DOCTOR OF PHILOSOPHY

A formal approach to modelling knowledge transfer processes

Tong, Jin

Award date:
2012

Awarding institution:
Coventry University

A Formal Approach to Modelling Knowledge Transfer Processes

Jin Tong

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE UNIVERSITY'S REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

January 2012

Acknowledgement

First of all I would like to thank my director of study Dr Siraj A. Shaikh for his continuous supervision and support throughout the undertaking of my PhD research.

This work would have not been possible without the support from the Faculty of Engineering and Computing at Coventry University and the contributions from all the individuals who participated in this research. In particular I thank Professor Anne James.

I also thank Dr Robert Ayres, Professor Peter Hill, Dr Amit Mitra and all of those who have helped me throughout my PhD study.

Finally, thanks to my husband Tolis and my parents, for their encouragement, patience and understanding during all these years. This work is especially dedicated to them.

List of publications

Tong, J., Shaikh, S. A. and James, A. E. (2011) “A formal approach to analysing knowledge transfer processes in developing countries”, *Proceeding of 9th International Conference on Software Engineering and Formal Methods*, Montevideo, Uruguay, 14-18 November, 2011.

Tong, J. and Shaikh, S. A. (2010) “ICT driven knowledge management in developing countries: a case study in a Chinese Organisation”, *Proceeding of the Wireless Communication and Information Technology in Developing Countries*, Brisbane, Australia, 20-23 September 2010.

Tong, J. and Shaikh, S. A. (2010) “A formal approach to modelling knowledge transfer processes”, *Proceeding of the 11th European Conference on Knowledge Management*, Famalico, Portugal, 2-3 September 2010.

Tong, J. and Ayres, R. (2009) “Knowledge needs: uncharted area in knowledge management”, *Proceeding of the International Conference on Organizational Learning, Knowledge and Capabilities*, Amsterdam, Netherlands, 26-28 April, 2009.

Tong, J. and Mitra, A. (2009) “Cultural influence on knowledge management practices: a case study in a Chinese manufacturing enterprise”, *Journal of Knowledge Management*, Vol. 13, No. 2, April, pp. 49-62.

Tong, J. and Ayres, R. (2009) “A low-level model for knowledge transfer”, *Proceeding of the IADIS International Conference on Information Systems*, Barcelona, Spain, 25-27 February, pp. 169-176.

Tong, J. and Mitra, A. (2009) “Knowledge maps and organisations: an overview and interpretation”, *International Journal of Business Information Systems*, Vol. 3, No. 6, pp. 587-608.

Tong, J. and Mitra, A. (2006) “Cultural influence on knowledge management practices: a case study in a Chinese manufacturing enterprise”, *Proceeding of the 3rd Asia – Pacific International Conference on Knowledge Management*, Hong Kong, 11-13 December, 2006.

Abstract

An organisation's competitive advantage depends on its ability to transfer knowledge effectively. Research suggests that knowledge transfer (KT) remains a problem for many organisations. The available literature on this subject is loosely associated and often decoupled from the context of KT processes. Consequently, very few of the existing studies can be used directly to diagnose any transfer problems and identify appropriate strategies in practice. This situation could be improved if we relate these studies to a low-level KT model. The aim of this research is to define KT at a detailed level and develop a model that can be used for analysing KT problems in practice.

By following a first principles approach, a graphical low-level KT model is developed. This model allows a wide range of KT processes to be represented by showing people's behaviours that take place in the course of transfer. It can be used in reviewing an organisation's KT practice and proposing suitable strategies for improvement.

A formal KT model is developed using Communicating Sequential Processes (*CSP*). The dynamics and complexities in the process of KT are well represented using the concurrency theory in *CSP*. This model is formalised based on the graphical model. It is verified through *CSP* model-checking technique using an *CSP* analysis tool – *FDR*. This formal KT model provides a precise and systematic framework in understanding KT.

The formal KT model captures people's behaviours in general KT situations. In reality, KT systems vary and often have deficiencies in many organisations. We also propose to analyse their KT problems using a formal approach which analyses their problematic transfer systems against our formal KT model. This approach is demonstrated in a case study.

The application of a process algebra in analysing KT is a novel idea. It explores a new direction of studying human knowledge related processes in the KM domain. We hope our effort serves to inspire new ideas and approaches to the wider KM community.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research context

While considering knowledge to be one of the most valuable and strategic assets, business organisations are keen to improve their knowledge management (KM) practice. KM is concerned with activities of knowledge generation, transfer, accumulation, adoption, and diffusion [17]. Among these KM processes, knowledge transfer (KT) attracts increasing attention because knowledge as an asset increases in value with use ([15], [38], [8], [30]). An organisation's competitive advantage depends on its ability to transfer knowledge effectively. However, the current research suggests that KT remains a problem for many organisations ([22], [24], [43], [15]).

1.2 Research problem

Davenport and Prusak [15] defined KT as involving two actions – transmission (sending or presenting knowledge to a potential recipient) and absorption by that person or group. In order to help organisations with KT difficulties, current studies in the KM domain mainly focus on two aspects. One perspective is focusing on key issues affecting people's activities in KT, such as trust and cultural issues. Another perspective is looking at various strategies that can facilitate and accelerate KT, such as setting communities of practice(CoPs) and implementing knowledge maps. The available understandings of KT remain at a high-level and they do not provide sufficient details of the specific steps and transactions that take place in the process of transferring knowledge. Their discussions on key transfer issues and mechanisms are often decoupled from the context of KT. Although their work could be very useful in studying KT practices at an organisational level, very few of them can be used directly to analyse particular KT problems at an individual level. This situation could be improved if we relate these studies to a more detailed understanding or model of KT. By KT at an individual level, we mean the transfer activities take place between individuals within the organisational environment.

Based on the researcher's understanding of relevant literature, the overall guiding question for this research can be defined as:

> ***RQ – How to analyse organisational KT problems at an individual level?***

The researcher is aware of the complexities in answering the primary research question. This motivates the definition of a number of more specific research questions derived from the main research question, which also needed to be answered during the research. These secondary research questions are:

- *RQ1 – What do we understand by KT?* By answering this question, the boundary and the context of this research can be defined. The focus of KT at an individual level can also be emphasised.
- *RQ2 – Why are the current KT models not suitable to be used in analysing KT problems at an individual level?* In order to answer this question, we first need to justify why a low-level KT model is needed for KT analysis at an individual level. Then we have to explain why the current models are not suitable for this purpose. This will lead to the suggestion of a need for a new KT model to be developed in this research.
- *RQ3 – How does KT take place at an individual level?* Having a good understanding of how KT takes place will allow us to develop our KT model which can be more applicable for the purpose of KT analysis in this research.
- *RQ4 – How can we analyse KT problems using a KT model?* Answering this question will allow us to generate a procedure of using our KT model in analysing KT problems within organisations.

1.3 Research aim and objectives

On the basis of the research problem identified in the previous section, the aim of this research is defined as:

> ***The definition and development a KT model that can be used for analysing organisational KT problems at an individual level.***

In order to achieve such an aim the following objectives were established:

1. Define what we mean by KT at an individual level and clarify when knowledge is considered to be transferred in particular.
2. Develop a low-level model for KT.
3. Assess the low-level KT model.
4. Develop a formalised version of the low-level KT model.
5. Verify the formal KT model.
6. Propose an approach for analysing KT problems on the basis of the formal KT model.
7. Demonstrate how to apply the formal approach for KT analysis within an organisation.

1.4 Research approaches

There will be the following milestones in this research. First a good understanding of the research problem and its context will be established. This will lead to the development of a clear definition of KT. Then a review of relevant literature will be carried out to help develop and assess a low-level KT model. The next milestone will be met when the formal KT model is developed and verified on the basis of the low-level model. And finally, the formal approach for analysing KT will be proposed and demonstrated.

Research methods and techniques selected to fulfill the above research objectives are listed in Table 1.1.

1.5 Contribution to knowledge

This research will mainly contribute to the domain of KM. It intends to refine the current understanding of KT and suggest a new way of analysing KT problems. Once the research aim and objectives are fulfilled, there will be three primary contributions to knowledge from this research. First, a low-level KT model will be developed. We will explain what is KT and how KT takes place in this low-level model. Second, a formal model for KT will be developed on the basis of the low-level KT model. Such a formalisation will increase our KT model's applicability in analysing KT. We also need to provide a guidance of how to use our model for analysing KT in this research. Therefore a formal approach for the purpose of KT analysis will be proposed as the third primary contribution.

In addition, our research will also contribute to the field of formal modelling, in particular the application of *CSP*. It aims to explore a new application area of formal methods.

1.6 Thesis structure

This thesis is structured as eight chapters. Each of these chapters is summarised as follows.

- The topic of KT and the outline of the research problem have been discussed in this chapter.
- In order to set the context of the research, What we mean by KT in this research is defined in Chapter 2. A review of the literature on two main areas in this domain – key issues affecting KT and major strategies for facilitating KT shows the limited strength of available studies in assisting KT analysis at an individual level. This suggests a low-level model for KT

Research objectives	**Planned research methods or techniques**
Objective 1 – Define KT	A review of relevant literature;
Objective 2 – Develop a low-level model for KT	• Data collection – KT examples collected from a review of relevant literature, semi-structured interviews and a diary study; • Data analysis – Encoding technique for analysing collected KT examples;
Objective 3 – Assess the low-level KT model	Use the same methods as for Objective 2
Objective 4 – Develop the formal KT model	• Formal modelling using *CSP* notation; • Analysis on *CSP* processes with the assistance of *ProBE*;
Objective 5 – Verify the formal KT model	*CSP* Model checking relying on *FDR*: • Failures refinement check; • Deadlock and livelock freedom check;
Objective 6 – Propose a formal approach for KT analysis	Apply the same model analysis method used for fulfilling Objective 5
Objective 7 – Demonstrate how to use the proposed formal approach in an organisation	• Further application of *CSP* model-checking technique as used for Objective 5 and 6; • A small case study at *Lotus*;

Table 1.1: A research plan for fulfilling the research objectives

may be helpful. After on a critique of the available KT understandings, the requirements for developing our own low-level KT model are also specified in this chapter.

- In Chapter 3, the overall research design and the methodological choices made in conducting this research are explained. The relevant literature on formal methods is also reviewed in this chapter.

- A low-level model for KT is presented graphically in Chapter 4. The development of this model is explained here. The renewed definition of KT at an individual level (as defined in Chapter 2) is presented as part of the low-level KT model. The assessment of this model is concerned with representing KT processes at a low-level. This reflects the model's strength in helping us to understand how an organisation practices KT at an individual level.

- Chapter 5 presents a formal KT model, which is developed based on the low-level graphical model. The *CSP* notation used in formalising the model is also introduced in this chapter.

- A verification of the formal KT model through *CSP* failures refinement checks with the assistance of the *CSP* model-checking tool (*FDR*) is described in Chapter 6. A set of desired properties of the formal KT model are specified during the verification. In addition, this formal KT model is also checked for deadlock and livelock freedom.

- The model analysis method used for verifying our formal KT model is proposed as a formal approach for KT analysis in Chapter 7. This chapter focuses on demonstrating this approach through examples taken from a Chinese organisation named *Lotus*.

- In the final chapter, the thesis concludes with a discussion of the path followed by the researcher, limitations of this study and the contributions delivered during the research. Areas for further work are also highlighted.

Chapter 2

Research Background

2.1 Introduction

This chapter explains the research context by reviewing current understanding of KT. It reveals the inconsistency of the definition of KT in related literature and develops a renewed KT definition in the context of this research. It also identifies a major gap in this field. The available discussion of key transfer issues and mechanisms is often related to a high-level understanding of KT. The work being done in this field is not sufficient in helping someone to refine his understanding of specific transfer problems at an individual level. This makes it difficult to facilitate effective transfer practices within organisations.

This chapter is organised as follows. First, the basic concepts in KT are briefly reviewed and what we mean by KT in this research is also defined in Section 2.2. Then in Section 2.3, the literature of two major aspects of current understanding of KT – key transfer issues and major transfer strategies is reviewed. The discussion based on the review reveals that the current understanding of KT is only useful in studying KT at an organisational level. To bridge this gap, a low-level model of KT would be useful. In Section 2.4, the need of developing a low-level KT model in this research is highlighted following a critique of available KT models in literature. Finally in Section 2.4 the focus aspect and the required features of this model are specified.

2.2 Knowledge and KT

KT is the focus of this research and there is much literature available on this subject. This section briefly reviews the basic concepts in this domain and introduces the definitions of knowledge and KT.

2.2.1 Debate on the definition of knowledge

The common understanding of knowledge in relative literature is grounded in Polanyi's [40] work, which extract the essence of Plato's original definition of knowledge as "Justified true belief". However, the debates based on this definition have been appeared in many researchers' work. For example, Nonaka [38] insists that one fact's truthfulness can only be judged by personal beliefs. Newell et al. [37] revisited this theme and argue that beliefs are not always truth, and some truths may not be believed in an organisational context even when full justification for them has been provided.

However no matter how different the definition of knowledge can be, it is widely agreed that knowledge can be split along different dimensions. Existing knowledge classification schema (e.g. Spender [52], [53], Blackler [6], Newell et al., [37]) within organisation studies more or less build on the premise suggested by Polanyi [40], distinguishing between tacit and explicit knowledge. Tacit knowledge is more subjective and experience based, consequently cannot be expressed easily. It always includes cognitive skills and technical skills. Explicit knowledge, on the other hand, is more rational knowledge that can be easily captured and communicated.

The data-to-information-to- knowledge chain is formed through the process of an observer or learner's distinction of different "objects." Davenport and Prusak [15, page 2] explain that "data is most usefully described as structured records of transactions" in organisational context. Individual pieces of data become information when they are classified or organized in meaningful patterns. Information also can be described as a message, because it can be passed around in organisations through organisational networks including both visible and invisible ones [15]. Then information is transformed into knowledge when a person understands the information and is able to apply it for some purposes.

2.2.2 Knowledge share, transfer and exchange

The term KT is often used in a generic sense to describe the knowledge flow between a source and a recipient. Other terms such as knowledge share (KS) and knowledge exchange (KE) are also used interchangeably with KT by some authors. Boyd et al. [8] focus on the misuse of these terms in current literature and list several examples (shown in Table 2.1) of existing definitions of KT (KS or KE).

In order to have the consistency in the present research, it is necessary to clarify the difference between these terms. Lee [31] defines knowledge sharing as "activities of transferring or disseminating knowledge from one person, group or organization to another" [page 324]. This definition is similar to Hall's [27] work that views knowledge sharing as a process where one party gives some knowledge (explicit or tacit) to another party (a person, a group or a repository). Definitions of this term in available literature more or less recognise it as the behaviour of giving away knowledge or making knowledge available to others. However, if the given knowledge has not been absorbed or used by the receiving party, such a sharing action has no value as knowledge is shared in order to make it available to the ones who need it to improve their work. Singley and Anderson [51, page 1] define KT among individuals as "how knowledge acquired in one situation applies (or fails to apply) to other situations". Their definition of KT emphasises the importance of the receiving party's actions following the knowledge sharing behaviour. Davenport and Prusak [15] also mention the necessity of such knowledge absorption in their definition of KT. They explain

Authors	Definition of KT (KS or KE)
Argote and Ingram [2, page 151]	The process through which one unit is affected by the experience of another
Watson and Hewett [61, page 143]	The codification and storage of existing knowledge into knowledge repositories or databases such that it can be accessed and reused
Gooderham [24, page 36]	The accumulation or assimilation of new knowledge in the receiving unit
Riege [43, page 48]	The application of prior knowledge to new learning
Christensen [13, page 37]	Identifying existing and accessible knowledge, in order to transfer and apply this knowledge to solve specific tasks better, faster and cheaper than they would otherwise have been solved

Table 2.1: Boyd et al.'s list of examples of existing definitions of KT (KS or KE) (Adapted from [8, page 139]

that:

> *"KT involves two actions: transmission (sending or presenting knowledge to a potential recipient) and absorption by that person or group. If knowledge is not absorbed, it has not been transferred. Merely making knowledge available is not transfer... The goal of KT is to improve an organization's ability to do things, and therefore increase its value." [15, page 101]*

According to Davenport and Prusak's [15] definition, knowledge sharing is one part of the complete KT process, while knowledge absorption and use by the recipients is the other one.

