Org senior management. This was particularly important because the initial plan to hire consultants from the vendor to configure the software had been rejected by the top executives. The reason for this rejection was that top management did not approve of acquiring a new system for asset management while SAP R/3 which was already in the process of implementation could be used for the same purpose. Therefore, they did not approve the budget for a new system and it was finally decided to accept Datastream’s offer regarding the purchase of 7i.

This project first started at DIVA and DIVB immediately followed. Historically all four sub-divisions of PubDiv had been using different systems which also had different names such as work management system, work notification system, and work notification tracking. This project was approached as a first step to implement a unified system across all four sub-divisions of PubDiv and eventually across the whole organization. This was the vision of the director of PubDiv; hence, the project was welcomed by him. The management at DIVA initiated a project by requesting 7i to perform an evaluation of the existing system. This was deemed necessary because the business processes were not clearly defined. In addition, the business processes were not standardized even within DIVA and even more so between DIVA and DIVB. The vendor conducted 18 workshops, several meetings and interviews for over 50 PubDiv staff from senior managers to field supervisory staff. This investigation took 16 weeks, the result of which was a report, which contained the following nine sections:

- **Section 1: Introduction**

- **Section 2: Strategies** comprised the main strategies and approaches PubDiv would use to achieve the project goals.

- **Section 3: Organizational Roles, Responsibilities and Structure** contained information about how PubDiv would organize and assign responsibilities to implement the project and change long-term maintenance practice.

- **Section 4: Business Processes** documented the three major processes (planned maintenance and inspections; responding to unplanned system events and performing IT support) developed by PubDiv operations, maintenance and materials staff to be implemented with the upgrade of the current work management system.

- **Section 5: Technology** documented the IT systems and changes needed to support efficient implementation of strategies, organizational roles and new business processes.

- **Section 6:** Identified the data requirements associated with the long-term use and management of the work management and related systems.

- **Section 7: Project Plan** documented a three-year project plan to implement change.

- **Section 8: Cost Estimate and Value** documented the expected costs over the three-year project and PubDiv's need for the projects successful completion.

- **Section 9: Appendices** contained a glossary of terms, organizational charts, and role descriptions referenced in the main document.

Datastream 7i is made up of several modules but not all were planned to be deployed. Table 4.2 exhibits all the modules available by 7i. As discussed above, management was concerned about having a more effective system in place which included additional functionalities such as Asset Tracking, Inspection, Inventory Management, Preventative Maintenance, and Project Management. However, because of the budget constraints and the decision to configure the system in-house, it was decided to initially merely implement functionalities for work order management and asset management[6].

[6] Besides the base and administration modules

Table 4.2: Different Modules of the Datastream 7i Software (source: vendor report)

Module	Description
Base	1) Functions as the prerequisite for all other modules 2) Provides key security and administrative controls over business processes: password aging control, classification codes, equipment hierarchy setup, and various operating accesses and permissions
Administration	1) Functions as the prerequisite for all other modules 2) Provides system administration functions, such as security, system-wide defaults, and the content functions and forms
Asset Management	1) Identifies, tracks, locates and analyzes physical assets 2) Associates documents, permits and other data with assets 3) Establishes and controls interactive relationships between assets and their associated business processes and usage
Data Collection	1) Allows for the creation of specific data collection activities and their distribution to mobile devices 2) Remote devices can be used to capture and transmit data back to the DataStream 7i database for validation and processing
Barcode	1) Allows users to design and print labels for assets and capture data faster and more accurately
Workflow	1) Creates pre-configured workflow for basic operations 2) Provides the ability to track sequences of events of various operations, such as work order and purchase order generation to automatically alert the originator or selected approver of the action required 3) Users may access workflow information to view notifications of pending actions and status changes and respond to action items 4) Instant notification and responses via e-mail gives users the ability to receive updates regardless of time or location

The project did not start immediately after the vendor's initial investigation. It took approximately 3 years from this initial investigation to the actual purchase and consequently the start of the configuration process. In April 2006, a Team of around 15 with representatives from DIVA, DIVB, and the IT department was selected to configure the system. This team was called Work Management System Task Force (WMSTF[7]). The team spent almost a year to develop To-Be business processes. This activity was carried out in order to come up with common business processes based DIVA and DIVB's current (i.e. As-Is) business processes. Each sub-division had previously identified their As-Is business processes (DIVA with the help of Datastream). During the process of mapping To-Be business processes, the group looked at the history of the work management in DIVA and DIVB, and the actors and their roles.

After the To-Be business processes were mapped out, they started configuring Datastream 7i based on these processes, continuously referring to the flow chart[8]. During the process of mapping the To-Be system and the actual configuration process workers from the field were not involved in the process. The project leader once mentioned "*we don't have people from the fields directly but people here interact with those… People from our team communicate what we do here [about the configuration of 7i] to the field workers and bring their feedback to us*". The initial plan was to equip the workers from the field with computer access (preferably laptops) so that they would track/document every work order from request to finished job, using the software. At the time that the

---

[7] WMSTF stands for Work Management System Task Force, which referred to a team of representatives from two subdivisions within PubDiv, the division which was implementing the new software. This is discussed further in the background provided for the case later in the chapter and in data analysis in chapter six.

[8] The flow chart was printed in a large scale and hung in the simulation room.

researcher was collecting data, field workers did not interacted with the software. They still used the hard copy printed work orders in order to get the work done. Maintenance supervisors are the ones who print out the work orders and provide the field workers with them. These roles are discussed in detail in chapter six (section 6.3.1).

When the researcher started her research at Pub Org (August 2007) the actual configuration of 7i had only been in place for a few months. Some of the basic features had been already configured. At this point the team was able to run queries to simulate a simple and straightforward process of work orders[9] from beginning to the end. The WMSTF team would meet bi-weekly to run simulations, discuss various features, and decide on how to configure them. Initially it was decided that for the tasks that IT team was unable to perform, the team would consult with the vendor. Additionally, the vendor was supposed to conduct periodic monitoring as a part of their maintenance deal which came with the purchase of the software. The role of the vendor faded quickly. During the time that this study was being conducted at Pub Org vendor consultants were never present in the configuration sessions or meetings. During this time one major meeting wherein the configured system would be demonstrated to the vendor and the team would discuss the difficulties in configuration with them was scheduled. This meeting was postponed several times and finally cancelled.

In our first interview, the project champion stated that the goal was to start using the system by September 2007 as a pilot test. The pilot version of the system was put in place (starting with a limited number of plants in each subdivision) four months later than

[9] In chapter six, work order process both theoretically (as business processes) and as set up in 7i is discussed (sections 6.3.1.1 and 6.3.1.2).

expected. At the end of September a demo of the “almost-configured” software was presented to the director of PubDiv. He was satisfied with the progress[10]. The software was expected to be configured further after receiving feedback through the pilot test while running parallel to MP5. The Datastream 7i software was then planned to be fully deployed in each sub-division once the configuration was complete. However, senior management was not satisfied with the system being called Pilot and believed that too much time had been spent on the process. Therefore, once actual data, including work orders, started being populated in the pilot system the decision was made to “dump” MP5 and continue with the pilot system as the actual system. From this point on, the process was referred to as “Rollout” instead of “Pilot”.

The case of Pub Org presented an ideal case for this study as the organization was implementing a software package. The sources of data and methods of data collection at Pub Org are discussed in the next section (4.3.1.2.2).

4.3.1.2.2. Data Collection at Pub Org

Data collection at Pub Org expanded from August 2007 to April 2008. Data were collected from different sources, which are listed in Table 4.3. Extensive technical documentation and other relevant organizational material were obtained. Digital copies of these documents were provided to the researcher. Review of documents, archival records, and organizational charts were beneficial because data about policies and norms, services, and organizational hierarchies besides the technical characteristics of the

[10] Around the same time Datastream was bought by Infor and the software was called INFOR 8.2 from this point on. In this study we keep the name Datastream 7i or 7i for short because majority of the actors continued to refer to the software as 7i.

application could be collected. As the main part of data collection, the researcher attended, as a neutral observer, twenty simulation sessions (the entire time period allocated for the simulations) in which fictitious processes were run for the configuration team (WMSTF) to decide about the settings of the software. Each of these sessions lasted for about five hours. During the breaks in each session short informal semi-structured interviews with the attendants were conducted. The researcher also attended ten biweekly meetings (WMSTF meetings), each lasting for approximately two hours, after the simulation time period. Meeting minutes for another ten WMSTF meetings which the researcher was not able to attend were also provided to her by the team. In addition, meeting minutes for eighteen IT training sessions which were conducted by IT and one representative from each sub-division to address the questions and concerns of WMSTF members were also obtained (this is referred to as IT Training minutes in Table 4.3). In addition, 7 semi-structured interviews, each about two hours long, were conducted with the project champion, team leader, and managers from each sub-division. During both interviews and the observation sessions, the researcher decided not to audio/video-tape the conversations to avoid distractions and self-censorship of the subjects. The researcher was also provided with copies of email correspondences of the implementation team related to this project. These emails included message exchanges with the vendor.

All the data collection was guided by the theoretical framework of SCOT, and the research questions related to various elements of it. For example, these questions were used to direct the interviews. Every note, including notes from interviews and simulation sessions, were transcribed the same day that it was obtained. The notes were kept as dated word documents.

Table 4.3: Sources of data

Data collection method	Quantity	Output
Neutral Observations and short informal interviews	20 simulation sessions	105 pages of text
WMSTF Meetings	20 meetings	55 pages of text
IT Training Meetings	18 meetings	40 pages of text
Semi-structured interviews	7 meetings	20 pages of text
Organizational Documents	10+ documents	500+ pages of text
Email correspondences	Over 100 emails	200+ pages of text
Data collection timeline: August 2007 - April 2008		

Multiple methods of data collection were used for the purpose of triangulation (Yin, 1986; 1994). This way the researcher attempted to address the issue of the validity (Klein & Myers, 1999; Yin, 1993). Yin (1994, 2002) suggests following a chain of evidence. Having multiple sources of data also helps with identifying a chain of evidence. In order to reach triangulation, obtaining multiple perspectives is recommended besides using multiple sources of data (Denzin, 1978). In this regard, perspectives of all the people involved in the project were sought through observation and interviews.

4.3.2. Data Analysis: A Hermeneutics Approach

Boland (1991) defined Hermeneutics as follows:

> "Hermeneutics is the study of interpretation, especially the process of coming to understand a text. Hermeneutics emerged as a concern with interpreting ancient religious texts and has evolved to address the general problem of how we give meaning to what is unfamiliar and alien" (Boland, 1991, p. 429).

Gadamer (1975) argues that by being in the world, as social actors, we are constantly involved in the act of hermeneutics giving meaning to every reading or hearing of a text through interpretation (Taylor, 1971; Saunders, 1981; Ricoeur, 1988). Conducting an interpretive observation, a social scientist interprets the constructed social reality for what it means to various actors (Lee, 1994). In doing so, the researcher gains an understanding of the way human subjects understand themselves and their surroundings. This kind of understanding can be gained through different approaches, namely, anthropological ethnography, participant-observation, and hermeneutics (Ibid). Social action can be comprehended as text (Ricoeur, 1981). Therefore, hermeneutics has been used extensively in social sciences to understand social phenomena (Butler, 1998).

Within the tradition of Interpretivism, Hermeneutics is a legitimate and common approach to analyze qualitative data (Boland, 1985; 1991). Through hermeneutics technique, the researcher finds meaningful insights from the text that correspond to the theoretical lens he is using. The term "text" can mean anything of a textual nature (e.g. activities of human actors) and not solely text-based documentation. Lee (1994) suggests that researchers have extended the notion of text to include "not just the documentary artifacts that human subjects create, but also their individual actions, group behaviors, and even social institutions, all of which, as text analogues, have meanings that can be read and interpreted" (p. 148).

In the context of this dissertation then the application of hermeneutics approach to analyzing the date was deemed appropriate since the goal was to find meanings in the textual data. The researcher intended to draw insights from the text which revealed how the subjects of the study interpreted the information systems they were dealing with and how these interpretations influenced the actions they took towards them. This approach seemed appropriate because the main purpose was to investigate the interpretation and sense making, which are highly contextual (Miles and Huberman, 1994; Yin, 2002).

In understanding the social action through hermeneutics, Butler (1998) describes the process as follows:

> "Social action, like a text, is a meaningful entity that must be construed as a 'whole'; however, an understanding of the 'whole' begins with an interpretive examination of its constituent 'parts' this again introduces the concept of the circle of understanding" (p. 291).

Throughout a hermeneutic process, the pieces of text that make sense to the researcher eventually evolve to a meaningful whole. Therefore, the researcher goes back and forth between the data and the theory. In other words, having a theory as a lens, the researcher finds pieces of text that correspond to the theory. The researcher then goes back to the theory and examines the pieces based on the theory. The iterative process of going back and forth between the text (i.e. data) and the theorizing finishes when the researcher makes sense of the phenomenon as a whole. At this point the “hermeneutic circle” is said to be closed (Taylor, 1971; Boland, 1985; 1991; Klein & Myers, 1999).

As discussed in chapter six, the approach of going back and forth between the ‘part’ and the ‘whole’ was taken when analyzing the data of this dissertation. In the case of Pub Org, this process resulted in development of a micro and macro mechanisms of the configuration process (presented in chapter seven); the ‘whole’ of the phenomenon. Development of these mechanisms started with analyzing the data through the lens of SCOT and identifying patterns that corresponded to the theory. During this process, for example, certain controversies that occurred during the configuration process of 7i were looked at, the analyzing of which revealed some themes regarding the mechanism. During the whole process, as each ‘part’ of the mechanism was being developed, the ‘whole’ picture of the configuration process was also taken into account. This process of going back and forth between the configuration mechanism and the data, and the theoretical themes continued till theoretical saturation was reached (Yin, 2002).

The philosophical approach of hermeneutics has been acknowledged as relevant to studying information systems in the IS field. Boland, for example, is one of the prominent researchers in taking this approach to studying IS (Boland, 1979; 1985; Boland & Day, 1989). He argued that study, and even use and design of IS is best understood as a hermeneutic process (Boland, 1985). He argues that the output of the use of information systems is text and is thus studied and interpreted by others in a hermeneutic approach. He believes that designing IS is also a hermeneutic task, in which designers read and interpret the organization and organizational actors. In line with the use and design of IS, Boland argues that studying information systems thus needs to be approached as a hermeneutic task. In this process, he believes that the researcher makes sense of the interactions during the use and design of IS in order to "interpret the significance and potential meaning they hold" (Boland, 1985).

Therefore, in both cases (Child Org and Pub Org) the hermeneutics approach was used to analyze the data since it is a legitimate approach; it corresponded well to the qualitative textual data of this dissertation; it offered an appropriate technique to extract themes related to the phenomenon of this study; it allowed for seeing the relationship of the parts and the whole; and it helped with theoretical generalization. More details about specific steps taken to analyze the data from Child Org and Pub Org cases are, respectively, presented in chapters five and six.

Chapter 5

Pilot Study -- Child Org Case Study

5.1. Introduction

As discussed in the case background in chapter four, the objective of the Child Org case study was to investigate employees' interpretations of CDT and CC Solutions, two of the main information systems at Child Org. Child Org was in the process of searching for an application that would integrate the functionalities of CDT and CC Solution. The need for this integrated solution stemmed from the need for extracting integrated information from CDT and Solutions. In addition, this need partly stemmed from dissatisfaction with the functionalities of CDT and CC solutions.

Therefore, the goal of the study at Child Org was to find the discrepancies in interviewees' interpretations in order to understand the reasons for dissatisfactions of employees with CDT and Solutions. Understanding the interviewees' interpretations of a software package that would replace CDT and Solutions was also of interest. Initially, Technological Frames of Reference (Orlikowski & Gash, 1994) was applied as a theoretical lens. Technological Frames of Reference (TFR) seemed promising because it investigates how people make sense of information systems. As discussed in chapter two (2.3.2.1.), one of the main premises of TFR is that members of the same group share the same frame of reference towards a technology. TFR also suggests that when in an organization frame incongruence among stakeholder groups exists difficulties in implementation and/or use of IS could arise. Therefore, Orlikowksi and Gash suggest that the incongruence of frames should be identified and decreased in order to address the conflicts.

Therefore, TFR was adopted as a theoretical framework in order to identify the congruence/incongruence of frames aiming at explaining the dissatisfaction with CDT and Solutions as well as the need for their integration. The initial analysis of the data proved to be challenging since an incongruence of frames of reference within groups were observed. This was contrary to the premise of TFR regarding sharing of frame of reference by individuals within a group.

Upon further investigation though, SCOT seemed more promising as a theoretical lens. Therefore, the case was re-analyzed adopting the lens of SCOT. This re-analysis of the same data was deemed appropriate since SCOT and TFR both aim at finding interpretations of groups. Therefore, had SCOT been adopted from the beginning of the study, the same process of data collection would have been followed. In other words, in a semi-structured approach, the interviewees would have been asked about their experiences and interpretations about CDT and CC Solutions as well as their perspectives about the need for an integrated system. Therefore, SCOT deemed to be appropriate in analyzing the already-collected data at Child Org.

In this chapter, first a discussion of the original analysis of data through the TFR lens is provided. Then, the re-analysis of the case through the lens of SCOT is presented. The chapter concludes by summarizing the lessons learned from this case study. These lessons not only supported the choice of SCOT as the driving theoretical lens throughout this dissertation but also shed more light on how to apply SCOT in the context of information systems research.

5.2. First Data Analysis of the Child Org Case

During the interviews, employees' concerns regarding a new integrated system and dissatisfaction with CDT and Solutions were repeatedly brought up. The search for a new system had already begun when the study started. The team in charge, which included IT personnel, managers, and representatives of clinicians, was in the process of viewing demos for different systems. Fifteen interviews were conducted from October 2005 to March 2006 with members of this team. The interviewees belonged to two groups: Support group, and Users group. The support group included the individuals who provided IT and managerial support for the use of CDT and CC Solutions. The Users group consisted of clinicians who were the direct users of CDT and CC Solutions (i.e. clinicians). Applying TFR as the theoretical lens, the interpretations of the interviewees regarding the use of CDT and CC solutions as well as the requirements for a new software package that would integrate the functionalities of these two were solicited.

Adhering to the theoretical lens of TFR, when analyzing the data those pieces of textual data that corresponded to the domains of TFR were highlighted. This was done separately for each group (Support and Users). Two of the domains of TFR, Nature of Technology and Technology in Use, were prevalent in the data. In addition, frame incongruence was present not only between but also within the groups. This observation was puzzling since one of the premises of TFR is that members of the same group share a technological frame of reference. The following section discusses the technological frames of Support and Users groups in the light of the two dimensions of TFR.

5.2.1. Technological Frames of Reference

As discussed in chapter two (section 2.3.2.1) technological frames characterize interpretations of different groups around the technology. According to the theory, members of each group share similar interpretations around the technology on specific domains (e.g. Nature of Technology), which reflects congruence of technological frames within a group. However, frame incongruence might exist among the frames of reference of various groups. Orlikowski and Gash (1994) found that when incongruence in technological frames exists across different key groups[1], conflict and difficulty can arise.

In this section, interpretations of Support and Users group are identified in terms of the ways the members of these groups make sense of the functionalities of CDT and CC Solutions (i.e. Nature of Technology), and the day-to-day use of these systems (i.e. Technology in Use). At the end of this section the status of frame congruence/incongruence within and between the groups are examined discussed.

5.2.1.1. Nature of Technology

As discussed in chapter two (section 2.3.2.1), Nature of Technology reflects people's understanding of functionalities and capabilities of the technology. When examining the data with the theoretical lens of TFR, those pieces of textual data that reflected interpretations of people about functionalities of CDT and Solutions were highlighted. These attributes were studied separately for each group (Support and Users), which are discussed below.

[1] For example, users, developer, and managers

5.2.1.1.1 Nature of technology: Support Group

The general belief of the support group was that CDT and Solutions have shortcomings in terms of their capabilities. However, they believed that these systems did a reasonable job in storing clients' information. In other words, this group mainly viewed CDT and Solutions as systems that offered simple functionalities. This is evidenced by the quotations below:

"It [Solutions] is word processing type of work, mostly typing to store and create reports on treatments".

"It [CDT] is a sophisticated system in terms of storing data. It [CDT] also generates reasonable reports … but there is just so much that the system can do to create good reports. It does not satisfy everyone. It works good for creating reports that satisfies general need of all departments but it is not capable of creating custom reports for each department".

Members of this group believed that not any system, though, could satisfy everyone because both systems have users from different departments and with different needs:

"Best what's good for everyone? … There never is that sophisticated enough in the world of computers [to fulfill everyone's needs 100%]".

"It's never going to be a simple system".

Overall, most members of the support group viewed both systems mainly as database systems, which were fairly good in storing information of clients, not offering sophisticated tools for reporting and analysis of data. Dissatisfaction of support group with CDT and CC solutions were also echoed in expressing a concern for the need of software that would integrate the functionalities of these two systems. Therefore, frame congruence in terms of Nature of Technology was present within the support group.

5.2.1.1.2. Nature of technology: Users Group

Similar to the members of support group, users of both CDT and Solutions viewed the systems falling short in terms of features. Most of the members of this group also believed that the systems were fairly good database systems for storing data where not offering many tools for analysis of data and reporting. One of CDT users, for instance, stated that he did not find the system useful because it did not provide the capability of drawing charts and graphs, which were greatly needed in his reports.

"What I've been looking for is charting and graph capabilities … reporting [in CDT] is very basic … I like graphs … These are data, number driven … more interactive charting is needed … they management] like graphs, they don't want to read".

In another comment, the researcher was told that both CDT and Solutions work efficiently as databases but lacked features when it comes to analyzing data. For example, the quote below reflects an interpretation of viewing the systems as data storing systems but lacking in capabilities for analysis and output.

"Both systems are efficient in the sense of giving data to them [but then in the case of output they are not]".

On the whole, the users expressed a major concern regarding the need for several features that would make the systems more effective and would address their needs in a better manner. Some of these concerns are echoed in the comments below.

"I get the information from CDT or Solutions and analyze it and make them look better with chart and graph and give it to the managers … so if CDT and Solutions had that feature, it would be nice … it would ease up things"

"CC Solutions has reporting tools too but most of the fields that I would like to use to get the report to create cheat sheets are not available to me … I create my own cheat sheets"

As it was observed, most users viewed CDT and CC solutions as inefficient systems. They believed that several functionalities needed to be added to both systems. Even though, the predominant interpretation of the nature of CDT and CC Solutions was inefficiency of these systems, some individuals believed that the systems were more capable than perceived by the majority of users. These individuals believed that both systems offered a lot of features but were not utilized by users. This interpretation was reflected in comments such as the ones below.

"Once we were able to work with the new system [CDT], we saw that there was a whole lot of benefits to it".

"There are many tools available that we don't use, we don't know how to use … we have a big packet of manuals [but do not use them]".

Therefore, within the users group incongruence of technological frame was observed in terms of interpretations around the nature of CDT and Solutions.

5.2.1.2. Technology in Use

Chapter two (section 2.3.2.1) defined Technology in Use as people's perception of the ways the technology works on a day-to-day basis and the conditions and consequences associated with this use. During the analysis, data that corresponded to the experiences of the two groups of employees (Support and Users) with the daily use of CDT and Solutions were highlighted. Sections 5.2.1.2.1 and 5.2.1.2.2 provide discussions of Technology in Use for these two groups.

5.2.1.2.1. Technology in Use: Support Group

Members of support group generally believed that CDT and CC Solutions were easy to use. Overall, the interpretation of this group was that the systems did not demand much from the users. They characterized the ease of use of the systems by "simple search and selection" process. This view is echoed in the comments below.

"It is just a few clicks. All they need to do is to narrow down their search and get what they need and then print it".

"They also have easy access to drop down menus to see what each code represents".

Additionally, support group members deemed the systems as becoming easier to use the more the users interacted with them. For example, the comment below about CDT reflects the belief that the more users have used CDT the more they have become familiar with it.

"People are more and more seeing this new CDT system friendly … to use the case notes all you have to do is to click".

Therefore, the overall impression of support group about CDT and CC Solutions in terms of the nature of their use was that they were extremely easy to use systems. Interestingly though, some individuals from support group sympathized with the users in believing that CDT and Solutions were indeed cumbersome to use for the users. The comments below regarding this interpretation were expressed by members of support group.

"Things take longer to do in the new system ... have to go through a lot of screens ... taking time to do things".

"A lot of clicks, menu to menu, page to page".

As can be seen, within support group's TFR, internal inconsistency existed; part of this group believed that the systems were easy to use and the other part believed that systems were cumbersome.

5.2.1.2.2. Technology in Use: Users Group

Most of the users complained that CDT and CC Solutions involved unnecessary steps to perform different tasks. Users felt that the system made their work more burdensome. The comments below reflect this concern about CDT and CC Solutions.

"I have made a lot of suggestions for the system [CDT] ... It would be really nice if they made the system simpler and got rid of the unnecessary things ... Anything that makes it simpler is greatly appreciated".

"Some stuff you can't access [in CC Solutions] (e.g. discharged clients profiles) after 30 days your access is terminated and then when you try to retrieve them the message is 'record not found' instead of saying something like 'you don't have access'".

Dissatisfaction with CDT and CC Solution's use was not limited to their difficulty of use. Some members found other attributes that resulted in their distaste for the systems. For example, one user simply did not find CDT visually appealing. Another user encountered difficulties, because of missing information, when using Solutions. The quotations below echo these concerns.

"I don't know, CDT is just not visually appealing … not interactive … Solutions is more user friendly) that one is not great either".

"Migrating of Solutions has been a nightmare … there are missing fields … for example the ID field (this is something I complain about a lot) … the problems, we don't know about them until we encounter them so it is kind of like trouble shooting".

As can be seen, most users found CDT and Solutions to be cumbersome to use; this resulted in an overall distaste for both systems in terms of their nature of use. Notwithstanding the prevalence of highly similar perceptions and understandings of the systems regarding their difficulty of use, some users expressed a quite different interpretation in this regard. For example, as it is reflected in the quotation below, the user finds CDT to be "very user friendly"; opposite to the common interpretation within users group.

"New CDT system is window-based and is a lot easier than the old ECHO system … There was easy transition … simple and good training … good to use … just using mouse … very user friendly" .

Therefore, technological frame discrepancy was present within users group in regards to users' interpretations of the use of CDT and CC Solutions. Table 5.1 summarizes the discussion in this section and exhibits the status of congruence/incongruence of technological frames within each group based on each element of technological frames of reference framework. As can be seen, incongruence of technological frames of reference is highly prevalent within groups, not in line with the TFR theoretical framework.

Table 5.1: Congruence/Incongruence within support and users groups

TFR element	Group	Congruence
Nature of Technology	Support	Yes
	User	No
Technology in Use	Support	No
	User	No

5.2.2. Frame Discrepancies and the Shawn Anomaly

The theory of technological frames of reference predicts within-group congruence and across-group incongruence of frames of reference (e.g. Orlikowski & Gash, 1994; Lin & Silva, 2005). According to this theoretical framework, organizational roles of individuals place them in specific groups. For example, in the case of Orlikowski and Gash's (1994) research the subjects of the study were categorized into groups of technologists, managers, and users; this was based on the organizational roles they performed. A specific technological frame of reference was shared within each group. However, different groups exhibited frame incongruence compared to one another. For example, technologists viewed the Lotus Notes groupware (that Orlikowksi and Gash were studying) to be extremely easy to learn on one's own but the users felt they had received inadequate training.

The findings of Child Org's case, however, proved to be inconsistent with predictions of TFR. As it was discussed in the previous section, frame incongruence was observed even within each group. In addition, to this inconsistency another anomaly was observed when analyzing Child Org's data. The researcher called this the Shawn anomaly. Shawn is the name of an individual who belonged to the support group in terms of his organizational role. However, he was one of those individuals who did not share either group's predominant interpretations regarding CDT and CC Solutions. In addition, Shawn was different from members of either group in terms of his familiarity with the systems. Technically, Shawn understood the system at the level of support group's knowledge of the systems. In terms of the business processes though, Shawn was highly familiar with the processes of CDT and Solutions. This technical as well business processes knowledge qualified Shawn in being able to assist the users with CDT and Solutions. This assistance was highly sought by the users. His assistance benefited the users the most especially when the facilitation was related to the integration of the two systems.

The researcher was told by many interviewees (from both groups) that Shawn was referred to frequently. For example, quotations below reflect that Shawn is highly sought after by the users.

"I normally talk to Shawn on the phone and explain things … we open the system at the same time and walk him through so that he can help me … and sometimes when we can't figure it out he says let me come down".

"If there is anything I don't know or can't figure out, I call Shawn … I am a little meticulous, so I want to make sure I do things right".

"Of course, I think I knew exactly what I want [and asked for specific things] because I had a table from before and I knew the names, the conventions … Maybe if I didn't have it I had to sit with Shawn and ask what they call things".

Shawn was overall sympathetic to the users; therefore, he assisted the users whenever he was referred to. This level of support was not a part of Shawn's job responsibility. Nevertheless, he provided the users the assistance they needed. Shawn was mindful of the need for the integration of the system. He also acknowledged that both systems could be confusing for users with minimal knowledge of the systems as well as the business processes. The former attribute of Shawn placed his interpretation of the nature of CDT and CC solutions within that of users group. However, the latter attribute made his interpretation quite different than either of the groups. This stemmed from the fact that importance of the familiarity with business processes in relation to the use of systems was not explicitly acknowledged by any other individual. Therefore, Shawn was sympathetic towards the users. This situation is called "Shawn Anomaly" by the researcher since TFR did not offer any explanation of it.

TFR also came short in explaining the incongruence of frames within the support and users groups. The anomalies here (Shawn, and within-group frame incongruence) pertained to the misfit of individuals within groups. Therefore, the researcher looked into other theories that could provide an explanation for the inconsistencies of the findings of Child Org case and TFR. Social construction of technology, which was discussed in chapter three, seemed promising for several reasons. SCOT encompasses technological frames of reference in addition to various other elements, which together explain interactions with and further development of the technology. Therefore, re-analysis of the data through the lens of SCOT was deemed appropriate. Furthermore, SCOT focuses on discrepancies and then identifies groups around the technology based on those interpretations. Therefore, SCOT seemed to offer coherence in the analysis of data compared to TFR. Re-analysis of Child Org's case in the light of SCOT is discussed in the next section.

5.3. Discovering SCOT: Explaining the Anomalies

One of the main premises of SCOT is that various social relevant groups are identified based on their interpretations of the technology. This is dramatically different from the traditional way of categorizing groups in IS studies based on organizational roles of individuals, which was also initially used in the study of Child Org. The researcher conducted a second data analysis on the data from Child Org applying SCOT as the theoretical lens.

The data was re-analyzed using the initial coding that was based on the elements of TFR. This made sense since these codes represented several problems around the technologies in place, CDT and CC Solutions. Data was categorized around the three main challenges related to CDT and Solutions that had come up in the first round of data analysis: *Update*, *Use*, and *Integration*. Update refers to the issue of upgrade of CDT and CC Solutions that had happened in the past. Use, refers to the direct experience with the systems on a daily basis. Integration refers to the need for finding software that integrated the functionalities of CDT and CC Solutions. These three categories are related to the aspects of the technology that were open to interpretation; in-line with SCOT approach.

The goal was to explain these three challenges through interpretations of the interviewees. In addition, one objective was to examine how an integrated solution was evolved to address the need to integrate CDT and CC Solution. In the current form, support group and specifically Shawn addressed most of the integration needs. However, the concern over the need to find an integrated software was brought up in most of the interviews. In addition, the interviewees' experiences with the past upgrades of CDT and CC solutions seemed to hover in most of the discussions the researcher had with the interviewees. This was an interesting observation since the topic of CDT and CC Solutions update was not initiated by the researcher. In other words, this topic was an emergent one which did not initially seem to be directly related to what the researcher was investigating. The experience with the use of CDT and CC Solutions was discussed in the first round of data analysis through the lens of TFR.

Therefore, in the second round of analysis through SCOT, the findings of the first round were organized to represent the *Use* of the system. However, this time, the researcher refrained from grouping the interviewees based on their organizational roles and then assigning an overall interpretation to each group. Instead, she found different categories of the interpretations of interviewees based on the two dimensions of TFR which had emerged in the first round of data analysis: Nature of Technology; and Technology in Use. These categories, as well as quotations from the data which reflect each category are shown in Table 5.3.

In addition, various interpretations around the update of CDT and CC Solutions were also identified to reflect different views about what interviewees experienced during the upgrade of CDT and CC Solutions. These interpretations along with excerpts from the data are exhibited in Table 5.2. Moreover, in order to capture the perspectives of the interviewees about an integrated solution, various interpretations were identified. These interpretations along with quotes that echo them are presented in Table 5.4. The findings pertaining to these three categories that are demonstrated in tables 5.2, 5.3, and 5.4 are explained in detail below.

5.3.1. Update

As it was discussed in chapter four, CDT and CC solutions had both gone through version updates in the past, respectively upgrading them from a DOS-based to window-based, and from a window-based to internet-based versions. The upgrade of each of these systems to the newer versions was perceived differently (in terms of difficulty) by different people. Interpretations were both in favor of and against update of CDT and CC Solution. For example, as it can be seen in Table 5.2, some individuals believed that the update of CDT went smoothly; whereas others believed that it was difficult and time consuming. In both cases (i.e. CDT and CC Solutions), negative interpretations were more prevalent. In this case the interpretations were influenced by roles of the employees.

One reason that investigation of interpretive flexibility around the issue of update was important was that employees' experiences from the updates influenced their views about the new system; i.e. integration of CDT and CC solutions. In the interviews, the researcher asked the subjects to elaborate on their perspectives about the need for the new system. The significance of the topic of update was substantiated by the repeated references made to the upgrades of CDT and Solutions during the interviews. Since almost every employee had had an unpleasant experience with either CDT or CC solution's updates, there was an almost unanimous apprehension towards adopting a new system that replaced CDT and CC Solution. This concern was reflected in comments such as: "We've been burnt out so much [that's why] we have searched a lot for a good solution".

5.3.2. Use

It was also important to look into the use of CDT and CC Solutions since their use directly influenced various interpretations over the nature of the system that would replace them. The predominant interpretation was that both systems were used as databases. In fact most of the employees referred to CDT and CC solutions as "database systems" or "storing systems". Some individuals believed that CDT and Solutions had more capabilities but were not used to their full potential. Therefore, in a new integrated system, employees desired "sophisticated" tools that would not only allow them to store data but also add data analysis and more importantly customized reporting capabilities.

When it came to use of the systems, there were two predominant beliefs: systems were difficult to operate; systems were easy to operate (Table 5.3). The first belief led the employees to rely on others (Shawn in particular) when they had difficulty using the systems. This reliance was more prevalent when there was a need to retrieve integrated data from both systems. Some individuals in this group acknowledged that by investing time users would have been able to learn the systems. Some individuals with the second belief were more sympathetic towards the users. These individuals acknowledged that the systems could be cumbersome to use. Other individuals with the second belief were not as sympathetic. This group believed that users simply did not invest time in learning the systems and it was only users' perception that systems were difficult to operate.

Various interpretations resulted in different actions. Those who believed the systems were difficult and that they had the right to ask for help went to Shawn (or IT) as soon as they had a problem, specifically those deemed "technical" by the users. Shawn specially felt sympathetic towards users; thus offered them assistance. The group who believed CDT and CC solutions were both easy to use and were not sympathetic towards the users blamed the users for not taking full advantage of the systems.

5.3.3. Integration

In the initial data analysis an emergent theme which was related to expectation of support and users groups from one another seemed to have offered some explanations for discrepancy of frames within groups. In the light of SCOT, this theme was in fact deemed to work as a closure mechanism. When users were not able to figure out how to accomplish tasks in either system and especially when the task required integration of both systems, they would call Shawn. In other words, a temporary solution for the integration problem was to ask Shawn for assistance. Overall, most groups (based on different interpretations about the systems, presented in Table 5.4) acknowledged the significance of the role of Shawn when there was a need for integration. Interestingly, this belief inspired the solution that finally was implemented to address the problem of integration.

Eventually, as had been planned, new software to replace CDT and CC Solution was not purchased. It was finally decided to build an interface between CDT and CC Solutions (a program equivalent of Shawn's role in the organization!) through in-house staff. Discussion of various interpretations about use as well as the update could partly explain this choice of closure. As it was repeatedly mentioned, most employees did not have a positive experience with the update of the systems. There were even comments which reflected the concern of not repeating the same mistakes if in fact a new integrated system was going to implemented. In addition, regarding use of the systems a lot of the comments revealed that both CDT and CC Solutions had more capability than perceived by most of the employees. Comments were made regarding the importance of investing time in learning the systems to not only overcome the difficulty of the use of systems but also learn how to use more sophisticated features of the systems.

Considering that the final solution which resulted in writing an interface between CDT and CC Solutions through in-house IT staff came two years after the initial study, the users had been more acquainted with the systems and the capabilities they could offer. In addition, since by writing an interface employees would still use the old systems (i.e. no need to get accustomed to a new system) this solution seemed to resolve the need for integration in the best of the interest of Child Org. In other words, it seemed that the final solution reflected a Shawn-equivalent which eliminate the need for adopting a new system, what most interviewees feared based on their experiences with update of CDT and CC Solutions. In addition, this solution addressed Shawn's frustration over losing a significant amount of time every day to assist the users with CDT, CC Solutions, and integration of the two.

5.4. Conclusion

The goal of conducting the case of Child Org was to become familiar with the processes of sense-making around IS in organizations. As it was discussed in the analysis of the data the initial choice of TFR as the theoretical framework proved to be insufficient in explaining the situation at Child Org, the incongruence of within-group technological frames. This theory would also not offer an explanation of the choice of final solution; i.e. bridging CDT and CC Solutions. However, when the data was re-analyzed through SCOT, the discrepancies around CDT and CC Solutions were not of concern any more. SCOT focuses on discrepancies and aims at explaining them in relation to the technological artifact itself. Therefore, the researcher was not only able to explain the prevalence of incongruence of frames but also the closure of the *Integration* controversy.

One lesson learned from the case of Child Org was that it was empirically observed that indeed in organizations various interpretations are usually formed around information systems. The flexibility of these interpretations are more prevalent when there are controversies (or problems) related to the information system. Another lesson was related to the categorization of individuals into various groups. As it was shown in this chapter, during the initial data analysis categorizing the individuals who were relevant to CDT and CC Solutions problem proved to result in inconsistencies with the theoretical lens of TFR. In this initial analysis the traditional approach was taken and individuals were grouped based on their roles within the organization. However, in the second round of analysis, SCOT's suggestion was taken and interpretive flexibility was used as a criterion for grouping. This approach helped explain the discrepancies in technological frames. Finally, from this case, it was learned that interpretive flexibility influenced the course of actions that was taken to address the problem of CDT and CC Solutions. In this case, the interpretation that found CDT and CC Solutions to be easy systems yet favored integration of the systems seemed to have influenced the final solution, building an interface between the two system.

In sum, the choice of SCOT to study how interpretations about an information system are formed and how those interpretations influence addressing controversies around the IS was substantiated by the lessons that were learned through the pilot case study. In the next chapter the main case study of this dissertation applying the theoretical lens of SCOT is discussed. In analyzing this case, the researcher investigated the process of the configuration of packaged software through the lens of SCOT.

Table 5.2: Aspects of CDT and CC Solutions Open to Interpretations- Update

Aspect of IS open to Interpretation	Interpretive Flexibility	Quotation
Update	***Difficult, Time consuming, "Nightmare", still dealing with issues related to the upgrades***	*"In-house staff took care of the migration [update], that's why it took a long time because they had to do their day-to-day job plus taking care of the transition … Next time we are going to have a project manager"* *"It we were to do something like that now, we would hire a project manager … For this new system we want to have a project manager"* *"Migrations [update] went terribly, especially CDT. In the case of CDT, we did not devote enough resources … Converting the old data to the new one was unnecessary … Too many people made decision rather than coming to consensus … vendor wasn't ready … talked to vendor to work … The problem was that in the beginning the system was slow … users could work, hit save button, go to lunch and come back before the work was saved ☺ … we asked them to speed it up"* *"There is always resistance to change … people live in their own comfort zone … they wanted us to make new CDT system like the old system … the codes, etc. they wanted to see it in the new CDT … but then later they see the mistakes and they realize that for example if they hadn't done this or that [with their resistance], we would've been ahead … they realize[d] that if they try[ied] to work together, things will [would have been] smoother"* *"we needed a project manager for the migration [update]… this would have prevented pointing fingers"* *"[For update, there were] a lot of administrative work… might have been a good idea to work parallel [old system and new system] for some time… [there was also] a disconnect between training and use"* *"Migrating from CC kids to CC Solutions has been a nightmare… there are missing fields … for example the ID field (this is something I complain about a lot) … the problems, we don't know about them until we encounter them so it is kind of like trouble shooting"* *"There were some bumps when migrating from CC kids to Solutions … Some*

Table 5.2 -- Continued

		pieces of information was lost …. They were not migrated to the new system" *"Not that I liked CC kids but this new version has some weird things about it … For example for tracking a client (e.g. a foster child to track back see where he/she came from and which places he/she went) you have to go through multiple steps which are unnecessary [I think] … Migration [update] didn't go as smooth as they thought"* *"Solutions after migration [update] is frustrating"* *"After migration[update] from CC kids to CC Solutions, clinicians find the database very difficult to use efficiently, some of the problems are not related to the database [though]… they are related to little things… [for example], 20-30 clinicians staff: not all of them have upgraded word processing* *"It was a lot of work, It's good that they upgraded, but it was a lot of work … we had to do a lot of cut and paste … a lot of stuff didn't integrate … the new system has a lot of steps … a lot of click boxes … it's very time consuming"* *"The new system used to be so incredibly slow in the beginning, that problem has been fixed now"* *"The migration [update] went parallel … there was a lot of cut/pasting involved, it was kind of like doing things from scratch … we had to have both systems open and cut and paste [the things that didn't migrate] from the old system"* *"So because of the difficult migration [update] from CC Kids to CC Solutions, I dread any other migration [update]"*
	Needed, "Smooth"	*"Throughout my [time here], I have gone through two migrations: one from CC Kids to Solutions and one from old Echo to CDT … the second migration [update] went better than that of Solutions"* *"Solutions' migration [update] went more smoothly"* *"CC solution's migration [update] rolled out better than CDT's … it was slow though in the beginning"* *"CC kids(or CC Solutions) went far better, Ms. Gonzalez managed it well … this new system helps them do the job better, it is internet based"*

Table 5.3: Aspects of CDT and CC Solutions Open to Interpretations- Use

Aspect of IS open to Interpretation	Interpretive Flexibility		Quotation
Use	NATURE of TECHNOLOGY	***Database system, Simple, Storing system***	*"It [Solutions] is word processing type of work, mostly typing to store and create reports on treatments"* *"It [CDT] is a sophisticated system in terms of storing data. It [CDT] also generates reasonable reports … but there is just so much that the system can do to create good reports. It does not satisfy everyone. It works good for creating reports that satisfies general need of all departments but it is not capable of creating custom reports for each department"* *"What I've been looking for is charting and graph capabilities … reporting [in CDT] is very basic … I like graphs … These are data, number driven … more interactive charting is needed … they [management] like graphs, they don't want to read".* *"Both systems are efficient in the sense of giving data to them [but then in the case of output they are not]"* *"I get the information from CDT or Solutions and analyze it and make them look better with chart and graph and give it to the managers … so if CDT and Solutions had that feature, it would be nice … it would ease up things"* *"CDT does not have capability for the clients [clients' parents] to enter information directly to the system, this feature is needed, this way data entry errors and time would be cut"* *"Another thing we need in both CDT and Solutions is ticklers. For example, there are things that have to be done every 90 days and there should be reminders for those … now there is a lot of manual tracking in the systems"* *"The system[CC Solution] has reporting tools too but most of the fields that I would like to use to get the report to create cheat sheets are not available to me … I create my own cheat sheets"*
Use	NATURE of TECHNOLOGY	***Variety of capabilities***	*"They are good systems … I just tend not to use them"*

Table 5.3 -- Continued

			"There are many tools available that we don't use, we don't know how to use … we have a big packet of manuals [but do not use them]"
Use	TECHNOLOGY in USE	***Difficult to navigate, Too many menus, Too many point and clicks, not visually appealing***	*"the new Solutions is more cumbersome to work with, tings take longer to do in the new system … have to go through a lot of screens … taking time to do things …"* *"Things take longer to do in the new system … have to go through a lot of screens … taking time to do things"* *"A lot of clicks, menu to menu, page to page"* *"[In CDT] for some things you have to go through a lot of steps which are unnecessary, you should be able to get to the data that you want directly"* *"I have made a lot of suggestions for the system … It would be really nice if they made the system simpler and got rid of the unnecessary things … Anything that makes it simpler is greatly appreciated"* *"Some stuff you can't access [in Solutions] (e.g. discharged clients profiles) after 30 days your access is terminated and then when you try to retrieve them the message is 'record not found' instead of saying something like 'you don't have access'"* *"I don't know, CDT is just not visually appealing … not interactive … Solutions is more user friendly) [of course,] that one is not great either… but [in CDT] no logical flow … Solutions more logical, I think … diversity of fields is better in Solutions""* *"One person had a problem with a field that contained the number that was assigned to each family, this field had not migrated and every time she needs that she has to go and look up that number from the old version while she could maybe once print all the numbers for all families (about 175) and just have her secretary re-enter those numbers in the new version. So this new version does not seem as intuitive as the older one to them but some of them do not want to make things easier and they just complain a lot. We've had two rounds of training and have created a very comprehensive manual also but still"* *"One problem now is that you have to know what type of note you are looking for a specific child before you can find it. There is not any option that you can*

Table 5.3 -- Continued

			search for that child and get the list of all reports and then choose what you want. You have to basically know what you want and are looking for… This is cumbersome for the supervisors" *"[In CC Solutions], It is difficult to get information out of it… you input the information to it but then it's really hard to get what you want out of it… you input and then when it comes to aggregating that info and reporting, it doesn't give you usable information … it doesn't have robust reporting system"* *"CDT is very slow … frustrating … For example sometimes I need to get just a small piece of information about a client and pass it to the Accounting people … I think those people in accounting should be able to access it themselves then that would ease up some time on me … Also for getting a small piece of information, why do I need to go through so many steps? … it is frustrating for me"*
Use	TECHNOLOGY in USE	***Easy to use, The more practice the easier it has become***	*"They also have easy access to drop down menus to see what each code represents"* *"It is just a few clicks. All they need to do is to narrow down their search and get what they need and then print it"* *"People are more and more seeing this new CDT system friendly … to use the case notes all you have to do is to click"* *"The new windows-based CDT is not as intimidating as the old one, much easier to train"* *"Once we were able to work with the new system [CDT], we saw that there was a whole lot of benefits to it… my coworkers also find it simple to follow …"* *"One problem is that they do not take the time to sit down and see what they want and tell us what they want us to put in the report for them … I cannot get them to tell me what they want to see on the reports … when I ask them they tell me they don't know … they ask me to tell them what there should be on the reports"* *"It is difficult for them to figure out how the search engine works, which is fairly easy … When looking at a bunch of menus and rows and rows of data, they feel uncomfortable working with the system"*

Table 5.3 -- Continued

			"No matter how many times you tell them, [they still forget] … They are like you have the ability (and systems have the ability) … why take the extra step … why not system does it" *"There were some issues associated with this change → not every field migrated with this new version and some users cannot handle that, even now after about 7 months… For example, one person had a problem with a field that contained the number that was assigned to each family, this field had not migrated and every time she needs that she has to go and look up that number from the old version while she could maybe once print all the numbers for all families (about 175) and just have her secretary re-enter those numbers in the new version. So this new version does not seem as intuitive as the older one to them but some of them do not want to make things easier and they just complain a lot. We've had two rounds of training and have created a very comprehensive manual also but still"* *"The new one [CDT] is windows-based, it is not as intimidating as the old one, much easier to train, it handles the information better, new case note is part of system, [it has] greater ease, more people connected"* *"new CDT system is window-based and is a lot easier than the old ECHO system … There was easy transition … simple and good training … good to use … just using mouse … very user friendly"*

Table 5.4: Aspects of CDT and CC Solutions Open to Interpretations-Integration

Aspects of IS open to interpretation	Interpretive Flexibility	Quotations
Integration	***Integrated system***	*"We talked to Solutions to develop CDT features into solutions … they haven't, they are under-capitalized … that's why we had to switch to new CDT … for a long time it was DOS based, now it's windows based"* *"It is an organization-wide system .. the features should be in a way that covers everybody's need and it cannot be customized just for few. "It's never going to be a simple system… There are so many factors involved, so many features needed [for both systems]"* *"Best what's good for everyone … There never is that sophisticated enough in the world of computers [to fulfill everyone's needs 100%]… databases are getting more* and more *sophisticated … difference between night and day … everything they work … not integrated enough… [there is] better communication between the systems"* *Therefore, systems have to also check for this. There is no system integration. Solutions does not have billing and schedule module. It has excellent features for behavioral but child welfare service features are missing.* *"We've been burnt out so much [that's why] we have searched a lot for a good solution. We've been actively looking for something for integration"* *"They have also tried to see if they can convert CDT to CC solutions or vice versa … but either way they will"* *"[In CC Solutions], It is difficult to get information out of it… you input the information to it but then it's really hard to get what you want out of it… you input and then when it comes to aggregating that info and reporting, it doesn't give you usable information … it doesn't have robust reporting system"* *"Some clients are in both and both systems should communicate. Otherwise it's a sloppy way of doing business".* *"My main concern is that everybody [that works with different pieces of data from the*

Table 5.4 -- Continued

		system] has to have access to both" *"No Bridge [between the two system], doesn't make economic sense … Also, not correct information … there are discrepancies for information and this is costly … we might have to invest heavily but we will benefit in long run"*
	More sophisticated tools, analysis tools	*"It's never going to be a simple system"* *"Another thing we need in both CDT and Solutions is ticklers. For example, there are things that have to be done every 90 days and there should be reminders for those … now there is a lot of manual tracking in the systems"* *"I get the information from CDT or Solutions and analyze it and make them look better with chart and graph and give it to the managers … so if CDT and Solutions had that feature, it would be nice … it would ease up things"* *"CDT does not have capability for the clients [clients' parents] to enter information directly to the system, this feature is needed, this way data entry errors and time would be cut"* *"Critical pieces [are that we have] tremendous amount of data … we just do tracking[though]… what we are doing[using our data] we don't do much analysis of data … we don't use it strategically"* *"Both systems are efficient in the sense of giving data to them [but then in the case of output they are not]"* *[There is a] need for better navigation… [also] Intuitiveness of the system to know 'what I need to do next'"*
	Dashboard	*"President recommended using dashboard, which would get the information from both systems and integrate it. This might be a good idea in short term, but if they make changes like update CDT from ECHO, they have to pay extra fee for dashboard"* *"We need a dashboard kind of system"*
	Reporting capabilities	*"Another thing is that there is no ability to get one type of report for the list of all kids"* *"I am quite satisfied for having learnt writing crystal reports … Crystal report is not something that anybody can write … I can, I think because I have that way of*

Table 5.4 -- Continued

		logical/analytical thinking … You should know the data and have analytical thinking to be able to do it … it's one of the nicest report mechanisms … I was never trained in Access, well I had received very basic training for querying in Access … Crystal reports is much more logical in my mind" *"We have IT department and not MIS … we mostly buy off-the-shelf products … our reporting needs customization"* *"The reports out of the systems are made depending on the user of it"* *"It [CDT] has some standard reports that do not meet all the needs… Each program (and department) needs its own customized reports… Now they are running crystal reports"* *"They send some of the staff to crystal report writing classes… Except for a few employees, others staff found it difficult and they do not use it (or hardly use it)… The staff also do not want to run reports themselves using search and narrowing down the search result based on the search criteria they need"* *"Some of them want the reports to be directly linked to their outlook or can see it in an outlook like format… The current reports don't work in outlook… as of now, they keep writing programs through Shawn who is responsible for CC Solutions [and very familiar with CDT… They get data from both systems and enter them to MS Access and make management reports… This is time consuming and sometimes it does not get done because Shawn is busy and they have not hired anyone else either"* *"I get the information, from CDT or Solutions, and analyze it and make them look better with chart and graph and give it to the managers … they don't want writings, they don't want to read, they like charts and graphs better … so if CDT and Solutions had that feature, it would be nice … it would ease up things"*
	Simpler	*"I have made a lot of suggestions for the system … It would be really nice if they made the system simpler and got rid of the unnecessary things … Anything that makes it simpler is greatly appreciated"* *"I don't understand why shouldn't many of the options [that I always use] be selected by default and I change them if I need to … Why should I scroll and select every time"*

Table 5.4 -- Continued

	Tools for parents	*"CDT does not have capability for the clients [clients' parents] to enter information [notes] directly to the system (Solutions has this feature) this feature is needed … this way data entry errors and time would be cut"* *"We have created manuals for Foster parents and it has worked quite well"* *"Parents also add some things, the progress report … they also have view access to treatment plan and med logs"* *"They need to be getting extensive training … some of them don't know how to do it"*
	Laptop use	*"Laptop versions of their system, so that they can enter their report and then upload it later … at their homes, car, off-site … this would give a lot more flexibility and efficiency"* *"Clinicians are 50-50 percent of time working remotely and on-site… it is internet based but it is faster on site and slower off-site… so, if they make the laptop use easier and faster, it will help a lot"* *"There is need for Portability of the system… Majority of the clinicians work off site (from homes, other agencies, etc.)… They have to be able to access the system off-site… Accountants are able to do that… Clinicians need to be equipped with laptops to be able to do this so that they can do the treatment, parents OK it [there and then] and then upload it"*

Chapter 6

Data Analysis -- Pub Org Case Study

6.1. Introduction

This chapter is allocated to the analysis of data from the main case study, Pub Org. First, the process of coding the data is described. Data coding involved an iterative process of reading and highlighting the text that corresponded to the process of configuration using SCOT as the theoretical lens. The data coding section (6.2) provides a detailed description of this process. The data was analyzed in two sets, the descriptions of which are provided after the coding process is discussed (section 6.3). The chapter concludes with a discussion of how various dimensions of the configuration process fit together. These conclusions set the stage for chapter seven, wherein a mechanism for the process of configuration is developed and discussed.

6.2. Data Coding

QSR NVivo was used for analyzing the qualitative data[1] of this study. In order to code the data, five nodes[2] were defined initially: Chronology, Work order, Features, Interpretive flexibility, and TFR. These were initial high level nodes derived from the theory, SCOT. The first two nodes were defined in an effort to find the data that described the narrative of the case and process of work orders. The other three corresponded to the technological artifacts, interpretive flexibility, and technological frames elements from SCOT. By creating the Features node the researcher aimed at finding data related to technical configuration problems that arose when configuring each feature. Through this node, the text (i.e. data) was scanned to look for various technical

[1] Please refer to Appendix B for a short description Nvivo qualitative data analysis tool for analyzing qualitative data.
[2] Nodes refer to categories that are defined based on the theory in order to organize the data in such a way that patterns and themes are discovered within the text (i.e. qualitative data).

features of the system. These data were also related to the closure and stabilization elements of SCOT. The Interpretive flexibility node was created to look for the data that described the prevalence of various interpretations for each problem.

In the initial stages of coding, three more detailed nodes for Interpretive flexibility[3] were created. These sub-nodes included: Disagreements; Philosophies; and Activities of individuals. Disagreements was used as a node to identify situations wherein explicit disagreements over issues existed. It was meant to find data related to disagreements around issues related to the configuration, especially the setting up of various features. This was in line with Bijker's approach for finding different interpretive flexibilities, and consequently various groups; i.e. based on existing problems.

The Philosophies node corresponded to the beliefs and perspectives of individuals when it came to the Datastream 7i system. Philosophies and TFR showed considerable overlapping data, the former including the data related to TFR and more. Therefore, it was decided to only code the data under Philosophies node. The Activities of individuals node focused on identifying instances when individuals took actions about the situations at hand. This node is similar to Philosophies in the sense that it looked for the positions that individuals took on various issues (e.g. problems, decisions). In other words, Philosophies was more about the perspectives and interpretations and Activities of individuals more about actions taken.

[3] These new sets of nodes emerged early on during the analysis. After this point no new node emerged all throughout the analysis. Once the new set of nodes was added, the researcher went back to the data that had been coded before these nodes had emerged and recoded the data.

After the coding was complete, the researcher started running queries[4] for each individual node. Querying allowed the researcher to view all relevant text to a node from various documents. In qualitative data analysis, this makes it easier to find patterns since the researcher is immersed in one context (e.g. Activities of Individuals). The next step was to analyze all the data and write the analysis under three categories: Chronology; Work Oder; and Features. For each category the researcher would run relevant queries, read the queried text, and write pertinent ideas in the form of comments.

Meanwhile, a search on all data based on relevant keywords to the topic was conducted. For example, when writing about the Asset tracking feature (discussed in section 6.3.2.7) the researcher conducted searches on all documents using keywords such as asset, inventory, tracking, material, and product. Additional searches were conducted if needed once some of the data that was extracted from the first set of searches[5] had been read. In addition, throughout the data analysis, when working on a specific theme, if the researcher observed data relevant to other themes she would take note of it[6]. At the end of the data analysis, those notes were incorporated in the analysis if they had been missed during the initial coding, querying and keyword search[7].

---

[4] In NVivo query refers to a report that is generated based on one or more nodes. In this report, pieces of text that were coded based on the nodes are extracted and put together.

[5] Keyword search was specifically useful when finding the themes for Work Order and Features because it was needed to find more narrowed down data on attributes of each (e.g. various stages of work order process or each specific system feature).

[6] This is in line with the hermeneutic nature of data analysis; the researcher is constantly working between the parts and the whole of the phenomenon.

[7] This could have occurred for reasons such as the data not necessarily containing explicit keywords or having missed the relevant data in initial coding.

6.3. Data Analysis

Data analysis was carried out in two sets. The first set of data analysis corresponds to the technical features of the system. This analysis revealed some controversies around various technical features of the system. For each feature various problems and interpretations were identified. The analysis then elaborates on how these problems were addressed based on the diverse interpretations. The second set of data analysis is related to the themes that emerged during the first set of data analysis. By emergent we mean themes that were not anticipated by the researcher from the onset. However, these themes emerged while the researcher was looking for themes more directly related to the configuration process in the light of SCOT.

The recurring themes during the second set of data analysis were named organizational constraints. As with the first set of data analysis, a SCOT approach was taken to conduct the second set of data analysis. After having been identified in the first set of data analysis, these controversies were further followed in the second set of data analysis, interpretations about them discussed, followed by interpreting the manner in which they were resolved.

In each set of data analysis the completion of the data analysis occurred when a theoretical saturation was reached (Yin, 1993, 2002). At the point of saturation, the researcher was unable to find new themes and further readings of the data were merely reiterating the same concepts.

6.3.1. First Set of Data Analysis

The first set of queries was run on the data coded for Chronology, Work Orders, and Features nodes. First, the goal was to write a narrative about the process of configuration from the time the software was purchased to when it was functional (i.e. roll out stage). This narrative, which was written based on the Chronology node, is presented in chapter four as the main case background (section 4.3.1.2.1). Next, the researcher queried (and conducted a search on) the data and wrote about the processing of work orders before and during implementation of the 7i system. This analysis enabled the researcher to compare the two and identify how business processes as well as features changed during the process of configuration. This analysis is presented in section 6.3.1.

Finally, the data on various features of the system was extracted. The focus here was on the more controversial features since the objective of the research was not on finding themes related to the configuration of every single feature. Instead, the goal was to investigate how various problems are approached and resolved in the process of configuration paying particular attention to the technological artifact (in this case by looking at various technical features of the system). Therefore, the more controversial features were examined. These were considered to be those whose involved disagreements. In addition, discussions around these features normally spanned several sessions. These features are described in section 6.3.2.

6.3.1.1. Work Order Process

Work orders (WOs) are requests made to address maintenance needs at PubDiv. A work order process involves completing a maintenance job from initiation to completion. The previous version of the system was called maintenance management system. This name was changed to the work order management system in order to refer to non-maintenance jobs as well. However, during the implementation of the 7i software only maintenance work related features were configured in the system. In addition, as explained in chapter four (section 4.3.1.2.1), one of the goals of this project was to implement the asset management system features of the 7i software as well, which was not completed successfully during the time period of this study. Even with this goal, the system was still called a Work Order Management System (WMS) and the team that was formed to configure the system was called WMS task force (WMSTF)[8].

6.3.1.1.1. Business Processes Before 7i

This section describes the planned (i.e. To-Be) business processes that were developed to be configured in the 7i software. These were based on the business processes in the old system, and the changes that were made to be configured in the new system. As discussed in section 6.2, the description of this section was completed based on analysis of the data related to the Work Orders node. The objective of this discussion is to gain an overview of business processes before 7i was configured. This is important because it provides the reader with an understanding of what the 7i system was expected to accomplish for PubDiv. Figure 6.1 exhibits the work order process as developed by WMSTF team.

[8] For the sake of brevity, from now on when referring to the whole group we will be using the term “WMSTF team” or merely “team”. When referring to individuals within the team we will be using the term “team members”. In order to identify other groupings of individuals we will be using the term “group”.