Knowledge exchange has been used as an alternative term of KT in some research papers (e.g. [1], [32]). During knowledge exchange, a knowledge owner passes on knowledge to another person with expectations that the recipient will reward him with a different piece of knowledge [8]. In practice, people can play different roles simultaneously in different KT processes. When a group of people conducts a set of complete KT processes, knowledge is exchanged within this group. Therefore knowledge exchange can be viewed as the consequence of a set of KT processes.

2.2.3 KT defined in this research

There are also different schools of thought concerning when exactly knowledge is transferred. King [30] summarised a few different views. For instance, some researchers insist that knowledge must be both communicated and applied before the transfer really takes place. Some other researchers believe that transfer can only occur if the recipient of knowledge has the capacity to apply it. Another view is that if the knowledge is understood by the recipient, it has been transferred. King [30] agrees that "each of these viewpoints appears to be useful in certain circumstances, so there is no universal agreement on which is best" [page 538]. Since there is no widely agreed view concerning when exactly KT can be said to have taken place (as explained in 2.2.2), it is necessary to address this issue in our model. We say knowledge is transferred successfully as soon as the knowledge seeker's understanding or knowledge is refined after communicating a set of messages with the providers either directly or through other people. In some situations, a seeker may not necessarily receive the exact piece of knowledge he is searching for. He may only gain some sort of information or knowledge relating to his original knowledge requests. If what he received can help him refine his understanding and guide him to retrieve valuable knowledge in later stage, then we believe that he was in a successful KT process. This also implies that what a seeker received in the transfer has to be understood or absorbed by him before it is successfully transferred.

In the context of this research, we focus on KT among individuals in organisational environments. We do not intend to study the KT processes at an organistaional level. This will be further explained later in Section 2.3. A KT process at the individual level involves a sequence of actions taken and decision made by people. It also involves interactions or transactions between people or between people and knowledge repositories. For example, knowledge providers first give away knowledge through different approaches, such as in person conversations, codifying personal knowledge into organisational knowledge base, using online discussion forum, and so on. Then knowledge seekers manage to access such knowledge, find it useful, and hence apply it. Davenport and Prusak's [15] definition

$$KT = \mathit{Transmission} + \mathit{Absorption} \tag{2.1}$$

intends to explain such KT situations (see more details in Section 2.2.2). It also represents a common view of current understanding of the KT process in available literature. However, their definition does not reflect the complete KT processes.

For instance, a knowledge seeker may realise his knowledge gap and try to search for relevant knowledge before it becomes available. If the knowledge seeker cannot generate required knowledge by himself, he will have to wait for its availability until it has been shared by potential knowledge providers. There are three subprocesses involved in this KT process – requesting new knowledge by the knowledge seeker (once knowledge needs are developed), sharing relevant knowledge by knowledge providers, and then absorbing and using available knowledge by the initial seeker. Received knowledge will be absorbed and used by the recipients only when they find such knowledge valuable. In the former KT example, knowledge recipients are aware of their knowledge needs when they decide to absorb the received knowledge. Knowledge seeking is treated as a part of 'absorption' in Davenport and Prusak's [15] definition. Although they highlight that purely making knowledge available to others cannot ensure the effectiveness and success of the KT processes within organisations, they did not explicitly emphasise that people's willingness to seek for knowledge can also affect the success of absorbing and using acquired knowledge. If we define the movement of knowledge as knowledge flow [48], our understanding of KT implies two different directions of knowledge flow embedded in transfer:

1. *The pulling mode* – First knowledge seekers develop knowledge needs and request new knowledge. Then providers give away knowledge upon requests. And finally, seekers absorb received knowledge and apply it in actions.

2. *The pushing mode* – Providers first give away knowledge, and then knowledge recipients develop knowledge needs. Finally, recipients absorb new knowledge and apply it in actions.

In summary, Davenport and Prusak's [15] definition of KT needs to be revised as the following:

$$KT_{PullingMode} = \textit{Requesting} + \textit{Sharing} + \textit{Absorption and Use} \tag{2.2}$$

$$KT_{PushingMode} = \textit{Transmission} + \textit{Awareness of needs} + \textit{Absorption and Use} \tag{2.3}$$

2.3 A gap in current understanding of KT

Current literature in the field of KT mainly focuses on two aspects. One is to identify key issues affecting the success of KT. The other is to propose strategies to facilitate KT. Although many researchers have contributed in studying both of the two aspects of KT, their work is often at a high-level and is more valuable in analysing KT problems at an organisational level. A low-level understanding or model of KT is more helpful in analysing KT problems at an individual level.

By briefly introducing the mainstream research on KT, this section aims to reveal the lack of a low-level model of KT in available literature.

2.3.1 Key KT issues

A few researchers attempt to summarise the major issues affecting KT practice in the current literature. For example:

- Davenport and Prusak [15] claim that KT cannot succeed without the support of technology, but a company's culture is more critical in determining how successfully knowledge is transferred. They also provide the following list (shown as Table 2.2) of the most common KT barriers derived from cultural factors. Possible ways of overcoming these barriers are suggested in their study too.
- Disterer [17] claims that the people issues are more critical than technical issues for the success of KT. He categorises the impediments to KT at two levels – individual and social. Individual transfer barriers include fear of losing power, fear of revelation, uncertainty of knowledge, and lack of motivation. Social barriers for KT include a lack of common language, attitudes of conflict avoidance, strong bureaucracy and hierarchy, and incoherent paradigms. He reviews empirical evidence available in existing literature (e.g. Ernst and Young survey by Ruggles [46]) and concludes that both individual and social barriers for employees can be seen as the consequence of cultural influence on people's behaviours and perceptions. He also suggests that a high-level of trust can help overcome the social and individual barriers to KT.
- Goh [22] presents a framework identifying the key factors that need to be considered to develop effective KT in organisations. In contrast to "hard" factors such as information technology and structured organisational processes, "soft" factors influencing KT are emphasised in this framework. These key soft factors include leadership, problem-solving/seeking behaviours, support structure, absorptive and retentive capacity, and types of knowledge. He suggests that a balance of "soft" and "hard" factors is required to facilitate the process of KT. The above key soft factors have been integrated in a framework (as shown in Figure 2.1) to highlight the relationships between them. He also emphasises a high level of trust between employees and a supportive culture are required to influence these key factors and ensure effective KT as shown in the framework.

Barriers	**Possible solutions**
Lack of trust	Build relationships and trust through face-to-face meetings
Different cultures, vocabularies, frames of reference	Create common ground through education, discussion, publication, teaming, job rotation
Lack of time and meeting places; narrow idea of productive work	Establish times and places for KT: fair, talk rooms, conference reports
Status and rewards go to knowledge owners	Evaluate performance and provide incentives based on sharing
Lack of absorptive capacity in recipients	Educate employees for flexibility; provide time for learning; hire for openness to ideas
Belief that knowledge is the prerogative of particular groups, not-invented-here syndrome	Encourage non-hierarchical approach to knowledge; quality of ideas more important than status of source
Intolerance for mistakes or need for help	Accept and reward creative errors and collaboration; no loss of status from not knowing everything

Table 2.2: Davenport and Prusak's list of transfer barriers and possible solutions (Adapted from Davenport and Prusak [15, page 97]

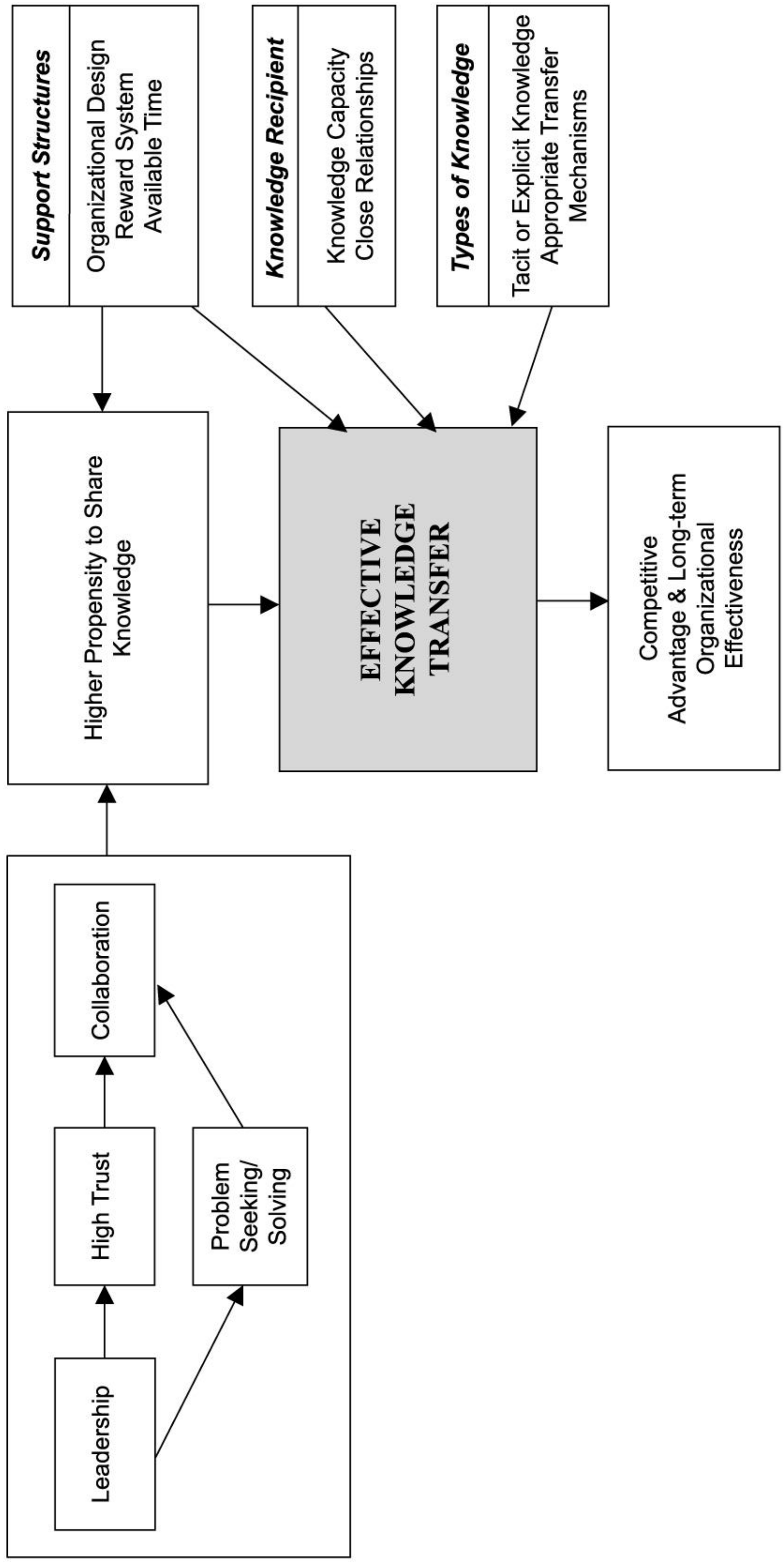


Figure 2.1: An integrative framework: factors influencing effective KT (adapted from Goh [22, page 28])

As we can see from the above examples, the emphasis on trust and culture is a common theme in many researchers' studies, although they may have investigated major transfer issues from slightly different perspectives. Culture is widely recognised as a critical factor affecting an organisation's KT performance (e.g. [15], [23], [12], [29]). Meanwhile it is a mainstream thought that trust can promote effective KT (e.g. [1], [25], [7], [15]). A particular transfer problem is often caused by a combination of several different factors. The available literature including the above examples would direct us to investigate the problem from the perspective of culture and trust framework, which would only allow us to understand the problem at an organisational level. However at an individual level, people make decisions and take a sequence of actions while participating in KT. Their actions are obviously affected by those key transfer issues at a lower level (e.g. at specific decision point or while taking a particular action). The majority of available studies do not break down KT and discuss these transfer issues at such a low level. A low-level understanding of KT needs to be related to these issues for us to further understand how people are influenced during KT.

2.3.2 KT strategies

Exploring effective transfer strategies is another major focus of the available KT literature. Many transfer strategies have been proposed to promote and facilitate KT in the literature, such as setting up communities of practice(CoPs). CoPs are self-organising groups of people who share common work practices, interests, or aims [9]. There is no standard means or formats of communication for people to transfer knowledge within CoPs. Although CoPs have existed in human societies for a long time, the term has just recently been introduced (cf. [64]). CoPs within organisations are frequently put forward as a way to promote effective KT (cf. [15], [64], [18]). Although CoPs are important for organisational KT, cultivating and maintaining them is still a challenge for many organisations (cf. [62], [63]). Currently no one has related the concept of CoPs to a low-level understanding of KT and clearly explained how or which part of the transfer process can be facilitated by this strategy. This prevents a complete view of what kinds of difficulties may occur while operating a CoPs. Without relating to the detailed process of KT, reasons why particular difficulties occur during the operations of CoPs cannot be well investigated. Although a detailed model of KT may not help us understand how CoPs can be created successfully within organisations, it can be used to explain at a low-level why some organisations succeed in maintaining them (cf. [15], [64]). Consequently, their lessons can be learned easily by other organisations.

Similar problem appears in current discussion of other transfer strategies too. Related discussion does not relate to specific transfer processes, so that we cannot sufficiently understand how these strategies facilitate KT. The available literature also does not allow us to review the roles of various transfer strategies in a common framework. This makes it difficult to understand each strategy's strengths and limitations in facilitating KT. In addition, an unclear explanation of how some organisations succeed in implementing particular transfer strategies also prevents the spread of effective use of these strategies in other circumstances. These limitations of current discussion of major transfer strategies suggest the need for a low-level model of KT, which can be used to simulate how a proposed strategy could potentially improve the transfer practice.

2.4 Need for a low-level model of KT

Earlier review on key transfer issues and current transfer strategies shows that KT is often studied at an organisational level in current literature. It also suggests that a low-level understanding of KT would allow these subjects to be discussed in a more specific context. It would allow us to understand how specific issues affect KT and how each strategy can be applied to facilitate KT at an individual level. By briefly reviewing the available KT models in current literature, this section aims to highlight that these models do not allow us to understand KT at a low-level. Therefore a low-level model of KT needs to be developed in this research.

Four different KT models from the current literature are reviewed here, including Szulanski's ([55], [56]) KT process model, Davenport and Prusak's [15] knowledge market model, Raymond's [46] gift economy model, and McGrath and Argote's [34] knowledge reservoirs framework.

2.4.1 Szulanski's KT process model

Szulanski ([55], [56]) presents a model of KT for analysing the difficulty of transferring knowledge within organisations, referred to as "the internal stickiness of KT". The notion of internal stickiness adapts the adjective sticky' used in management strategy and marketing literature, which means difficult to imitate or difficult to sell (cf. [41], [20]).

In Szulanski's study ([55], [56]), KT is recognised as "a process in which an organisation recreates and maintains a complex, causally ambiguous set of routines in a new setting" ([56, page 10]). This process can be viewed as consisting of four major stages: initiation, implementation, ramp-up, and integration.

- The initiation stage includes all events that lead to the decision to transfer.

For instance, a need for knowledge is recognized and the potential solution for satisfying that need is identified, then the feasibility of transfer related knowledge is explored.

- Once a decision to transfer knowledge is taken, the implementation stage begins. In this stage, communications between the knowledge source and the recipient are activated, social ties between them are established, the related knowledge flow between them. Transfer communications are customized to suit the recipient's needs and to avoid problems encountered in previous transfer processes.
- Implementation activities cease once the recipient decides to accept and start to use the received knowledge in the next stage – ramp-up. This stage also includes the recipient's attempts at identifying and resolving unexpected problems that arise while using the newly received knowledge to meet the post-transfer performance expectations.
- In the final stage, transferred knowledge becomes routinely used and institutionalized within organisations.