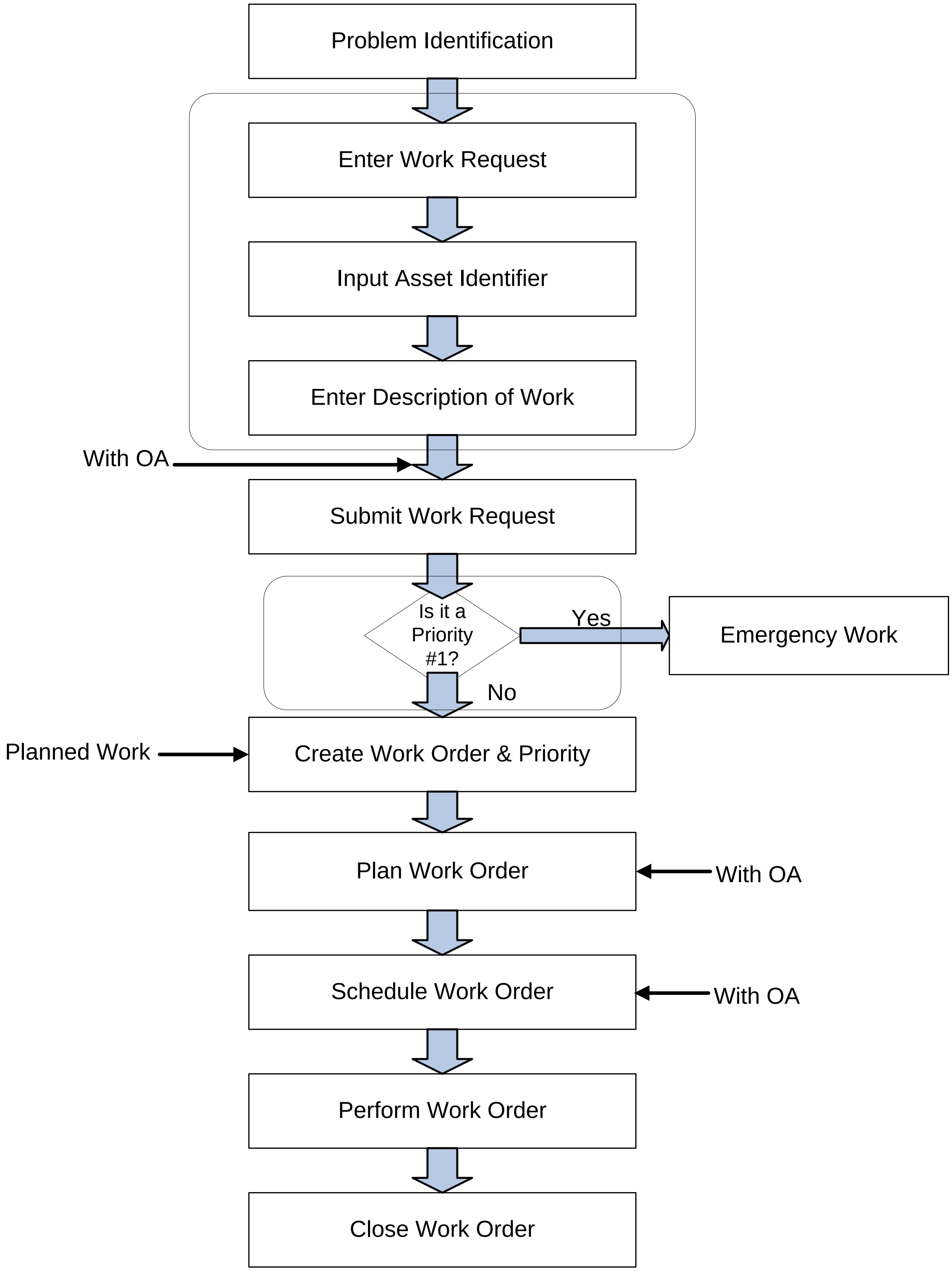

Figure 6.1: Work Order Process

A work order is initiated by a general requester (GR). A GR is an employee at any level or physical location in the organization and who views a maintenance problem with equipment. The GR fills in a form in which he describes the problem. Alternatively, the worker verbally (whether in-person, or via phone or email) communicates the problem to a supervisor and the supervisor issues a work order request. This request is sent to PubDiv's central data center and entered in the database. The operator who enters the work order in the system gives a confirmation number to the requester, which can be used as a point of reference to the work order request in the system[9]. The priority (degree of importance and urgency) of the work order is set either by the worker or his supervisor. At this stage the work order status is said to be "Open".

The open work request then goes to an operational authority (OA). An OA is a senior project manager who has authority over the operational work of a facility. Each facility has only one OA but one OA can oversee more than one facility. The OA checks the work request for validity of the problem and its priority. If a priority has not already been established for the work order, the OA assigns one. The OA also checks for overlapping or redundant work orders and cancels them if there are any. At this point the work order is ready to go to a scheduler and its status is set to "Released". OAs can view all the work orders but are able to only change those that have been referred to them. During the configuration process, the necessity of each OA being able to see all the work orders was questioned. The majority of team members believed that OAs should be able to view only those work orders related to their own facility. This was, for example, one of the issues

[9] This is most of the time done via phone calls.

that presented itself as a controversy to the team (explained in more detail in section 6.3.2.1).

After an OA releases a Priority 1 (i.e. emergency) work order, it receives immediate execution. If the work order is not of Priority 1, it goes to the planner, a senior project manager. In this stage basically it is determined whether the work order can be processed without the need for additional in-house resources, needs to be planned by allocating additional resources, or needs to be contracted to outside contractors. At the end of this stage the status of the work order is set to “Planned”.

After the work order is planned it goes to a Scheduler, a project manager. At this stage, the Scheduler assigns the work order to a maintenance supervisor (MSUP) from a relevant department in the order of the priority (2-5) of the work order. At the end of this stage the work order is said to be “Scheduled”. In the next stage, the MSUP assigns relevant labor to perform the work order. When the necessary tasks are performed and relevant tests are completed the work order is said to be “Ready for acceptance” or “Completed”[10]. The work order then goes back to the OA for closure. After the OA has checked to make sure all work order information is updated and complete, the work order is set to “Closed”. At this point, the maintenance work is complete and the equipment is functional (i.e. the problem has been resolved).

[10] The whole process can take any time between a few hours to weeks. Some work orders are never completed. These all depend on the criticality of the work order, and the resources.

In practice, most work order processes are not as straightforward as described above. Exceptions can happen at any stage of the process. For example, often equipment cannot be released for maintenance because it is being used for normal operations. This complicates the process especially when the priority of the maintenance is high and a delay in the repair job could result in the work order turning into backlogs (i.e. delayed work orders).

If a work order is not complete within the time allocated by Scheduler, it becomes a backlog. In MP5 (and before MP5) if resources are not available to address the backlogs, the work order can be closed (by PubDiv's central data center staff) and reopened at a later time. However, this is not possible in 7i. One reason is that in MP5 the entire process was still practically manual and the software was mainly used to keep a historical account of the work order. This resulted in some inoperable equipment sitting on the floor for a significant amount of time. One goal of 7i is to prevent this from happening. In 7i the objective is to track the work order as it is being processed (i.e. 7i as a tracking tool) and to eliminate backlogs as much as possible. Therefore, the controls set in 7i do not allow for closure of work orders unless they have actually been completed.

At the time of data collection, there were around 60,000 records in MP5, about 8,000 of which were open work orders. Some WMSTF members believed that the data in MP5 was "far behind" and that it needed to be "massaged down" to be usable in 7i. In this regard, this group believed that MP5 needed to run as a standalone system parallel to 7i for a period of time before being shut down completely. In addition they felt that the data from MP5 needed to be "looked at and cleaned up" before transferred to 7i. They also believed that the old open work orders needed to either be closed or canceled in order to organize the data.

This concern came from the fact that a similar issue had occurred when upgrading an older version of the system to MP5. In other words, a lot of "unclean" data had been transferred to the new system. Therefore, one big concern of some team members was to start off the 7i system with accurate data, as expressed in the following comment: *"I just hope that we learn from errors of the past because we have made mistakes in the past and have lost a lot of data in MP5"*. Overall, many of the members believed that understanding data was key in assuring successful migration to the new system. For example, one of the team members stated that *"incoming and outgoing data are extremely important… workers really don't know the data and IT doesn't care about the data… If lay people knew MS Access tables and the way they link maybe that would've helped a little… but now these problems delay the project*". This team member was referring to the importance of understanding the data that underlies the work management system in order to be able to configure the system more accurately.

6.3.1.1.2. Business Processes During 7i Implementation

Datastream 7i is web-based work management software; this is a major difference between Datastream 7i and MP5. The upgrade from MP5 to 7i enables a large number of concurrent users access to the system, as well as reduces the number of configurations (i.e. configuring the system on the application server versus each individual computer).

The software was purchased packaged which meant it did not allow for customization. The software, however, needed to be configured to reflect business processes of PubDiv. IT personnel mainly used FlexSQL and PL/SQL when configuring and maintaining the system. They use two servers: application server, and reporting server. The application server is web-based.

In 7i, upon logging in depending on his role, a user sees different settings on the first screen (referred to as Inbox). Figure C.1 in Appendix C exhibits the Inbox configuration. The work order process is not a linear process as a WO can go back and forth to the same person multiple times during the process (please see Figure C.2 in Appendix C). Any employee with access to the system can issue a work order. However, the work order can be issued only for those equipments that are already input in the system. This is another upgrade from the MP5 system.

OAs check the system daily to see a list of “standing” work orders which are waiting to be released by OAs. In 7i OAs still see all the work orders of all the facilities when they log in. In order to filter down the records to only view work orders of his facility an OA needs to write a dataspay[11]. One of the desired upgrades from MP5, however, was to have the system automatically filter the records of WOs based on facilities. As discussed in section 6.3.2.1, team members encountered disagreements over the scope of the work orders that each OA is allowed to view. This issue was not resolved during this study. After the OA filters the records for the WOs in his facility, he checks for duplicates and

[11] Filters are called dataspies in 7i.

makes sure the work order is indeed a valid request. By accepting the request OA "releases" it. At this point the WO is ready to be "scheduled".

A Scheduler checks the released WOs to decide whether the work order needs to be performed by a Contractor. If enough in-house resources are available to complete the work order, the scheduler changes the status of the work order to "scheduled". If the work order requires outside contractors for the repair job, the scheduler makes a comment about it. What is interesting is that scheduler does not actually "schedule" the work order. A maintenance supervisor assigns the appropriate maintenance crew to the work order and makes comments about the estimated costs of the maintenance work. At this stage if the MSUP confirms that in-house crew is not able to perform the repair job, he sends it to contractors. Otherwise, similar to the old system, he/she prints out the WO and hands it to the maintenance workers. He also has to make comments about the details of the maintenance such as costs and hours booked. After the MSUP has printed out (or contracted out) the work order, he changes the status to "work in progress". This part of the process is still done manually (i.e. when the work order is in progress and being worked on by the field workers) even though the system has been configured for this feature.

The least straightforward stage of the process is the "work in progress" stage. Firstly, there is no simple and clear procedure for contracting the work to outside contractors. No specific feature was configured for contract work orders. Moreover, sometimes regardless of initial guess of the scheduler and MSUP that in-house crew would be able to complete the work order, in-house resources might prove insufficient in the middle of the process; and thus, there would be a need to contract the WO at this stage. In addition, frequently a maintenance crew would realize that the actual problem is related to an asset other than what has initially been reported in the work order request. This complicates the process since the work order has to travel back to the initial point and be reissued with the new information. This has become more problematic in 7i since 7i imposes more restrictions on the work order process. When the work order is completed the workers return the print-out to the MSUP. The MSUP checks the equipment to ensure that the problem is resolved and then changes the status of the work order to “Ready for Acceptance”.

At this point, the WO goes back to the operational authority. OA also verifies that the equipment has been repaired (usually by sending a maintenance worker to check the equipment). If the problem is resolved, OA changes the status to “Closed”. Some of the specifications that OA needs to check for their entry validity at this point include equipment ID, hours of labor, and costs. OA makes the final decision by also reviewing all the comments that were left by others during the process. All along the process, at each stage, the person in charge is required to comment on the details.

Overall, in the follow-up interviews it appeared to us that since the processes had changed significantly; team members seemed to be confused about them. In the initial observations and interviews people appeared to have a clearer understanding of the whole work order process. Some team members mentioned that one reason for this confusion was that 7i is designed for more streamlined process and that is not the case at PubDiv. For example, the researcher heard comments such as:

"On a smaller scale this would have worked perfectly… It's a very good system it just doesn't work for DIVA".

"We don't have the chain of commands necessary for such a process… it is really frustrating… For example, Maintenance crew work for OAs in one facility but they actually report to schedulers in another facility… so there is no accountability [referring to the fact that the line of authority is not as straightforward; thus, it is not easy to hold employees accountable]… and it is all because we are so spread out and our processes are not as streamlined and straightforward… if things were more central and if there was a leadership on the top and more accountability this system would work perfectly".

"We need a hierarchy in place otherwise this system needs to be configured for each facility separately… in that case, it could also be implemented perfectly but as it is not".

It was only in the configuration process that the team realized that the real-life business processes were not as straightforward as they appeared when developing To-Be business processes; even with consideration of the exceptions that can happen in the process. The researcher believes that viewing the work order business processes on the actual configured system made the team members' more conscious about the issue that the work order business processes are not as straightforward as they may have appeared when they were developing the To-Be business processes. From the comparison of the processes before after 7i is configured, it can be seen that even thought the new system offers more features for the computerization of the processes, it can also restrict how the processes are completed. This is interesting because it calls for a dynamic and mutual change in business processes and the configuration process while software is being configured. This comparison was also important in the analysis of data since it explains why the WMSTF team faced challenges when configuring the software. This discussion will be elaborated on in the next chapter after the configuration process of 7i and business process development of by WMSTF team is discussed in this chapter. Next section (6.3.1.2) is allocated to a discussion of systems features, where the features that became controversial during the configuration process are discussed.

6.3.1.2. System Features

In this section several of the system features that were the most challenging to configure are discussed. These features were often the focus of discussions during configuration sessions. In the end though, many of these features were not configured the way they were desired by the team.

In order to configure 7i, a group of around 15 representatives from DIVA and DIVB came together as the work management system task force. The WMSTF team met in a simulation room twice a week to discuss the configuration issues. The team would run various scenarios in order to understand the system, test the features, and offer solutions for updates. Normally, they would run a work order from requesting to closing. Each person would assume one or more roles during each simulation and they would take turns running the whole process. Simulations were basically hypothetical scenarios; each scenario paying particular attention to a specific feature.

While configuring the software there were various environments (i.e. copies) to work with: Production, Development, and Testing. The Production environment was the version that came packaged from the vendor. This environment "was kept clean and original", in order to maintain an un-configured copy as a backup. The Development environment was a copy of software that IT personnel worked with to configure the software. The Testing environment was where simulations took place to configure and test the software. When the process of configuration started these three environments each had a separate database. However, towards the end of the project they all shared the same database (Oracle 10G) and there were merely different data schemas for each environment.

Below, a discussion of several of the system controversial features, which gained special importance during the configuration process, is provided. These features were prevalent when the notes from simulation sessions, meetings, and interviews were reviewed at the time of data analysis. For each feature, an explanation of the tool is provided. This explanation is followed by a discussion of the controversies about the feature, its interpretive flexibility, and the way controversies were resolved.

6.3.1.2.1. Inbox and KPI

The first interface that every person views upon logging in 7i includes an Inbox on the left side and Key Performance Indicators (KPIs) on the right side (Figure C.1 in Appendix C). Inbox "i*s kind of like a mailbox… It's the first screen that anyone sees when they first sign into the system*". The Inbox contains three sections, separated with different tabs. Depending on the role of the person logging in the system, the relevant tab opens. A list of work orders categorized based on their status is shown to the user which are relevant to his role. For example, a list of what a maintenance supervisor sees includes: Scheduled; Return to Maintenance; Work in Progress; Extension Scheduled; Immediate Release; Ready for Acceptance; Extension Denied.

KPIs are dashboard graphs showing gauges that have three colors (Red, Yellow, Green) that reflect the status of work orders. The status green signifies that there is no backlog. Yellow is an indicator of the work order being close to becoming a backlog whereas red indicates the need for immediate attention to the work orders that are backlogs. Backlogs refer to jobs that have been delayed.

In order to configure the Inbox page, the team drew a table on the board and wrote down all the needed items for Inbox. This began with a concise high level list of items and was continuously refined. Table 6.1 is the initial table they drew in order to configure the Inbox. It started with discussing what each person (role) should see upon logging into 7i. They went back and forth between the flow chart and this table. They followed the same process between the software and this table when actually configuring these features.

Table 6.1: Initial Design of Inbox by WMSTF team

Operations Authority			Scheduler			Planner			Maint Supr		
Ops	Maint	Mgt	Ops	Maint	Mgt	OPs	Maint	Mgt	OPs	Maint	Mgt

Table 6.1 was expanded (and updated) further as the team discussed the Inbox feature. This table became the point of reference for the team during the process of configuration of Inbox.

It was discussed that in the older systems if there were any backlogs and there were not enough resources to complete the work order, they were allowed to close the work order and issue a new request at a later time to avoid cluttering the system[12]. In order to avoid this in 7i, the team had to come up with criteria for the number of days allowed for completion of work orders that would be reasonable enough. The team spent a considerable amount of time coming to an agreement about the number of days to consider a work order as a backlog. Finally the days shown above were agreed upon contingent on further consideration at a later time (i.e. "good for now… we'll run scenarios later and we'll look into it").

In the initial setup, some of the team members noticed several problems with Inbox and KPIs. One of the main problems brought up repeatedly was the need to partition the Inbox and KPIs based on different facilities. Initially every user would see all the KPIs in all facilities and in order to view the work orders relevant to his facility a user had to run a dataspy (i.e. filter). Even a seemingly simple solution such as this did not seem to be an apparent solution to everyone in the beginning. Only after a few members proposed it, the filtering of the data became a norm for everyone. However, the need for automatic filtering of the records through the Inbox was repeatedly mentioned in simulation sessions, group meetings, and individual interviews. This issue was brought up by one individual and eventually became a concern of the majority of the member. The following comment demonstrates concern over this issue:

[12] However, this was apparently not practiced commonly because as mentioned by one of the members, the old system suffered from data integrity and organization to a large extent.

"Inboxes were all wrong because everyone sees the same things. For example, a mechanical person sees all the mechanical and electrical problems altogether, and that's why it is hard to see which one's theirs. They might not even have any backlogs for their own work but because of this problem, it will show as if they do and unless they go and check the list they will not know and this is not the point of KPIs or Inbox".

In the second quotation below, one of the team members brings to another member's attention that his interpretation of how Inbox should work does not accurately reflect that of the team leader (Bob) or what had been previously discussed, pointing to the board that they usually used to write down notes, and draw tables and models.

First person: *"everybody's Inbox shouldn't look the same but Bob said that Inbox for everyone should look the same"*.

Another person replied: *"Bob didn't say that [pointing to the expanded Inbox table drawn on the board] according to the board, the Inbox configurations aren't the same"*.

The quotes above reflect how even coming in agreement about existence of a problem (in this case the Inbox) could be challenging. Furthermore, in the beginning when this problem was noticed, the members did not seem to be clear about their concerns/requests. However, as time passed the members became clearer about their requests. This was observable in the terminology they used to refer to this problem as well. For example, at one point they worded this concern (in a meeting memo) in the following way: *"Each Inbox item is linked to a dataspy that each user needs to create. Each item needs to be linked and relevant to the user. Therefore, this needs to be done through queries, to filter the relevant data"*. The way this problem is worded in the quotation, it seemed that at this point the team did not have a clear idea about the technical capability of the system to automatically present only the relevant data to each user.

In the follow-up interviews, it was inquired if this issue was addressed in the roll out version of 7i. Interestingly, when the pilot version of 7i had already been implemented in several facilities, this issue became clearer. For example, this process was called partitioning, as seen in this quotation: *"DIVB would like for the Inbox Codes and the Dashboard KPI's be partitioned per facility"*. The clarity about the definition of this process was acknowledged explicitly by the team (in a meeting memo):

"Discussed the proper term for what was referred to as "Routing", the proper term for the request is "Partitioning". Partitioning is considered a two step process which requires input from the front and back end. The option was given to create classes in an effort to perform this action".

As shown in the above quotation, the team had reached a better understanding of how to address the problem of KPIs though technical configuration of the system, rather than finding a work around (e.g. manually filtering the data). Therefore, the Inbox and KPIs were configured in a way that, based on various classes, each user would automatically view records related to his department (or even as detailed as a particular plant) upon logging in the system. However, it was also allowed for the users to have the ability to view records from other departments as well if they chose to do so[13]. Therefore, in the beginning a role such as general requester had a limited view. However, with the latest

[13] This was not configured in the Inbox; however, it was configured that through a link in Inbox the user could filter the records to view work orders from other departments.

configuration, it was decided that they would have a more expanded view but merely related to their location.

Other minor problems with the Inbox tool were also prevalent. One example was related to the number of days allowed for an order, and the start date[14] of work orders used to determine backlog status, reflected in the first and second quotations below. In this case, while configuring the Inbox, it was realized that some of the numbers assigned as the criteria for backlog status were in fact incorrect. In the case of start date, it was realized that the way this feature was set up, it was possible to set the start date of a work order such that before it had reached a planner the work order would already be considered a backlog. Another minor problem was related to the terminology used on the interface related to various types of work orders: non-backlog work orders, new work orders, and backlog work orders. The third and fourth quotes below demonstrate this situation:

“They are actually wrong… instead of >3 days being backlogs, it has to be <3”.

“Start dates need to change. Otherwise when it gets to planner sometimes the work is already a backlog… the minute they create it, it sets a start date, that’s not correct. I told them either fix it or just take the KPIs out, it’s a waste of time”.

“Non-backlog work order, oh that’s too wordy… it should simply say: new work orders… no, it should just say work orders because you know it’s either new or backlog”.

“This is wrong… It is just words. It should be: total work requests, new work requests, backlog, duplicates and possible backlog”.

[14] The date from which the number of days for backlog starts to be counted

The problem about the start date (represented by the second quote) was brought up at the time that the team thought they had reached a relative closure on the case of Inbox and KPIs. Comments such as *"who brought up this, him? I need to see you outside… see, you stir the team up and leave"* were repeatedly heard from the group, reflecting the frustration of some members over opening the discussion about KPIs and Inbox again.

In this particular situation, regardless of the unwillingness of some members to continue working on the configuration of the Inbox and KPIs, the issue was discussed further. However, after some discussion, it was unanimously agreed that the backlog start date (explained above) was in fact setup incorrectly and had to be changed. It was decided that the planner would be responsible for setting the start date of work orders to prevent premature backlog status. The terminology was also corrected to avoid being unnecessarily verbose. In the tools the term Work Order was used. Different statuses of a work order, such as total, backlog and duplicates would be shown for each specific work order as a report, which would be created based on relevant criteria not as a feature to be set by the user.

In sum, the team members were divided into two groups in their opinion of the Inbox and KPI features. One group believed in the appropriateness of having a generic view for these features across all divisions and facilities. In other words, this group believed that each user should be able to view all work orders. The second group argued that the appropriate configuration would allow each user to only view work orders relevant to his division. The final configuration of this feature allowed for both. According to this configuration, a user would view work orders pertinent to his division upon logging in the system. However, he would also be able to view other work orders if he chose to.

6.3.1.2.2. Priority

The Priority feature (shown on Figure C.2 in Appendix C) was configured to be able to set orders of precedence for work orders, in which the maintenance jobs would be completed. The team initially proposed that a priority level would be set by the general requester for each work order. This priority level would define in what order various work orders needed to be performed. The purpose of having a feature for priority was so that the Maintenance Supervisor would be able to schedule the work orders based on their degree of importance and time-sensitivity. This seemed to be a cultural norm in the organization since the team also usually had a list of configuration issues with set priorities that needed attention, discussion, and agreement. Even though this is probably a common practice, mentioning it here is important because, as was observed, the team as a whole took a similar approach in resolving the configuration issues in general. This approach seemed to be ad-hoc rather than systematic. Even though the team would develop a list of prioritized configuration issues that needed to be addressed during each session, they would address those problems in no particular order.

The configuration of the Priority feature was brought up several times during the process without much agreement. This was inevitable because as different people mentioned, there could be a political dimension to this feature. In other words, the concern was that setting priorities for work orders could, at times, be "*subjective and based on relationships*". For example one of the team members once mentioned:

"Let me say something [and he went to the front of the room and pointed to the flow charts]… the main goal here is to move from reactive to planned work orders. Now all are reactive. Everyone thinks their work is more important that's why there is a possibility that they set the priority to 1 even if it is not really 1. That's why the OA needs to see it to make sure it is really an emergency and then send it back. You guys are all missing the whole point".

Overall, members were divided on the choice for the scale used for the Priority feature, the wording of each option, and automatic change of status for work orders with high priorities. The most important concern was disagreements over the scale. Opinions ranged from the need for 1-3 to 1-5 scales. What was interesting about this discussion was that people disagreed on the choice of the Priority scale based on different grounds. For example, the group that voted in favor of a 1-5 range (1 being the highest and 5 being the lowest priority) argued for each priority setting in terms of the period of time allowed for the work order with that priority to be completed (e.g. allowing more than 36 months for priority 5). On the other hand, the groups that favored 1-3 scale were concerned about the wording of each priority setting. For example, Emergency, Urgent, and Important were proposed as the terms used for priorities 1, 2, and 3 respectively. The belief regarding the appropriateness of each scale was reflected in a specific interpretation. This has been elaborated on later in this section.

In the follow-up interviews the researcher observed that the 1-4 level, the middle ground suggestion, was adopted with the following descriptions:

Priority 1: Emergency (e.g. work orders related to regulatory issues, and health of the personnel, which need to be completed immediately)

Priority 2: Urgent (work orders that need to be completed within two hours since they are requested)

Priority 3: Important (more common work orders)

Priority 4: Engineering Capital Improvement project (CIP)[15] (to address minor issues)

Some people believed a 1-5 range was too broad for the priorities. This was especially of concern since in the old system, the users were used to a 1-3 scale and this new range would make it confusing for them to decide what priority to assign to a work order. The following dialogue demonstrates this concern.

Smith[16]*: "the priorities used to be 1-3 scale and now it's 1-5… we're forcing them to use this new definition".*

Jane: "but if they have a clear definition, this should not be a problem".

Smith: "but it used to be 1-3 now with 1-5 scale, that's why a lot of WOs are sitting and are incomplete, because people will resist completing them if we try to change their habits".

Smith's reason for voting for a 1-3 scale was based on his belief of keeping the format of features as consistent as possible with the old system. This was of concern for people particularly for features that dealt with the users' judgments.

An interesting observation during the discussions of the 'Priority' feature was that some team members would use the flow charts and refer to the processes constantly to argue

[15] CIP refers to "works based on long range plans"
[16] Names used are all fictitious name

about their choices. For example, one other issue that divided team members was whether work orders with high priorities would have to be immediately released to maintenance supervisors by eliminating the first step, that of going to the operational authority first.

A group believed that the problem regarding the setting up of Priority feature was not of a *"political"* or *"subjective"* nature but due to lack of understanding the processes. Indeed, the comments below demonstrate that there was not an agreed upon definition of Priority settings.

"We have documents that tell you what the definition of each priority is. Scheduler should be allowed to warn or question but not change".

"No one downstream, of OA should be allowed to change the priority and we spent a lot of time talking about that. You should certainly be able to challenge it but not change it. If you have a high ratio of priority 1, that is really high, means that you do a lot of waiting through a number 1 priority".

These quotes reflect the situations wherein some members attempted to explain the definition of the Priority feature to others. Hearing comments such as above at that late of a stage in the process of configuration shows how the team had not reached an agreed upon definition of the Priority feature. This lack of clarity of definitions is also exhibited in the example below.

On one occasion a team member, John, in order to prove his point about the problem of lack of understanding of the processes related to Priority, went up to the flow chart and asked one of the other team members, Bob, various questions about Priority settings.

John asked other team members not to interrupt and not to give opinions to avoid biasing Bob's opinion. Bob confirmed that the MSUP is the one who schedules the work orders based on their priority. He added that the OA is the one who checks to make sure the set priority for each work order is legitimate. Most team members agreed on these two processes. Agreeing on these processes though did not indicate these processes had the same meanings for all. For example, one team member interpreted the importance of completing work orders with priority of 1 as completing the work "out of courtesy" for the supervisor who assigns the work. However, priority of 1 meant a sense of urgency for most other team members. For example, in one discussion when a team member who did not initially seem to believe it necessary for the work orders with priority 1 to go to OA[17] told that this was "out of courtesy". Whereas others believed that it was necessary for the OA to check for the legitimacy of the work order priority before it could proceed.

However, when John asked who actually decides what priorities to set for work orders, team members were divided or did not have an answer. This division among the team members was interesting since the whole group seemed to have agreed that the OA was the person who had the power to change priorities when checking for their legitimacy. John basically was trying to prove that team members themselves did not know the processes. At this point John defended himself by bringing up the issue that the reason for not having a clear rule for who (i.e. which role) would have the authority to change the priorities was that processes at DIVA and DIVB were different[18].

[17] Some believed that priority 1 work orders needed to be immediately released.
[18] The difference in the structure of DIVA and DIVB has been elaborated in section 6.4.2. However, it was necessary to mention it here in the context of the issue of Priority.

The issue of processes being different at DIVA and DIVB came up quite often in different discussions. Many blamed reasons for not achieving the goals of the project in trying to standardize the processes of DIVA and DIVB to common business processes between the two sub-divisions. John's reason was that centrality of the structure of DIVB made the work order process more streamlined; thus, easy to follow the chain of command. However, in the case of DIVA the fact that field workers had to report to supervisors out of their facility (due to DIVA's structure) made it more difficult to track the chain of commands. Therefore, it was difficult to come up with a common ground for DIVA and DIVB to set the 'Priority' feature.

In addition, it was said that business processes at DIVA were far less structured when compared to those of DIVB. In fact the facilities at DIVA were more spread out. This was believed to be one reason for less needed communications at DIVA, which made the work order process easier. A member from DIVB mentioned:

"Approach at DIVB is different. Over there people know each other and if one sets a priority 1 they don't doubt it. They have a better communication over there… they know exactly what's going on. Therefore, when there is a priority 1, they do it right away, they don't sit down look at it to figure out first if it really is an emergency or not. But at DIVA, since they don't know who does what because of it being so spread out and each division having its own maintenance, some people might just set priority one because they want their own job to be done".

This implies that the Priority feature had the potential to be used more for political purposes at DIVA. Team members from DIVA also acknowledged the challenges due to the unstructured nature of their sub-division.

The discussion over the Priority feature spanned several sessions. Many were not satisfied with the amount of time allocated to discussing and attempting to configure the 'Priority' feature. In fact, they finally settled on the 1-4 scale setting and that is how the feature was setup on 7i. However, as mentioned before, in the follow-up interviews the researcher was told that Priority remains a dysfunctional feature (i.e. it is still used subjectively and/or politically).

In other words, as it was of concern during the configuration process, the completion of work orders did not necessarily adhere to their priorities. The system as configured did not stop some people from exercising a subjective approach to completing work orders. The interviewees made statements such as:

"Theoretically though there are 4 types of priorities… However, practically it depends on the relationship of OA and MSUP really… Priorities are not really working and at this point they are subjective".

As reflected in this quotation, similar to the old system, many MSUPs still decide on the order of completion of work orders subjectively, and based on relationships with people who have requested the work. This was exactly what had concerned some of the groups during the configuration process: *"organizational issues influence who prioritizes and how"*. The interesting point here is that even though a valid concern, the subjectivity of the setting of the priorities or even the order of the execution of work orders based on priorities (i.e. whether really adhering to the priorities when executing the work orders) did not seem to be influenced by the feature itself. As can be seen, this problem would have still persisted regardless of the scale and wording used for the priority feature. This is substantiated with the observation of the similar issue in the context of the abovementioned issue of approaching prioritized configuration issues. Similarly, a "subjective" approach was taken by the team when tackling configuration issues in general. In other words, even though for the sake of objectivity, efficiency, and effectiveness priorities were set for the issues to be tackled, a somewhat ad-hoc approach was taken in order to complete the tasks.

Another important observation here is that even though a feature was set in the system, it was still practiced in a subjective manner. In other words, the system, as configured, did not prevent the users from using the feature the way they desired; thus the initial concern that started all the controversy about this feature.

The reason that the team was not successful in finding a solution to the problem of the Priority feature could be attributed to the commonly held attitude of the team members: *"we're tired of spending more time"* on this feature and they desired *"to move on"*. Comments such as *"ohhh, lets not set this work order to priority 1! [when running simulations and having the knowledge that this would open up the discussion over Priority feature again]"* and *"we need to move on… I don't even know why we keep discussing this"* were repeatedly heard in the simulation sessions. When frustrated with the progress of configuration such as in the case of the Priority feature some members would go even further to blame the delays on more significant issues such as organization of the project and leadership. The following quotation was made by a frustrated team member during one such discussion.

"I think the main issue was lack of organization. The project leader needs to be here now and He needs to have structure. I haven't seen anything slower than this [referring to the process of implementation]. It took a year to finish those business processes [pointing to the flow chart]".

Interestingly at these times usually the blame for having invested a significant amount of time on the configuration process would also go back to having spent a significant amount of time on flow charts[19]. Therefore, it could be that team members' desire to progress and continue the configuration process could have led the team to adopt a "moving on" mechanism for the closure of the Priority issue.