Szulanski ([55], [56]) also identifies four sets of factors that can cause difficulties (stickiness) in transfer. They are:

- Characteristics of the knowledge transferred, including the causal ambiguity and "unproveness" which refers to the knowledge without a proven record of past usefulness
- Characteristics of the source of knowledge, including the lack of motivation and a perceived unreliability
- Characteristics of the recipient of knowledge, including the lack of motivation, the lack of absorptive capacity, and the lack of retentive capacity
- Characteristics of the context, including a barren organisational context and an arduous relationship which means the lack of intimacy between the source and the recipient

He claims that the internal stickiness can be measured by evaluating the impacts of the above factors at each stage of the transfer process. Based on data collected from a two-step questionnaire survey, he conducts statistical analysis and gives a weight to represent the influence of each factor at each transfer stage. These weights show the stickiness caused by the corresponding factors. The larger weights represent the more significant transfer barriers.

Szulanski ([55], [56]) provides a constructive way to study the issues in the process of KT. However, this process model has two main limitations. First, the explanatory power of this model claimed by Szulanski is limited. Based on the internal stickiness analysis, he suggests that the three most important barriers to KT are lack of absorptive capacity of the recipient, causal ambiguity of the transferred knowledge and an arduous relationship between the source and the recipient. However, he does not relate his stickiness measurement results to any empirical examples and explain why these particular barriers are significant. Consequently, it is difficult to understand particular influence of these barriers. Second, this model is also lack of detailed descriptions of how people behave in the transfer process. For instance, there is insufficient detail to describe the implementation stage where people interact with others to transfer knowledge. Szulanski also ignores the complexity of communications between knowledge source and recipient while identifying the major transfer barriers. He quotes Shannon and Weaver's [50] work and classifies the transfer barriers based on their signalling metaphor specifying five basic elements of a transfer: source, channel, message, recipient and context. However, he does not sufficiently justify why Shannon and Weaver's work is appropriate here to be used in identifying the transfer barriers in his model. Furthermore, while defining transfer barriers based on the signalling metaphor, he does not discuss the characteristics of the channel of a transfer representing people's activities and interactions. No explanation is provided to specify why attention is only paid to barriers in source, message (knowledge), recipient and context.

2.4.2 Davenport and Prusak's knowledge market model

The concept of a market (cf. [19]) is adopted in Davenport and Prusak's model of KT [15]. They suggest that KT has similarities with transactions in an ordinary market.

They claim that like ordinary markets, the knowledge market has buyers and sellers who negotiate to trade goods and services (knowledge). It also has brokers who guide the buyers to the sellers, like brokers in the normal markets. An individual can play all three roles simultaneously in different transactions.

- Knowledge buyers are the seekers who try to solve a problem by acquiring new knowledge or information from others.
- Knowledge sellers are actually the knowledge providers or sharers, as they share what they know with the ones who need it.
- Knowledge brokers can also be seen as gatekeepers or boundary spanners. They make connections between those who seek knowledge and those who can provide it. Librarians in organisations are good examples of the knowledge brokers, as their responsibility is to provide information guide (i.e. a

library catalogue) to help the buyers and the sellers finding each other.

In ordinary markets, transactions occur only when all the trading parties get or at least believe they will get benefits from the transaction. Davenport and Prusak [15] highlight that people in the knowledge market also expect they will benefit from exchanging the knowledge (like goods and services in the normal markets). This expectation is the driving force of these knowledge transactions. The benefits for a buyer to participate in a knowledge transaction are that it will help them in solving a problem or improving their judgement and skills. Benefits for sellers and brokers are various and not as obvious as for the buyers, such as gaining reputation as a knowledgeable person. These benefits motivate people in the organisation to transfer knowledge.

Davenport and Prusak [15] also argue that benefits gained from knowledge transactions can be measured by a price system as in the ordinary market. They claim this price system involves three key factors – reciprocity, repute, and altruism, to represent the possible payment of exchanging knowledge in the knowledge market. Like the amount of money affecting trading parties' behaviours in ordinary market, the above three factors in the knowledge market can also affect behaviours of buyers, sellers and brokers. Reciprocity affects a knowledge seller's expectation on the buyer's willingness to reciprocate when they exchange roles in the future transactions. People are willing to spend scarce resources (i.e. time, energy and knowledge) only when they believe they will be offered a favour back in the future. Repute is critical when a knowledge seller tries to give away his knowledge in order to gain reputation as a knowledgeable person or an active seller. Their increased reputation can help them getting more offers in the future when they become buyers. Repute is therefore related to the reciprocity. The third factor, altruism triggers a seller's motivation to share knowledge for just helping others or for the benefit of the organisations they belong to. These factors also affect the buyer and the brokers in a similar way.

While reviewing this knowledge market model, Davidson and Voss([16, page 104]) emphasise "the idea that I will share my knowledge with you because you – or someone you know – will have access to the knowledge I may need in the future". They believe that this is based on the theory of social exchange that "the actions of individuals are motivated by the return that these actions are expected to bring". Although there are different views on market' due to a long history of research (cf. [19], exchange theory has been widely viewed as one basic principle in the concept of market. Without going too deep in exploring what is a market, we adopt McMillan's [35] definition to represent one of the classic views of an ordinary market. According to him, a market is a forum for carrying out exchanges that are voluntary. These exchanges are market transactions. Each party can veto the transaction and also freely agrees to the terms subject to the rules of the marketplace.

However, if we have a more close view on the features of a market exchange in the available literature, it shows that KT has some particular features compared to transactions in the ordinary markets. First in the ordinary markets, once a product is purchased, it is not available for the same seller to sell it in another transaction any more because the ownership is shifted. By contrast people's knowledge can be transferred in infinite times to anyone while they still own such knowledge. According to some economy researchers (cf. [36]), people's knowledge can be seen as being non rivalrous. Cheung [11] also argues that the very nature of knowledge is that an idea or creative work can be made accessible to anyone at little or no additional cost. Second, the legal contexts between KT and ordinary market transactions are different. In the ordinary markets, the rules of the marketplaces are compulsory for all trading parties once they are voluntary to be involved in the transactions. Meanwhile, their rights are under the protection of specific transaction policies or lows. However, people's behaviours are not strictly regulated by any legal requirements while transferring knowledge. Finally, unlike products in ordinary markets are protected using intellectual property rights such as patents and copyrights, knowledge (i.e. an idea, a different understanding of something) can be shared with anyone without the deprivation to its original creator (cf. [11]). With the above differences in mind, we cannot understand KT in terms of a market transaction.

In addition, Raymond [42] emphasises the difference between exchange and gift cultures and explains KT practices in terms of a gift culture. His work is reviewed in detail in the next section (2.4.3). Raymond's gift economy model also suggests that KT cannot be viewed as transactions within ordinary markets.

2.4.3 Raymond's gift economy model

Based on Mauss' [33] research on the concept of gifts in anthropology and economics (cf. [5]), Raymond [42] explains the KT practice within open source software communities in terms of a gift economy model.

KT within open source software communities can be seen as a gift economy because the relationships within these communities are transformed to interdependencies based on the idea of reputation. People's reputation is gained through giving away what they see as gifts, such as pieces of source code or drawbacks of current software discovered by them.

Raymond [42] insists that in order to understand the role of reputation in open-source communities, it is necessary to examine the difference between an exchange culture and a gift culture. In an exchange economy, scarce goods are allocated in a decentralized way through trade and voluntary cooperation, and people's social status is primarily determined by having control of thing to use or trade. Raymond believes that gift cultures are adaptations not to scarcity but to abundance. In a gift economy, people normally do not have a significant material-scarcity problem with survival goods, so that social status is determined not by what they control but by what they give away. This means that the society of open-source developers can be described in terms of a gift culture. Raymond explains that software is freely shared within these communities and that there is no serious shortage of the survival necessities', such as disk space, network connections and bandwidth. Due to such abundance, community members' competitive success can only be measured based on their reputation among peers.

Since Raymond [42] only briefly refers to Mauss's work, Bergquist and Ljungberg [5] summarise Mauss and his followers' work on gift economy theories and interpret gift giving as a way to build and maintain relationships of power between groups and individuals. According to Mauss' study, giving away a gift brings forth a demand for a return of a gift. The returned gift can be either an object or an abstract reward, such as an enhanced reputation or recognition of the earlier contribution. Therefore, gift giving creates social interdependencies and a social structure is organised.

To advance Raymond's presentation of a gift economy model [42], Bergquist and Ljungberg [5] provide more details of KT operations within open source communities and attempt to explain these operations in the context of a gift economy. According to their understanding, OSS development "relies on gift giving as a way of getting new ideas and prototypes out into circulation" [page 305]. They claim that gift giving within open source communities is important because it creates openness and organises social relationships between people in a certain way.

Bergquist and Ljungberg [5] conduct a detailed analysis of power relations involved in gift giving in the open source communities. It shows that gift giving in open source communities takes place in a digital world, and cannot be understood completely using Mauss's classic theories of primitive' culture that is based on the giving of material objects. The character of digital information makes open source gifts unique. For example, contributors in open source communities can give away an infinite number of copies of the same gift (e.g. a document or software) without losing it or diminishing its value. The efforts or cost of giving a gift to one person is almost the same as giving this gift to many people. These phenomena do not exist in gift economies based on material objects.

Bergquist and Ljungberg [5] also argue that the classic theories of gift economies cannot be used to clarify how gifts are treated as parts of quality assessment processes within open source software communities. In open source communities, people give away gifts because they believe that others' return of gifts (e.g. feedbacks on the gift) can help them improve the quality of the gift given away. With this in mind, they claim that gift giving in open source communities can be related to another kind of gift economy – the peer review system in academic societies. Academics share knowledge in their fields because this is a way of career progression. They give away knowledge as gifts in return of status and reputation. The acceptance of a gift implies recognition of the research status and feedbacks from the gift recipients can help them improve their academic work.

2.4.4 McGrath and Argote's knowledge reservoirs framework

According to McGrath and Argote [34] Knowledge resides in three basic elements of an organisation – members, tools, and tasks, and the subnetworks formed by these elements, including member-member, tool-tool, task-task, member-tool, member-task, task-tool, and member-tool-task subnetworks. They used the term reservoir' to indicate these knowledge repositories.

Members are the human component of organisations while tools are the technological component. Tasks reflect the organisations' goals, intentions, and purposes. The member-member network is the organisation's social network. The task-task network is the sequence of task or routines the members follow. The tool-tool network is the combination of technologies used by the organisation. The member-task network shows who is responsible for which task. The member-tool network assigns members to tools. The task-tool network specifies which tools are used to perform which tasks. The member-tool-task network specifies which members perform which tasks using which tools.

Knowledge embedded in these different repositories determines the group's ability to transfer the knowledge. They suggest an effective way to transfer knowledge is to change the knowledge reservoirs or to move the subnetworks. However in practice, compatibility of members, tools, tasks moved from one unit to another can be problematic. Moving networks is also difficult to do because they embody interactions that may not fit the new context. Therefore, compatibility with a new context is a necessary precondition for successful KT in organisations according to McGrath and Argote.

McGrath and Argote's model [34] is high level and no detail is available to explain how a transfer takes place. They suggest that knowledge can be transferred when knowledge reservoirs are moved or changed. However, they did not provide any details of how to move or change these knowledge reservoirs. In addition, they only focus on KT at the organisational level. Transfer of knowledge between individuals is not been explained at all in the model.

2.4.5 Discussion

None of the above four models explore KT at a detailed level. Although Szulanski's model ([55], [56]) recognises different stages within the transfer process, it is not specific enough to capture people's major activities and their interactions with others. His statistical evaluation of the stickiness can only show which transfer factors are more significant, but not identify particular points during the transfer process where these factors matter. Davenport and Prusak's knowledge market model [15] only identifies the key players involved and their role in the KT, but not provides any details of how these key players behave to take their roles during the transfer processes. This market model was developed in relation to their understanding of common transfer barriers. Since they explored KT at an organisational level in this model, their discussion of key transfer issues remains at a high-level (as introduced in Section 2.3.1). Raymond's gift economy model [42] and other researchers' extensive work relating to this model can be used to explain why people are willing to be involved in KT practice, especially within virtual societies. The gift driven nature of KT is emphasised in this model. However, it is a high-level model providing no detailed description of the process of KT. Similarly, McGrath and Argote's model [34] also does not give us any low-level details of KT.

In short, most available models of KT in current literature do not allow us to understand what exactly happened during KT, particularly people's decisions and actions. Without these low-level details about KT, our understanding of the key transfer issues and major KT strategies remain unclear. There is a need of developing a low-level KT model which breaks down the transfer process and points out particular points or sub-activities in this process (e.g. when and why 'trust' makes people behave in certain ways during the transfer processes).

2.5 Requirements of the KT model

The review of KT model in current literatures suggests that we need to develop a low-level KT model. People's activities during KT are normally complicated and it is impossible to capture all aspects of KT in our model. Therefore, the main focus and the required features of the model need to be identified first. These requirements can also be used as the validation criteria to check if our model does what it should do.

2.5.1 The procedural aspect of KT as the model focus

The complexity of KT at the individual level determines that it is impossible to capture all aspect of KT while studying it at a low-level. This could be one major reason that current studies on KT remain at the organisational level. There are several aspects of KT that we could explore, such as:

- Outcomes or results obtained from KT, such as a changed practice or organisational culture, or a person's changed understanding;
- People's perceptions toward KT , such as their understanding of the difference between tacit and explicit dimensions of knowledge, or their intentions (or expectations) when participating in KT;
- Procedural details describing how KT takes place, such as the time factor, the sequence of actions that people follow to share what they know with others, or available options they have when searching for required knowledge;

Which one from the above optional aspects should be the focus of our model? In order to develop a low-level KT model, the focus needs to be on what happens during KT rather than the effects of this process. This means that outcomes of KT do not need to be represented in the model while the procedural details of KT are definitely a necessary aspect. A person's perceptions can affect his activities during KT. Including the aspect of people's perceptions could potentially complement the procedural details represented in the model and provide a richer picture of what exactly happens during the process of KT. However, it is not realistic to represent this aspect in our model. People's perceptions are various and very subjective, so that it is impossible to describe these details in a simple and standard way. People's intentions when participating in KT can be seen as an example here. While capturing their activities, the reasons (including social and cognitive issues) why they behave in certain ways are various. Attempting to represent the above examples will only make the model over complicated. In addition, with the absence of people's perceptions, the model can still sufficiently represent KT processes at a low-level.

Unlike other existing KT understandings or models, our low-level model avoids defining what kind of knowledge is transferred in the process, but focuses on the procedural aspects of the transfer. Specifying what is transferred in this process could potentially complement the procedural details and provide a rich picture of what happens during the transfer processes. However, it is not realistic to represent this aspect in the model. Personal knowledge has been classified on two dimensions in general – explicit and tacit [40]. However because of the lack of a widely agreed distinction, people's understandings of the difference between tacit and explicit knowledge vary. Therefore distinguishing people's activities of transferring those two types of knowledge becomes difficult. In addition, it is not necessary to specify what kind of knowledge is transferred in our model because of the motivation of our research (specified in Chapter 2). With the absence of this aspect, the model can still sufficiently represent KT processes at a low-level.

Focusing on the procedural aspect of KT alone does not mean that we ignore the rest of the KT aspects (such as the content of KT). We intend to develop a low-level KT model which could be used as the foundational work to understand other KT aspects. Our model can be seen as a framework explicitly showing the procedural KT details with other transfer aspects hidden at the back end. It will be precise but comprehensive enough to represent the overall view of KT at an individual level. In short, our model will focus on the procedural aspect of KT.

2.5.2 Critical features of the model

Although the procedural aspect of KT is chosen to be the focus of this model, not all procedural details are necessary. Only specific procedural details that are critical to the model in fulfilling its research purposes will be represented. For example, the time-line during KT could be represented in the model. Including time could potentially enhance the developed model and provide more details of how a transfer process takes place. However without the representation of time, the model can still be used in the two ways required by the research motivation. Therefore, time will not be considered as a critical feature in developing this model. For the same reason, there are other non-critical procedural details, including modes of communications used by people to interact with others and locations of participants of the transfer processes.

The following behavioural patterns of the procedural aspect of KT will be captured as critical features in this model:

- People's decisions during KT

  Including people's decisions corresponds to the motivation of developing this model – to allow key issues to be related to particular points in the transfer process. To reflect these decisions in the model, three different kinds of details need to be captured. First is the major decision points when people make their choice. The second kind is people's available options at particular points during KT. And the final one is consequent actions or outcomes of people's choices at each decision point. Taking different options at particular decision points leads people to different KT results. Their earlier decisions have already determined their next actions or even next decision points that they have to go through. Capturing these consequent actions or outcomes can show what happened following people's decisions during KT and also link decision points in a logical sequence.

- People's interactions with others during KT processes

  In the process of KT, people can interact with others either directly or indirectly (through some sorts of knowledge resources). Since a KT process normally involves several people, their interactions must be captured in our model.

2.6 Summary

This chapter suggests that the available literature of KT often discuss key transfer issues and major transfer strategies at an organisational level. This does not allow us to analyse any transfer problem at an individual level. This issue may be addressed if KT is studied at a low-level where people's activities during KT are discussed. However, a review of the major transfer models available in the literature shows that the required low-level transfer model does not exist currently. Therefore this study aims to develop such a model. The procedural aspect of KT is identified as the main focus and the model will emphasise on people's decisions and their interactions with others during KT. A definition of KT at the individual level is also given at the beginning of this chapter as the conceptual foundation of this study.

Chapter 3

Methodology

3.1 Introduction

As explained in Chapter 2, a model for KT is the major focus of this research. This chapter is concerned with outlining and justifying methodological approaches to develop and validate such a model. It is organised as follows. An overview of the methodologial design of this research is provided in the next section. The design of the overall sequence of steps followed in this research points out that there are two major procedures to achieve the research purposes. One is to develop a low-level KT model, while the other one is to formalise this low-level model. Then in the third section, a first-principles approach is proposed to develop the low-level KT model. A detailed methodological design of this approach is also explained in this section. The formal method adopted for developing the formal KT model is then presented in the fourth section. Within this section, formal methods and related concepts are introduced and the choice of required tools and techniques for developing the formal KT model such as *CSP* and model-checking are also justified.

3.2 Overview of the methodological design

Figure 3.1 below provides an overview of the methodogical design of this research. It shows a sequence of the steps followed during the research, while all techniques applied and tools used at each step are also explained.

3.2.1 Overall sequence of the research

First, we justify the need for a low-level KT model based on the investigation of the previous studies in the domain of KM. Such a justification allows us to clarify the focus of this research and also specify the requirements of the KT model we intend to develop.

After addressing the research ethical issues and obtaining required ethical approval for this research, we move onto developing the low-level KT model. This is one major research procedure designed to complete this research. The chosen methodology for this procedure – a first-principles approach is justified in the following section (Section 3.2.2), while the detailed design of this procedure will be explained in Section 3.3.

Next, we need to identify the shortcomings of our low-level KT model and justify the need to formalise such a low-level model. The methodology we choose for such a formalisation – formal method, is first introduced in Section 3.2.3. Related notation and concepts of this methodology are then explained in Section 3.4. Developing the formal KT model is another major procedure in this

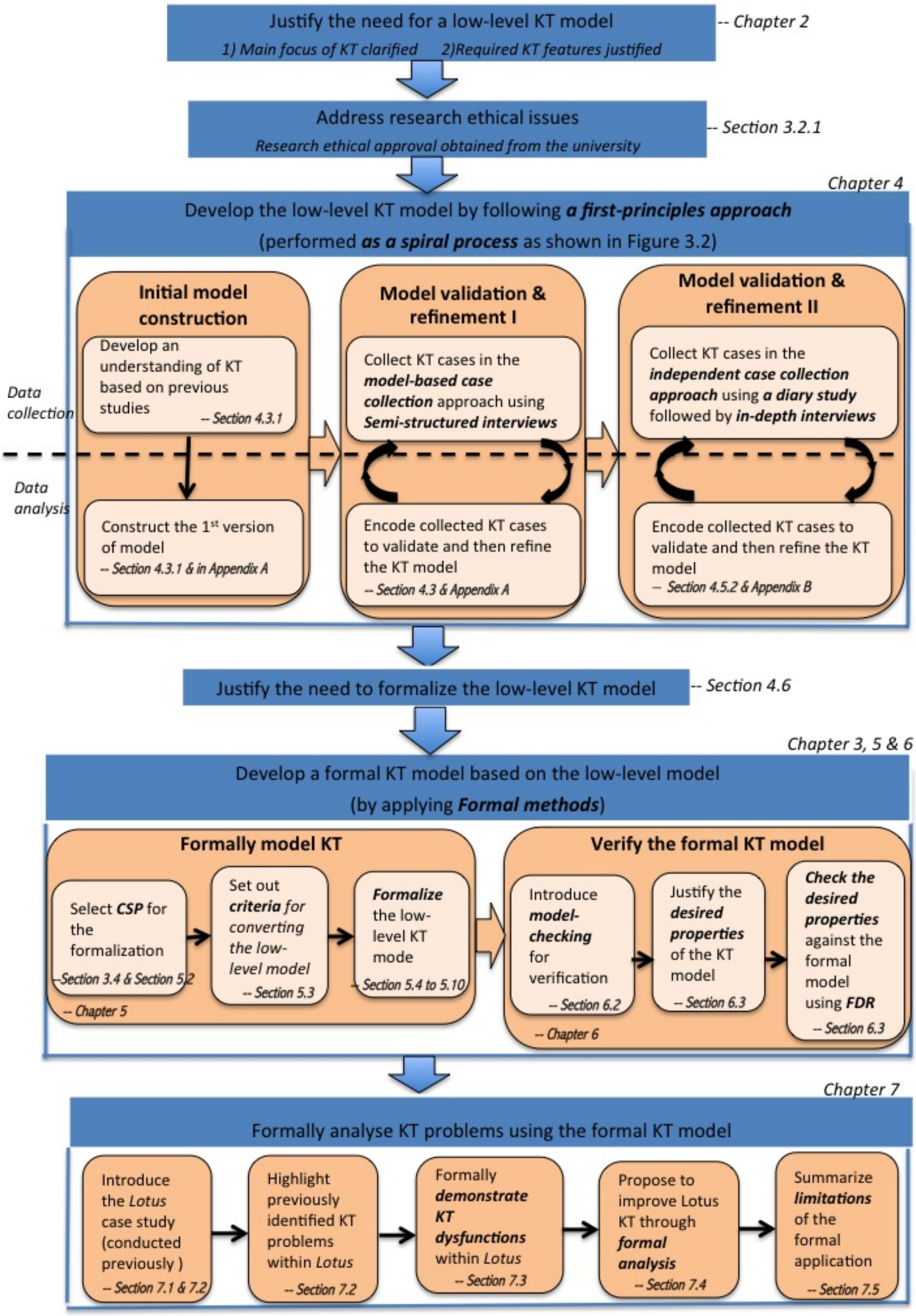


Figure 3.1: Overview of the research methodology

research. Tools and techniques required in this procedure are also justified in Section 3.4.

Finally once the formal KT model is developed, we will use it to formally analyse KT problems in a small case study. By conducting such a basic formal analysis, we will be able to demonstrate very simple but significant value of this research.

The above overall sequence of this research (as shown in Figure 3.1) is also reflected in the organisation of this thesis. For example, the previous chapter provided the justification of the need for a low-level KT model. The development of this model is then presented in the next chapter. Chapter 5 and 6 covers the justification of the need for formalising the low-level model and also the development of the formal model. The last chapter before the conclusion in this thesis then focuses on the work of the formal analysis of KT problems.

3.2.2 Developing a low-level KT model

Based on the justification of the need for a low-level KT model given in the previous chapter, a low-level KT model can be developed based on some first principles. Before explaining the detailed desgined of this approach, we briefly clarify why we do not choose a grounded theory study as the main methodology in this research.

A grounded theory study seeks to generate a theory from an area "where there is lack of theory and concepts to describe and explain what is going on" [44, page 90]. It involves going out into the field which relates to the particular situation forming the study focus and collecting the data. The narrative form of a grounded theory is normally a theory or a theoretical model, and is applicable to a wide variety of phenomena. The focus of this research is what is going on in the process of KT and the anticipated outcome is a low-level KT model, so that a grounded theory study seems appropriate. One key feature of grounded theory study is that researchers should seek to enter the field without any theoretical preconceptions [44]. In traditional social research, researchers studying any social processes normally use an inductive, naturalistic approach that starts the investigation without any predetermined assumptions about the process ([39], [44], [54]). However, it is almost impossible to follow this research tradition in the context of our research for the following two reasons. On one hand, the process of KT commonly exists in people's everyday life. Almost everyone including the researcher have a certain level of experience of KT. On the other hand, we study KT from a novel perspective (transaction-oriented) where current literature cannot provide much conceptual guidance. Without a pre-analysis of the KT process, it is difficult to predict how much workload this research requires and what kind of details are needed from the field investigation. In other words, a grounded theory study is not appropriate in this research.

3.2.3 Developing the formal KT model

KM is a multidisciplinary subject area. People have various research background (i.e. Computer science, Business studies, Sociology or Psychology) and often focus on different perspectives of KT in this domain. Therefore it is critical to combine different methodologies and approaches that derive from various disciplines. The application of formal methods could be particularly useful in formalisation of definitions, concepts and models in this domain. Such formalisation can set the foundational grounds of the discipline and provide a rigorous common language with which researchers and industrial practitioners can communicate and interact.

In the context of this research, KT can be viewed as a concurrent system in which people interact with each other to transfer knowledge. Formal methods are appropriate in studying such concurrent systems. More justifications of the choice of formal methods in our research are given in Section 3.4.

3.3 A first-principles approach

The first principles used to develop the low-level model are identified in this section. Based on the design of the low-level KT model, a process to develop this model is outlined. Necessary tasks and required methods are also justified here.

3.3.1 A spiral process to develop the low-level model

Three major tasks need to be completed to develop the low-level KT model:

- *Initial model construction*
- *Model validation* – to validate the model against various KT cases to check if it has fulfilled the predefined model requirements
- *Model refinement* – to refine the model to deal with the unrepresentable KT cases identified in the ongoing model validation

It is not realistic to develop the required model in one go – constructing the model in one go and then validating it. In this research, the above major tasks can be completed in the following sequence:

- First, the task of initial model construction needs to be completed.
- Then the tasks of model validation and model refinement will be integrated together to form a spiral process to produce a final model:
    1. In this spiral process, the initial model needs to be validated first to explore any KT cases that it fails to represent, so that necessary revisions can be made to refine the model.
    2. Then the refined model needs to be assessed again. If this validation shows the refined model failed to represent any KT processes, then further revisions need to be made.
    3. Repeated validation-refinement-validation' steps will gradually refine the KT model until it reaches a stable state when no further revision is required according to previous validations.

When new KT cases that the current model cannot represent are identified in a model validation, the model will be refined to deal with these cases. Then both new and previously identified KT cases will be used in the next model validation. The overall sequence to complete the above major tasks can be presented in Figure 3.2 below.

There are two major issues in conducting this spiral process – one is how to collect KT cases and the other issue is how to use the collected KT cases to develop and validate the model.

3.3.2 KT cases collection

A triangulation strategy will be used through this spiral process. Triangulation "is a method of finding out where something is by getting a 'fix' on it from two or more places" [44, page 371]. This strategy will be applied to determine the resources and the approaches for case collection. Using more than one resources or approaches can avoid bias of collected cases and address the limitations of any single resource or approach.

3.3.2.1 Two resources for collecting KT cases

Two major resources will be used in this study to collect adequate KT cases for model validation:

- *The field* – Similar to any other research, the required KT cases for this research will be collected through field studies.
- *The researcher's own understanding of KT* – The researcher's own understanding of KT will be used to construct the initial model of KT. However,

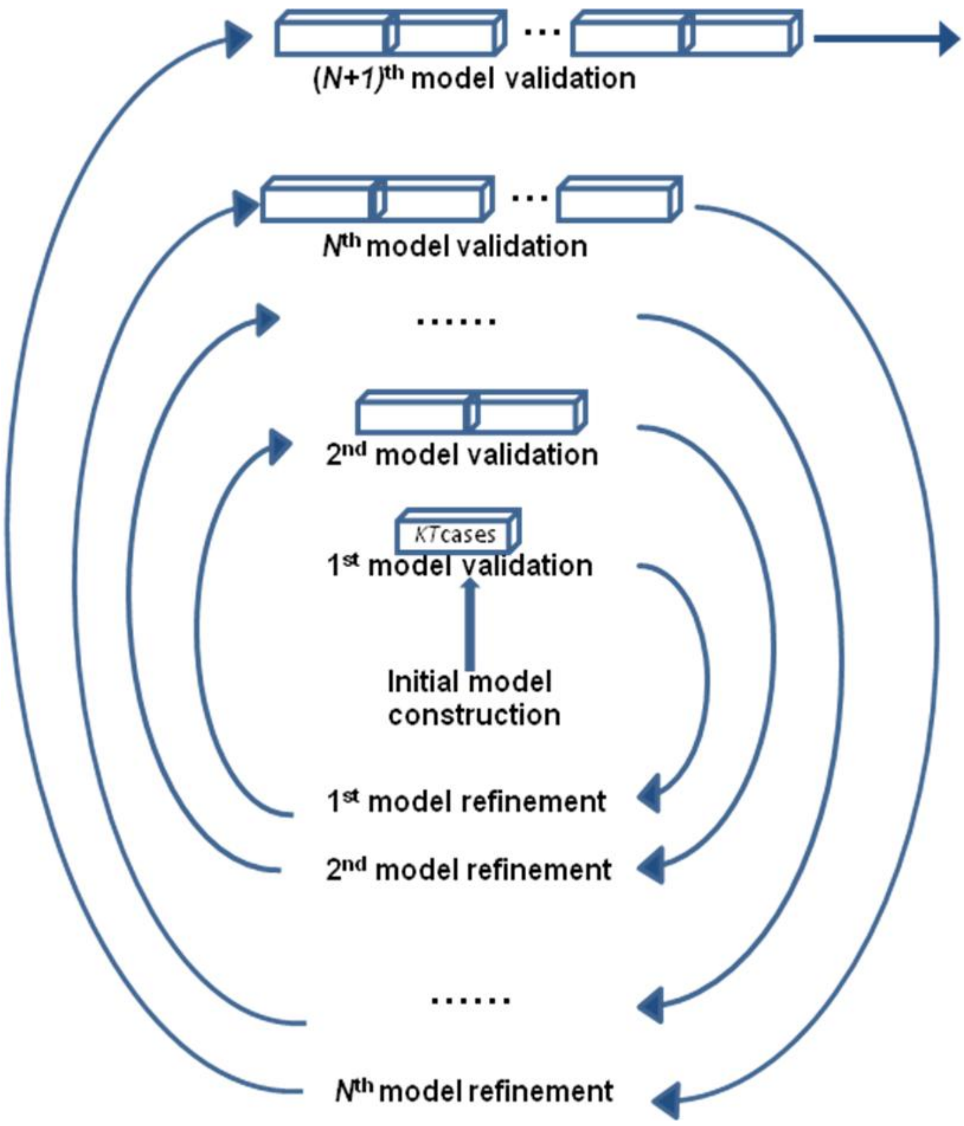


Figure 3.2: The overall sequence of completing the major tasks

such an understanding will gradually change with the repeated validations and refinements of the developed model. This means that some new hypothetical KT cases can be generated based on the researcher's refined understanding of KT.

It is not necessary to use cases from all resources to validate the model through the entire validation process. We can name the KT cases that pass model validations as positive cases and the failed ones as negative cases. Positive cases gradually increase in each validation within the spiral process. It is reasonable to use cases from one resource first, and then increase the resources while the developed model becomes more and more robust.

3.3.2.2 Two approaches for collecting KT cases

Two broad approaches will be used to collect KT cases in the field. One is a model-based approach, and the other one is an independent approach. This is also an application of the triangulation strategy. They can complement each other in the KT case collection for model validation.