In sum, three main concerns seemed to influence interpretations of different team members regarding the Priority feature: political use, standardization of the scale, and change in habit. Political use referred to "subjective" use of the feature. The group for whom this was a concern argued that the priorities of work orders would be set not based on their order of precedence but rather on relationships of people. A second group was concerned with the inappropriateness of standardization of the priority scale across DIVA and DIVB. This group argued that because of the differences in the structures of DIVA and DIVB, use of a common scale for both would render the feature meaningless. A third group resisted the idea of changing the scale for the Priority feature arguing that it would require a change in habits of the personnel. This group argued that this change could result in a resistance towards the use of the feature. For the final configuration it was decided to choose a middle ground solution; i.e. use a 1-4 scale.

[19] An entire sub-section (6.3.2.1) is allocated to the discussion of flow charts since this topic similar to the topic of business processes was a controversial one.

6.3.1.2.3. Reject/Cancel

Reject and Cancel work orders are two features intended to allow an operational authority to deny and halt a work order before it had started (please see Figure C.2, Appendix C). The two features presented an interesting case since some team members were not familiar with them. In addition, most of the members who expressed some familiarity with them did not show consistency in their interpretations.

A discussion about the Reject and Cancel features was started by a team member who was not able to distinguish between these two features. Other team members were also divided over their interpretations of these two features. Overall, there was an agreement that one of these features was used for deleting duplicate work orders and the other was used for canceling unnecessary work orders. Those work orders that were rejected would automatically disappear from the system without leaving a record. However, in the discussion about this issue some members used the term Cancel and some used Reject to refer to the act of denying a work order by an OA; i.e. there was confusion over the meaning of these two features. Therefore, the first problem was to establish the definition of each of these features. The second concern was that even in the case of Reject there needed to be an audit trail left after the command was executed.

The confusion over these features was apparent when some team members appeared to have one interpretation about them in one session and a different interpretation in another. For example, some members suggested that the Cancel feature had to be taken out whereas, upon explaining, it was apparent that they really were referring to the Reject feature. The reason for believing that the Reject feature had to be removed was that it would “kill the records” (i.e. eliminate the records without leaving a trace) when executed. Other members responded to this suggestion stating that this feature was needed so that an OA had the chance to reject duplicate work orders. Overall, there was confusion and thus division among team members about the meaning of these features and whether they were needed at all.

The discussion about the configuration of the Cancel and Reject features spanned four sessions. The team came to the consensus that the Cancel and Reject features were both required; Reject to be used for denying duplicate work orders by an OA; Cancel to be used to eliminate an unnecessary work order by the person who issues it and by an OA; and Reject command needs to leave an audit trail for tracking purposes. As a team member explained, *“IT set up the audit trailing capabilities to track all actions that occur on a work order and assets. Anytime you wish to track who made what change, right click that screen and select “Event Log”, a box should appear entailing who made the changes”.* The biggest problem in this situation was related to the definition of the Reject and Cancel features.

In sum, interpretation of one of the groups stemmed from a conservative point of view: the ability to track work orders. This group defended the idea of having an audit trail for the Reject feature. Another group had a more liberal interpretation. This group stressed the freedom of users. The idea behind this interpretation was to allow users to create work orders when deemed necessary and cancel them once they realized they were unnecessarily issued. There was also a group of people who took more of a middle position. This group was initially under the impression that they had understood the features but acknowledged that they really did not once these features were explicitly explained. This group accepted the final solution without any resistance. Finally, there was a fourth group who were indifferent about both of these features. This group in essence did not have a specific interpretation of the features. Members with this interpretation either did not pay particular attention to the discussion or they followed it with confusion and in the end accepted the dominant interpretation.

6.3.1.2.4. Calendar/Book Labor

For each work order the relevant trade and the number of hours of labor is assigned (please see Figure C.3, Appendix C). The MSUP assigns the relevant trade (e.g. electrician) to each work order from a tool called “Book Labor”. Originally the scheduler was the one booking the labor. However, in the configuration of 7i it was decided that since the MSUP was more knowledgeable about the trades relevant to various work orders, he would be in a better position to assign the trades (i.e. activities, specialty: e.g. plumbing). Some team members believed though that both the MSUP and the scheduler needed to be able to perform each other’s tasks if needed; hence, the need for both of them being able to assign trades. Under the ‘Book Labor’ tool, the MSUP and scheduler assign an estimated number of hours for the labor work.

One of the configuration issues related to Book Labor was related to creation of a calendar type tool that showed the availability of workers, workers' trades and employee information. It was suggested that the MSUP and the scheduler would be able to better plan the work orders based on availabilities. This would help in reducing backlogs. This idea was received favorably by the whole team. However, it took at least a month before the idea was openly discussed for implementation. The discussion of a Calendar feature, which was also mentioned under the names Activity and Labor Hours came up quite often after the initial proposal. A calendar was added to the configuration; however, this tool was not functional when the researcher conducted the follow-up interviews. In other words, labor was still booked through the old method of assigning the trade and hours without the knowledge of availability of specific maintenance personnel. This was done manually (i.e. adding a comment) by the MSUP and scheduler rather than using the calendar as a tool to book the labor.

One main problem, which made the task of configuring the Calendar feature difficult, was ambiguity of the definitions (i.e. the problem) in general. For example, the researcher was told by one team member that *"we discussed the job definitions back and forth and they kept changing their definition to make them as clear as possible… descriptions were meant for documentation and only the list of them would be on the system… they need to clearly and precisely define things"*. This person was more concerned about the ambiguity of the definitions more than the specific details of the calendar. This member had created a list in which he had documented the trades, their codes, and descriptions to use when working with scheduling work orders. He would use the codes to access the corresponding trades on the system. This was a temporary solution in order to be able to complete the simulations on work orders.

Another example of the ambiguity of the definitions was observed when the researcher inquired about the meaning of various colors (green, yellow, and red) used to highlight different days on the calendar. One explanation given was that red corresponded to full availability of the workers being booked, yellow corresponded to 80% of the availability booked and green meant booked way below total availability. Another explanation was that green meant less than 6 hours booked for each labor force, yellow was equivalent to 6 hours, and red color corresponded to 8 hours booked. In addition, scheduling was initially set to be done based on trade. It was then discussed to set the scheduling based on department. The reason for this second suggestion was that the initial method did not allow for calculation of the cost of a work order. However, when scheduling through the second method it would be possible to calculate the costs[20].

Besides the critical issues, disagreements, and delays in configuring the Calendar feature, this process also suffered from data entry errors. The tool was not fully functional until trades, employees, and their hours (i.e. availabilities) were entered in the system. Entering availabilities especially went through significant data entry errors. At the time of configuration, only a small amount of employee information had been entered in the system for the purpose of running simulations. However, this problem was noted by only a handful of members. For example, in one of the seemingly irresolvable discussions over the issue of the calendar, one member stated that *"now in the system they only have hours*

[20] Work orders that are contracted to vendors follow a separate rule for calculation of costs.

only for one employee. IT and these administrative staff should input the hours for all employees to have a realistic view of how many hours are available". On another occasion, another member explicitly told the group that *"this is a data issue that needs to be fixed"*. This individual was referring to the fact that some of the data entered in the testing environment in order to run simulations were fictitious or partial. This team member suggested that actual or complete data needed to be entered in the system in order to run and troubleshoot the Calendar feature. This suggestion was not taken seriously as the team did not follow up on the idea of checking the integrity of the data. This could have potentially partly addressed (or at least clarified) the problem. This resulted in a lot of confusion over whether the feature was set up correctly or not; hence, the tension in the group and a delay in the configuration process.

Another issue that added to the problems over the 'Calendar' feature was, similar to several other issues, related to the difference between the structures of DIVA and DIVB. The most significant difference was use of the different calculation methods for trade rates (in order to calculate a total cost for each work order) at DIVA and DIVB. At the time, DIVB was using an average trade rate for their maintenance work and DIVA did not have a system in place. Initially, it was suggested to use a 40% mark-up (i.e. adding an additional 40% to the average trade rate) on all DIVB's trade rates to reflect the nature of maintenance work at DIVA. This solution was met with significant resistance by DIVA members. The reason for this resistance was that this would inflate the rates for DIVA and if implemented would face significant resistance from DIVA workers. Finally, upper management decided that DIVA and DIVB were required to adopt a single trade system. It was decided to use an average rate, which would not reflect the true trade rate but would make it unique for both sub-divisions.

In sum, configuration of the Calendar did not satisfy the team as a whole. The issue of the Book labor feature, in general, demonstrated a situation where the problem at hand was not clearly understood. The vagueness of the problem is prevalent in the examples discussed above. This was also substantiated by comments that were made later on about the issue of Book Labor. For example, during the time when the pilot system was being launched, some members attributed the problem of the Calendar to unsuccessful migration of some of the configurations from training to the pilot system. These individuals did not recall that Calendar feature never became functional. The researcher realized that the issue of migration of configuration came up in a few cases. This led the researcher to believe that sometimes when no explanation was readily available to members, some tended to explain away the situation by issues that seemed irrelevant to the topic. This seemed to be more of a coping strategy. More examples of this type of coping strategy were observed in configuration of other features. As an example, politics at times seemed to be irrelevantly considered as a hindrance to the process of configuration.

6.3.1.2.5. Failure/Problem/Action Code

In order to route detail information about the failure of a piece of equipment, its problem, and the relevant actions to be taken to fix the problem, three different codes were configured (please see Figure C.2, Appendix C). These codes include: Failure, Problem, and Action. These codes were meant to make it possible for schedulers and planners to assign the labor, and number of hours for a maintenance job more accurately. Another purpose of these features was to standardize the processes and reduce errors as much as possible. These codes collectively[21], at a higher level, were meant to be used to collect data on the failure of equipment to be used for long term capital planning. There was a belief that information collected over time would allow failure analysis to identify weaknesses in business processes, design of systems, and operational utilization.

A *failure code* refers to a typical failure of a piece of equipment (e.g. valve failure, pump failure). *Problem code* refers to the type of work that needs to be done on a work order. Since the system is based on equipment class, it was deemed logical to establish problem codes. A problem code reflects the type of maintenance work performed on an asset (e.g. pump, valve). Examples of problem codes include GM for General Maintenance and PM for Preventative Maintenance. As the name suggests, an *Action Code* denotes the specific maintenance work required in order to complete a work order. In this sense, an action code is an instance of a problem code class. An example of an action code is INST for New Equipment Installation.

There were several problems around the failure code, the problem code, and the action code. Overall, in order to standardize these codes, the team seemed to take an ad-hoc approach. For example, during the configuration process the team started creating a list of common failures to be able to standardize them and input them in the system as failure codes. As one of the team members mentioned, *"you don't wanna create a big list but*

[21] These codes are collectively called analysis code.

you would want to come up with as many as possible to have a list and standardize it. That's what we are trying to do". As this quotation demonstrates, the approach they took did not follow a systematic way of standardizing failure codes. In other words, the team did not define a set of criteria based on which they would create the list. The idea of creating a list was merely mentioned and different team members started populating that list according to their preferences.

One main concern in the configuration of the problem, failure, and action codes was that of making these codes required fields. Another issue was to convince IT to configure them as required. The team was divided between two camps. People in the first camp believed that schedulers should not be forced to choose a code because that would limit how they could utilize the system. The reason for this limitation was that if schedulers encountered a problem that did not fit the definition of any of the codes from the already-standardized list of codes, they would not be able to complete the work order. This group of team members argued that planners and schedulers had to not only be guided how to use the system but given the freedom to plan/schedule the work orders without restrictions such as problem and action codes. This concern is reflected in the following quotation.

"All we need to do is to tell schedulers and planner to go about their businesses in the system... we only need to show them this process and to tell them how they got there but they need to decide on the rest".

People in the second camp believed that making these fields required would bring more standardization to the process and reduce the errors. This group further reasoned that this was an inevitable part of PubDiv's business process standardization strategy, by implementing 7i. A comment such as *"this goes back to having a business rule, it is logic, as simple as that"* was repeatedly heard from this group, referring to the importance of having a business process in place for every feature configured on 7i. Eventually, interpretation of this group was a more convincing reason for the team to act on.

Another problem that the team seemed to be grappling with was related to clarity of definitions. This kind of problem appeared in most situations. Overall, even though the team had spent several months in developing To-Be system business processes, they did not seem to have worked out the details about the definitions of various elements in each business process. For example, the situation explained below demonstrates confusion over terminology and the resolution of this confusion.

Some team members used the term Cause code interchangeably with the Problem code. Initially, when discussing the Problem code, this did not seem to create any concern for any of the team members. However, eventually one of the members asked for clarification between the two codes (i.e. problem and cause codes). He asked, "*if you don't know the cause how would you fix it?*" Some members agreed and others went on to explain that *"as long as you see the problem and know how to fix it why does it matter to know what caused the problem"*. The team leader gave a few examples. For instance, he said *"when you see a problem with a piece of equipment (for example you see a pipe is leaking) and you know how to fix it, you go ahead and fix it but you don't necessarily* *need to know how the leaking happened"*. These explanations by some team members seemed to convince the group that believed there needed to be a distinction between a problem code and a cause code. The person who brought this issue up was convinced and expressed that he understood the differences. Noting this issue here is intentional. From the review of documents by the vendor, the researcher found out that cause code is in fact a separate code within the Analysis code class, referring to the root cause of a failure of a piece of equipment.

In the above scenario, the confusion over the similarity or difference between problem and cause codes was merely an issue of clarity. This kind of problem arose repeatedly regardless of the effort of some team members to emphasize on the importance of the clarification of definitions. This kind of concern is reflected in comments such as: *"The important issue is the interpretation of the Action code, I really want us to know this clearly and be able to use it"*. This situation also demonstrated a lack of use of the existing information in order to clarify the meaning of various tools. Another example of a recurring problem, which was also brought up in the case of problem codes, was the differences between the structures of DIVA and DIVB. Once again, some team members argued that because of the simplicity of the business processes in DIVB and centrality of its structure, it was possible to adopt standard problem codes. Even though the researcher heard comments about concerns such as *"in DIVB it is all in one place, everyone knows everyone else, so, it is easy to know how to assign jobs… DIVA and DIVB have been using different methods; for example, DIVA uses the problem code whereas DIVB uses job type to define a problem".*

In sum, issues around the configuration of problem, failure, and action code were several and of different nature. One problem stemmed from not having reached agreed upon definitions of these codes and their functions. Another problem was over whether or not to standardize the list of these codes. The team members were divided over this issue because they interpreted the amount of freedom that the users of these features would have to be given differently. A group of people believed in total freedom of the users in describing the codes. Another group argued against this idea because they believed that by standardization (which meant limiting the freedom of the users) they would achieve more standardizations of the work order management in general.

6.3.1.2.6. Material Purchase

There are three different ways to purchase and use material/part/items. This process is referred to as Material Management System, EPO, Material Requisition or Bench Stock. These three ways include:

1) Purchase order (PO) in-house stores/stock items: These items have a Pub Org part number.

2) Purchase order non-stock item (a non-stock PO): This method is usually used for large items, which are more costly (e.g. motors). These items have a manufacturer number.

3) P-Card: This is a credit card like purchase in order to purchase outside the system. There is a spending limit for every person. If the spending limit is not sufficient for the purchase of the item required, a non-stock PO is used.

Configuring the Purchase feature proved to be challenging. The main reason for this challenge seemed to be not having a clear business process for purchasing in place. This was explicitly acknowledged by the team. For example, the researcher observed notes such as *"the WMSTF team has determined the need to establish business rules dealing with different Purchase Types"* in meeting minutes. MP5 did not have a specific feature for purchases and the MSUP entered the material and purchase information manually as a comment. When done in this way, no detailed information about specific material used for each work order was provided. For example, information such as "25 ft cable" or "switches" would be entered in the system. In addition, the exact amount of material used for each specific work order would not be specified. For example, a bulk purchase of specific material would be made to use the material for a WO and the left over material would be stocked for later use. The specifications of how much of the material was used/left would not be entered in the system.

Therefore, one of the objectives was to configure 7i in a way that it would make it possible to tie back each WO to specific material (whether stock or non-stock items) and their costs. Such a feature would estimate the cost of each work order from the onset. The team appeared to *"know the problem but not know the solution"*. The team leader for example stated *"we are trying to find commonalities between how DIVA and DIVB conduct purchases and implement that in 7i… We are not trying to create stores [referring to one possible solution to address Purchase issue] and all that … I don't think that DIVA has that much of detail information about their purchasing process though… they don't use parts [referring to purchase of parts for equipments]"*. These comments and similar issues that came up during this configuration demonstrated that there was no clear common business process in place for the Purchase feature. This had not been addressed in the flow charts either.

In one of the sessions when the team was "*determined to address the issue once and for all*", different members suggested different solutions. First, the team leader asked for "drawing a logic tree"[22]. This suggestion was not taken seriously and the discussion continued. One member, Joe, proposed creating a list of inventories and populating it with products (to represent different parts) that could be used later. He proposed this because he believed that *"90% of times the products used are repetitive"*. This idea was opposed by another member, Kaleb, stating that they *"don't have a warehouse, the guys use P cards (purchase cards) and they keep purchasing"*. He also mentioned that *"various products are from different manufacturers, and had different prices and it did not make any sense to input them in the system"*.

To defend his idea about creating a list of inventories, Joe insisted that accumulating various products in the system would pay off because eventually it would contain all possible manufacturers and various products that are used on a regular basis. Therefore, he told the rest of the group that meanwhile *"you have to keep creating them"*. Kaleb suggested an alternative: creating a fictitious warehouse and assigning a certain number of units to each product, and assigning prices to them. His reason was that if Joe's suggestion was implemented, they would have to create a new product every time and that would lead to *"creating a million dollar list and that if an inventory-like list is used*

[22] As will be discussed in the second set of data analysis (section 6.3.2), some team members were highly in favor of the use of the flow charts and diagrams to help understand the underlying processes for configuring the features.

every time they use a product it would decrease it from the inventory which would eventually run out". Kaleb acknowledged that if this alternative used, they would not be able to tie back their work orders to their inventory. He argued that because of their various attributes the products would be hardly repetitive ones. For example, if two valves have the exact same specifications but are purchased from different vendors, they are considered different items. For this reason, Kaleb argued that the chances of using the exact same items repetitively would be low; thus there would be no value in creating a long list. However, Joe argued that if the team would be persistent in creating the list, eventually they would be able to come up with one that contained most of their needed products. He reasoned that this was the case because the products as well as the number of vendors with whom Pub Org conducted business were limited in number. Even though most other team members were skeptical about this solution, he believed "*it would pay off*".

The team leader later on discussed the second solution (i.e. creating a fictitious warehouse). He argued that implementing this solution would be too time consuming. He emphasized the importance of understanding what each way of acquiring material meant. He said: *"I want someone to become subject matter expert in material acquisition. We know the front part"*. He also explained that they needed to merely emphasize on non-stock material in this project. He added that,

"in order to be able to tie the costs of the material back to work orders in current format we have to be able to bridge to our SAP system and financials. That's a big project and cannot be done with this group. We can probably handle the P card and PO cases but stock one we won't be able to… The key is to [eventually] see if there is any one module

to get all this done, unless we are gonna connect to SAP and get a snapshot of what's there and a snapshot of what's here, etc. all the transactions we go through we have to come back and track them in the most simple way. I don't want us to jump to different screens. Then this tab becomes "Purchases" [instead of current P card purchases], just purchases and then we go from there… For now, I guess we would have to do it manually and give reports to the financial department".

Thus this issue was resolved. This meant creation of a cost report manually for each work order after the work order was complete. In other words, this solution did not address the problem of work order cost estimate through configuration of the Purchase feature but through a work around.

Some examples of decisions over the Purchase feature that were addressed through configuration included grouping purchase types, and choosing proper terminologies. Initially, each purchase type had been configured under a separate tab. Some team members found having different tabs unnecessary and asked for all of the purchase types to be under one tab, Purchase (as had previously been suggested by the team leader as well), and adding a drop down menu for various purchase types. In addition, it was unanimously acknowledged that the initial use of a title such as non-PO purchases was not appropriate because it was unnecessarily wordy. It was agreed that non-PO purchases was actually the same as P-card purchases. Therefore, the use of P-card purchases was suggested as it denoted a clearer and simpler wording. In another example, the team realized that the initial terminology chosen to reflect various cost types was not meaningful to any of the team members. These categories included: Own, Contractor, and Part. The team deliberated over the meaning of these terms. The team had a guess for the meaning of each: Own referring to in-house work, and Contractor denoting work orders that were contracted out. The team was not able to guess what the meaning of Part was.

After a long debate, the team decided "*to understand what each option meant and to find better replacements*". This example shows that the configuration process, for most part, seems to have had followed a "patch work" and ad-hoc style. In other words, the team seemed to at times configure a feature without much reflection on its meaning and terminology. This resulted in confusion later on when the team ran simulations.

6.3.1.2.7. Asset Tracking

One of the main goals of this project was to configure 7i as an asset management besides a work management system. In fact the software was initially called Asset Management System by the team. An asset management system is one that identifies, tracks, locates, and analyzes physical assets. Over time, this name was replaced by Work Management System, (abbreviated as WMS) which was also officially used by the team and in all correspondences. In MP5 no data was available on assets. MP5 merely stored the names of the equipment and any further details needed to be checked manually. Asset was the smallest unit in a facility. The asset hierarchy includes[23]: Facility or Location (i.e. a specific geographic location), System (e.g. piping), Position (e.g. pump), Asset (e.g. valve). This is referenced in the system by LSPA an abbreviation corresponding to Location, System, Position, and Asset. LSPA was configured to be a form (demonstrated by four tabs on the system) in order to upload information on the assets throughout the

[23] The examples provided here together denote: a valve on a pump in the lighting system of a specific facility.

whole hierarchy of each. Even though documenting the hierarchy of the assets was initiated in the system and specific features were configured, asset management never became functional in 7i. In other words, currently, the software does not function as an asset tracking system.

The team was not able to configure any tools for managing assets. In fact, even though the team repeatedly mentioned the importance of asset management and the need for configuring relevant features, only one initiative was taken regarding this problem. Categorizing the assets based on their criticality was proposed by the team. DIVB became proactive on this issue. They developed a model for ranking and rating the criticality of pieces of equipment in order to upload them to 7i. The following comment was found in the team's meeting minutes regarding this issue:

"The rating is to indicate the relative importance of an asset to the overall operations and production… it was argued that this rating may assist for managing asset by criticality and will assist the planners and schedulers to prioritize the workload… The team defined characteristics that cover a wide range of the industry attributes that should be used to analyze each maintainable asset. These characteristics are currently being redefined/modified". At the time, these characteristics included:

- Mission and customer impact
- Safety and environmental impact
- Ability to isolate single-point-failures
- Preventive Maintenance History
- Corrective Maintenance History
- Mean-Time-Between-Failures or "Reliability"
- Probability of failure
- Spares lead time
- Asset replacement value
- Planned Utilization rate

Operations and maintenance managers (or their delegates) were then asked to *"to provide weights using a scale from 0 to 10 to identify significance to our business or to simply assign an Overall Criticality Ranking on a scale from 0 - 100, with 100 being the most critical"*. However, at the time of this study, DIVB personnel did not identify any of their assets as critical, thus *"reports based on the equipment criticality entity could not be generated"*. This process did not proceed further. Even though the asset tracking module was a subject of the configuration process from the beginning and was one of the main objectives, it did not become functional during this project.

In the follow-up interviews, the researcher inquired whether a new solution had been found regarding configuration of asset management. It was found that 7i was not yet being used as an asset management system. One of the interviewees, Mike, discussed a theoretical solution he had developed to address the issue of asset management. However, since the system was already in place and since in the past the team had not taken his ideas seriously, Mike did not consider discussing this solution with them. He explained his theory as follows.

"The theory in my head for asset management is that in every position, we focus on position [from LSPA hierarchy which was explained above] for work order purposes and not go to asset because in one position there are at least 4-5 assets (e.g. pump, valve). When you multiply there are so many assets. That's why we could not manage it. And that's why DIVB was more able to manage assets because they are smaller, they are less spread out, there are fewer assets and the processes are more straightforward".

Therefore, Mike believed that it would be more convenient to manage the assets in the 7i system at the level of their Positions rather at the level of the Asset. This was particularly the case at DIVA because of the larger size of DIVA and its spread-out structure.

Mike added that *"in order for asset management to work, they would need to approach them like work orders as well and asset management people do work orders on them and record everything (i.e. track assets with WOs)"*. He emphasized that *"Asset tracking and WOs for maintenance are separate and un-integrated that's why the attempt to implement both on 7i was unsuccessful… there is basically no tracking mechanism for assets"*. In his view, *"our biggest problem is that they tried to put the configuration together too quickly. Also, they all of a sudden populated the system with data"*. Mike believed that asset management had to be handled in a similar manner to work management. However, he argued that first a tracking mechanism (similar to that of work order management process) had to be put in place. In addition, Mike was concerned that inputting real data to the system before the configuration of the system was completely satisfactory made it difficult to make systematic changes, the kind that asset management configuration required. Interestingly, other members had a different idea about populating the system with data: *"we try to put as much information there as possible so that they can extract what they need"*. This group believed that the sooner the data was input in the system, the sooner they would be able to work with the system and identify the problems.

In sum, Mike seemed to be pessimistic about any further improvements to the system. He repeatedly mentioned that the system could have potentially become a sophisticated system that would have addressed the needs of PubDiv. However, he believed that the configuration of 7i did not achieve this goal. The nature of the challenge in the case of asset management was different from the types of challenges the team faced when configuring other features; yet, the team approached it in the same manner as other configuration problems. In other words, asset management itself represented a whole process (i.e. a system) similar to the work order process. The team seemed to have needed to define the process clearly first; perhaps by developing business processes and flow charts. However, the team considered it as another individual feature that needed to be configured on the 7i system. Therefore, it appears that not considering a greater scope for asset management resulted in not fully grasping the problem and eventually ignoring the configuration of this feature altogether.

6.3.2. Second Set of Data Analysis: Organizational Constraints

The second set of querying/analysis was not detached from the first set since several of these themes emerged while the researcher was writing about Chronology in chapter four (4.3.1.2.1), Work orders (section 6.3.1), and Features (section 6.3.2). During the first set of data analysis, the researcher noticed the prevalence of three organizational dimensions of the configuration process: flow charts, business processes/rules, and training. These are discussed in sections 6.3.2.1 through 6.3.2.3. These themes were more salient when the researcher was querying/analyzing the data related to interpretive flexibility, more specifically related to three nodes: Philosophies, Activities of Individuals, and Disagreements. This meant that these three organizational dimensions were major recurring attributes in the data that divided the team members were salient in the discussions/decision making. These attributes are discussed in sections 6.4.1 to 6.4.3.

6.3.2.1. Flow Charts

Flow charts (e.g. work order process flow char exhibited in Figure 6.1) were used to describe work processes and used as the basis to configure the software. Flow charts were one of the most controversial topics during the discussion of the configuration of the system. Team members' interpretations were divided about flow charts. A group of individuals found flow charts a necessary tool worth the time invested in developing them. The other group saw no value of the flow charts; this group felt that the time spent on developing flow charts was "a waste of time". They believed the team would have benefitted more if this time had been spent on the actual configuration of the software.

The flow charts reflected the To-Be system. The WMSTF team started with the As-Is system, which was described in the reports prepared by the vendor at the beginning of the 7i implementation project. The flow charts consisted of one main chart, which mapped the processing of a work order from beginning to the end (Figure 6.1). The team spent a significant amount of time developing the flow charts. One of the reasons for the lengthening of the process of 7i implementation that was mentioned were the disagreements over what the new (To-Be) processes would have to be. These disagreements were partly due to the disbelief of some team members in the value of flow charts. Another major issue that led to disagreements about the To-Be business processes was the difference in the structures of DIVA and DIVB. The main goal of developing flow charts besides having the processes mapped out for the configuration was development of common business processes for DIVA and DIVB.

Therefore, one group believed that there was considerable value in having flow charts. This group argued that it was natural to spend time on their development. They saw value in it; the necessity of developing flow charts was a given fact for this group. They believed that they needed to know the "ins and outs" of various processes. It was repeatedly suggested that *"team members should view the flow chart to get acquainted with all roles and duties"*. Other such comments were made frequently.

"We own this process [pointing to the flow charts]… it took a long time, we spent months for business processes, flow charts, and especially the descriptions… we beat this horse to death but it was necessary… flow charts are the visual part of the configuration… IT definitely needs this to be able to configure based on the processes mapped out here… Understanding the To-Be system and the processes are very important; even in technical stuff, we have to keep going back to the flow charts/business processes".

"We need to keep remembering what we are doing, what we are striving for, what we have to stick to, It has taken us a long time to come up with these processes as a group and we want to remember that in all our steps".

Later on, when training was being investigated, members of this group believed that flow charts and business processes were also essential in training and that they needed to be included as a part of the training package[24]. The idea was to compile the flow charts, a document about the background of the processes, and the definitions of priorities as a training package.

[24] This is explained in detail in Training section (6.3.2.3).

However, some members of the first group believed that the only value of the flow charts was in regards to the configuration process. They acknowledged that once the system was rolled out, *"people are not even gonna see these flow charts, they just see the parts that are related to themselves"*. This implied that the sole use of flow charts was for configuration purposes. This interpretation also meant that this group, like the third group, acknowledged that flow charts were not usable by regular users of the system, including field workers. This group believed that flow charts did not reflect the whole process. For example, one person was concerned that *"there is a second layer of flow, which is the human paperwork side. This is not captured by flow charts… These need to clearly be identified such as who does what at each point of the flow chart"*. However, this group found it extremely necessary for the configuration of the system. Their reasoning was that the configuration team needed to know the foundation of the system, i.e. the business processes.

The second group, consisting of those individuals who did not view flow charts as valuable, believed that the functionalities of the system were being compromised because they *"were trying to fit the system to the flow charts [or business processes in general]"*. They believed that they *"should've looked at the system and see what it does and then fit/change our processes"*. This group added that the interpretations of the field workers had not been incorporated in developing them. They argued that *"this was a waste of time and workers were not included and did not and would not understand it… and that this was not a good depiction of what was going on because it is very different when you are out there in the field"*. This group resented the flow charts even after the software was already under configuration. Additionally, this group believed that the team as a whole did not hear their voice regarding dissatisfaction with flow charts. For example, individuals made comments such as: *"we agreed to disagree to move on... This was just shoved down our throats, we didn't want it"*.

The second group was particularly opposed to the idea of using flow charts as a training tool. They were not convinced that other employees, especially field workers, would understand or benefit from learning the processes through the charts. One of the group members went even further to blame the leadership for this problem. The following comment reflects this concern.

"I think the leadership was wrong. Bob was the wrong person for this project. We spent 1.5 years[25] *on the flow charts which to me really don't have any use. To line workers those are meaningless"*.

A third group that was more neutral towards the flow charts did not necessarily believe that development of the flow charts was essential. However, this group accepted the use of flow charts after they were developed and made frequent references to them. This group was optimistic about how the process of the development of the flow charts had reached a stable point. This mind-set was noticeable from comments such as: *"we had a lot of heated discussions… you missed all the arguments [followed by a laughter]…. but there is light at the end of the tunnel… there is more agreement now"*.

[25] Interestingly the researcher was given a variety of numbers for the period of time that it took for the team to develop the flow charts; from a few months to 2 years!

This group believed that disagreements were addressed and was surprised to find out that the controversy still persisted. For example, the comment below is an example of comments made by this group reflecting this belief.

"I thought we were doing this in a unified way not DIVA vs. DIVB. I don't know what happened along the way. So these business processes (showing the flow charts) didn't you guys agree on it? Wasn't it a unified process?"

The first and third group acknowledged that there had been a lot of disagreements throughout the process; however, these groups believed that the disagreements over the flow charts had been resolved. Numerous complaints were heard regarding the flow charts from the second group. However, it was a surprise to us that this group did not voice their dissatisfaction more explicitly. This could have been due to the fact that the flow charts had already been developed and that voicing their dissatisfaction would not change the situation. The researcher believes that the second group's hesitance in discussing their dissatisfaction did not reflect a power issue (i.e. fear of voicing their opinion because of their power disadvantage). This belief stems from the observation that in other situations some individuals from this group had been vocal about their disagreements. In addition, the members of this group did not communicate concerns about power disadvantages in this case.

In sum, regardless of the disagreements, flow charts were used extensively when configuring the software. Generally, flow charts were used as visual pictures of the processes, a simplifying tool, a point of reference, and representative of the unified To-Be business processes. Usually, when the group was simulating a scenario on the system they would refer to the processes on the charts to make sure they were covering all the processes and steps. Reference to the flow charts was more frequent when there was a problem in the configuration process (e.g. when configuring the Inbox). Flow charts were also used often to attract the attention of the team to particular configuration problems by keeping them focused on the processes. During the process of configuration, the team would constantly go back and forth between the software and the business processes depicted by the flow charts.

6.3.2.2. Business Processes/Rules

Issues around business processes went hand in hand with the controversy of flow charts. However, there were some distinct attributes related to business processes that divided team members. As discussed before, one of the objectives of this project was to develop standard business processes common to both DIVA and DIVB. The main concern shared by most team members was that the business processes at DIVA and DIVB were not similar. This dissimilarity indeed created some obstacles for the configuration of 7i.

The group that felt more strongly about the difference of the structure between DIVA and DIVB complained that *"the business processes were initially not standardized at all… there was no standard, no guidelines, no policies on how to standardize them"*. A vendor that had worked with Pub Org in the past was hired to conduct an evaluative study of the As-Is system at DIVA; "*to study everything*". As-Is business processes at DIVB were identified internally by a team from DIVB.

The WMSTF then developed the To-Be business processes based on the As-Is processes of DIVA and DIVB. In this process the team looked at the history of both DIVA and DIVB and the actors/roles within both sub-divisions. Team members signed off on the developed To-Be business processes. The goal of the configuration was then to map these processes to the system. As was discussed in the process of development of the flow charts, the WMSTF was divided on their interpretation of To-Be business processes.

Unlike what occurred with the flow charts, there was no group that took a neutral position towards the issue of business processes. One group was satisfied with the processes that were developed and the other group was not. The former believed that it was feasible to have common processes for DIVA and DIVB. This group is called Optimists in this section. The latter group believed that these DIVA and DIVB were inherently different and, thus, had to "do business differently". This group is called Pessimists. The difference of interpretation, unlike that of flow charts, was not about the belief in the essentiality of the process. Both groups acknowledged that developing To-Be business processes was necessary, though challenging; "it was a monumental task", said one person from Optimists group. Other similar comments, are depicted below.

"The biggest challenge was standardizing the processes even with the groups being involved... every individual has his/her point of view of how things should be done… There were a lot of disagreements".

"There were a lot of complications in coming up with an agreed on To-Be system…Still some people who have totally different interpretations of the whole process".

Other reasons for disagreements that were mentioned partly stemmed from differences in interpretations of different individuals. Interestingly, one member explained this difference in interpretations in the form of altruistic versus individualistic perspectives:

"You have to have this in mind whether you look at it from an altruistic perspective or an individualistic perspective… if you are only concerned about your own task or you think about the whole organization or at a higher level".

"We touch people's lives. The end users go way beyond the users in the field. In our line of work, our end users are people… some people do not look beyond their own work and to how we impact people's lives and how this system and these processes go all the way and affect that".

Other comments regarding perspectives of individuals towards the system were made:

"Some people think that this system is a policy device; it is here to watch them… but no, this will help increase efficiency… It helps validate exactly what you need… It goes back a lot to business justifications (e.g. system report)".

"Some people see this as a tracking device but they need to somehow be able to see who is accountable for what. For example, when a work order keeps coming back to a person for redistributing, now that counts as work/hours for that person… I also think that's not efficient when the work orders kept going back to one person so that the right person is found to do the job and this is extra work hours. There has to be a way to match the right person to the right work order in the first place".

This quotation refers to cases when work orders are unnecessarily delayed by going to employees who are not relevant to the work. This delay causes extra work hours and thus inefficiency. In addition, those individuals who receive the work order but are not relevant to the work are able to bill work hours. The concern in this quote is to prevent individuals from taking advantage of the system (i.e. by gaining work hours irrelevantly).

However, what divided the two groups was the belief that since DIVA and DIVB had different structures they needed to have different To-Be business processes. In other words, the Pessimists group believed that the system needed to offer flexibility, expressing this interpretation by statements such as:

"the configured system needs to be more flexible to work in different organizational structures… even if you go down the hall some people might do things differently".

Since coming up with standardized business processes for both DIVA and DIVB meant that some roles and responsibilities had to be adjusted to fit the new processes, the Pessimists group was not satisfied with adopting such an approach. The individuals from the Pessimists group were from both DIVA and DIVB. These individuals shared the same dissatisfaction, not believing in common business processes for DIVA and DIVB. One source of dissatisfaction could be that having to change business processes to fit within the common standard made the job of configuration more challenging. However, if an individual was from DIVA, his dissatisfaction would probably also stem from the fact that that DIVA had to comply more and compromise more on business processes. To-Be business processes reflected more of DIVB's processes; this could be because the processes were more straightforward at DIVB than at DIVA. This situation reflects the case when one interpretation can have different meanings for different individuals. In other words, standardization of business processes meant something different to individuals from DIVA and DIVB even if these individuals all shared the pessimist interpretation of business processes.

However, the team as a whole recognized the need to improve To-Be business processes. Members from the Optimists group emphasized the importance of business processes repeatedly. This group believed that developing the business processes was the foremost priority. Comments such as the following were made by this group:

"I would like to see the assessment of the business rules. Training comes second… We need the system to be robust first and foremost that's where the budget should go first… I wanted to find flaws and problems but I want them to learn all these processes really well".

The team believed that eventually the vendor would assist them with improving the business processes. The researcher repeatedly heard comments that expressed this hope. This is reflected in the following comment.

"Datastream will help with improving the business processes also… these guys know the best business practices of the industry… they should know how to do the processes the best way… they know all the tricks".

However, during this study the assistance from the vendor was not sought by PubDiv mostly due to budgetary limitations and partly due to mismanagement (i.e. management not following up with this request).

The general idea seemed to be that the system was quite flexible with a lot of capabilities. The prevalent belief was that the WMSTF was not able to take advantage of the system. In other words, they attributed the reasons for not being able to configure the system as desired to organizational rather than technical issues. This belief was echoed by the team members in comments such as the following.

"When looking for a new system, they gave a demo for this one… I found it quite flexible and easy to configure… We met with other vendors too… the systems were limited… they did not conform to our processes… the screens were dark and hard to see… this system was the closest to our processes and what we need".

"We still need to change some of our processes to conform to the system because the system does not do everything exactly we do our processes and we have to change some of our processes for those limitations but still this with this system we had do to the least compromise (this is my word). Of course first we studied and developed our new processes independently of the application".

"This system was also easier to manage in-house without a lot of support needed from the vendor… The team as a whole agreed that this was a better system… I also saw ease of interpretation, and ease of use. The interface is much easier than other systems… Others were monochromatic… I don't need glasses but I had a hard time to look at the screen with other applications".

The Optimists believed that it was natural to spend a significant amount of time on developing To-Be business processes: *"It takes some time to reach a common understanding".* The Optimists believed that they had to get the work done somehow even when they did not have business processes/justifications in place. For example, comments such as below were made.

"There is no business rule to support this logic[26] *so meanwhile it has to be done this way… this also goes back to having a business rule in place to get this logic… for now we have to go the inefficient way".*

In sum, the reasons for disagreements over the issue of business processes were various. However, the main point that clearly divided the team members was the difference

[26] This referred to a decision regarding routing the work orders through the OA rather than the MSUP by taking the option of "ready for closure" out of the MSUP's menus. This decision did not have a business rule in place. In fact, it was mentioned that the MSUP needed to be able to bypass the OA in the cases of emergency.

between the structures of DIVA and DIVB. A group of optimist individuals believed that regardless of the differences between DIVA and DIVB developing common business processes for the two was possible. This group believed strongly about standardization of the processes because this was one of the major goals of the 7i implementation project as a whole. A second group, members of which seemed to be relatively more pessimistic, believed that it was necessary to maintain the differences between the business processes of DIVA and DIVB in the 7i system. This group believed that business processes needed to closely reflect the way these two sub-divisions were structured.

6.3.2.3. Training

The significance of training was almost unanimously acknowledged. The researcher repeatedly heard that "*when implementing, the key is training*".

The initial plan was to contract the training to an outside vendor. Several vendors submitted request for proposals. The WMSTF team had meetings with a few vendors. If contracted, the vendors would have been required to initially develop a manual *"of interactive model training 20-30 people… the manual could even be a benchmark of business rules aspect of it the system"*. From this quotation, it is apparent that the emphasis of the group advocating understanding the business rules was prevalent in issues around training as well. Vendors promised to provide DVDs and web-based training material, as well as to train an initial group of employees representing various sites. The WMSTF team preferred to contract the training to Datastream since they were the developer of 7i and had implemented the system in other organizations. However, due to budgetary limits, no outside consultant was hired to conduct the training. Disapproval of the budget for hiring outside contractors for training was partly due to the fact that the system was still being configured. Management argued that the priority was on the functionality: "*functionality first, then training*". Therefore, the idea of spending money on training was not appealing to the management at this point.

Once it was realized that the training had to be carried out with in-house staff, different suggestions were made on how to go about the training. Some of these suggestions are listed below.

- Teaching employees from DIVA and DIVB the business processes using the flow charts through a one-day seminar. The presenters would be from both DIVA and DIVB. The business process training would be followed by technical training of the employees on the 7i system.
- Developing manuals with snapshots of the screens and explanations of the tools and processes. The reason for this suggestion was: "*if you write the steps and show them the snapshots, it should be really easy for everyone to figure out"*.
- Developing a training packet, which included documentation of what the WMSTF team had done, flow charts, backgrounds of the business processes, priorities, definitions, and explanation of the Inbox.

Disagreements over the training were based on several issues. One issue centered around the priority of training the employees on business processes versus the system. This concern was explicitly voiced in this quotation: *"First, I would like to see the assessment of the business rules, training comes second"*. Another issue revolved around the priority of preparing training documents versus training employees on the actual work management system.

Another point that divided the team was, similar to most other issues, related to the difference between the structures of DIVA and DIVB. One example of the distinction between DIVA and DIVB was that maintenance supervision at DIVB merely existed in the context of mechanical (and not electrical) work. However, it was decided that in the new system there should not be any distinction or deliberation on the consequences of the differences between DIVA and DIVB.

The main difference in the structure of DIVA and DIVB was related to the centrality of the structure at DIVB. DIVA was scattered through several facilities. Therefore, the processes were more straightforward at DIVB. The group that believed the team was not ready for training was specifically concerned with employees at DIVA. The processes at DIVA had changed the most. Besides, 7i had been already used to some extent at DIVB. Therefore, this group was concerned about the effectiveness of using the same training material as DIVB. For example one individual from this group stated that *"we are not comparing apple to apple, we are comparing apple to grapefruit… why don't you guys start from a division that hasn't used 7i before, then we would be comparing more of apple to apple… in DIVB is all in one place, everyone knows, in DIVA is all over the place"*.

Another major disagreement about training was the readiness to start the training (and pilot test). The group that advocated speeding up the process believed that they were ready, as reflected in the following: *"It's easy. We have them all. It just needs to be* *compiled".* When others argued that the employees' readiness depended on how they would perceive the system, a member from the first group replied *"it's not about how people perceive it. It's about how we make it to be… we are the champions".*

From the group that believed they were not ready, comments such as the one below were heard:

"What are you gonna train them on… the system is not ready. We are not anything near ready… no, we are not ready. All we need to focus now is to focus on business processes… what are you trying to accomplish, what do you want to train them on? The processes or the system? The system is not ready… they need the system, they know the processes… the processes take at most one day to cover… train on what? the processes might change and thus the system… we need to work the bugs out first… we are not ready for the training".

The other group would respond with comments such as:

"It's never gonna be ready. It is at a stage to put the training material together… You don't need the system to train on the processes. It's just the processes; system just enables the processes… If Tania learned it they can learn, too [referring to the fact that Tania has just recently joined and she didn't know anything about the system and had not worked with MP5]… all we need to do is to tell OAs and MSUPs to go about their businesses in the system… yeah we need to show them this process to tell them how they got there but they need to get the process ready to teach them… we just put a training manual based on what we have… for the rest this team will help once the email notification about training and pilot has gone out".

The insistence on launching the pilot system and starting the training created a lot of tension between the two groups. This topic was discussed in several sessions. However, finally when one of the senior managers visited and expressed the need to speed up the project, it was decided that the training would start and the pilot system would be launched. The senior manager stated:

"We've been involved in this project for almost 2 years now. I know there are a lot of personality clashes… we don't want a double standard here [though], some people are ready for training and some people aren't… I had no idea of the management of this project… you don't have enough people… but I'm not looking for the whole thing to work… I want us have something for the pilot… the purpose of the pilot is to find the flaws… and most people in this room will become trainers later on".

Several others had the same idea and expressed that *"all we want is to put this behind us"*. Thus, it was decided that the training material would be put together as soon as possible and the pilot system would be launched. The group who supported the idea of launching the pilot and starting the training argued that *"we don't need a whole lot of training for pilot because that's the point of the pilot to learn for the actual implementation… why don't we do one plant, these guys go bring you information from the field, use it then go to other plants"*. The group which was opposed to the idea of the readiness for training argued back that *"what are you gonna leave them with after you have trained them? Even you, who is so brilliant, do not have that graphic of a memory to remember everything. These people need to take something with them so that they can go back to and remember. These people haven't been involved with the process like we have and even we don't remember everything so how do you expect them to learn without any training material. They need some written material to take with them"*.

The group in favor of launching the pilot and starting the training, was concerned that they would need some time to develop training material and conduct comprehensive training sessions. Members from the first group argued that the concern over training material was baseless: *"that's the problem, you guys think you should leave something with them…the flow charts can be used as the guide and inbox shows all they need to be able to use the system…. pilot becomes about people"*. They added that in the first stages of training they would start with employees who would be more ready for training; thus, little time would be needed for preparation. This quotation reflects this opinion: *"you start with the more apt people first. You don't try to crack the toughest knot first"*.

The disagreement over the readiness for training became so intense in one session that the members of different groups demonstrated explicit resistance to one another's ideas. For example, when one person from the group in favor of starting the training tried to explain his reasons by pretending that they were all in a training session[27], others who opposed the idea of starting the training refused to participate. The tension during this session was so high that the group against training said they *"felt that they had been stabbed in the back and that decisions had been made behind [their] back"*. A member from this group stated that *"everyone's is setting their own individual thing (agenda)... Bob shouldn't just stand up there and say it there because some of us don't come to all the sessions so we don't get informed about a lot of things. We don't have good communication"*. This was referring to a few occasions when some individuals had communicated important issues merely verbally in the meetings without notifying others about them. This session ended without arriving at any solid conclusion about how to approach training.

The final decision about training was to come up with a solution that would satisfy both groups. Therefore, it was decided that they would in fact have a training manual. The final training material/plan, however, still met more of the pro training group's expectations. It was decided to develop training manuals based on different roles (e.g. general requester, operation authority) and conduct hands-on training sessions site by

[27] Bob started to explain as if he is teaching from the board that has the Inbox configurations.

site. The training material was not as comprehensive as what the group against readiness for the training had desired. It was said that *"this is just a simple training document, actually I call it a guide. So it's going to be just the straight process"*. The training material indeed merely included snapshots of the steps of the main process (such as those exhibited in Figures C.1 to C.3 in Appendix C) accompanied with minimal explanations. The motto that drove the process was "*to train the trainers*". The idea was to train an initial group in each site and then for those groups to train new groups of employees.

6.4. Summary

In this chapter several of the controversies that happened around the configuration of 7i, whether of a technical or organizational nature, were discussed. In each case, the situation, in line with SCOT approach, was followed from the initiation until the closure of the problem. The first step in analyzing each situation was to identify various interpretations around the issue at hand (e.g. Priority feature, or Business processes). Then the ways that the controversies were addressed based on interpretive flexibility of the issue were investigated.

This analysis was presented in two sets. The first set focused on the analysis of the technical features of the system and their configuration. The first set included the Inbox and KPI, Priority, Reject/Cancel, Calendar/Book Labor, Failure/Problem/Action Code, Material Purchase, and Asset Tracking features. The second set examined the controversial organizational issues which influenced the process of configuration. These organizational dimensions were identified while the first set of data was being analyzed.

More specifically, the topics of flow charts, business processes/rules, and training were repeatedly brought up during the configuration process. These were not issues that, from the onset, the researcher had anticipated observing based on the theory and attributes of software configuration. They emerged when the configurations of the technical features were being examined. When studied more closely (in the second set of data analysis), their influence on the configuration process was indeed clear.

In chapter seven implications of the findings of chapter six are discussed. Based on the insights gained in chapter six about the configuration process, a mechanism for the configuration process is developed in chapter seven. This mechanism, accompanied with a definition of the configuration process of packaged software, aims at an analytical generalization of the themes that emerged from the empirical analysis of the data in chapter six.

Chapter 7

Discussion and Implications

7.1. Introduction

In chapter six, the qualitative data of Pub Org case was analyzed through the theoretical lens of SCOT, remaining faithful to the overarching research questions of this dissertation:

1) Why is packaged software configured a specific way and not another?
2) How does this configuration occur?

In line with SCOT, the data analysis revolved around the controversial technical features and organizational issues related to 7i. The objective of chapter seven is to synthesize the understandings gained through this study. In doing so, the goal is to reach a theoretical generalization about the configuration process. Accordingly, a mechanism for the configuration process is constructed based on the findings of this study. This mechanism, accompanied with a definition of what constitutes the configuration process, addresses the two main research questions.

In section 7.2, the development of a mechanism for the process of configuration and its technological, organizational, and individual elements is discussed. In section 7.3, contributions of this study are assessed according to Lakatos's (1976) research programme. Implications of this study for research and practice as well as the limitations are discussed in section 7.4.

7.2. Discussion

The first part of this section (7.2.1), discusses the development of a mechanism for the process of configuration based on the literature. Section 7.2.2 then delves into the discussion of the development of a mechanism of the configuration process based the findings of this study which were examined in chapter six. Section 7.2.3 then discusses the findings of this dissertation about the concept of the information systems user.

7.2.1. A Mechanism of the Configuration Process Based on the Literature

Mayntz (2004, p. 237) defines mechanisms as "*recurrent processes generating a specific kind of outcome*". In this regard mechanisms represent theoretical propositions and depict the ontology of a phenomenon. In other words a mechanism depicts the objects of a phenomenon as well as the process through which these objects are intertwined and interact with each other to reach a specific outcome. Since the configuration is a process (i.e. it starts with a goal and ends in an outcome), a mechanism would be an appropriate way to represent it. In this section, a mechanism for the configuration processes is depicted based on its definition from the literature discussed in chapter two. The definition synthesized from the literature portrays the configuration process as a series of activities that include: technical adaptation, organizational adaptation, and social adaptation.

As discussed in chapter two (section 2.2.3), configuration of software requires setting up technical features such as the parameters, databases, user interfaces, and reporting format (Bamford & Deibler, 1995; Dart 1991). However, the configuration process does not relate solely to the technical adaptation of the software. Defining (and often re-defining) the business processes is an inevitable aspect of the configuration process. The process of configuration involves constant adaptation of software to the business needs, or modification of business processes to adjust to the software. Sommerville calls this constant adjustment co-design of software and business processes (Davenport, 1998; Davidson & Chiassen, 2005; Sommerville, 2008). Finally, the literature acknowledges the social challenges that could be faced during a software configuration process. Usually, different groups of individuals from various functional roles/areas are involved in the process of configuration of software. In essence, the configuration process of software involves social interaction of these groups. Sommerville (2008), for example, identified understanding configuration semantics as one of the most challenging activities of the process of configuration. This problem could be exacerbated when various groups are involved in the process. The understanding based on which the software is configured is as a result of social interaction of these groups. Therefore, social adaptation plays a key role in shaping the final configuration.

Based on the literature, we synthesized a mechanism for the configuration processes. This mechanism is shown in Figure 7.1.

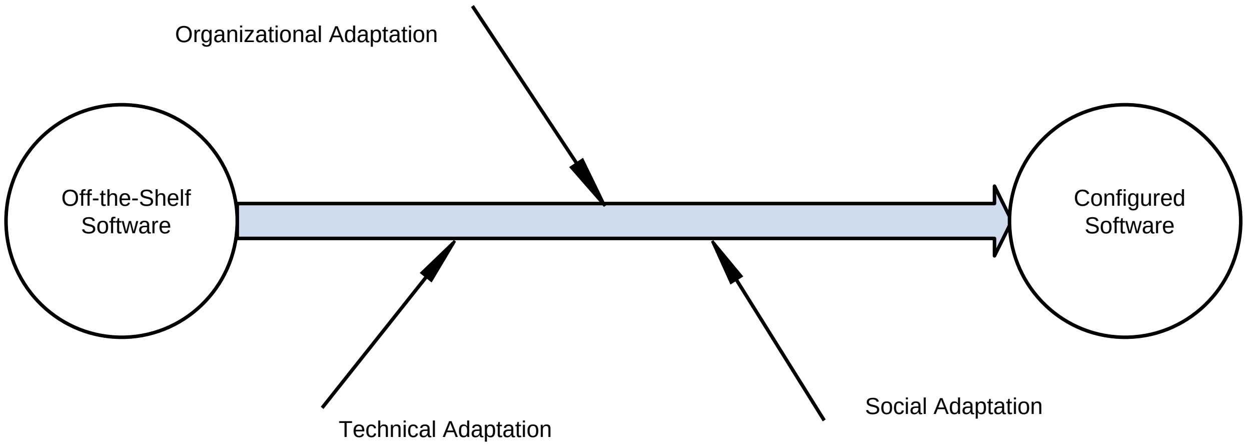

Figure 7.1: A mechanism of the configuration process based on the literature

This mechanism does not reflect the dynamics of each dimension of the configuration process. Specifically, the roles of people involved in the process are not clear. Therefore, in this dissertation the goal was to delve into these dynamics through the lens of SCOT. The configuration mechanism was developed after completion of data analysis. This mechanism is discussed in section 7.2.2.

7.2.2. A Mechanism of the Configuration Process based on the Findings of this Dissertation

In this section, the development of a mechanism of the configuration process is described. This is carried out in three steps. Each step corresponds to a dimension of the process of configuration: technological dimensions, organizational constraints, and discourse forces. The first dimension (section 7.2.2.1) refers to those elements of the mechanism that address the configuration of the technical features of 7i software. The second dimension (section 7.2.2.2) touches on organizational constraints that influence the process of configuration. Finally, the third dimension (section 7.2.2.3) elaborates on the discourse forces that could influence the course of actions in the process of configuration.

7.2.2.1 Technological Dimension of the Configuration Process

In the initial stages of data analysis some preliminary ideas about the configuration mechanism were formed. In other words, from the first steps of data analysis the researcher began to visualize the mechanism. One of these initial models is represented in Figure 7.2. According to this mechanism in the case of each controversy various interpretations were present. These various interpretations offered alternative solutions to the configuration problem. The choice of a final solution brought closure to each controversy. As it is reflected in the schematic model, interpretive flexibility around the problem exists in the path between the problem and the availability of alternative solutions. A final configuration solution is reached through a closure mechanism. All along the process, technological frames guide the actions taken by different groups to solve the problems. This mechanism applies to the less or non-controversial technical features as well. In these cases the process might result in fewer alternatives and closure can be reached faster.

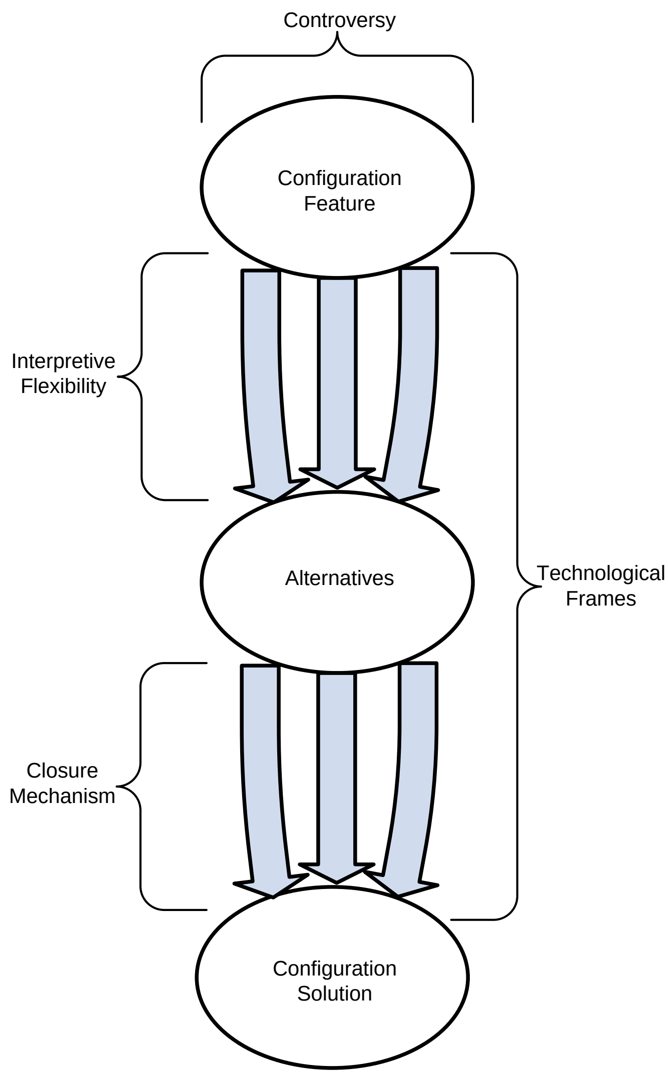

Figure 7.2: An initial mechanism of the configuration process

This configuration mechanism continued to be developed as the data was analyzed further. After the entire analysis section was written and an initial mechanism was developed, it was realized that configuration is made up of a micro and a macro mechanism, each of which has a trigger, a process and an output. In the case of the macro mechanism of the configuration process (Figure 7.3), the mechanism is triggered by the need for configuration of the packaged software, the process is the one through which the software is configured. This process is surrounded by problems and controversies around the technical and organizational features of the software. The output of this mechanism consists of the configured software as used in the organization.

The macro mechanism portrays the entire configuration itself, which comprises: the un-configured packaged software, setting up features, and the configured software. The setting up of the features is represented by a path, which is covered by dots and clouds[1], each of which represents a specific feature. Clouds refer to the controversial, and dots to the uncontroversial features. Within each cloud there is a micro mechanism, which illustrates how the controversies around a feature are resolved. This micro mechanism is shown in Figure 7.4.

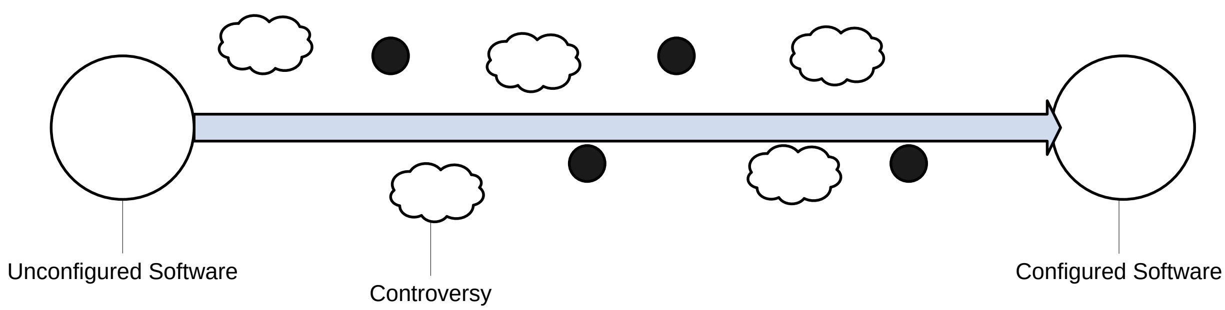

Figure 7.3: Macro mechanism of the configuration process

The micro mechanism is triggered when a configuration problem arises. In the case of PubDiv, this situation usually occurred when an individual would bring a problem to the attention of the entire team. As the process developed (i.e. the interpretation and re-interpretation of the problem occurred) individuals took stances towards the problem.

[1] Special thanks to my colleague Reza Vaezi who pointed out the appropriateness of the use of cloud symbol to reflect the fuzziness of the situation at hand and prevalence of interpretive flexibility.

Finally, the output was the closure mechanism for the problem. This is the stage where the technical feature was set up (or configured). In order to develop the micro mechanism, each controversy was re-read and compared to the other controversies. This process was an iterative one. In other words, the researcher went back and forth between the model and the narratives of the controversies. The analytical generalization of this process is illustrated in the model shown in Figure 7.4. The macro problem (M-problem, shown on Figure 7.4) refers to each controversy. As shown in this model, around each M-problem there is an interpretive flexibility.

The circular representation of M-problems, m-problems, and Interpretation is meant to depict the dynamic nature of these elements of the model. The process of interpretation and re-interpretation of M-problems and m-problems is also represented by circles. This circular, thus dynamic, nature is well reflected in an instance such as when the Inbox feature was being interpreted and re-interpreted. As discussed in chapter six (section 6.3.1.2.1.), when the issue of the Inbox was initially brought up, the team members were not clear about the nature of the problem. However, as the problem continued to be discussed and its interpretations developed, the problem was verbalized more clearly. This example also echoes the surfacing of more minor problems (depicted in Figure 7.3 as m-problems) in the process of interpretation and re-interpretation of an M-problem.