- The model-based case collection approach

  The model-based approach is to collect KT cases using the developed KT model as a starting point. Using this approach, the developed model will be presented to all participants before any KT cases are collected from them. The main advantage of using this approach is that the participants can spot any faults or weakness of the developed model in a very short time. They do not need to provide all KT cases they can think of but the ones that they believe the model cannot represent. However, the researcher's subjective view may affect their understanding and the KT cases they provide may have bias in this way. This is because the model is initially based on the researcher's personal understanding of KT and it has been introduced to the participants in advance.

- The independent case collection approach

  The independent approach is to collect cases independently from the developed model. This means that participants will not be informed with the developed model, so they can provide KT cases without being influenced by the researcher's personal views. Although following this approach may be time consuming, it allows KT cases without bias to be collected.

3.3.2.3 Applied data collection techniques

There are several different data collection techniques that may be applied in the context of the above two KT case collection approaches, including observations, interviews, questionnaires and diaries.

- Observations

  Observations are normally used to collect data about people's behaviours in the field. However, observing behaviours is not a suitable enquiry technique in this study. First, as an ethical requirement, permissions from all participants need to be gained before starting the observation. It is impossible to predict who will be involved in a transfer process, so that getting permission from all involved parties in advance becomes a problem. Second, any KT process involves at least two people and direct observation cannot focus on all participating parties simultaneously.

  In addition, using observations would encounter particular problems while following either of the above two case collection approaches. On one hand, informing every participant with the developed model is required in the model-based collection approach. However, it is impossible to do so using observations, since the involving parties of a transfer process cannot be pre-identified. On the other hand, observations depend on the researcher to record people's activities and they are not completely independent. The nature of the independent approach determines that only self-reporting activities followed by the participants can avoid the influence of the developed model. This means observations are not a suitable case collection technique for the independent approach.

- Interviews

  Interviews are flexible and adaptive, so that they can be used to explore KT cases that cannot be directly observed. Using interviews, permissions can be gained from either one party or retrospectively from all parties involved in a transfer process. Considering the requirements of the above two case collection approaches, interviews seem more appropriate in the context of model-based case collection. The developed model can be presented to interviewees during the interviews.

  In regards to the independent case collection, interviews are not a suitable technique. Since interviews involve direct interactions between the interviewer and the interviewees, it is very difficult to avoid the researcher's influence on participants' responses to the interview questions. This is against the basic requirement of this approach. In addition, the independent approach is time consuming, so that it is not realistic to interview each participant individually. However, interviews have the advantage of allowing researchers to validate or clarify particular details of the collected cases. Therefore, while choosing another technique as the major option in the independent case collection, interviews can still be used as a complementary technique.

There are different forms of interviews, including fully structured, semi-structured, and unstructured interviews. Since the purpose of interviews is clearly defined in this study, key questions in each interview will be predetermined. Considering everyone's experience and understanding of KT is different, the researcher needs to change the wording of the questions or explain the questions according to the interviewees' personal situation. This means semi-structured interviews are appropriate in this study.

- Questionnaires and diaries

  Questionnaires are a self-completing data collection technique. Without direct interaction with the researcher, respondents to questionnaires can have the maximum control over their participation. Questionnaires also allow participants to seek out the information before responding to the questions. Given the self-reporting nature of questionnaires, they are only a suitable technique to explore KT cases using the independent approach.

  Diaries are normally used to study people's behaviours by asking participants to report their daily activities. This data collection technique has not been given a consistent definition or format in current literature. In this study, the diary is seen as a form for carrying out the questionnaires in the independent case collection.

  In the diary study, participants will be asked to record their KT activities on a pre-printed log form. Instead of breaking the day into brief intervals, the log here will only ask participants to review and report their major KT activities at the end of each day. The activities will be initially recorded using the participants' own descriptions.

  A diary study can potentially reduce the case collection time. It can provide "the means to generate a very substantial amount of data with minimal amount of effort on the part of the enquirer" [44, page 258]. The participants can complete it over a period of time. As a complementary technique to the diary study, follow-up interviews will be necessary in the independent case collection. These in-depth interviews can be used to clarify any unclear or missing details of the KT cases collected in the diary study.

In summary, semi-structured interviews will be used in this research to conduct the model-based transfer case collection. In the independent KT case collection, a diary study followed by in-depth interviews will be deployed.

3.3.3 KT cases analysis

The developed KT model will be used as a coding scheme [44] to check whether or not a particular behaviour (each step in the model) has occurred in the collected KT cases. People's actions in each case will be encoded into a series of steps. If all actions within a collected case can be encoded using the model, then this is a positive case and the current model does not need to be refined.

If there are uncodable actions within a collected case, this case can be either positive or negative. All uncodable actions need to be analysed first to check if they are behaviours relating to the critical features of KT process, including people's decisions points, their decisions, consequent actions following their decisions, and their interactions with others or other knowledge resources. If no uncodable action relating to these required features is observed, the model does not need to be revised. Otherwise a refinement of the model is necessary.

3.3.4 Three stages within the spiral process

Because of the use of two different approaches while conducting the repeated validation-refinement-validation steps, the overall spiral process (as shown in Figure 3.2) for developing the low-level KT model can be viewed as three separate stages. These three stages are also illustrated in Figure 3.1 earlier.

- Stage 1: Initial model construction

  An initial KT model can be constructed based on the researcher's pre-existing understanding of KT (which may be established based on the relevant literature and her personal KT experience).

- Stage 2: Model validation and refinement I

  In the second stage, the model validation and refinement mainly rely on collecting and analysing KT cases collected using the model-based case collection approach. Through semi-structured interviews, all participants will be able to overlook the ongoing model development and provide KT cases that they believe the presented model cannot represent. Necessary refinements thus can be made to our model. A variety of symbols need to be defined to represent different behavioural patterns which were identified as the required features of the KT model in the previous chapter. These symbols will allow us to encode the collected KT cases in a relative standard manner. At the end of this stage, our low-level KT model should reach to a relatively stable state where no further revision can be suggested by the participants.

- Stage 3: Model validation and refinement II

Similar tasks of model validation and refinement will be conducted in this stage as in Stage 2. However a different approach – the independent case collection approach will be used to serve the purpose. Data collection will mainly rely on a diary study followed by in-depth interviews in this stage. Since our model should have already reached to a stable state at the previous stage, work to be done in this stage will mainly for assessing and validating our KT model.

3.4 Formal methods

The relevant literature on formal methods is briefly reviewed and key concepts (such as formal specification and verification) are introduced in this section. It is also justified that KT can be seen as a concurrent system and *CSP* is a suitable process algebra for modelling KT systems.

3.4.1 Related literature on applied formal methods

Formal methods are mathematically based languages, techniques, and tools for studying and developing complicated systems. They provide frameworks for people to specify, develop and verify systems in a systematic manner. Using formal methods can greatly increase people's understanding of complicated systems by revealing inconsistencies, ambiguities, and incompletenesses that are not easy to detect [14]. When used in early stages of the system development, formal methods can help reveal design flaws that otherwise may be discovered at later testing and debugging stages. This could improve the cost effectiveness. In later stages of the system development, they can be used to determine whether the system is implemented correctly and check if there are different but equivalent implementations available.

Several well-known surveys in the current literature identified challenges and potentials of applying formal methods in software engineering industry. For example, Hall [26] defended formal methods as an engineering approach and discussed 'seven myths' about their applications. Wing [65] provided a very good introductory to the underlining concepts and principles of formal methods. The cost effectiveness of the industrial applications of formal methods was discussed by Thomas [57] from a CEO's point of view. Austin and Parkin [3] conducted a questionnaire survey on the use of formal methods in both research and application, and aimed to explore the reasons for their low acceptance in industry. In addition, Clarke and Wing [14] also gave a brief introduction to the notions in formal methods. Based on a list of notable industrial applications, future directions for the formal methods community were also proposed in their study.

Although these surveys took different viewpoints and adopted different survey approaches, there is an agreement in their conclusions that scalable applications of formal methods still face significant challenges. The major difficulties of using formal methods in practice included that "the notations were too obscure, the techniques did not scale, and the tool support was inadequate or too hard to use" [14, page 626]. There were very few successful case studies and they were not convincing enough. In addition, few people had the sufficient training to use formal methods effectively on the job [14].

More recently, a thorough survey exploring the current state of art and trends of formal methods applications was carried out by Woodcock et al. [66]. They looked into relevant studies and applications over the past 20 years and provided a richer picture of how formal methods were applied in different parts of the system development life-cycle in various domains. They also revisited concerns raised in previous applications and identified the progress and trends in this subject area. Through their ongoing studies, they intended to produce some hard evidence to support Hoare's [28] positive vision on computer software engineering in a future world. Increasing studies begun to offer a more promising picture of formal methods applications and more successful case studies can be found in the available literature in recent years [14].

However we could not find any previous studies on applying formal methods in the domain of KM, so that the adoption of this methodology in this research is an experiment. More justification of the choice of particular formal method used in this research will be given in Section 3.4.4, Section 3.4.5 and Section 3.4.6.

3.4.2 Formal specification

A formal method is based on some well-defined formal specification languages. Wing [65] claimed that a method is formal if it has a mathematical basis which is normally given by a formal specification language. People often use formal methods to specify a system's desired behavioural and structural properties. Such a process is a formal specification. Its value is that it allows the external behaviours of a system to be described without specifying its internal implementations. Since this process is the act of writing things down precisely, it contributes a deeper understanding of the system being specified [14]. Compared with informal ones, formal specifications often have the following advantages [65]. First because of its mathematical basis, a formal specification is more precise and usually more concise than the informal one. Second, they are amenable to machine analysis and manipulation which are not applicable to informal specifications.

In the context of this research, a formal specification of KT would help us gain a deeper understanding of how KT takes place and identify people's major behaviours during this process.

3.4.3 Formal verification

A formal verification goes one step beyond the specification. This is a process to analyse and prove the system for desired properties. Two major techniques are normally used for formal verification, including model-checking and theorem proving.

Model-checking is a technique to check if a finite model of a system has a desired property. There are two general approach to model-checking [14]. The first one is temporal model-checking. In this approach specifications of a system are expressed in a temporal logic while the system is modelled as a finite state transition system. Then "an efficient search procedure is used to check if a given finite state transition system is a model for the specification" [14, page 630]. The second approach is to formulate the specification as an automaton. The system is also modelled as automaton in this approach. Then they are compared to determine that if the system's behaviour conforms to that of the specification.

While using the theorem proving technique, both the system and its desired properties are expressed as mathematical formulas. Their mathematical logic is "normally given by a formal system which defines a set of axioms and a set of inference rules" [14, page 633]. Theorem proving allows a proof of a property from the axioms of the system to be identified.

Theorem proving often requires interactions with a human user. This process is often slow and error-prone because of this reason. Unlike theorem proving, model-checking is completely automatic and fast. In addition, model checking can be used to check partial specifications. This means a system's correctness can be checked even if the system has not been completely specified [14].

Because of the above two advantages of model-checking, it is planned to use model-checking to verify our formal KT model in this research. A detailed plan of model verification will be presented in Section 6.2, such as the choice of model-checking using *CSP* failures model. The model-checking technique will also be further discussed in that section.

3.4.4 KT as a concurrent system

Schneider [47] emphasises that concurrent systems are complicated. Within a concurrent system, many components (also called processes) may execute in parallel. The complexity of a system arises from the combinations of ways in which its components interact. These interactions sometimes cause phenomena not present in a sequential system, such as deadlock and livelock. A concurrent system is deadlocked if no components can make any progress, simply because they are waiting for a communication (interaction) with other components before they can continue their own actions [45]. This means that some of their actions will never take place if their related components are not available. A deadlock is undesirable as it ultimately halts the execution of a system. Livelocks arises when components within a concurrent system descend into infinite loops, never to interact with their environments again [45]. This type of behaviour is also called divergence. Livelocks are undesirable as the system performs an infinite unbroken sequence of internal actions and they often result in an unnecessary consumption of resources. Both problems arise not due to the design of individual components within a system but the way they are combined [47]. The design of a concurrent system "requires ways of keeping these interactions under control"[page ix].

A KT system often has several parties playing different transfer roles at the same time. The interactions among them vary and can often cause transfer problems. Both deadlocks and livelocks are common in a problematic KT environment. For example, a person may request knowledge from others but never get any responses, or another person may repeatedly use an internal database to search for unavailable answers. The former example can be seen as a transfer deadlock while the latter one serves as a KT livelock example.

Another key feature of a concurrent system is nondeterminism, which a sequential system does not have. A system exhibits nondeterminism if two different copies of it may behave differently when given exactly the same inputs. A nondeterministic system is uncontrollable, unobservable from the outside and untestable in principle. Establishing any property of such a system can only be done through formal understanding and reasoning. Since we only focus on the observable aspects of KT (particularly people's transfer choices and decisions), the feature of nondeterminism is not applicable to the KT system we are concerned. Therefore we do not discuss this feature in detail.

In short, KT processes can be seen as concurrent systems while the concurrency theory provides a way of understanding and thereby representing the dynamics and complexity in the process of KT.

3.4.5 Process algebra

One of the purposes of studying a concurrent system is to verify statements about processes. Such statements allow us to correctly simulate a specified process. Since we need to determine if the real process and the simulated process are equal, it is necessary to have a criterion for identifying processes. This criterion determines the semantics of the concurrency theory. Process algebra theories equipped with different semantics have the capacity to express equality on different levels. Process algebras are particularly suitable for specifying and verifying the key features of concurrency, including communication between parallel processes, deadlock behaviour, abstraction from internal steps, fairness, nondeterminism, priorities in the choice of actions, tight regions, etc [45].

Baeten [4] defined the term of *process algebra* in his study. He referred the word 'process' as the behaviours of a system. Anything showing behaviours can be seen as a system, such as a software, a machine or even a human being. A system's behaviours are events or actions that this system performs with the order in which they are permitted to execute. We normally describe certain aspects of a system's behaviours, so that our description of this system is abstract or idealized. In Baeten's [4] definition, the word 'algebra' means that an algebraic approach is taken in describing a system's behaviours. In general, a process algebra is an algebraic approach to the study of concurrent processes. Its tools are algebraical languages (also recognised as concurrency theories or concurrency models) for the specification of parallel processes and the formulation of statements about them, together with calculi for the verification of these statements [21].

Baeten [4] also briefly summarised the history of process algebra. Research on this topic began with Robin Milner's seminal work on the Calculus of Communicating Systems (CCS) during the period from 1973 to 1980. C.A.R. Hoare's Communicating Sequential Processes (*CSP*) first appeared in 1978, and was subsequently developed into a fully-fledged process calculus during the early 1980s. There was much cross-fertilization of ideas between CCS and *CSP* as they developed. In 1982 Jan Bergstra and Jan Willem Klop began work on what came to be known as the Algebra of Communicating Processes (ACP), and introduced the term process algebra to describe their work. CCS, *CSP* and ACP are the three main process algebra languages in this field. The majority of the other process algebras theories can trace their roots to one of these three calculi. For example, LOTOS (Language of Temporal Ordering Specification) is a newer addition to the process algebra family. It was initially based on CCS, while some notation and concepts were later introduced from *CSP*.

Since process algebra languages are developed for studying a concurrent system (such as KT), it is necessary to select one of them to develop our formal KT model in this research.

3.4.6 The choice of *CSP* in this research

CSP is the chosen process algebra language in this research to develop our formal KT model. *CSP* is a notation for describing concurrent systems with component processes interacting with each other and their environment by means of communications. With a collection of mathematical models and reasoning methods, *CSP* is particularly good at understanding and analyzing systems whose primary interest arises from the ways in which their component processes interact at the level of communication [45]. we choose *CSP* mainly because of the following three reasons:

- First, key *CSP* features make it applicable in describing KT at the required level. More explanation is given in Section 3.4.6.1.
- Second, *CSP* is a rich mature language that has been studies for a few decades and its supporting tools for analysis are advanced and reliable (see Section 3.4.7).
- Third, the closest subject area of applying formal methods to our research in available literature is human-machine interactions. These relevant studies (e.g. [49], [10]) mostly applied *CSP* and they can be used as a useful foundation to progress the formal modelling in this research.

Several key features of *CSP* make it applicable in describing KT in the context of this research. These features include the concept of *CSP* processes and events, communication, recursion and abstraction. How these features apply to KT is explained as following.

3.4.6.1 *CSP* processes and events

A system's interacting components are presented as processes in *CSP*. These processes are considered as "independent self-contained entities with particular interfaces through which they interact with their environment" [47, page 3]. Two or more processes can be combined to form a larger system, which can also be treated as a self-contained entity with interfaces. In other words, this larger system can also be seen as a (larger) process. This is the conceptual framework taken in *CSP* for analysing the world.

The only things that we can observe about a process are the events that allow this process to communicate with other processes or its environment. Therefore the most fundamental object in *CSP* is a communication event. In order to construct a *CSP* process, we need to decide on an alphabet of communication events. It should include all events that the process and its related processes might perform. Roscoe [45] believed that the choice of this alphabet is the most important decision we need to make while modelling a real system in *CSP*. It determines both "the level of detail or abstraction in the final specification and also whether it is possible to get a reasonable result at all" [45, page 13].

In the context of this research, a KT system is formed by several involving parties who communicate for the purpose of KT. They are recognised as taking different transfer roles during KT. These roles can be represented as *CSP* processes. The set of events within each process is the list of actions people follow while taking individual transfer roles. Describing KT in terms of processes and events is consistent with the motivation of developing the KT model in this research (as explained in Chapter 1 and 2). The procedural aspect of KT can be well represented at a low-level in this way.

3.4.6.2 Communications

Roscoe [45] clarified that we need to think of a communication as a transaction or synchronization between two or more *CSP* processes rather than simply a transmission of data in one way. A component process often has to cooperate with others in the performance of an event. The way *CSP* models this event assume that it only happens when all its participants are prepared to execute it. This event is also abstracted and instantaneous in the *CSP* description. This means *CSP* only represents the event as happening at the moment when it becomes inevitable (when all involving parties have agreed to execute it), although the 'real' phenomenon of the modelled event might take some time. A process' interaction with its environment also takes the same form as that between two processes, in which events only happen when both involving sides agree.

There is a basic collection of *CSP* operators that allow us to model processes that simply describe patterns of communications. In particular, the operators of choice, composition and synchronisation are very useful in modelling KT in the context of this research. First, the choice operator can be used to represent people's available transfer options at different decision points during KT. The Composition operator allows different transfer activities to be combined so that people's transfer actions can be modelled in a logical sequence. The operator of synchronization is particularly useful in describing how different transfer parties interact with each other and highlighting the synchronized events between them. This could help identify and analyse specific KT problems within organisations which often arise when people interact with others.

All above operators relevant to our research will be introduced in detail in Section 5.2.

3.4.6.3 Recursion

CSP offers a precise way of defining recursive processes. People often have to repeat certain actions or even repeatedly take certain transfer roles within KT. The *CSP* recursion operator can help avoid repeating unnecessary events while describing KT, so that our formal KT model can be minimized. The less events within a *CSP* model, the more efficient the formal verification of this model will be in later stages.

3.4.6.4 Abstraction

Abstraction is another useful *CSP* feature for modelling KT in this research. It allows the system to be modelled in a way that only relevant aspects of process behaviours can be seen. Irrelevant details can be abstracted (hide) away from a process by applying *CSP* constructs [45]. The KT model required in this research only focuses on the procedural aspect. Since other KT aspects (Such as social and cognitive issues in KT) are not relevant, they can be hide away through abstraction.

This idea of abstraction is particularly useful in formulating a variety of specifications and check if a system satisfies them. The correctness of this system can be proved in this way [45]. We also plan to use the feature of abstraction to produce required specifications for verifying our formal KT model in this research. Examples of how an abstraction can be used will be given in Section 6.2.2 (where the concept of *CSP* failures refinement is introduced).

3.4.7 *CSP* tools

Compared with other process algebra theories, *CSP* has more advanced and reliable tools to support model specification and verification.

There are quite a few automated proof tools for *CSP*. The main proof and analytic one at present is called *FDR* (standing for Failures Divergences Refinement). The existence of *FDR* has led to a revolution in the way *CSP* is used [45]. Several other tools have similar external functionality but were developed based on very different algorithms. Among these automated proof tools, *FDR* is recognised as the most powerful and complete one at the moment [45].

There are also other tools serving as simulators or animators to allow the human user to experiment with *CSP* processes. Using these tools, the user can explore the studied system and interact with the *CSP* processes in reality instead of having to imagine doing so. Simulations using these tools can not prove results about processes. They only provide a form of implementation that allows experimentation. However, these tools can be used to assist formal analysis using *FDR*. *ProBE* (Process Behaviour Explorer) is one of this kind of tools [45].

Both FDR and ProBE are free and there are sufficient examples and exercises of using them in the available literature. Therefore, the *CSP* model verification carried out in this research will mainly rely on those two tools. More details of how both of them are used in this research will be given in Section 6.2.3.

In summary, the choice of *CSP* is very suitable in the context of this research. The level of analysis for concurrency required for modelling KT is well supported in *CSP* in terms of communication, recursion, abstraction, divergence and deadlock. Communications of events can be modelled both sequentially and concurrently along with introducing choice, composition and synchronisation. *CSP* also provides a mature framework for analysis on KT including model checking, which allows us to check for refinement, deadlocks and livelocks, all of which are relevant KT situations. The reliable tool support for *CSP* is also helpful.

3.5 Summary

A detailed methodological design of this research was explained in this chapter. A first-principles approach will be adopted to develop the graphical low-level transfer model. Individual tasks and required techniques and tools within this approach were also specified. This low-level KT model will be presented in the next chapter. Meanwhile, *CSP* will be applied in developing and verifying the formal KT model. The choice of *CSP* was justified in this chapter and how it is used in formalising our KT model will be explained in Chapter 5. The model-checking technique introduced here will be used in verifying our formal KT model in Chapter 6. It will also be used in Chapter 7 to show how the formal approach can be applied in practice. Model-checking will be further discussed in those two related chapters.

Chapter 4

A low-level Model of KT

4.1 Introduction

By following the first-principles approach described in Section 3.3, a low-level model for KT was developed. This chapter aims to describe this model in detail. It is organised as following. In the next section, we describe the conceptual foundation of this model which was built based on a theoretical investigation in Chapter 2. Then how the model was gradually developed is explained in the third section. In the fourth section, this low-level KT model is presented graphically and described in detail. An example transfer process is also presented in this section to demonstrate how the model can be used to represent a variety of KT processes. Next, an assessment of this model is described in the fifth section. And finally the need to formalise this low-level KT model is highlighted before this chapter concludes.

4.2 Conceptual foundation of the model

Before the interviews were conducted for the development of the low-level KT model, a clear understanding of KT was established in Chapter 2, such as what we mean by KT, how knowledge flows in the transfer processes, and when exactly knowledge is transferred. Meanwhile, the main focus of our KT model was justified and the reasons why only particular aspects of KT were considered in this model were also explained in Chapter 2. In this section, we briefly summarise related discussions and present them as the conceptual foundation of our low-level KT model.

On one hand, we built our understanding of KT on the basis of Davenport and Prusak's [15] work and revised their definition as the following:

$$KT_{PullingMode} = \textit{Requesting} + \textit{Sharing} + \textit{Absorption and Use} \tag{4.1}$$

$$KT_{PushingMode} = \textit{Transmission} + \textit{Awareness of needs} + \textit{Absorption and Use} \tag{4.2}$$

This revised KT definition indicates two different directions in which knowledge flows in the process of KT. They were described as the pulling mode and the pushing mode in Chapter 2. We attempt to represent both two knowledge flow directions while developing the low-level KT model. In our definition of KT, we recognize knowledge is transferred successfully as soon as a person's understanding or knowledge is refined after communicating a set of messages with another person either directly or through other people. A clear understanding of this concept is necessary particularly when we attempt to model a complete process of KT.

On the other hand, we attempt to look into what happened during KT but not how and why they happened. Our low-level KT model only focuses on the procedural aspect of KT, particularly people's decisional or interactive behaviours during KT. The above two behavioural patterns were identified as the critical features of our KT model. A detailed justification of this choice is given in Section 2.5. In order to present the above two critical features , three different kinds of details need to be captured in this model. They are:

- Major points when people have to make decisions;
- People's available options at each decision point;
- The consequent actions or outcomes following each option;

4.3 Evolution of the model

Six interviews were conducted using the model-based transfer case collection strategy (described in Chapter 3). A low-level transfer model was initially constructed based on the researcher's pre-existing understanding of KT. And then this initial model was gradually refined until the present version is produced based on these interviewees' personal experience in KT. The evolution of our transfer model is presented in Appendix A as a list of different versions.

4.3.1 The initial version of the model

Based on the researcher's pre-existing understanding of KT, the initial version of the KT model was constructed. First KT was presented as a logical sequence in terms of people's decisions, consequent actions following these decisions, and their interactions with others or other knowledge resources. Since a KT process often involves several people, its logical sequences can be very complicated. These logical sequences can be represented in a graphical form, so that people can understand them more easily. The choice of the symbols used in constructing the initial model was not finalised yet at this stage. More symbols were introduced in later versions as more complicated KT processes were encountered. A full list of the symbols used in our KT model is provided in Section 4.4.1.

Second, people's decisional and interactive actions were identified as the critical features of our KT model. However, representing the critical features alone may not always be sufficient to model KT processes at the individual level. In order to make our model work as planned, some supplementary non-critical features may need to be introduced in our model. For example, people take different roles during KT. Their roles are not critical to the model's value. However, representing people's activities in terms of their roles could simplify the model presentation. People may repeat their activities while participating in KT, so that placing these activities into the categories of different roles may avoid the duplication of the activities represented. In additional, relating people's activities to their roles could be a logical way to represent these activities in a more specific context. This is particularly useful in presenting similar activities people take at different stages within KT. Davenport and Prusak's [15] understanding of different KT roles was adopted while developing the initial version of our KT model. It simply included the roles of knowledge seeker and knowledge provider. Those two transfer roles were also common seen in other researcher's work in the KM domain.

Third, the process of KT was presented in one integrated framework in the initial version. Although a knowledge seeker's actions and a provider's actions were captured in two different boxes, a set of arrows representing the interactions between them was used to integrate these two parts into one single framework. It was decided to break down this single framework into separate units and present several major stages of KT in later versions. An explanation is given in the next section to show how this change was made in related versions of our model.

In addition, the two directions of knowledge flow were not explicitly highlighted in the initial version of the KT model. However, both knowledge flow directions could be traced by following the set of arrows which were used to connect individual transfer actions or decisions.

4.3.2 Major revisions to each version

The 1st interviewee provided a scenario in which a provider may provide knowledge without knowing others' knowledge needs. This means a person could receive new knowledge without being aware of the need for such knowledge first. Therefore a new role – knowledge recipient was added and a new connection was made between 1.B and 1.D in the 2nd version. In a similar approach, more revisions were made to each of the later versions of the model based on individual interviewees' contributions. These changes gradually refined the model and they are briefly summarised in Table 4.1.

One significant decision taken during the model evolution was the inclusion of extra transfer roles. For instance, the 2nd version started to include the role of knowledge recipient. The role of absorber was added in the 4th version and the role of needs assessor was first introduced in the 6th version (as the role of needs discoverer). In the 9th version, the role of needs transmitter was included.

These roles were added mainly to increase the capability of our KT model in representing complicated KT processes. For example, the 9th interviewee provided a KT example that the previous eight versions of our model could not correctly represent. Within this example, a seeker requested knowledge from a knowledge provider, but this provider decided to pass such request onto another person after assessing it (by taking the role of needs assessor). By including the new role of needs transmitter, the 9th version of our model could clearly represent this KT process. Another reason to add extra roles in our model evolution procedure was to simplify the graphical presentation of our earlier models by removing duplicated parts and present them as new transfer roles. These newly added roles were defined in order to reflect the nature of one's actions while taking these roles. The names of these roles were mostly chosen by the researcher except the role of knowledge recipient was a commonly used term in other KM researchers' work.

Another major decision taken during this evolution procedure was the breakdown of a knowledge repository. Apart from people, KT processes often involve knowledge repositories. Knowledge can be transferred between people or between people and repositories. In our model, the knowledge repositories are presented as containing two kinds of contents – direct knowledge and indirect knowledge. Without going into depth in specifying different kinds of knowledge, we view a person's knowledge as either direct knowledge or indirect knowledge. Direct knowledge is people's knowledge in the ordinary sense. It is exactly the kind of knowledge a knowledge seeker is after. Indirect knowledge is something related to direct knowledge. It may guide a knowledge seeker to locate and gain the direct knowledge or help a knowledge provider to describe (hence to share) his direct knowledge. Such a simple category allows us to reflect people's different behaviours while using different transfer mechanisms without the necessity of including the explicit and tacit dimension of knowledge. The terms of direct and indirect knowledge were finalised in the last version of our model, although the breakdown was firstly attempted as early as in the 2nd version.

The final major decision taken was to break down an integrated framework into several units to present KT in our model. This is mainly because more complicated KT examples were collected in later interviews while developing the model. These more complicated KT cases indicate that people may need to repeat their activities until knowledge is successfully transferred. In order to be able to represent these KT processes without having too many duplicated parts in our model, it is better to present the captured transfer details in several modular units rather than in one integrated framework. These individual units can be used as the basic elements to represent complicated KT processes. The first attempt of such a breakdown in producing the 3rd version. In the final version of our KT model, each unit was an individual transfer role and we have got six different units in total.

Version	**Changes made to the previous version**
2nd version	• Add the role of knowledge recipient; • Add $1.D$ to allow a knowledge recipient to receive new knowledge without being aware of related knowledge needs first; • Decide to break down the knowledge repository into a lower level (but not finalised yet);
3rd version	• Explicitly highlight the two knowledge flow directions – the knowledge pulling mode and the pushing mode; • Decide to present the KT model as several individual units instead of an integrated framework and start this with the units of seeker and provider in two knowledge flow modes;
4th version	• Simplify the previous version by adding the role of knowledge absorber ; • Take a knowledge recipient's actions out from the unit which describes a seeker's actions (although those two roles are normally taken by the same person); • Add extra steps to present a knowledge provider's actions, including $K.2A$, $K.2B$, and $P.1A$;
5th version	Remove duplicated steps from the previous version, particularly in the units of knowledge absorber and knowledge recipient

6th version	• Remove the presentation of two knowledge flow directions (the pulling and the pushing modes) and represent a seeker and a provider's actions in those two modes as their decisional actions during KT; • Add the role of knowledge needs discoverer; • Represent each transfer role as one individual unit in our KT model;
7th version	• Remove duplicated steps in the unit of knowledge recipient; • Add more steps in the unit of knowledge seeker, including $S.3A$ and $S.3B$;
8th version	• Combine the recipient of knowledge and the recipient of knowledge needs as one and present it as the role of recipient; • Refine the unit of knowledge needs discoverer and rename it as the role of knowledge needs assessor;
9th version	Add one extra step $D.3B$ in the unit of needs assessor to allow our model to represent the situation when a knowledge provider needs to be a seeker first before he could provide the requested knowledge to another seeker

Final version	• Add the role of knowledge needs transmitter to allow a person to pass on a received knowledge request to another person after taking the role of needs assessor; • Add one extra step $Option3 - ND2$ in the unit of needs assessor to allow one to have an option of passing a received request onto another person (another potential provider); • Finalise the breakdown of knowledge repository and categorise it as direct and indirect knowledge;

Table 4.1: How the low-level model was evolved

4.4 Graphical presentation of the final model

There are six different roles identified in the final version of our model. They are:

- *Knowledge Seeker* – Someone in the seeker role searches for knowledge to address his knowledge needs.
- *Knowledge and Needs Recipient* – In the recipient role, someone receives knowledge or requests for knowledge from others.
- *Knowledge Absorber* – Someone in the absorber role assesses received knowledge and decides whether to use it or not.
- *Needs Assessor* – In the role of a needs assessor, someone assesses received requests and decides how to respond.
- *Needs Transmitter* – Someone in the needs transmitter role passes on received requests to others.
- *Knowledge Provider* – In the provider role, someone provides knowledge to others directly or shares it through some repositories.

This low-level KT model is presented as Figure 4.1 below.