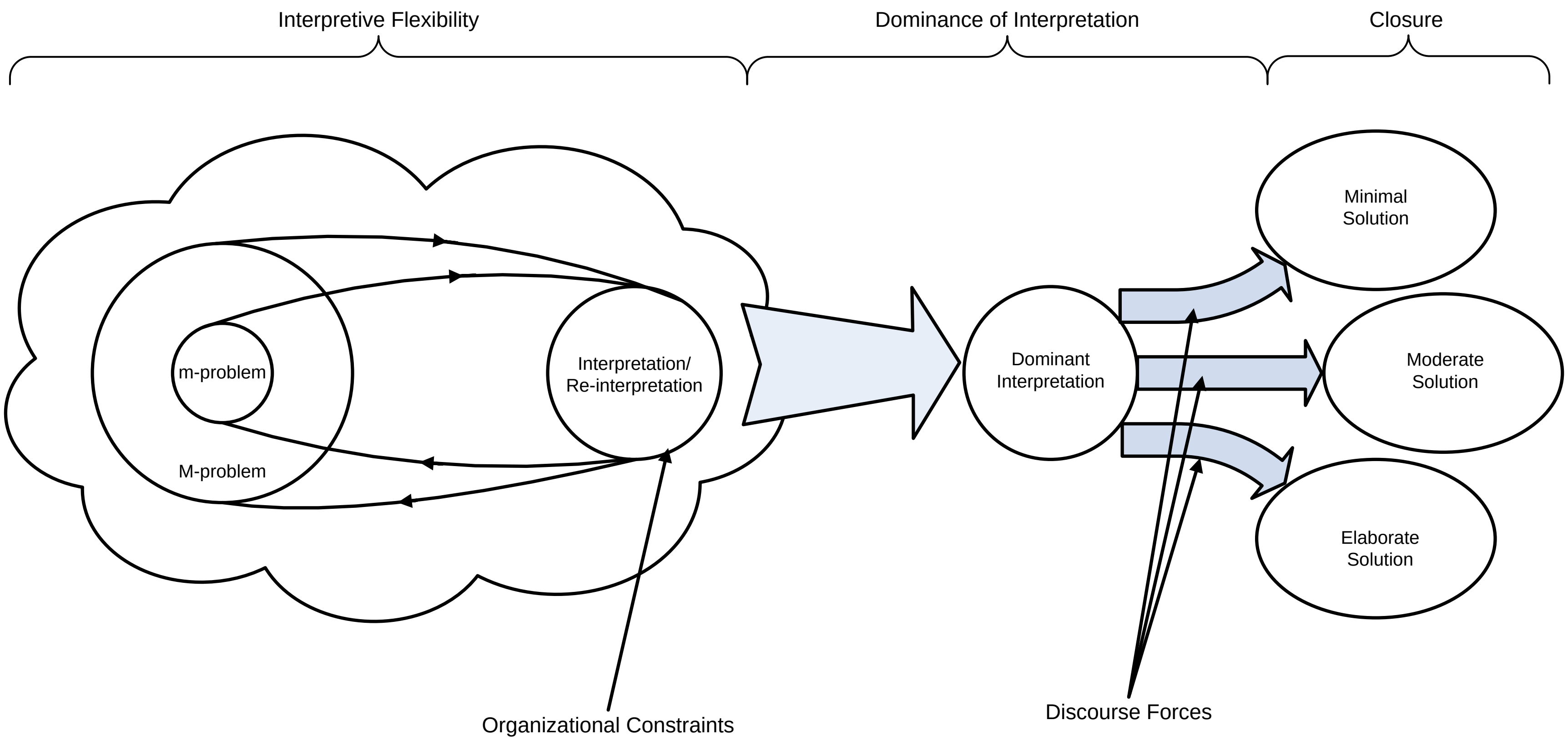

Figure 7.4: Micro mechanism of the configuration process

This was discussed in section 6.3.1.2. For example, in the case of the Inbox/KPI controversy, minor problems such as establishing a start date and the allowed number of days for backlogs were discovered. These types of problems are depicted as micro problems, represented as m-problems in Figure 7.4. In addition, these m-problems themselves were subjected to interpretive flexibility. For example, the suggestion to re-investigate backlog start dates was proposed by one of the individuals. In the beginning, the rest of the team members were not in agreement with this individual. It was only after further speculation and interpretation that others accepted this individual's solution. The solution consisted of changing the date to start counting backlog status from the creation of the work order to when the planner set the date. In another example, one m-problem that arose from the dynamic interpretation of the Inbox M-problem was the realization of the need for a Routing mechanism. Other examples of m-problems which surfaced and were resolved in the case of the Inbox configuration involved refining the terminology to refer to different types of work orders, and setting up the number of days as the criterion for qualifying a work order as a backlog. Thus, it is through these circular, dynamic processes of interpretation/re-interpretation that controversies and the minor problems within each are discussed by members of the team. This stage is called *Interpretive Flexibility Stage*.

The classification of individuals in the process of configuration is based on what happens during the Interpretive Flexibility Stage. Membership in an interpretation group is dynamic. This is because the problem is constantly re-interpreted, accompanied by relevant discussions among the entire team. During these interpretations the perspective of some people might change. In other words, the membership in each category (of interpretation) is dynamic and is constantly changing[2]. Eventually the process of interpretation/re-interpretation slows down and one interpretation becomes dominant; this is depicted as a stage (*Dominance of Interpretation*) as well as an outcome (*Dominant Interpretation*) in Figure 7.4. At this stage the team that has interacted with the technology, interpreted it, and approached the controversies around it, finds a solution. Technological frames of reference guide the team in this process.

The outcome of the micro mechanism is a solution which corresponds to the Closure mechanism in SCOT and/or a configured feature on the software. This final stage is called the *Closure Stage* in the mechanism. Based on the data from Pub Org, these closure mechanisms were identified. These mechanisms are categorized as: *Minimal*, *Moderate*, and *Elaborate*.

A ***Minimal*** closure mechanism refers to a situation where a minimum solution was implemented in order to be able to progress with the configuration process; this type of closure does not offer a satisfactory solution to a majority of team members. For example, configuration of the Calendar feature (6.3.1.2.4) reflects a minimal solution. Even though the Calendar feature was still not functional, the team decided to proceed to other configuration issues. The team as a whole decided that this feature would be used. This was a means to close the issue for the time being, with the goal of working on it later.

[2] Even if we insist on a rigid categorization based on various interpretations around the controversy

We called this type of closure mechanism, which was used on various occasions as a "good for now" or "moving on" closure mechanism. In these situations the objective was to implement a minimal configuration (or ignore the issue altogether) with the goal of revisiting the problem in the future. An example of a situation where the issue was ignored is when comments such as *"we're tired of spending more time… we need to move on… I don't even know why we keep discussing this"* were heard in the case of the Priority feature. In this case, the team did not further develop the Priority feature to prevent subjective use of this tool.

Minimal closure also refers to circumstances in which the implemented solution proved to be insufficient when used. The reason this type of closure is placed in the Minimal closure category is that a solution of this kind stemmed from the lack of a systematic approach a specific configuration issue. For example, when configuring the Priority feature even though a significant amount of time was invested in this process the final solution did not meet any group's expectations. This example demonstrates one of the limitations of information systems in accounting for ambiguous meanings. For example, in the case of the Priority feature, politically contentious aspects of the software could not be standardized even though the Priority feature was meant to overcome political use of the system to begin with!

In summary, some of the approaches taken in a Minimal closure situation include ignoring the problem, arriving at ad-hoc solutions, and sometimes even engaging in irrelevant blaming. The last approach was prevalent when no explanation was readily available to members facing the challenges. It seemed that some individuals tended to explain away the situation by issues that seemed irrelevant to the topic. This seems to be more of a coping strategy. As discussed in chapter six (section 6.3.1.2.4), one example of this situation was when an individual blamed the improper working of the Calendar feature on improper immigration of the features from the training (i.e. simulation) to the pilot system. This individual did not recall that the Calendar feature was subject to a lengthy debate and was not configured appropriately in the first place. Other examples such as blaming configuration challenges on other issues such as politics, leadership, flow charts, and business processes were also discussed in chapter six.

A ***Moderate*** closure mechanism refers to situations where the implemented solution satisfied some groups to a high degree but did not completely convince others. For example, problem/action/failure codes (i.e. collectively called analysis code) were configured to a moderate degree in the system. These fields were eventually configured as required fields. This was against the interpretation that planners and schedulers should be given the freedom to plan/schedule the work orders without restrictions. Closure of the analysis code is considered Moderate since it addressed the concern of standardization of the process by recommending the creation of a list of all the codes (i.e. failure, problem, and action) and populating them, but did not address the concern of providing planners and schedulers the freedom to plan/schedule. This closure is also considered Moderate because the problem itself was of a moderate nature. In other words, not configuring the analysis code feature on the system would not pose a hindrance to the flow of work orders. As discussed in chapter six (section 6.3.1.2.5), even before configuring these features some individuals were able to complete work orders using the lists (paper-based) they had created themselves.

An ***Elaborate*** closure mechanism refers to situations wherein the adopted solution satisfied the majority of the team members. It is in this case that various interpretations converged to more or less one dominant interpretation. For example, in the case of configuration of the Inbox, the final solution addressed everyone's concern: it was configured to not only show the work orders relevant to the role of the person who logs into the system, but also offer the capability to the user to view work orders from other departments/facilities. This was one of the rare situations wherein the interpretation and reinterpretation of the problem resulted in a high degree of clarity. As discussed in chapter six (section 6.3.1.2.1), in the beginning when the Inbox was being configured, the members did not seem to be clear about their concerns/requests. However, later on members became clearer about their request as a result of more clearly understanding the Inbox feature. This was also observable in the terminology they used to refer to this problem. For example, the first quotation below demonstrates how the Inbox was described during the initial stages of configuration while the second reflects the same at later stages. As can be seen, the terminology in the second quotation is more specific. It was during this process when the team realized that there was a need for routing mechanisms for work orders in order to be able to configure the Inbox properly.

"Each Inbox item is linked to a dataspy that the user needs to create… each item needs to be linked and relevant to the user… therefore, this needs to be done through queries…"

"Team discussed the proper term for what was referred to as "Routing", the proper term for the request is "Partitioning". Partitioning is considered a two step process which requires input from the front and back end. The option was given to create classes in an effort to perform this action".

Another example of an elaborate closure was the case of configuration of Cancel/Reject. In this case, the final configuration addressed the concerns of liberal as well conservative interpretations. As discussed in chapter six (6.3.1.2.3), one group's interpretation stemmed from a conservative point of view: the ability to track work orders. This group highly defended the idea of having an audit trail for the Reject feature. Another prevalent interpretation had a more liberal nature; this group emphasized the freedom of users. The idea behind this interpretation was to allow users to create work orders when deemed necessary and cancel the same records once it was realized that they had been unnecessarily issued. The end result addressed both of these concerns.

The mechanisms discussed in this section address the main research questions of this dissertation: Why is packaged software configured a specific way and not another?; and How does this configuration occur? The mechanisms offer an explanation by tracking the configuration features one at a time. Each mechanism reflects the dynamics of different stages of the configuration process from identification to the resolution of each problem. Depending on how each configuration problem is interpreted and re-interpreted, a different solution might be reached. The mechanisms also account for those influences such as organizational constraints and discourse forces. The former is influential during the interpretation and re-interpretation of each configuration problem. The latter influences the choice of final solution when an interpretation becomes relatively dominant. As can be clearly seen, a configuration occurs in a way that depends on various influences. Examining the configuration process through this mechanism offers a stage by stage explanation of this process as happens at a micro level.

The development of a mechanism is important because it offers an explanation for some of the gaps in the literature on packaged software configuration. This has been discussed in section 7.3 where the contributions of this research have been assessed through Lakatos's (1967) concept of research programme.

7.2.2.2. Organizational Constraints of the Configuration Process

As discussed in chapter six (section 6.3.2), certain organizational issues such as Training, Business Processes, and Flow charts constrained the configuration process. These issues, as with the issues regarding the technical features of the system, were brought to light in the form of controversies. Moreover, interpretive flexibility around these controversies was also present. For example, when it came to the issue of Flow charts, the WMSTF team members were divided over the belief about the essentiality of flow charts. As discussed in the chapter six (6.3.2.1.), one group believed that flow charts were essential, and were the basis of configuration. Another group was against the flow charts and even resented them. There was also a group that was neutral about the flow charts. Even though the development of flow charts had long been over, the discussion over them was still prevalent during most of the configuration process. The issue of flow charts even created friction among various groups and at times slowed down the process. Business processes and Training were two of other organizational issues which impacted the configuration process.

These organizational issues influenced the configuration process and approached in the same way as the controversies around the technical features. The former (i.e. influence on the configuration process) was studied in this research since it occurred at the time of configuration. The latter (i.e. resolution of these controversies) had occurred before the configuration process had started. Therefore, these controversies were not directly studied in this research. For that reason, they are not depicted in the configuration mechanism. However, the influences of these controversies were an important dimension of the configuration process and are thus reflected in the model. These influences are demonstrated in the form of *Organizational Forces* in the micro model (Figure 3). In this study the impact of organizational forces was observed on controversial configuration features. Therefore, in the macro model the arrows representing organizational forces are shown to impact controversial features.

This finding is in line with criticisms of SCOT discussed in chapter three. One of the criticisms of SCOT is related to the influence of broader societal structural relationships on technological developments. For example, Winner (1993) argues that issues such as gender, race, class, and ethnicity might affect the course of development of a technology. Douglas (1990) criticizes SCOT for disregarding political economy regarding technological developments. Even though Bijker’s (1995) work addressed some of these concerns, his theory does not specifically focus on these forces. SCOT acknowledges the significance, for example, of race, class, and the social and political environment on the development of technology. However, Bijker and his colleagues refrain from classifying social relevant groups based on such influences (e.g. race). They emphasize on the classification of groups based on interpretations and meanings. In this regard, the role of these influences in the path of development of a technology is not clearly understandable through the lens of SCOT.

In the empirical study of this dissertation, some organizational forces (i.e. Business processes, Flow Charts, and Training developments), which were not the direct topic of configuration but influenced the configuration process, were identified. These forces exhibited the greatest influence over the process of interpretation and re-interpretation. Therefore, the mechanism of configuration that was developed in this dissertation not only follows the technological artifact but also takes into account higher level influences.

7.2.2.3. Discourse Forces in the Process of Configuration

In the second set of data analyses (discussed in chapter six) a series of individual attributes were identified. Upon re-interpretation of the data, when developing the mechanism of configuration, the relationship between these attributes and the mechanism was revealed. As was discussed before, one of the findings of this study was that the grouping of individuals made more sense based on their interpretations rather than on their individual differences. However, a consistent theme in attributes of individuals was identified as well. Based on this theme, perspectives of those individuals who were involved in this project at Pub Org are classified into three categories: *Optimism*, *Indifference*, and *Pessimism*. Two points have to be noted here. One, these attributes do not reflect an individual's personality in general; investigating this issue was not an objective of this study. Two, these attributes were discovered as an underlying theme when it came to taking stances about configuration controversies.

Analysis of the data revealed that dominance of *optimism* usually resulted in an *Elaborate* closure. Prevalence of *pessimism* most likely resulted in *Minimal* Closure. Finally, pervasiveness of *indifference* normally resulted in more *Moderate* Closures. These categories are shown on the configuration mechanism (Figure 7.4) as Discourse Forces that influence the reaching of solutions. An example of elaborate closure was the case of the Inbox configuration. When configuring this feature overall optimism in individual attributes was dominant. Therefore, the final solution addressed most team member's concern. In the case of configuring the Priority feature, overall team members were pessimistic about the capability of this feature. This prevalent pessimism resulted in not systematically discussing and addressing the Priority problem. When configuring the analysis code most team members did not seem to be treating this feature as a critical. Perhaps the reason for this indifference was that the system would work regardless of the presence of this feature. The final solution to configuration of the analysis code was a minimal one. What is important to note is that interpretation and re-interpretation of the problem affects the overall pessimism and optimism. In the situations explained in chapter six, in each configuration controversy, when the overall interpretative process resulted in high degree of clarity, optimism seemed to have become more prevalent.

Based on the discussion in this section (section 7.2.2) a working definition for the process of configuration of packaged software could be offered. This definition reflects the micro mechanism that was developed in this section.

The configuration process of packaged software is defined as an interpretive process in which different features are set up for the software. The final setup of each feature could reflect a minimal, moderate, or an elaborate solution. The choice of this final solution is the result of interpretation and re-interpretation of each configuration option under the influence of relevant organizational constraints. The dominant interpretation under the influence of discourse forces such as optimism, pessimism, and indifference guides the choice of the final solution.

7.2.3. Redefinition of the Concept of the Information Systems User

In the initial stages of reading the data on ‘Philosophies’, the researcher found various attributes of interpretive flexibility. At this stage it was possible to define various groups with distinct attributes. Initially, the researcher attempted to place each individual within a specific group. Upon further analysis it was realized that individuals would move from one group to another in different contexts related to each specific problem. This validated the concept of interpretive flexibility as emergent, and is in line with SCOT. What added to the validation was the fact that even though after initial grouping the researcher tried to keep each individual within a specific group for each problem, the task proved to be impossible. In other words, for each problem new groups would emerge.

Therefore, although the researcher did not categorize the actors based on the traditional way of classifying individuals based on their functional areas (e.g. Finance vs. Marketing), she still attempted to adhere to a rigid grouping initially; this proved to be impractical. This was a verification of the fact that interpretive flexibility can change from one context (here, a problem) to another. Therefore, in this case it was also observed that the concept of interpretive flexibility and inclusion are dynamic and context specific. Thus, confirmation for appropriateness of categorization around problems was also found. However, certain individual characteristics seemed to remain consistent through various problems. These individual characteristics were grouped as in shown in Table 7.1. These categories were named as: *Pragmatist*, *Perfectionist*, *Solver*, *New Comer*, *Indifferent*.

The influence of these characteristics was prevalent on reaching closure mechanisms; this was discussed in terms of the discourse forces of the configuration process. As can be seen in Table 7.1, each combination of characteristics reflected an overall optimism, pessimism, or indifference attribute to each group. While these attributes could have influenced individuals' interpretations of various controversies, their influence was more explicit on action than on interpretation (i.e. TFR vs. interpretive flexibility). In addition, when it came to the interpretative flexibility of various features, individuals who shared the same interpretations could come from different groups (i.e. groups 1 to 5).

Table 7.1: Characteristics of individuals in the context of 7i configuration process

Group	Attributes
1) Pragmatist	- Is mostly optimistic - Has grand/visionary ideas - Encourages reaching consensus - Believes in documenting the events/activities - Likes to "move on" from problems quickly even if the solution is short term - Does not believe in a rigorous training - Believes that a pilot system needs to be launched as soon as possible

Table 7.1 -- Continued

	- Believes in the usefulness of flow charts - Believes in the value of investing time on the development of the business processes
2) Perfectionist	- Is mostly pessimistic - Has attributes very similar to those of Group 3 members - Pairs up with Group 1 members on a lot of issues - Works through problems patiently - Keeps repeating the scenarios - Knows the system really well - Does not find the old system as problematic - Finds the new system useful - Is concerned about data - Has knowledge of the systems and processes in narrower areas but at a deeper level than group 3 - Is neutral about the usefulness of flow charts - Believes in the value of investing time on the development of the business processes - Is proactive, takes initiatives - Is ahead of other team members in completing tasks - Acknowledges the differences between DIVA and DIVB a lot - Believes in extensive training on the system
3) Solver (or Solution Provider)	- Is mostly optimistic - Is experienced in general (in the business processes) - Is comfortable to ask questions and to keep the discussion on specific issue for as long it takes to address the problem (opposite of group 1) - Knows broader topics than Group 2 members but with less depth - Believes in the flow charts - Acknowledges the difference between DIVA and DIVB - Group 3 would agree (share) with group 2 a lot. Therefore, people from these two groups (which together made the majority in number as well) really influenced the way the system was configured. - Believes there needed to be training maybe on system but mostly on business processes - Is key in preparing the training material
4) Newcomer	- Is mostly indifferent - Is still learning (i.e. is not familiar with the system or the business processes) - Is a newcomer - Is not familiar with the flow charts - Is sometimes intimated by the discussions
5) Indifferent	- Is mostly indifferent - Does not seem to be involved in the process much - Usually does not offer any solutions - Accepts the solutions

Finding the attributes shown in Table 7.1 as well as the insights gained regarding interpretations of individuals in the process of configuration calls for a redefinition of users in the context of IS research. This is also in support of the proposition made by Lamb and Kling (2003) with regard to the reconceptualization of users as social actors in IS research. Lamb and Kling argue that individualistic treatment of the concept of IS users merely based on cognitive models limits our understanding. They call this type of IS user as "socially thin" and argue that "the socially thin user construct limits our understanding of information selection, manipulation, communication, and exchange within complex social contexts" (Lamb & Kling, 2003, p. 197).

Lamb and Kling (2003) believe that the concept of the user prevalent in IS research is based on the bounded rationality notion defined by Simon (1955). According to this view a user is defined as "an atomic individual with well-articulated preferences and the ability to exercise discretion in ICT choice and use, within certain cognitive limits" (Lamb & Kling, 2003, p. 198). In the context of the Datastream 7i project individuals exhibited a high degree of uncertainty in articulation of their requirements.

Instead of a rational view of users based on cognitive individualism, Lamb and Kling (2003) describe user as a contextualized social actor with four dimensions: *affiliations* (i.e. organizational and professional relationships), *environments* (i.e. institutionalized practices of organization), *interactions* (i.e. resources and media of exchange), and *identity* (i.e. individual and collective identities as portrayed by the user) (Fidel et al., 2004). Kallinikos (2004) similarly finds the rationalistic treatment of users limiting in his study of organizational and behavioral implications of ERP systems. He argues that realistically organizational actions do not unfold as “prearranged sequences of steps”. Rather, Kallinikos argues that these actions entail “holistic patterns of cognition and open encounters marked by sidesteps, unpredictable turns and improvization that defeat straightforward procedural standardization” (Kallinikos, 2004, p. 11).

Kallinikos’ depiction of how users encounter organizational situations well explains how the WMSTF team members approached the each controversy in the configuration process. This view of behavior of organizational actors could elucidate how the processes of interpretation and re-interpretation, as depicted in the micro mechanism (Figure 7.3), happen. This view of a social actor could also explain the choices made during the closure stages of the configuration process. Moreover, this view justifies the choice of minimal or moderate closures even if these choices do not make sense rationally.

In this dissertation the use of the notion of “users” is avoided. This is as a result of valuing the interpretation of individuals involved in the process of configuration rather than their roles in the organization. Therefore, in line with the ideas of Lamb and Kling (2003) and Kallinikos (2004) the notion of organizational actors, whose actions are influenced by their interpretations (of technological dimensions), context (of organizational constraints), and attributes (of discourse forces) is employed.

This observation is important since the limitation of defining actors based on their functional roles in an organization has been seen in IS literature. An example of this case was introduced in chapter five when the pilot case study of this research was discussed. In this example, Shawn was identified as an individual who did not fit in any of the pre-specified groups (Users and Support) in terms his interpretation of CDT and CC Solutions. Another example from the IS literature, more specifically in the implementation of packaged software, was the observation made by Yeow and Sia (2008). Yeow and Sia examined the incongruence of the socio-cognitive (Orlikowski & Gash, 1994) technological frames of different groups in the process of implementation of packaged software. The goal of the study was to investigate how the 'best practices' of the packaged software were negotiated by different groups, and how these groups reached agreements.

The researchers classified the participants of the project based on their roles (e.g. end-users, Financial policy department (FPD)). Their findings from a pair-wise comparison of various technological frames demonstrated that several frames overlapped on some elements. For example, they found that both end-users and FPD groups' frames emphasized audit and control issues. This finding is similar to the premise of SCOT that various groups take different approaches towards the technology. However, based on a SCOT approach the classification of the participants of the project would have occurred after identifying their perceptions about the software and its issues.

7.3. Contributions

Contributions of this research are manifold. Chapter two synthesized the literature on packaged software and well as on its configuration. This synthesis was complemented by the development of a mechanism of the configuration process, depicted in Figure 7.1. In addition, this study examined and compared two major theories. This comparison was conducted at the theoretical (discussed in chapter two) as well empirical level (the pilot case study). Empirical contribution was also offered to the theory of SCOT. In this dissertation a linkage between SCOT and IS research was made.

This section offers a discussion of the contribution of this study in addressing some of the gaps in the literature on packaged software configuration. In doing so, an assessment of this contribution based on Lakatos's (1967) research programme is undertaken.

7.3.1. Assessment of the Contributions According to Lakatos's Research Programme

In this section contributions of this dissertation are highlighted. Lakatos's (1976; 1977) concept of a Research Programme is used to evaluate the contribution of this research. Lakatos argues that theories develop as a result of succession of slightly different theories and empirical content over time. Lakatos called this succession of theory enhancement *Research Programme*. According to Lakatosian research programme, in order for a programme to be progressive, there has to be consistently progressive **theoretical** as well as **empirical progress**. The former may be verified immediately but the latter cannot be verified immediately. According to Lakatos, a research programme has three domains: ***Negative Heuristic***, ***Positive Heuristic***, and ***Anomalies***.

7.3.1.1. Negative Heuristic

At the heart of a research programme, there is a theoretical core. “The **negative heuristic** specifies the ‘hard core’ of the programme which is ‘irrefutable’ by the methodological decision of its protagonists” (Lakatos, 1977, p. 135). The negative heuristic “forbids us to direct the *modus tollens* at this ‘hard core’” (p. 133) and discourages working on theories that are inconsistent with the main theory of the programme. This is why defining the ontological stance of a research programme from the beginning is extremely important. It is critical to define what is it that the researcher is attempting to study. He should clearly specify what the theory under study explains and what it does not. Therefore, Popper’s (1959) concept of falsification plays an important role in the process of research because it obliges the researcher to remain clear about the scope of research. Lakatos (1977) argues that the hard core of the research programme should be protected until the programme becomes degenerative (i.e. does not progress any more). At this time the theory is replaced.

In the case of this dissertation, the researcher adhered to the concept of the negative heuristic. Firstly, the phenomenon of the study was clearly defined since the beginning. This was reflected in the two research questions:

1) Why is packaged software configured a specific way and not another?
2) How does this configuration occur?

This research was clear about its ontological stance from the onset. As discussed, the goal of this study was to examine the configuration of packaged software through the theoretical lens of social construction of technology. Therefore, all throughout the study the explanandum (i.e. the phenomenon to be explained (Rosenberg, 2000)) remained the same - the configuration process. In addition, the goal throughout the study was to address some of the unanswered questions of configuration phenomenon; thus remaining faithful to the theoretical core of the literature on the configuration process. Explanans are defined by Rosenberg (2000) as general laws for deductive explanation and are law-like and follow formal logic to imply the explanandum. In this study explanans stemmed from the SCOT theory.

7.3.1.2. Positive Heuristic

Positive Heuristics are auxiliary hypotheses, which form a protective belt around the theoretical core. A positive heuristic guides the researcher towards the paths to pursue. Lakatos argues that we must direct modus tollens towards these. These hypotheses go under tests and are adjusted or replaced to defend the hard core of the programme. “The positive heuristic consists of a partially articulated set of suggestions or hints on how to change, develop the ‘refutable variants’ of the research-programme, how to modify, sophisticate, the ‘refutable’ protective belt” (Lakatos, 1977, p. 135). He adds, “content-increasing auxiliary hypotheses turn a chain of defeats-with hindsight- into a resounding success story, either by revising some false ‘facts’ or by adding that each step of a research programme be a *consistently progressive theoretical problemshift*" (Lakatos, 1977, p. 134).

In this study, contribution was made to the positive heuristic of the research program with regard to configuration through the findings that were discussed in the first section of this chapter. These findings together formed the mechanism of the configuration process. Development of this mechanism was one step towards adding to a generative research program, and was in line with Lakatos's idea of *model* by simplifying the process (and thus saving the researcher from becoming confused by the "ocean of anomalies"). Lakatos defines a model, which is meant to simulate the reality, as "a set of initial conditions (possibly together with some of the observational theories) which one knows is *bound* to be replaced during the further development of the programme, and one even knows, more or less, how" (Lakatos, 1977, p. 136).

As acknowledged earlier in this chapter, the mechanism of the configuration process serves as a stepping stone for further theoretical development of theories for understanding the phenomenon of the configuration process. Development of the mechanism of the configuration process assisted the researcher in providing a holistic view of the configuration process. Macro and micro mechanisms depicted a holistic as well as an atomistic view of the configuration process. These mechanisms as well as theoretical explanations of their elements offer an initial theoretical perspective on the configuration process, which could be studied further and empirically tested. These attributes of the mechanisms qualify them as a part of a positive heuristic of a research program on the configuration process.

7.3.1.3. Anomalies

Kuhn (1962) called unexplained puzzles Anomalies or counterexamples. Anomalies "must lead to changes only in the 'protective' belt of auxiliary, 'observational' hypothesis and initial conditions" (Lakatos, 1977, p. 133). Anomalies are never completely exhausted. However, researchers should refrain from thinking that these anomalies are taken randomly and that the protective belt is built in a miscellaneous way.

One of the objectives of this dissertation was to address some of the anomalies in the configuration literature. As discussed in chapter two (2.2.3.2.), understanding the meaning of configuration of software can be difficult. One of the biggest challenges, identified by Sommerville (2008), in the configuration of packaged software includes understanding the options available for configuration. He argues that this can be a problem even in a seemingly simple case such as that of MS word. Somerville (2008) mentions that, for example, even after twenty years of experience in using MS word he finds it difficult to express what some options (such as preference or organizer screen) available in MS word mean and how they interact.

Sommerville argues that even when developers identify and understand different configuration options available, they face the decision of which configuration to choose. Sommerville also emphasizes the importance of understanding the relationship between designing the business processes and configuring the software. Sommerville is one of the pioneers of highlighting the gaps in the configuration literature. However, his work does not delve into the theoretical development to address these gaps. He also repeatedly emphasizes the importance of organizational and social challenges of the configuration process; however, his study does not tap into these issues deeply. For instance, he focuses mainly on developers (i.e. programmers) in organizations. Therefore, roles of non-programmers in the process of configuration are not clear. The purpose of Sommerville's study is to bring the importance and criticality of the challenges of configuration to the fore. Sommerville calls for more attention from the research community to the topic of the configuration of packaged software.

In this regard, the researcher believes that the findings of this dissertation help with addressing some of the gaps that appear in the configuration literature. This was explained to some extent at the end of section 7.2.2. Additionally, in this study, organizational constraints of the configuration process tapped into the influence of organizational issues (such as business processes) on the process of configuration. The approach to studying the configuration process was aimed at understanding the configuration process from a holistic view, which shed light on some of the puzzles of the configuration process. The choice of SCOT warranted this holistic view. A fundamental difference between SCOT and other views is the emphasis on 'thick description' (Geertz, 1973). In this type of study, a detailed description of the technological artifact itself and its surrounding context is provided. By providing a thick description of the technology, the researcher follows the design of the artifact as it takes form and evolves (Bijker et al., 1987). This approach seems to be appropriate when studying a process when the goal is to discover influences on the process. SCOT proved to be appropriate in this study where the goal was to study the process of configuration.

By discovering various interpretations around controversial configuration features, relevant groups were identified. One of the premises of SCOT is that even ‘working’ and ‘nonworking’ of a technology are also socially constructed rather than being inherent attributes of a technology (Bijker, 1995). This was in fact observed during the process of the configuration of the 7i software. The workable forms of the features that were discussed in chapter six resulted from social interaction of WMSTF team members. “In this way, the “working” and nonworking” are now being treated as *explanandum,* rather than used as *explanas* for the development of technical artifacts” (Bijker, 1995, p. 75). Configuration of 7i was the explanandum in this study, remaining faithful to the epistemology of SCOT - that of the social construction of technology.

In this section, the main contributions of research in terms of addressing some of the gaps of the literature on packaged software configuration were discussed. This discussion was carried out in the light of Lakatos’s research programme approach and the two overarching research questions of this study. Based on the assessment of the contribution of this research (section 7.3.1), according to Lakatos’s research programme, this dissertation is considered generative. In other words, the study achieved the goal of providing some answers to gaps that exist in the literature on the packaged software configuration.

In addition, this implies that there are still unanswered questions in this research area, presenting further areas of research. Therefore, investing in research on the topic of configuration is warranted. This presents several opportunities for future research. In section 7.5, limitations of this study and areas of further research are discussed.

7.4. Implications for Practice

Sommerville (2008) summarized it nicely when arguing that configuring a system can be time-consuming and expensive. He identified one of the biggest challenges in this process to be trying to understand a configuration. Statements of this sort are a testimony to the value of providing practitioners with an analytical tool by which they can understand and manage the process of packaged software configuration. The configuration mechanism developed in this chapter could serve as such an analytical tool.

The mechanism accompanied by the discussion of its development could assist the management with understanding the process of configuration. This understanding could result in being better informed about the options available for the configuration. In addition, by knowing the dynamics of each stage in the process of configuration, managers could provide the kind of support that would accelerate each stage. For example, managers, due to a better understanding of what could influence the configuration process, could support the type of discourse forces that would result in more fitting solutions.