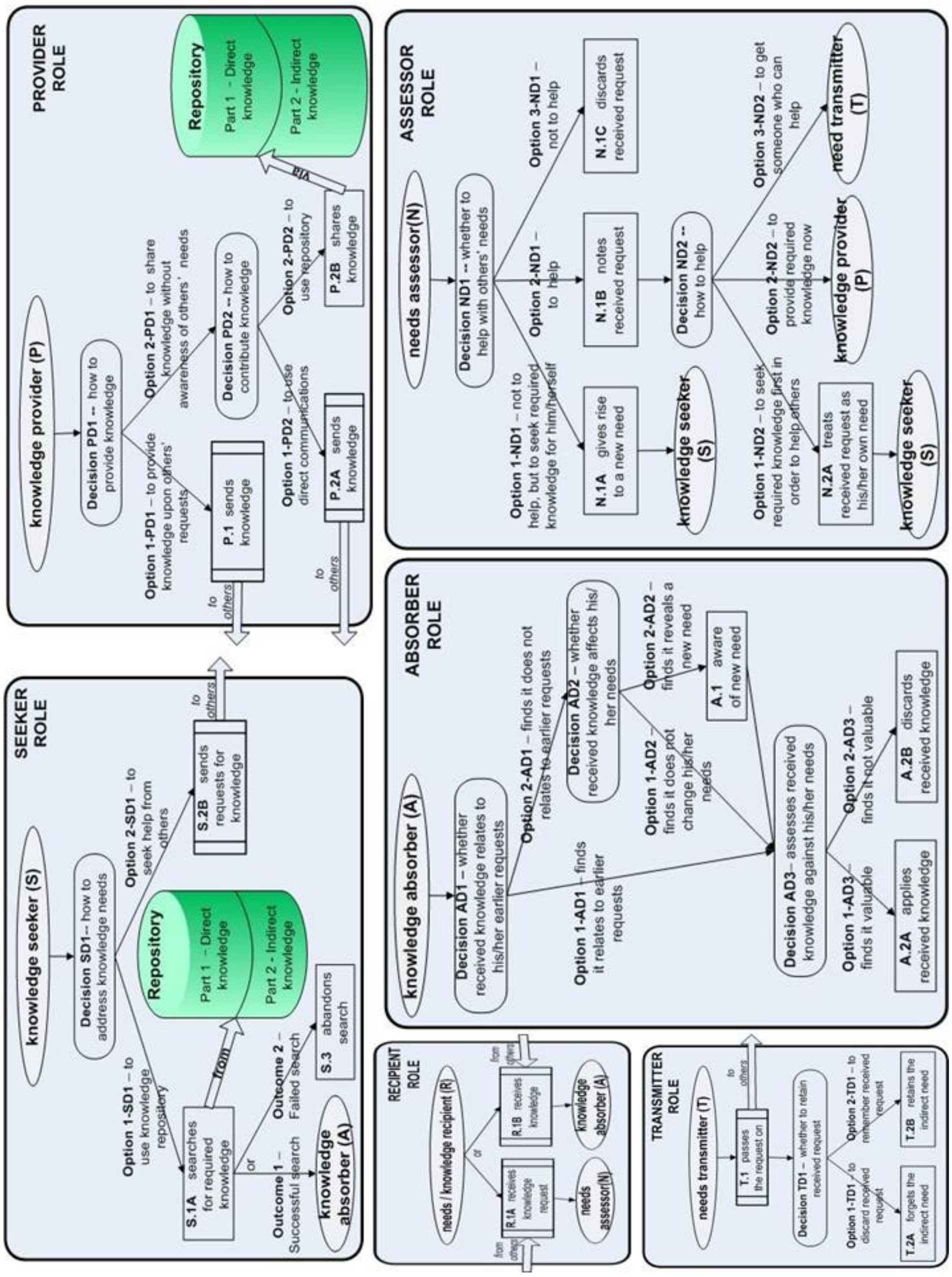


Figure 4.1: A low-level model for KT

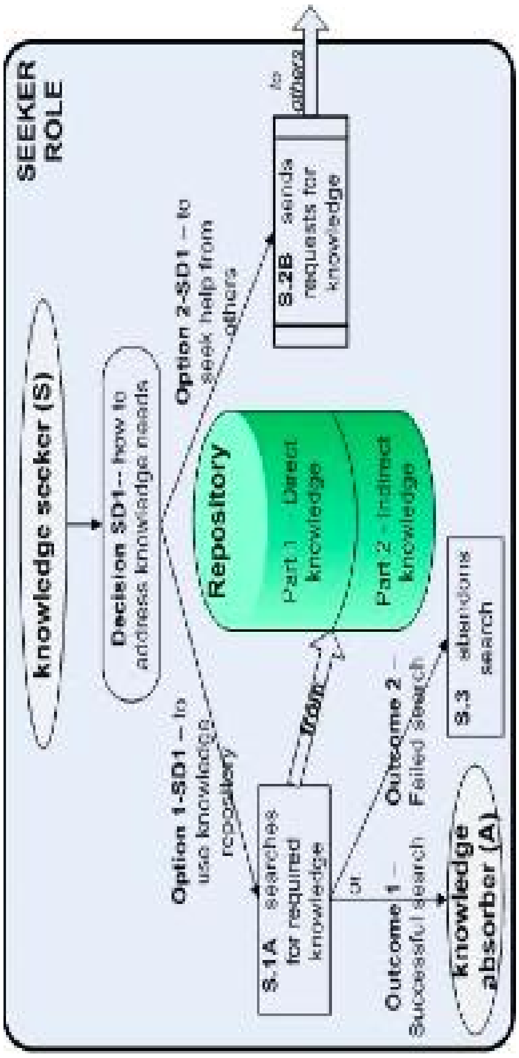


Figure 4.2: The knowledge seeker role

4.4.1 Graphical symbols used in the model

There are several graphical symbols used in constructing this model. Each of them are described in Table 4.2 below.

4.4.2 The role of knowledge seeker

A *Knowledge Seeker* must decide whether to search for knowledge in a *Knowledge Repository* or seek help from others ($Decision\,SD1$). If he decides to use a *Knowledge Repository* ($Option\,1 - SD1$), there are two possible outcomes of his search ($S.1A$). If he discovers new knowledge ($Outcome\,1$), he then becomes a *Knowledge Absorber*. But if he does not find anything ($Outcome\,2$), he may

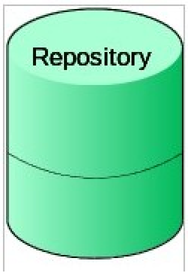
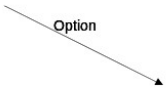
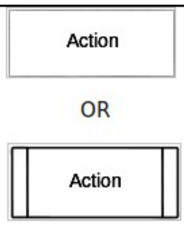
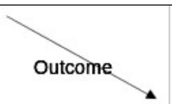
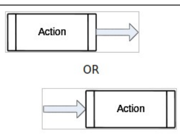
Symbols used	**Purpose**
	Transfer role
	Knowledge repository
	Decision points
	Options at each decision point
	Consequent actions following a chosen option
	Consequent outcomes following a chosen option
	Interactions with others

Table 4.2: Graphical symbols used in the low-level transfer model

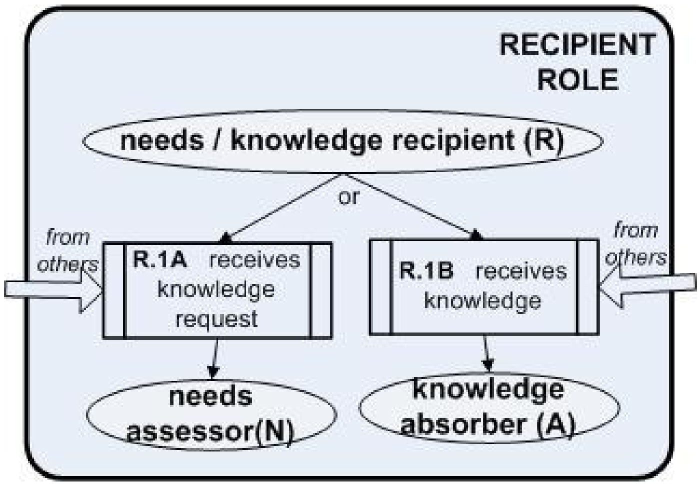


Figure 4.3: The knowledge and needs recipient role

abandon the search ($S.3$). On the other hand, if he had decided to seek help from others ($Option\,2 - SD1$), he would have made a request to another person (shown as the thick arrow from $S.2B$). Once the seeker has abandoned the repository search or made a request to another person, he returns to the starting point where he is not currently playing any role in the transfer and waiting to start another *Seeker*'s attempt.

4.4.3 The role of knowledge and needs recipient

A person in this role either receives knowledge requests from others ($R.1A$) or receives knowledge from others ($R.1B$). If he receives knowledge requests, he then becomes a *Needs Assessor* to decide how to deal with such requests. If he receives knowledge instead, he then becomes a *Knowledge Absorber* to decide if the received knowledge is worthy to be absorbed.

4.4.4 The role of knowledge absorber

A *Knowledge Absorber* first needs to decide whether the received knowledge relates to his earlier knowledge requests ($Decision\,AD1$). If the received knowledge does relate to the earlier requests ($Option\,1 - AD1$), he then has to assess its value against his knowledge needs ($Decision\,AD3$). If he finds the knowledge valuable, he then applies it to solve his work problems ($A.2A$). If the received knowledge is not valuable, he then discards it ($A.2B$). However, if he finds the received knowledge does not relate to any of his earlier requests ($Option\,2 - AD1$), he needs to check if this has helped him reveal a new knowledge need ($Decision\,AD2$). If it does not affect his knowledge needs

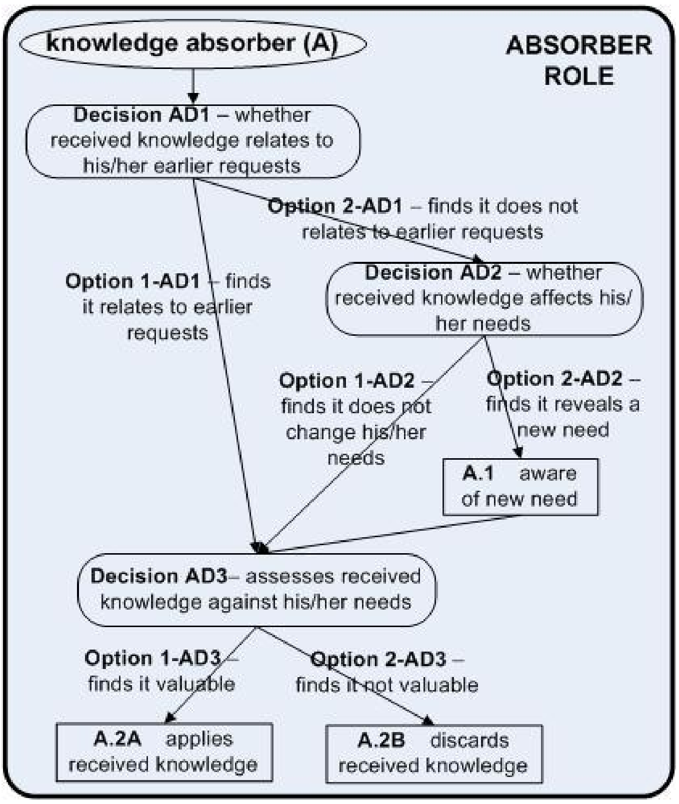


Figure 4.4: The Knowledge absorber role

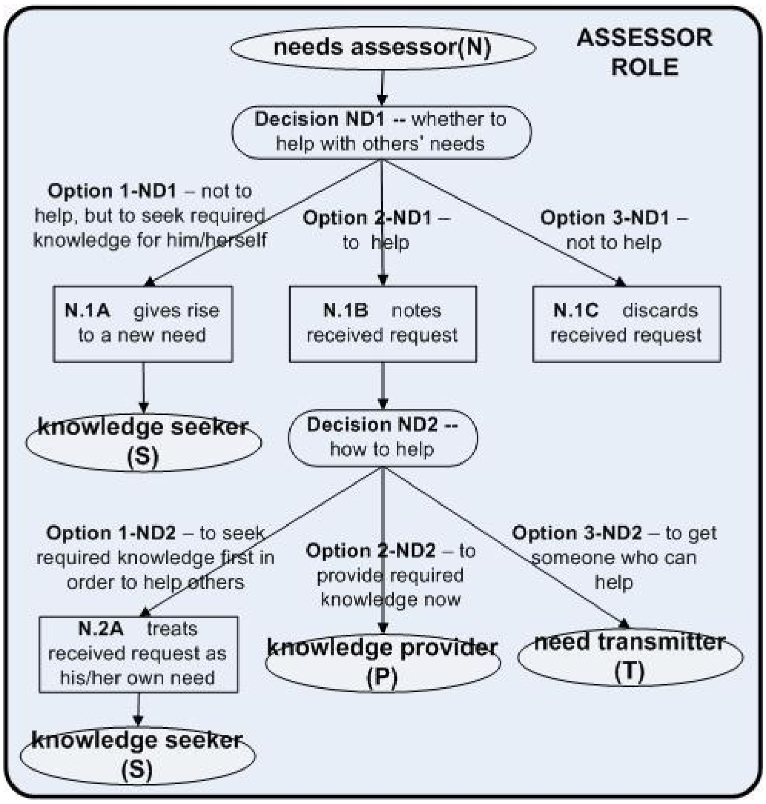


Figure 4.5: The needs assessor role

($Option\,1 - AD2$), he moves forward to assess its value ($Decision\,AD3$); but if it gives rise to a new need, he then has to reflect his renewed knowledge needs ($A.1$) before assessing the value of the received knowledge.

4.4.5 The role of needs assessor

When a person takes the role of *Needs Assessor*, first he needs to decide whether to help with others' needs or not ($Decision\,ND1$). He has three options here. One is not to help ($Option\,1 - ND1$) but to seek the requested knowledge for himself, because this may have given arise to a new knowledge need for himself ($N.1A$). The second option is to help the requester ($N.1B$), and the last one is not to help and also ignore the received request ($N.1C$). If he took the first option, he then becomes a *Knowledge Seeker*. If he had taken the second option which is to help the requester, he then has to decide how to help ($Decision\,ND2$). He has three options here. First although he is willing to help, he may not have the required knowledge at the moment. Therefore in this case, he has to treat the received request as his own need ($N.2A$) and act as a *Knowledge Seeker* to

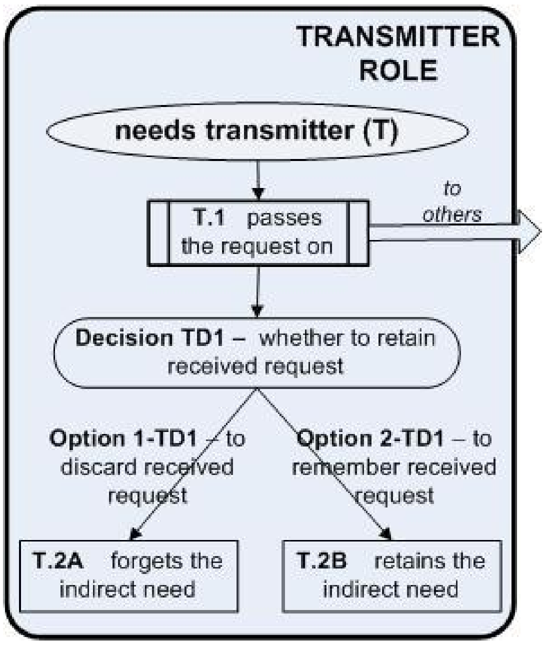


Figure 4.6: The needs transmitter role

search for the required knowledge. The second option (*Option* $2 - ND2$) for him to help is to act as a *Knowledge Provider* to provide the required knowledge directly. The last option (*Option* $3 - ND2$) is to act as a *Needs Transmitter* to pass on the received request to someone else who is more capable of providing direct knowledge.

4.4.6 The role of needs transmitter

A *Needs Transmitter* first passes on a knowledge request (that was received earlier from someone else) to others (shown as the thick arrow from $T.1$). He then has to decided whether to retain this request or not (*Decision* $TD1$). He can either discard it immediately (*Option* $1 - TD1$) or retain it in case of providing further help to the original requester.