In general, practitioners interested either in the acquisition or development side of packaged software products could benefit from contributions of this research. On the acquisition side, the insights gained could be beneficial in familiarizing organizations with the dynamics and issues related to implementation and configuration of packaged software. This study could also benefit the IS development side because it could help software development companies familiarize themselves with the type of challenges their customers face when implementing their products. These insights could be applied by packaged software developers in order to build software that is more understandable by end-users.

7.5. Limitations and Future Research

Remaining faithful to the epistemological stance of this dissertation, it is acknowledged that the findings of this research relate to analytical, rather than statistical, generalization. The purpose of this study was to contribute to theory on the configuration of packaged software. Pub Org offered an exemplary case (Yin, 1993; 2002) to study such a topic. The interpretive approach of this study and the hermeneutic technique of data analysis allowed for studying the phenomenon (i.e. the configuration process) holistically, the result of which are the macro and micro mechanisms developed in the earlier sections of current chapter (section 7.2.2). As acknowledged previously, this mechanism is to be treated as a stepping stone for further theorizing. Future research would greatly add to the endeavors of this dissertation by building on the concepts of macro and micro mechanisms in a variety of ways.

First, each of the elements of the mechanisms could be further refined and/redeveloped. For example, by studying various organizational contexts, researchers could identify other organizational forces. In addition, such studies could add to better understanding of the role of organizational constraints of the configuration process. Discourse forces as well as the technological dimensions of the configuration process could receive the same treatment. In this regard, studying other types of software (besides the one studied in this dissertation) could greatly contribute to understanding of complexities of the technological dimension. Moreover, other team dynamics could shed more light on the discourse forces that influence the configuration process.

Additionally, it would be of great value to fine-tune the definitions of various elements of the models. Providing further empirical support for the models would enhance this understanding. For example, a future study could aim at defining various closure mechanisms in further detail. There would also be of value in the refining the relationship among M-problems, m-problems, and the interpretation/re-interpretation process.

Second, the mechanisms and concepts developed in this study could be the subject of studies of positivist nature. These studies could potentially test the mechanism of the configuration process, partly or as a whole, by developing measurable constructs. The results of such studies could add to the validity of this research by statistical generalization. For example, a future study could investigate the interactions of individual forces and various closure mechanisms.

Third, another limitation of this research stems from the limitation in time scope of the research. Ideally, conducting a longitudinal study to follow the configuration process from the time of purchase to training, to a period post implementation of software would enrich the findings. This is a limitation of the research design in this dissertation, one as a result of the scope of this research.

Finally, considering that this dissertation is one of the first to attempts to adopt SCOT holistically in the context of information systems research, future studies focusing on the application of SCOT in IS research are warranted. One of the goals of this dissertation was to refine the theory of SCOT in the context of soft technologies (e.g. IS) rather than what SCOT was developed through - hard technologies (e.g. bicycles). As discussed earlier in this chapter, this dissertation has shed some light on the applicability of SCOT in the context of the configuration process of software. It is our hope that the findings of this study regarding SCOT will be of value in future studies.

Appendices

Appendix A: IS Literature on Technological Frames of Reference

Table A.1: Literature review on technological frames of reference

Research	Study context	Frame domains	Frame structure	Groups	Incongruence	Frame change	Study of power
Barrett (1999)	- investigates adoption of EDI service in the London insurance market - longitudinal study of development, introduction, system use - snapshot of frames derived from cultural assumptions	- nature of technological change - nature of business transactions - importance of market institutions	N/A	- IT professionals - senior managers - brokers and underwriters (users)	- problematic or negotiated	N/A	- highlights the power relations embedded interpretations that form the basis of frames and cultural norms
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Cano (2003)	- a recursive construction of technological frames is proposed to understand how technological understanding is constructed by an organization - at moment *t*, technological frame (MT(t)) is the result of the acknowledgement of interaction prior to (*t-1*) from other individuals (i.e. MT(t)=MT(t-1)	N/A	N/A	N/A	- social incongruities are the problematic ones - in order to decrease incongruities of technological frames, complexity of interaction of each individual with the technological topic has to be taken into account	N/A	N/A

Table A.1 -- Continued

Research	Study context	Frame domains	Frame structure	Groups	Incongruence	Frame change	Study of power
Davidson (2002)	- studies a sales information system at an insurance company - longitudinal study of requirement and pilot implementation - frame change over time	- IT delivery strategies - IT capabilities and design Business value of IT - IT-enabled work practice	N/A	- system developers - system constituents - executives	- incongruence may vary across frame domains - incongruence may vary across relevant social groups - incongruence may vary over time - salience of domains shift in response to triggers for change	- frame change due to shifts in frame domain salience (triggered by environmental stimuli, reorganizations, or technology changes) rather than in changes in domain content - excessive frame stability could result in unwarranted escalation of commitment to IT projects	- when some frames become dominant, interpretive power is brought to the fore
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Gallivan (2001)	- studies reskilling IT professionals for client/server technologies at a telecommunications company - an implementation project - examines a snapshot of frames	- vision of reskilling/type of change	N/A	- change managers - IT professionals - others	- problematic or negotiated - suggests that identifying incongruent frames before the implementation begins can help with communication among different groups	- using various techniques related to frame change to prepare employees for organizational change	N/A

Table A.1 -- Continued

Research	Study context	Frame domains	Frame structure	Groups	Incongruence	Frame change	Study of power
Iivari & Abrahamsson (2002)	- studies organizational culture of a small software development company during implementation of UCD. - investigates the difference of frames of different subcultures - examines a snapshot and then a follow-up after a year	- nature of user-centered systems development - motivation and criteria for success - use of user-centered systems development	N/A	- software engineers - managers - usability specialists	- problematic or negotiated - mutual understanding and cross-cultural dialogues needed in order to reach frame alignment - suggests offering workshops for crating mutual understanding - emphasizes on early identification of incongruences in order to create understanding and frame alignment	- does not study frame change but suggests that in order to reach alignment there needs to be communication	N/A
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Khoo (2001)	- studies design of peer review policies for the Digital Library for Earth System Education (DLESE) - studies a design and implementation project - examines a snapshot	N/A	N/A	- does not identify separate groups a priori and looks into different interpretations - similar to SCOT view but SCOT is not acknowledged explicitly	- problematic or negotiated - incongruence in frames is not counterproductive per se - incongruence in frames has to be acknowledged and looked into for design purposes.	N/A	N/A

Table A.1 -- Continued

Research	Study context	Frame domains	Frame structure	Groups	Incongruence	Frame change	Study of power
Khoo (2004)	- investigates the prevalent technological frames to understand different views about digital libraries - studies a design and Implementation project - a longitudinal study	technologically oriented view of digital libraries - utilitarian view of digital libraries	N/A	- principal investigators (PIs) - working groups	- problematic or negotiated	N/A	N/A
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Law & Lee-Patridge (2003)	- aims at making sense of different views about the use of a knowledge management system through construction of social cognitive structures - a 3-phase analysis approach: theme-based, frame-based, and lens-based analyses - examines a snapshot	- knowledge sharing at the organizational and individual levels (IS group) - methods and approaches for knowledge capture (HR group) - knowledge capturing and documentation (HR group)	- cognitive maps of factors influencing knowledge sharing and knowledge acquisition/ capture	- IS group - HR group	- incongruence might be problematic consequences - does not delve into investigating this issue	N/A	N/A
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Lin & Cornford (2000)	- studies replacement of an e-mail system in a financial institution - Pre-	- the nature of problems - requirements for the system	N/A	- office IS group - user group - management	- problematic or negotiated -draws on Actor Network Theory	- traced back past frames retrospectively through social translation process	- through Actor Network Theory explained how one group can manipulate other groups' frames

Table A.1 -- Continued

	implementation - snapshot of frames	- images of implementation - issues around use		group	- suggests that alignment of frames can be achieved by translation of each other's frames towards a group's own frame		
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Lin & Silva (2005)	- Studies framing and reframing as a social and political process - studies an implementation project - examines a snapshot	- requirements solution to the problems - understanding of the problem - understanding of the project	N/A	N/A	- problematic or negotiated	N/A	- suggests that understanding, interpretation, and expectation of information systems are framed and reframed through the exercise of power
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
MacLeod & Davidson (2007)	- proposes integrating analysis of narratives to analysis of TFR to - argues that technological frames shape what we hear and what we interpret - examines a snapshot of the use of an IS	N/A	N/A	- the focus is on individuals	N/A	N/A	- importance of power is acknowledged - narratives reflect individuals' perception of power

Table A.1 -- Continued

Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
McGovern & Hicks (2004)	- investigates how political processes influence implementation of information systems - conducts a longitudinal study of selection and implementation of IS	- nature of technology Technology strategy - technology-in-use - type of partnership	N/A	- research team - managing director	- problematic or negotiated - argues that the success of the project stemmed from working around the dominant frame	N/A	- exercise of power by managing director by having technological frame dominance
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
McLoughlin et al. (2000)	- studies implementation of a team-based cellular manufacturing technology in three companies - longitudinally examines the process of frame closure and stabilization	- treated as unidimensional	N/A	- various stakeholders	- problematic or negotiated	- examines the evidence of frame closure and stabilization	- discusses political processes that were used to bring dominance to specific frames
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Orlikowski (1992)	- investigates how implementation of a new groupware system changed the nature of work and the social interactions - results showed that people's mental model (or technological	N/A	N/A	- managers - seniors -administrators	- not discussed	- no exclusive and comprehensive discussion on frame change - the author suggests that education and communication is needed in order to change	- some of the subjects feared loss of power (mostly in terms of skills and core competencies) - no specific and explicit discussion of power

Table A.1 -- Continued

	frames) and organizational structure influences the implementation of a groupware system - results showed that in the absence of mental models that are familiar with the technology, people use their understanding of previous familiar technologies - an implementation project was studied, taking a snapshot of the project					people's frame to understand the new technology	
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Orlikowski & Gash (1992)	- proposes a framework (grounded in shared meanings people attach to information technology) for diagnosis and understanding intended and unintended changes around the use and development of information technology	- philosophy towards technology - issues around initiation - issues around implementation - issues around use - criteria of success	N/A	- managers -technologists - users	- congruence /Incongruence along a continuum - interested in incongruence in kind not degree of frame - frames and incongruence based on three different organizational changes (i.e. first, second, and third	- initiates the discussion about frame shift over time but calls for more empirical work to track these shifts	N/A

Table A.1 -- Continued

		- impact - relations with other players in the computing social world			order changes)		
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Orlikowski & Gash (1994)	- studies implementation of Lotus notes at a large consulting firm - early, ongoing implementation - snapshot of frame	- nature of technology - technology strategy - technology-in-use	N/A	-technologists - managers - users	- problematic or negotiated	N/A	N/A
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Ovaska et al. (2005)	- observation of an E-commerce project - suggests that systems requirements are interpreted and shaped through a dynamic process of filtering, negotiating, and shifting	- business value of systems development - systems development strategy - systems development capability - systems development resource allocation	N/A	- internal development unit - business unit (interestingly in the later episodes there is lesser and lesser distinction between the two groups and… when referring to	- problematic or negotiated - reaching agreement is not necessary - recognizing and explicitly acknowledging various attitudes and expectations is necessary	- frames are investigated in episodes… there is frame shift	- argues that differences in expectations and attitudes are more concerned with the political system of the organization

Table A.1 -- Continued

Research	Study context	Frame domains	Frame structure	Groups	Incongruence	Frame change	Study of power
				frames, the authors talk about frames of participants not IDU or BU separately)			
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Puri et al. (2006)	- studies development of the National Spatial Data Infrastructure (NSDI) in India - a longitudinal study	- user participation - access to NSDI - contents of NSDI	N/A	- the concerned scientists, experts, and technocrats - geo-spatial industry groups - end users	- is not concerned with identifying the incongruence of frames among different groups - suggests focus on bringing different frames of meaning around a technology to the same understanding - suggests that understanding socio-cognitive processes in the development of IS provides insights to initiate social interactions of different groups	N/A	N/A
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Sahay et al. (1994)	- examines implementing of GIS technology - implementation - Snapshots of frames	-identified a detailed list of issues/problems with GIS technology	multidimensional analysis of frames related to geographic information systems	- Experts (IT) - Users	- assessing the degree of incongruence through quantitative multidimensional scaling	N/A	N/A

Table A.1 -- Continued

Research	Study context	Frame domains	Frame structure	Groups	Incongruence	Frame change	Study of power
Shaw et al.(1997)	- studies effectiveness of computer systems support at an elevator company - investigates the effects of technological frames on end-user satisfaction - examines snapshot of frames	- technology in use - technology strategy - ownership of technology - nature of technology	N/A	- MIS staff - management - end-users	- problematic or negotiated - shows that the level of congruence is related to user satisfaction	N/A	N/A
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Weilenmann (2001)	- studies the ways a group negotiated the use of a new mobile awareness device (Hummingbird) for skiers - examines snapshots	- when to use - where to use - wense of shared ownership	N/A	- instructors - users	- does not specifically address this issue - negotiation was done through talk and action.	N/A	N/A
Research	**Study context**	**Frame domains**	**Frame structure**	**Groups**	**Incongruence**	**Frame change**	**Study of power**
Yoshioka et al. (2002)	- explores the challenges of adopting a MOO-based technology to support a virtual working environment	- view of technology - rationale for technology - use of	N/A	- Toki HQ - Toki US - Toki APG	- problematic or negotiated - a shared interpretive scheme is referred to as community-	N/A	N/A

Table A.1 -- Continued

	- examines snapshots	technology			based interpretive scheme - argues that difference in interpretive schemes across sites, nationalities, languages, and roles as well as over time exists - differences in interpretive schemes can lead to differences in the use of the system		

Appendix B: A Short Description of NVivo Data Analysis Tool

1) Figure B.1 exhibits a snapshot of an opened document within NVivo.
2) The top part of the screen exhibits the Interpretive Flexibility Node and its sub-nodes.
3) The bar at the right side of the screen exhibits the coding strips. This shows which parts of the data are coded under different nodes.
4) The figure also shows a piece of text that is highlighted and is being coded.
5) Once the data is coded, queries can be run to extract only those pieces of the text that correspond to specific nodes in one document.

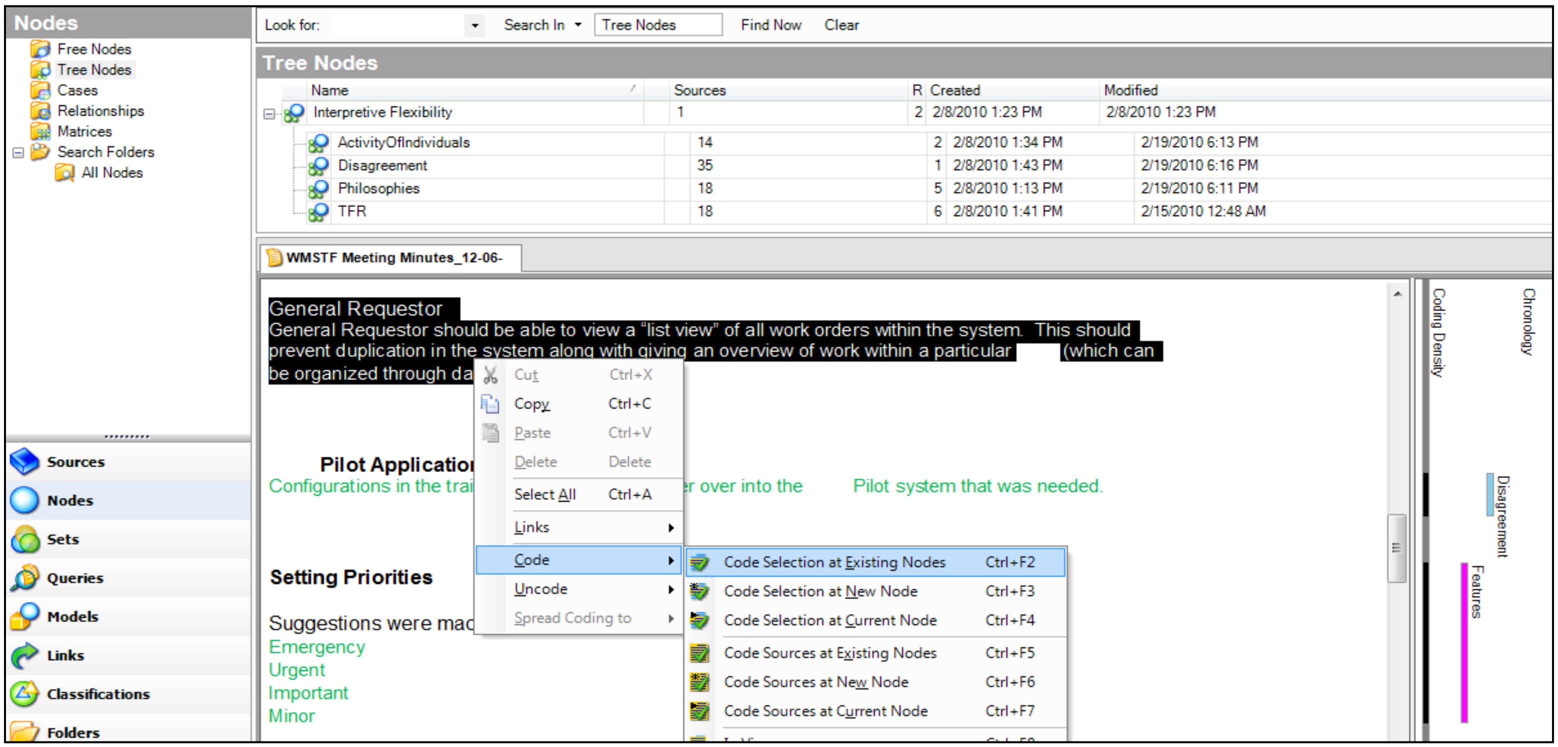

Figure B.1: A Snapshot of NVivo

Appendix C: Snapshots of Different Pages of Datastream 7i

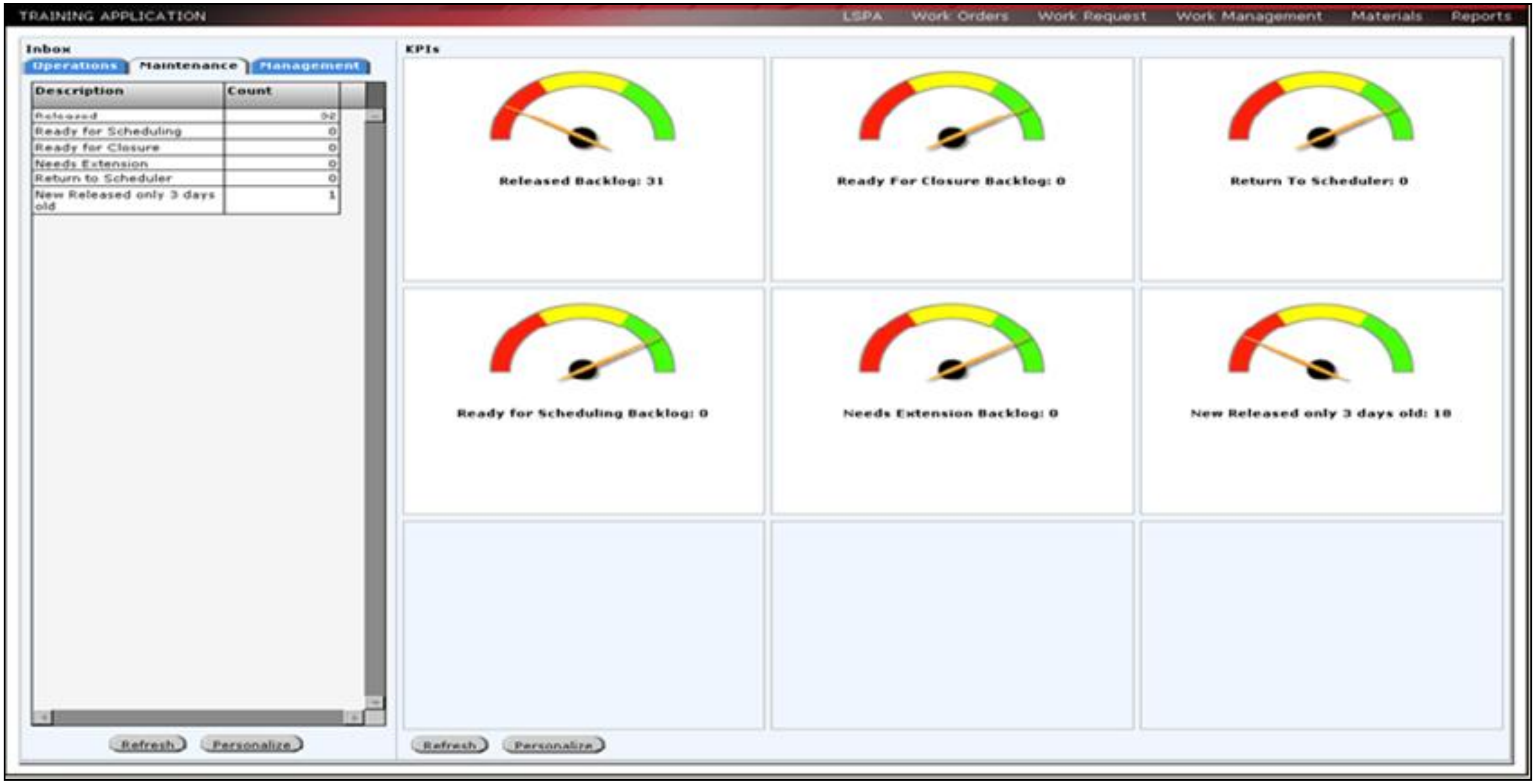

Figure C.1: Inbox (Left) and KPI (Right) Configuration on Datastream 7i

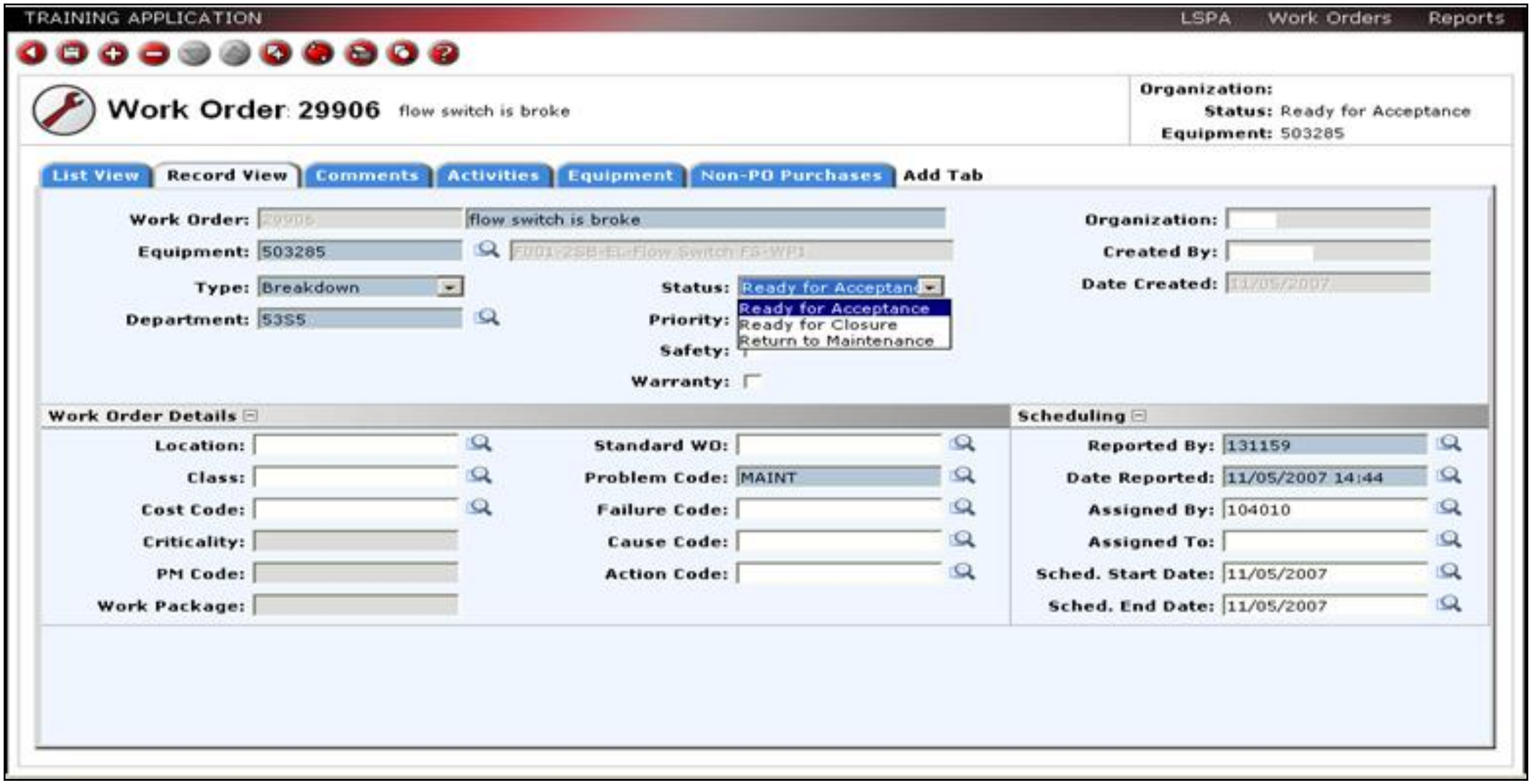

Figure C.2: Details of a Work Order Record Configuration on Datastream 7i

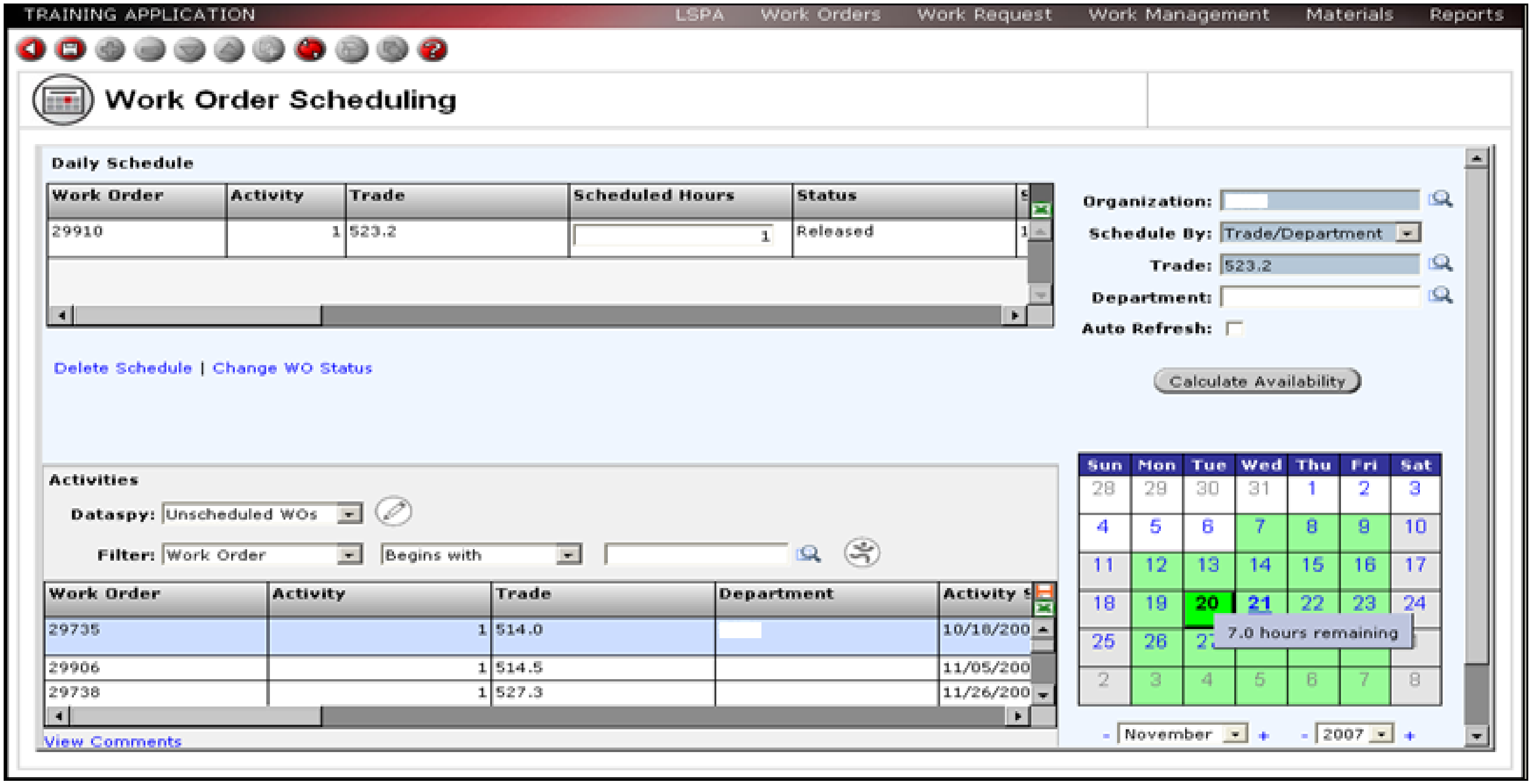

Figure C.3: Calendar Feature Configuration on Datastream 7i

Bibliography:

Abelson, R.P. (1981) "Psychological status of the script concept," *American Psychologist*, 36(7), pp. 715-729.

Allen, J.P. (2000) "Information systems as technological innovation," *Information Technology and People,* 13(3), pp. 210-221.

Anderson, E. (1990) "Choice models for the evaluation and selection of software packages," *Journal of Management Information Systems*, 6(4), pp. 123-138.

Ashby, W. R. (1956) *An introduction to cybernetics*, London: Chapman & Hall.

Ashforth, B.E. & Y. Fried (1988) "The mindlessness of organizational behaviors," *Human relations*, 41, pp. 305-329.

Azad, B. & S. Faraj (2007) "Examining alignment of frames using actor-network constructs: The implementation of an IT project," *Proceedings of the 13th Americas Conference on Information Systems (AMCIS),* Keystone, Colorado, August 2007.

Azad, B. & S. Faraj (2008) "Making e-government systems workable: Exploring the evolution of frames," *Journal of Strategic Information Systems*, 17(2), 25.

Bamford, R. & W.J. Deibler II (1995) "Configuration management and ISO 9001," *SSQC*, Software Systems Quality Consulting, available at: http://www.ssqc.com/do25v6new.pdf.

Barrett, M.J. (1999) "Challenges of EDI adoption for electronic trading in the London insurance market," *European Journal of Information Systems*, 8, pp. 1-15.

Bartlett, F. C. (1932) *Remembering*, Cambridge, University Press: Cambridge, MA.

Bartunek, J.M., Gordon, J.R. & R.P. Weathersby (1983) "Developing 'complicated' understanding in administrators," *Academy of Management Review*, 8, pp. 273-284.

Bartunek, J.M. (1984) "Changing interpretive schemes and organizational restructuring," *Administrative Science Quarterly*, 29, pp. 355-372.

Basili, V. & B. Boehm (2001) "COTS-Based Systems Top 10 List," *IEEE Computer*, 34(5), pp. 91-93.

Bijker, W. E., T. Hughes and T Pinch (Eds.), (1987), *The Social Construction of Technological Systems: new directions in the sociology and history of technology*, The MIT Press: Cambridge, MA.

Bijker, W. E. (1992), “The Social construction of fluorescent lighting, or how an artifact was invented in its diffusion stage,” In Bijker, W. E. & J. Law (Eds.) *Shaping Technology/Building Society*, pp. 75-102, The MIT Press: Cambridge, MA.

Bijker, W. E. (1995) *Of Bicycles, Bakelites and Bulbs: Toward a theory of sociotechnical change*, MIT Press: Cambridge, MA.

Bijker, W. E. & Pinch, T. J. (2002) “SCOT answers, other questions: A reply to Nick Clayton,” *Technology and Culture*, 43(2), pp. 361-370.

Bloor, D. (1973) “Wittgenstein and Mannheim on the sociology of mathematics,” *Studies in History and Philosophy of Science*, 4, 173-191.

Blumer, H. (1969) *Symbolic interactionism: Perspective and method*, University of California Press: Berkley, CA.

Boczkowski, P.J. (1999) “Mutual shaping of users and technologies in a national virtual community,” *Journal of Communication*, pp. 86-108.

Bodewitz, H.J.H.W., Buurma, H. & G.H. de Vries (1987) “Regulatory science and the social management of trust in medicine,” In Bijker, W. E., T. Hughes and T Pinch (Eds.), *The social construction of technological systems: new directions in the sociology and history of technology*, The MIT Press: Cambridge, MA.

Boehm, B. & C. Abts (1999) “COTS Integration: Plug and Pray?,” *Computer*, pp. 135-138.

Boje, D. (1991) “The storytelling organization: A study of story performance in an office-supply firm," *Administrative Science Quarterly*, 36, pp. 106-126.

Boland, R. J. (1978) “The process and product of system design,” *Management Science*, 24(5), pp. 887-898.

Boland, R.J. (1979) “Control, causality and information systems requirements,” *Accounting, Organizations, and Society*, 4(4), pp. 259-272.

Boland, R. J. (1985) “Phenomenology: A Preferred Approach to Research in Information Systems,” In *Research methods in information systems*, E. Mumford, R. A. Hirschheim, G. Fitzgerald and T, Wood-Harper (Eds.), Amsterdam: North-Holland, pp. 193-201.

Boland, R. J. & W. F. Day (1989) “The Experience of System Design: A Hermeneutic of Organizational Action,” *Scandinavian journal of management*, 5(2), pp. 87-104.

Boland, R. J. (1991) “Information system use as a hermeneutic process,” In *Information systems research: contemporary approaches and emergent traditions*, H. E. Nissen, H. Klein and R. Hirschheim (Eds.), Amsterdam: North-Holland, pp. 439-458.

Boland, Jr., R., and R. Greenberg (1992) "Method and Metaphor in Organizational Analysis," *Accounting, management, and information technology*, 2(2), pp. 117-141.

Bostrom, R.P. & J.S. Heinen(1977a) "MIS problems and failures: A socio-technical perspective, Part I-The causes, *MIS Quarterly*, 1(3), pp. 17-32.

Bostrom, R.P. & J.S. Heinen (1977b) "MIS problems and failures: A socio-technical perspective, Part II-The application of Socio-Technical Theory, *MIS Quarterly*, December, 1977, pp. 11-28.

Brooks, F. (1995) *The mythical man month: Essays on software engineering*, Addison-Wesley: Reading, MA.

Brown, J.S., P. Duguid (1991) "Organizational learning and communities-of-practice: toward a unified view of working, learning, and innovation," *Organization Science*, 2(1), pp. 40-57.

Brownsword, L., Oberndorf, T. & C. Sledge (2000) "Developing new processes for COTS-based systems," *IEEE Software*, pp. 48-55.

Butler, T. (1998) "Towards a hermeneutic method for interpretive research in information systems," *Journal of Information Technology* 13 (4), pp. 285-300.

Cano, J.J. (2003) "Technological frames recursive construction approach: a systematic theory for information technology incorporation in organizations," In Cano, J.J. *Critical reflections on information systems: a systematic approach*. Hershey, PA: Idea Group Publishing, pp. 31-45.

Carmel, E. (1997) "American hegemony in packaged software trade and the culture of software," *Information Society*, 13(1), pp. 125–142.

Carmel, E. & Sawyer, S. (1998) "Packaged software development teams: what makes them different?" *Information Technology & People*, 11(1), pp. 7-19.

Carney, D. & F. Long (2000) "What do you mean by COTS?," *IEEE Software*, pp. 83-86.

Chateaubriand, O. (2003) "Quine and ontology," *Principia* 7 (1/2), pp. 41-74.

Clayton, N. (2002) "SCOT: Does it answer?," *Technology and Culture*, 43(2), pp. 351-360.

Collins, H. M. (1981) "Stages in the empirical programme of relativism," *Social Studies of Science*, 11, pp. 3-10.

Collins, H. M. (1983) “An empirical relativist programme in the sociology of scientific knowledge,” In Knorr-Cetina and Mulkay (Eds.), *Science Observed.* London: Sage Publications.

Daft, R.L. & K.E. Weick (1984) “Toward a model of interpretation systems,” *Academy of Management Review*, 9, 284-295.

Dagwell, R. & Weber, R. (1983) “Systems designers’ user models: A comparative study and methodological critique,” *Communications of ACM*, 2(11), pp. 987-997.

Dart, S. (1991) “Concepts in configuration management systems,” *Proceedings of the 3rd international workshop on Software configuration management*, Trondheim, Norway, pp. 1-18.

Davenport, T.H. (1998) “Putting the Enterprise into the Enterprise System,” *Harvard Business Review*, pp. 121-131.

Davidson, E.J. (1996) “Framing Information Systems Requirements: An Investigation of Social Cognitive Processes in Information Systems Delivery,” *Doctoral Dissertation*, Sloan School of Management, Massachusetts Institute of Technology.

Davidson, E.J. (1997) “Changing Frames or Framing Change? Social Cognitive Implications of Organizational Change during IT Adoption,” *Proceedings of the 30th Hawaii International Conference on Systems Sciences (HICSS)*, 6, pp. 475-484

Davidson, E.J. (2002) “Technology frames and framing: A socio-cognitive investigation of requirements determination,” *MIS Quarterly*, 26(4), pp. 329–358.

Davidson, E.J. & D. Pai (2004) “Making Sense of Technological Frames: Promise, Progress, and Potential,” *IFIP International Federation for Information Processing*, 143, pp. 473-491.

Davidson, E.J. & M. Chiasson (2005) “Contextual influences on technology use mediation: a comparative analysis of electronic medical record systems,” *European Journal of Information Systems*, 14, pp. 6–18.

Davidson, E. (2006) “A technological frames perspective on information technology and organizational change,” *Journal of Applied Behavioral Science*, 42(1), pp. 23–39.

Davis, F.D. (1986) “A Technology Acceptance Model for Empirically Testing New End user Information Systems: Theory and Results,” *Doctoral Dissertation*, Sloan School of Management, Massachusetts Institute of Technology.

Denzin, N.K. (1978) *The research act: A theoretical introduction to sociological methods.* New York: McGraw-Hill.

Denzin, N.K. and Lincoln, Y.S. (1998) *Collecting and interpreting qualitative materials*, SAGE Publications: Thousand Oaks, CA.

Dosi, G. (1982) "Technological Paradigms and Technological Trajectories: A Suggested Interpretation of the Determinants and Directions of Technical Change," *Research Policy*, 11, pp.147-62.

Douglas, S. J. (1987) *Inventing American broadcasting*, 1899-1 922, Baltimore.

Douglas, S.J. (1990) "The social construction of technological systems: New directions in the sociology and history of technology," *Isis*, 81(306), pp. 80-84.

Dreiling, A., Rosemann, M., van der Aalst, W., Heuser, L. & K. Schulz (2006) "Model-based software configuration: patterns and languages," *European Journal of Information Systems*, 15, pp. 583–600.

Eden, C. (1992) "On the nature of cognitive maps," *Journal of Management Studies*, 29(3), pp. 261-265.

El Sawy, O.A. & T.C. Pauchant (1988) "Triggers, templates and twitches in the tracking of emerging strategic issues," *Strategic Management Journal*, 9, pp. 455-473.

Erran Carmel, E. & S. Sawyer (1998) "Packaged software development teams: what makes them different?," *Information Technology & People*, 11(1), pp. 7-19.

Estublier, J. (2000) "Software configuration management: a roadmap," In *ICSE '00: Proceedings of the Conference on the Future of Software Engineering*, ACM Press, pp. 279–289.

Feblowitz, M.D. & S.J. Greenspan (1988) "Scenario-Based Analysis of COTS Acquisition Impacts," *Requirements Engineering*, 3, pp. 182-201.

Finkelstein A., Spanoudakis G. & M. Ryan (1996) "Software package requirements and procurement, In *Proceedings of the 8th International Workshop on Software Specification and Design*, IEEE Computer Society Press, Washington, DC, pp. 141-145.

Ferguson, E. S. (1974) "Toward a Discipline of the History of Technology," *Technology and Culture*, 15, pp. 13-30.

Fischer, C. S. (1992) "America Calling: A Social History of Telephone to 1940, University of California Press: Berkeley, CA.

Fiske, S.T. & S.E. Taylor (1984) *Social cognition, teading*, Addison-Wesley: MA.

Gadamer, H.G. (1975) *Truth and method*, The Seabury Press: NY.

Gallivan, M. J. (2001) "Meaning to change: how Diverse stakeholders interpret organizational communication about change initiatives," *IEEE Transactions on Professional* Communication, 44(4), pp. 243-266.

Geertz, C. (1973) *The interpretation of cultures*, Basic Books: NY.

Giddens, A. (1984) *The constitution of society,* Polity: Cambridge, MA.

Gioia, D.A. (1984) "Scripts in organizational behavior," *Academy of Management Review*, 9, pp. 449-459.

Gioia, D.A. & C.C. Manz (1985) "Linking cognition and behavior: A script processing interpretation of vicarious learning," *Academy of Management Review*, 10, pp. 527-539.

Gioia, D. A. (1986) "Conclusion: The state of the art in organizational social cognition: A personal view," In H.P. Sims & D. A. Gioia (Eds.), *The thinking organization: Dynamics of organizational social cognition,* Jossey-Bass: San Francisco, CA, pp. 336-356.

Gioia, D.A., & H. R. Jr. Sims (1986) "Introduction: Social cognition in organizations," In *The thinking organization.* Jossey-Bass: San Francisco, CA, pp. 1-19.

Goetz, M. (2001) "Memoirs of a software pioneer: Part 1," *IEEE Annals of the History of Computing*, pp. 43-56.

Goodman, P., Griffith, T.L. & B. Fenner (1990) "Understanding technology and the individual in an organizational context," In *Technology and Organization*, Jossey-Bass, San Francisco: CA, pp. 45-86.

Gummesson, E. (1991) *Qualitative methods in management research*, Sage: London.

Hedgecoe, A. & P. Martin (2003) "The drugs don't work: expectations and the shaping of Pharmacogenetics," *Social Studies of Science*, 33(3), pp. 327-364.

Hirschheim, R. & M. Newman (1991) "Symbolism and information systems development: Myth, metaphor and magic," *Information Systems Research*, 2(1), pp. 29-62.

Humphreys, L. (2005), "Reframing social groups, closure, and stabilization in the social construction of technology," *Social Epistemology*, 19(2-3), pp. 231-253.

Hughes, T.P. (1987) "The Evolution of Large Technological Systems," In: Bijker, W.E., Hughes, T.P. and Pinch, T., (Eds.), *The social construction of technological systems: New directions in the sociology and history of technology*, The MIT Press, Cambridge: MA, pp. 51-82.

Huff, A.S. (1990) *Mapping strategic thought*, Wiley: Chichester.

Iivari, J. (1990) "Implementability of in-house development vs. application package based information systems," *DATABASE*, pp. 1-10

Isabella, L.A. (1990) "Evolving interpretations as a change unfolds: How managers construe key organizational events," *Academy of Management Journal*, 33(1), pp. 7-41.

Janson, M.A. & A. Subramanian (1996) "Packaged software: selection and implementation policies," *INFOR*, 34(2), pp. 133-151.

Johnston, R. (1972) "The Internal Structure of Technology," In P. Halmos (Ed.), *The sociology of science*, Sociological Review Monograph, (18), pp. 117- 30.

Keen, P.W., Bronsema, G.A., & S. Zuboff (1982) "Implementing Common Systems: One Organization's Experience," *Systems, Objectives, Solutions*, 2(2), pp. 125-134.

Khoo, M. (2001) "Community design of DLESE collections review policy: A technological frames analysis," In *Proceedings of the First ACM/IEEE-CS Joint Conference on Digital Libraries*, Association for Computing Machinery: New York, pp. 157-164.

M. Khoo, (2004) "The use of visual artifacts in the user-informed development of an educational digital library collection," *Journal of Digital Information* 5(3).

Kiesler, S. & L. Sproull (1982) "Managerial response to changing environments: Perspectives on problem sensing from social cognition," *Administrative Science Quarterly*, 27(4), pp. 548-570.

Klein, H. K., and Myers, M. D. (1999) "A Set of principles for conducting and evaluating interpretive field studies in information systems," *MIS Quarterly* 23(1), pp. 67-94.

Kilker, J. & G. Gay (1998) "The social construction of a digital library: A case study examining implications for evaluation," *Information Technology and Libraries*, 17(2), pp. 60-70.

Kline, R. & T. J. Pinch (1996) "Users as agents of technological change: The social construction of the automobile in the rural United States," *Technology and Culture*, 37(4), pp. 763-795.

Kuhn, T. S. (1962) *The structure of scientific revolutions*, Chicago: University of Chicago Press.

Kunda, D., & L. Brooks (2000) "Identifying and classifying processes (traditional and soft factors) that support COTS component selection: A case study," *European Journal of Information Systems*, 9, pp. 226–234.

Langfield-Smith, K. (1992) "Exploring the need for a shared cognitive map," *Journal of management studies*, 29, pp. 349-368.

Law, J. (1987) "Technology and heterogeneous engineering: The case of Portuguese expansion," In Bijker, W. E., T. Hughes and T Pinch (Eds.), *The Social Construction of Technological Systems: new directions in the sociology and history of technology*, The MIT Press: Cambridge, MA.

Law, Y., & J. Lee-Partridge (2003) "Towards a strategy for sense-making of empirical knowledge management perceptions-The TFL methodology," In *Proceedings of the 36th Annual Hawaii International Conference on Systems Science*, IEEE Computer Society Press: Los Alamitos, CA.

Layton, E. (1977) "Conditions of technological development," In I. Spiegel-Rosing and D.J. deSolla Price (Eds.), *Science, Technology, and Society*, Sage: London and Beverly Hills, California, pp. 210.

Lee, A.S. (1989) "A scientific methodology for MIS case studies," *MIS Quarterly*, 13(1), pp. 33-52.

Lee, A.S. (1994) "Electronic mail as a medium for rich communication: An empirical investigation using hermeneutic interpretation," *MIS Quarterly*, pp. 143-157.

Lee, A.S. & Baskerville, R. L. (2003) "Generalizing generalizability in information systems research," *Information Systems Research*, 14, pp. 221–243.

Light, B. (2005a) "Going beyond 'misfit' as a reason for ERP package customization," *Computers in Industry*, 56, pp. 606-619.

Light, B. (2005b) "Potential pitfalls in packaged software adoption," *Communications of ACM*, 48(5), pp. 119-121.

Light, B. & S. Sawyer (2007) "Locating packaged software in information systems research," European *Journal of Information Systems*, 16, pp. 527-
530.

Lin, A. (2000) "Initiating system innovation: A technological frame analysis of the origins of groupware projects," Doctoral Dissertation, Department of Information Systems, London School of Economics.

Lin, A. & T. Cornford (2000) "Framing implementation Mmnagement," *Proceedings of the International Conference on Information Systems*, Sydney, Australia, pp. 197-205.

Lin, A. & L. Silva (2005) "The social and political construction of technological frames," *European Journal of Information Systems*, 14, pp. 49–59.

Lord, R.G. & R.J. Foti (1986) "Schema theories, information processing, and organizational behavior," In H.P. Sims & D.A. Gioia (Eds.), *The Thinking Organization: Dynamics of Organizational Social Cognition*, Jossey-Bass: San Francisco, CA, pp. 20-48.

Lucas, H.C., Walton, E.J. & M.J. Ginzberg (1988) "Implementing packaged software," *Management Information Systems Quarterly*, 12(4), pp. 537-549.

Mackay, W. E. (1988) "Diversity in the use of electronic mail," *ACM Transactions on Office Information Systems* 6(4), pp. 380–397.

Markus, H. & R.B. Zajonc (1985) "The cognitive perspective in social psychology," In G. Lindzey & E. Aronson (Eds.), *Handbook of social psychology,*Random House: New York, pp. 137-230.

Markus, M.L. & N. Bjørn-Andersen (1987) "Power over users: Its exercise by system professionals," *Communications of ACM*, 30(6), pp. 498-504.

Mason, R. (1991) "The role of metaphors in strategic information systems planning," *Journal of Management Information Systems*, 9(2), pp. 11-30.

Martin, M. (1991) ""Hello Central?" Gender, technology and culture in the formation of telephone systems, McGill-Queens: Montreal.

MacKenzie, D., & J. Wajcman (1985) "The social shaping of technology," Open University Press: Milton Keynes, England.

McGovern,T., & C. Hicks (2004) "How political processes shaped the IT adoption by a small make-to-order company: A case study in the insulated wire and cable industry," *Information and Management*, 42, pp. 243-257.

MacLeod M. & E. Davidson (2007) "Organizational storytelling and technology innovation," *Proceedings of the 40th Annual Hawaii International Conference on System Sciences* (HICSS).

McLoughlin, I., Badham, R., & P. Couchman (2000) "Rethinking a-political process in technological change: Socio-technical configurations and frames," *Technology Analysis & Strategic Management*, 12, pp. 17-37.

Miles, M. & A. Huberman (1994) *Making good sense: Qualitative data analysis: An expanded sourcebook*, Sage publications: Thousand Oaks, CA.

Mohamed, A., Ruhe, G., Eberlein, A. (2007) "MiHOS: an approach to support handling the mismatches between system requirements and COTS products," *Requirements Engineering*.

Montazemi, A.R., Cameron, D.A. & K.M. Gupta (1996) "An empirical study of factors affecting software package selection," *Journal of Management Information Systems*, 13(1), pp. 89-105.

Moon, S. Reviewed work(s): Of Bicycles, Bakelites, and Bulbs: Toward a theory of sociotechnical Change by Wiebe E. Bijker Source: Science, Technology, & Human Values, Vol. 22, No. 1, (Winter, 1997), pp. 127-130.

Morgan, G., Smircich, L. (1980) "The Case for Qualitative Research," *The Academy of Management Review*, 5(4), pp. 491-500.

Morisio, M. & M. Torchiano (2002) "Definition and classification of COTS: a proposal," *ICCBSS 2002*, Orlando (FL), February 4-6.

Mumford, E. (2006) "The story of socio-technical design: reflections on its successes, failures and potential," *Information Systems Journal*, 16, pp. 317-
342.

Munitz, Milton K. (1981) *Contemporary Analytic Philosophy*, Macmillan: New York & London.

Neisser, U. (1967) *Cognitive Psychology*, Appleton-Century-Crofts: New York.

Norman, D.A. (1993) "Cognition in the head and in the world: An introduction to the special issue on situated action," *Cognitive Science*, 17(1), pp. 1-6.

Oberndorf, T. (1997) "COTS and open systems–An overview," available at: http://www.sei.cmu.edu/str/descriptions/cots.html#ndi

Oldfield, R.C. & O.L. Zangwill (1942) "Head's concept of schema and its application in contemporary British psychology," *British Journal of Psychology*, 32, pp. 267-286.

Orlikowski, W.J. (1992) "Learning from notes: Organizational issues in groupware implementation," *Proceedings of the Conference on Computer Supported Cooperative Work*, Toronto, Canada pp. 362-369.

Orlikowski, W.J. & Gash, D.C. (1992) "Changing frames: Understanding technological change in organizations," *Sloan School of Management*, Working Paper #3368, Massachusetts Institute of Technology: Cambridge, MA.

Orlikowski, W.J. & Gash, D.C. (1994) "Technology Frames: Making Sense of information technology in organizations," *ACM Transactions on Information Systems*, 12(2), pp. 174-207.

Orlikowski, W. & J. Baroudi (1991) "Studying information technology in organizations: Research approaches and assumptions," *Information Systems Research*, 2(1), pp. 1–28.

Orlikowski W. J. & C.S. Iacono (2000) “The truth is not out there: an enacted view of the ‘digital economy’,” In *Understanding the digital economy: Data, tools, and research*, (Ed.) E. Brynjolfsson, B. Kahin, MIT Press: Cambridge, MA, pp. 352–80.

Ovaska, P., Rossi, M. & K. Smolander (2005) “Filtering, negotiating and shifting in the understanding of information system requirements,” *Scandinavian Journal of Information Systems* 17, pp. 31-66.

Pettigrew, A.M. (1985) “Contextualist research: a natural way to link theory and practice,” In Lawler, E.E., Mohrman, A.M., Mohrman, S.A., Ledford, G.E. and Cummings, T.G. (Eds.), *Doing research that is useful in theory and practice*, Jossey-Bass: San Fransisco, CA.

Pinch, T. J. & W. E. Bijker (1984) “The social construction of facts and artefacts: Or how the sociology of science and the sociology of technology might benefit each other. *Social Studies of Science*, 14(3), pp. 399-441.

Pinch, T. J. & W. E. Bijker (1987) “The social construction of facts and artifacts: Or how the sociology of science and the sociology of technology might benefit each other,” In *The social construction of technological systems: New direction in the sociology and history of technology*, W. E. Bijker, T. P. Hughes, and T. J. Pinch (Eds.), Cambridge, MA: MIT Press.

Poole, P.P., Gioia, D.A. & B. Gray (1989) “Influence modes, schema change, and organizational transformation,” *Journal of Applied Behavioral Science*, 25, pp. 271-289.

Porac, J.F. & H. Thomas (1990) “Taxonomic mental models in competitors definition,” *Academy of Management Review*, 15, pp. 224-240.

Puri, S.K. (2006) “Technological frames of stakeholders shaping the SDI implementation: A case study from India,” *Information Technology for Development*, 12(4), pp. 311-331.

Quine, W.V.O. (1953) “On what there is,” as reprinted in *From a logical point of view*, Harper & Row: New York.

Quine, W.V. (1969) “Epistemology Naturalized,” In W.V. Quine, *Ontological Relativity and Other Essays*, Columbia University Press: New York and London.

Ricoeur, P. T(1978) “The model of the text: Meaningful action considered as a text,” *Social Research*, 38, pp. 529-562.

Ricoeur, P. (1981) “The model of the text: Meaningful action considered as a text,” In J.B. Thompson (Ed.) *Hermeneutics and the human sciences*, Cambridge University Press: England, pp. 197-221.

Roberts, F. S. (1976) “Strategy for the energy crisis: The case of commuter transportation policy,” In R. Axelrod (Ed.), *The structure of decision: The cognitive maps of political elites*, Princeton University Press: Princeton, NJ, pp. 142-179.

Rogers, E.M. (1995) *Diffusion of Innovations*, Free Press: New York

Rosenkopf, L. & M.L. Tushman (1998) “The coevolution of community networks and technology: Lessons from the flight simulation industry,” *Industrial and Corporate Change*, 7, pp. 311-346.

Russell, S. (1986) “The social construction of artefacts: A response to Pinch and Bijker,” *Social Studies of Science*, 16, pp. 331-346.

Sætnan, A. R. (1991) “Rigid politics and technological flexibility: The anatomy of a failed hospital innovation,” Science, Technology, and Human Values, 16(4), 419-447.

Sahay, S., Palit, M. & D. Robey (1994) “A relativist approach to studying the social construction of information technology,” *European Journal of Information Systems* 3, pp. 248-258.

Saunders, P. (1982) “Phenomenology: A new way of viewing organizational research,” *Academy of Management Review*, 7(3), pp. 353-360.

Sawyer, S. (2000) “Packaged software: Implementation of the differences from custom approach to software development,” *European Journal of Information Systems*, 9(1), pp. 47-58.

Scott, S.V. & E.L. Wagner (2003) “Networks, negotiations, and new times: the implementation of enterprise resource planning into an academic administration,” *Information and Organization*, 13, pp. 285–313.

Schneider, S.C. & A. DeMeyer (1991) “Interpreting and responding to strategic issues and the impact of national culture,” *Strategic Management Journal*, 12, pp. 307-320.

Schneider, S.C. & R. Angelmar (1993) “Cognition in organizational analysis: Who’s minding the store?,” *Organizational Studies*, 14, pp. 347-374.

Shaw, N., Lee-Partridge, J., & J. Ang (1997) “Understanding end-user computing through technological frames,” In J. J. DeGross & K. Kumar (Eds.), *Proceedings of the 18th International Conference on Information Systems* (pp. 453-459). Atlanta, GA: Association for Information Systems.

Shrivastava, P. & I. Mitroff (1983) “Frames of reference managers use. A study in applied sociology of knowledge and knowledge use,” In R. Lamb (Ed.), *Advances in strategic management*, 1, JAI Press: Greenwich, CT.

Sharp, L. (1952) “Steel axes for stone age Australians,” In *Human Problems in Technological Change*, E. H. Spicer (Ed.), Russell Sage Foundation: New York, pp. 69-72.

Sia, S.K. & Soh, C. (2007) “An assessment of package-organisation misalignment: institutional and ontological structures,” *European Journal of information Systems*, 16, pp. 568-583.

SIIA (2004) “Packaged software industry revenue and growth,” *Software and information industry association*, Available at: http://www.siia.net/software/pubs/growth_software05.pdf

Sims, H. R. Jr. & D.A. Gioia (Eds.) (1986) *The thinking organization*, Jossey-Bass: San Francisco, CA.

Skinner, B. F. (1953), *Science and Human Behavior*, Free Press: New York.

Smith, B. (2003) “Ontology,” In Luciano Floridi (ed.), *Blackwell Guide to the Philosophy of Computing and Information*, Blackwell: Oxford, 2003, pp. 155-166.

Soh, C., Sia, S.K. & J. Tap-Yap (2000) “Cultural fits and misfits: is ERP a universal solution?,” *Communication of the ACM*, 43(4), pp. 47-51.

Soh, C. & S.K. Sia (2004) “An institutional perspective on sources of ERP package-organisation misalignments,” *journal of Strategic information Systems*, 13(4), pp. 375-397.

Soffer, P., Golanya, B., Doria, D. & Y. Wand (2001) “Modeling off-the-shelf information systems requirements: an ontological approach,” *Requirements Engineering*, 6, pp. 183–199.

Sommerville, I. (2008) “Construction by configuration: challenges for software engineering research and practice,” *19th Australian Conference on Software Engineering*, Perth (Western Australia), March 25-28, IEEE Computer Society.

Suzanne Moon (1997) review of “*Of Bicycles, Bakelites, and Bulbs*,” *Science, Technology and Human* Values, 20, pp. 127–130.

Taylor, C. (1971) “Interpretation and the Sciences of Man,” *Review of Metaphysics*, 25 (7), pp. 3-51.

Tenkasi, R., and R. Boland (1993) “Locating meaning making in organizational learning: The narrative basis of cognition,” *Research in organizational change and development,* 7, pp. 77-103.

Trauth, E. & E. Cole (1992) “The organizational interface: A method for supporting end users of packaged software," *Management Information Systems Quarterly,* 16(1), pp. 35-53.

Ulkuniemi, P. & V. Seppänen (2004) “COTS component acquisition in an emerging market,” *IEEE Software*, pp. 76-82.

Van den Belt, H. & A. Rip (1987) “The Nelson-Winter-Dosi model and synthetic dye chemistry,” In W.Bijker, T.Hughes & T.Pinch (Eds.), *The social construction of technological systems*, pp. 135-158. Cambridge, MA: MIT Press.

van Nierop, A. & Blankendaal, A. C. M. & C. J. Overbeeke (1997) “The evolution of the bicycle: A dynamic systems approach,” *Journal of Design History*, 10(3), pp. 253-267.

Vigder, M., Gentleman, M. & J.Dean (1996) “COTS software integration: State of the art,” *Technical Report No. 39190, National Research Council of Canada*, pp. 1-25.

Vigder M. & J. Dean (1998) “Managing long-lived COTS based systems,” *NRC report no. 41587, National Research Council of Canada*, August 1998.

Visker, G. & A. H. Bree (1987) “Standard software: A survey,” *Information*, 29(10), pp. 902-907.

Von Hippel, E. (1988) *The sources of innovation.* Oxford University Press: New York.

Wagner, L. Scott, S.V. & Galliers R.D. (2006) “The creation of 'best practice' software: Myth, reality, and ethics” *lnformation and Organisation*, 16, pp. 251-275.

Walsh, J.P. (1995) “Managerial and organizational cognition: Notes from a trip Down memory lane,” *Organization Science*, 6(3), pp. 280-321.

Walsham, G. (1993) *Interpreting information systems in organizations*, John Wiley: Chichester.

Walsham, G. (1995) “Interpretive case studies in IS research: Nature and method,” *European Journal of Information Systems*, 4(2), pp. 74-81.

Walsham, G. (2005) “Learning about being critical,” *Information Systems Journal*, 15, pp. 111-117.

Walsham, G. (2006) “Doing interpretive research,” *European Journal of Information Systems*, 15, pp. 320-330.

Walz, D., Elam, J. and B. Curtis (1993) “Inside a software design team: knowledge acquisition, sharing, and integration,” *Communications of the ACM,* 36(10), pp. 63-77.

Weick, K.E. (1979) “Cognitive processes in organizations,” In B. Staw (Ed.), *Research in Organizational Behavior,* 1, *JAI Press: Greenwich,* CT, pp. 41 -74.

Weick, K.E. & M.G. Bougon (1986) “Organizations and cognitive maps: charting ways to success and failure,” In H.P. Sims & D.A. Gioia (Eds.), *The Thinking Organization*, Jossey-Bass: San Francisco, CA, pp. 102-135.

Weilenmann, A. (2000) “Negotiating use: Making sense of mobile technology,” *The Journal of Personal Technologies* (4), Springer Verlag.

Winner, L. (1993) “Upon opening the black box and finding it empty,” *Science, Technology, and Human Values,* 18(3), pp. 362-441.

Wingerd, L. & C. Seiwald (1997) “Constructing a large product with jam. In *ICSE ’97: Proceedings of the SCM-7 Workshop on System Configuration Management*, Springer-Verlag, pp. 36–48.

Woodforde, J. (1970) *The Story of the Bicycle*, Routledge and Kegan Paul: London.

Woodworth, R. S. (1981) *Experimental Psychology,* Henry Holt: New York.

Xu, L. & S. Brinkkemper (2007) “Concepts of product software,” *European Journal of Information Systems*, 16, pp. 531-541.

Yeow, A. & S. K. Sia (2008) “Negotiating ‘best practices’ in package software implementation,” *Information and Organization*, 18, pp. 1–28.

Yoshioka, T., Yates, J., & W. Orlikowski (2002) “Community-based interpretive schemes: Exploring the use of cyber meetings within a global organization,” In *Proceedings of the 35th Annual Hawaii International Conference on Systems Science*, IEEE Computer Society Press: Los Alamitos, CA.

Yin, R.K. (1986) *Case study research*, Newbury Park, Sage: CA.

Yin R.K. (1993) *Applications of Case Study Research,* Sage: Thousand Oaks, CA.

Yin R.K. (1994) *Case Study Research: Designs and Methods*, 2nd ed., Sage: Thousand Oaks, CA.

Yin, R.K. (2002) *Case Study Research, Design and Method*, 3rd ed., Sage Publications: Newbury Park.

Zuboff, S. (1988) *In the Age of the Smart Machine*, Basic Books: New York.