4.4.7 The role of knowledge provider

When a person takes the role of *Knowledge Provider*, his first decision (*Decision* $PD1$) is either to provide knowledge to others according to their requests (*Option* $1 - PD1$) or just provide whatever he thinks necessary (*Option* $2 - PD1$). If he decides not to provide knowledge regarding to people's knowledge requests, he then needs to decide the means of sharing his knowledge with others (*Decision* $PD2$). He can either choose to communicate his knowledge with others in person ($P.2A$)

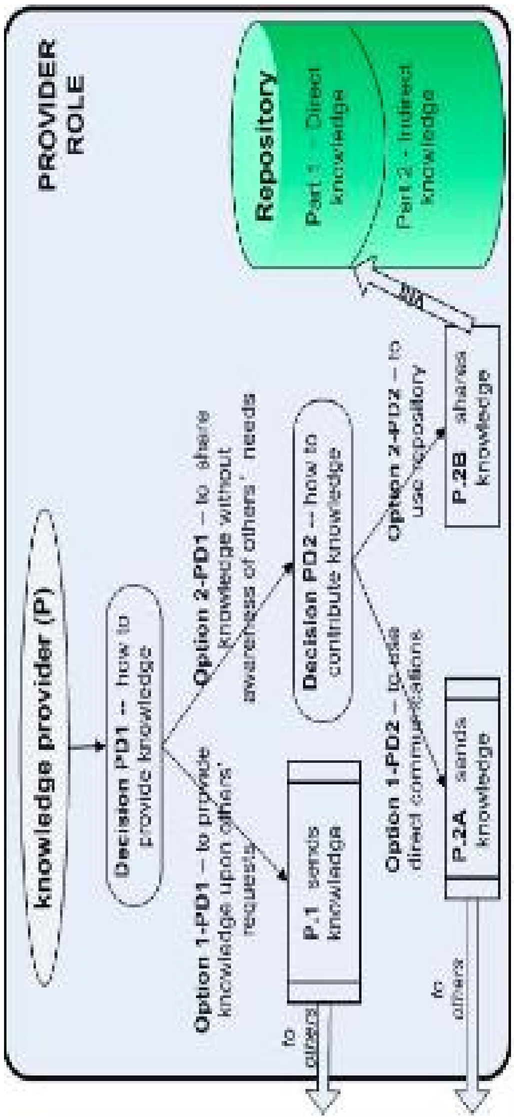


Figure 4.7: The knowledge provider role

or just share it through a *Knowledge Repository* ($P.2B$).

4.4.8 An example of KT

The model can be used to represent a variety of KT processes. For example Figure 4.8 shows an instance of KT involving three people. This KT example was provided by one of the interviewees who helped developing our low-level KT model.

In this KT example, Person 1 initiates the transfer by acting as a *Knowledge Seeker* (S) after developing some needs. He decides to seek help ($SD1$) from a second person – Person 2 ($S.2B$). Person 2 is then in the role of a *Needs Recipient* (R) as a result of the query from Person 1 ($R.1A$). Next, Person 2 becomes a *Needs Assessor* (N) and decides to help person 1 ($N.1B$). He then decides to act as a *Needs Transmitter* (T) and passes on the query to a third person – Person 3 ($T.1$). Person 2 then finally chooses to retain this query ($T.2B$) because he may provide further help later. Meanwhile, Person 3 becomes a *Needs Recipient* ($R.1A$), then assesses the request ($ND1$), and decides to help ($N.1B$). Next, he chooses to become a *Knowledge Provider* (P) and respond directly to Person 1 ($P.1$). Person 1 becomes a *Knowledge Recipient* (R) as he receives the response from Person 3 ($R.1B$). Then he takes the role of a *Knowledge Absorber* (A). To decide if he is going to use the received knowledge, he first checks if it relates to one of his earlier requests ($AD1$), before assessing the received knowledge against his needs ($AD3$), and deciding to use it ($A.2A$). The transfer process is now complete.

4.5 Assessment of the model

Our low-level KT model is designed to help people understand KT processes in various situations, so that related transfer problems can be analysed. In order to check if our model has fulfilled this purpose, both theoretical and empirical assessments are conducted in this section. Although the model validation and refinement were planned as a combined task in the spiral process (Section 3.3), the assessment of the model is still described separately in fifth section. During the model development, our model reached to a stable state when no further revision was required as soon as the model-based transfer case collection and analysis were completed. Therefore the independent transfer case collection and analysis were used mostly for assessing this model.

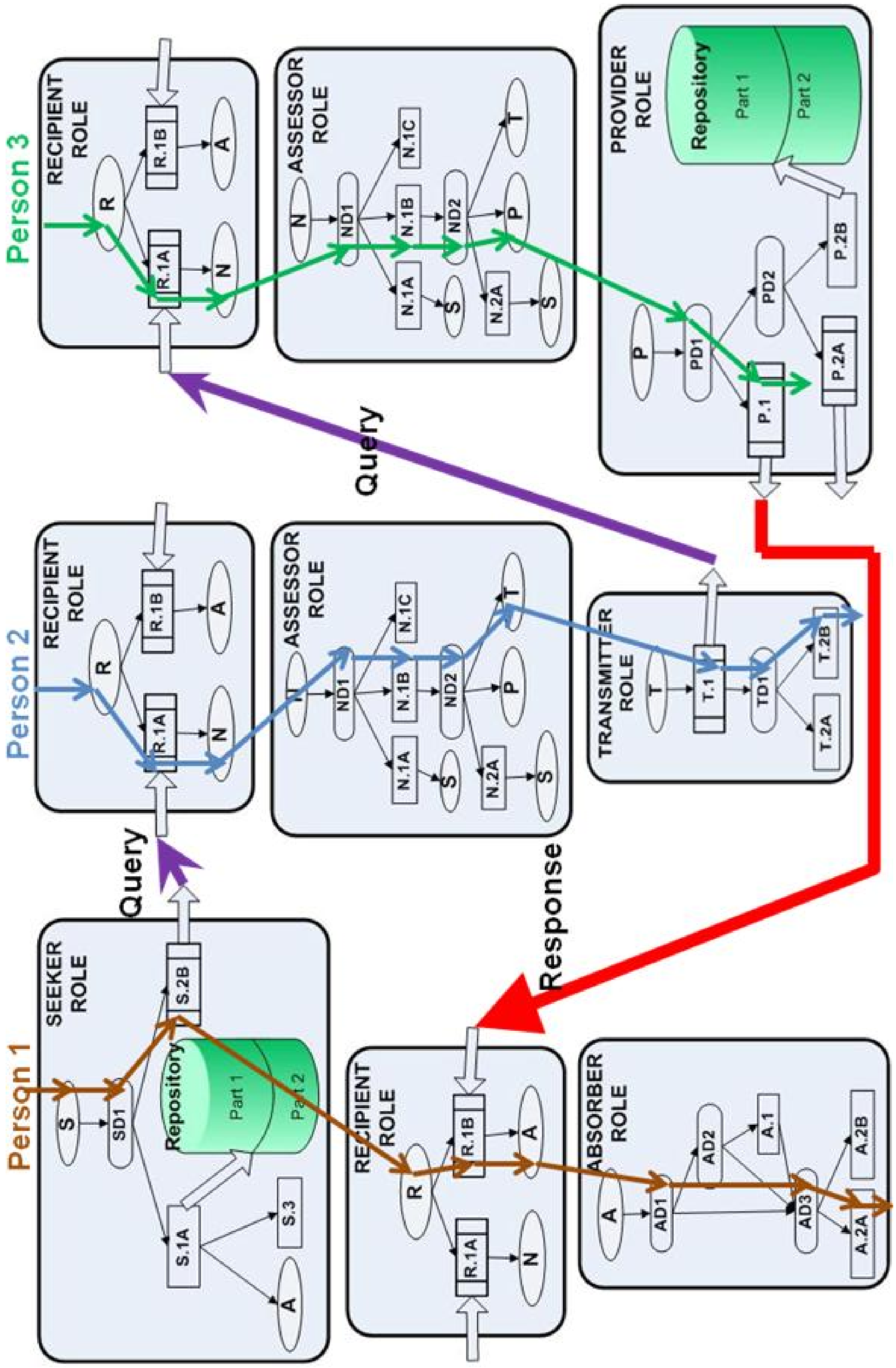


Figure 4.8: An KT example illustrated using the low-level model

4.5.1 Theoretical assessment

How different KT strategies help facilitate KT processes is frequently discussed in available literature (see more discussions in Section 2.3.2). One way to assess our low-level model is to check if it allows these strategies to be represented in detail by showing steps and transactions that take place in the course of transfer. KT processes used in this assessment are mostly generated from current literature based on the researcher's understanding toward a variety of KT strategies. Therefore, such an assessment is at a theoretical level.

For example, a community of practice is an informal strategy for KT which is intended to facilitate direct communications [64]. The model can be used to demonstrate these communications. Figure 4.8 (shown earlier as the example process of KT illustrated using our low-level model) is also one example of how people may interact in a community of practice to transfer knowledge.

4.5.2 Empirical assessment

Although empirical assessment cannot prove the model to be true, it can potentially reveal weaknesses or faults in the model. In practice, it is difficult to observe a full KT involving several people. Therefore this assessment has been conducted by collecting data from individuals about their transfer experience and checking whether the steps they followed are consistent with the 29 steps shown in the model (Each thin arrow leading to an action box or a decision point in the roles, shown in Figure 4.1 is counted as one step).

4.5.2.1 A diary study

The diary study(described earlier in Chapter 3) was conducted. Eight volunteers completed diaries about their knowledge seeking and knowledge providing experiences. A total of 74 reported KT cases were collected. All behavioural phrases within each case were highlighted. And then these cases were paraphrased as sequences of actions by using our KT model as a coding scheme that contains predetermined categories which allows us to encode whether or not a particular behavior (each step in the model) has occurred. A complete list of the encoded transfer cases can be found in Appendix B.

It is quite common that the participants may not explicitly mention every single decision they have made during KT. However, knowing their actions afterwards allows us to specify which option they have taken at these implicit decision points. For example, one KT case was described as the following: "When having a problem, the participant tried to seek help from a colleague first, but the response received did not help solve the problem. Then he decided to seek help from another colleague, and was still waiting for a response". Using our KT model, this example can be encoded as involving the following steps: S − $SD1$ − **S.2B** − R − **R.1B** − A − $AD1$ − $AD3$ − **A.2B** − S − $SD1$ − **S.2B** − *still waiting*; The above steps not in bold represent several decision points that have not been explicitly mentioned in the participant's diary, but can be inferred. The last part of this example case "still waiting for the response", cannot be encoded using the model.

From all the collected cases, a total of 197 steps can be encoded. Some of the uncodable actions described in the reported cases involve behaviors that do not relate to KT (e.g. a sequence of actions of trying different IT techniques to solve one problem). The other uncodable actions are mainly concerned with time which is not taken into account in the model (e.g. the step of "still waiting" as shown in the above example). No other uncodable actions relating to KT were observed.

4.5.2.2 A follow up interview survey

A follow up interview survey was conducted to cover KT situations that the diary study could not explore. On one hand, only 18 of the 29 steps in the model appear in the collected cases in the diary study. The following 11 steps in the model were not observed: three steps of providing knowledge without any requests ($PD1 - PD2$; $PD2 - P.2A$; $PD2 - P.2B$); four steps of absorbing knowledge without any earlier requests ($AD1 - AD2$; $AD2 - AD3$; $AD2 - A.1$; $A.1 - AD3$); three steps of dealing with received requests after passing them to others ($T.1 - TD1$; $TD1 - T.1A$; $TD1 - T.1B$); and one step of treating other's needs as his own even after deciding not to help ($ND1 - N.1A$). This may be because the questions in the logbooks only focus on how knowledge is transferred when people have immediate problems to solve. The above unobserved situations were not considered in the design of the logbooks. Therefore follow up interviews were necessary to further investigate into these steps.

On the other hand, out of the 74 collected cases, 15 cases relate to complicated knowledge transactions involving more than two parties. Only details of one person's steps are described in the collected cases. The full details of transactions between different parties are not captured in the data. Therefore the follow up interview survey was also conducted for tracking down the complete knowledge processes.

Five of the diary study participants volunteered to answer the follow up interview questions. These questions were designed to look into their personal experience in unexplored transfer cases. Their answers are recorded and encoded in a similar way as the diary study data. According to the these interviews, people's transfer experience is still consistent with the low-level model. In addition, no other uncodable actions relating to KT were reported.

4.5.2.3 Discussion

Although the above results are largely consistent with the model, several weaknesses of the model are apparent. There are certain aspects of the transfer process that the model does not attempt to represent. For example, as discussed in the empirical assessment, time is not reflected in the model. However, representing time was not one of the design objectives of the model. In addition, the social factors affecting people's decisions are not shown in the model, though it highlights the major decision making points. However, it is not a primary objective to include these cognitive factors in our transfer model. In short, both theoretical and empirical assessments of the model show that the model has captured sufficient detail of KT processes regarding to the requirements of this research (see Chapter 2 and Chapter 3).

4.6 The need to formalise the model

There is a potential problem in presenting our KT model using a graphical form. People may misinterpret the graphical sequence of the captured transfer details of the model. One approach to address this issue is to have a precise and formal definition of the process sequences. It can provide a standard interpretation of the graphical presentation of the model. Because of the enhanced formality, a formal presentation of our KT model can be used in analysing problematic KT environments potentially. This would be a novel way of analysing KT in the KM domain.

4.7 Summary

The low-level KT model is presented in this chapter. In this model, a clear definition of KT is provided, so that it can be distinguished from other misused terms, such as knowledge share and knowledge exchange. The moment knowledge is considered to be transferred is also clarified in this model. It provides a rich picture of how a KT process takes place by capturing people's roles, their major decisions and interactions with others during this process. The model can be used to represent various KT processes at a detailed level. It also allows a wide range of KT mechanisms, including knowledge maps and communities of practice, to be represented in a common framework.

This low-level KT model was assessed both theoretically and practically. The assessment showed its strength in representing a variety of KT processes. The results also revealed some aspects that the model can not represent, such as the time and social factors of transfer. However with the absence of these aspects, the model still sufficiently represents KT at a low-level and includes the transfer details required for this study.

Although this model captured rich details of KT and can potentially help people understand different KT situations, its graphical presentation determined that it could have the following two main shortcomings. First, people may misunderstand the transfer details in this model because of its informal presentation. Second, how people adopts this model in understanding KT processes may vary since there is no standard way of using it. In order to address this issue, this low-level transfer model needs to be formalised. Such a formalisation will be described in the next chapter.

Chapter 5

A Formal Model of KT

5.1 Introduction

As highlighted in Section 3.2, the graphical low-level transfer model presented in the previous chapter is very informal. People may misinterpret the graphical sequence of the captured details of KT and misuse this model. A formalisation of this model could help address the above issue. This chapter is mainly concerned with how this model can be formalised using *CSP*. The choice of *CSP* was justified in Section 3.4.6. This chapter is organised as following. First, the basic building blocks of *CSP* describing the notation and features of the language relevant to our usage are introduced. Then several simple criteria for formalising the graphical model are set out. Each of the individual transfer processes from their model is then formalised using *CSP* in the following six sections. In the tenth section, we present the entire KT model showing critical synchronisations between individual processes.

5.2 *CSP* Notation

A *CSP* system is modelled in terms of *processes* and *events*. The *CSP* expression $a \rightarrow P$ describes a process that initially performs event a and then behaves as process P. For example, a process *SALE* shows a single transaction of a vending machine selling chocolate. The vending machine accepts the right amount of coins and sells a chocolate. This single transaction process can be represented as

$$SALE = coin \rightarrow choc \rightarrow STOP$$

This describes a process that can perform the event *coin* and *choc*, after which is simply *STOP*. Process *STOP* is the simplest *CSP* process that can be described; it has no event transitions and does not engage in any events. Process *SALE* would simply make no further progress once it reaches STOP. We develop the process further to describe a vending machine recursively as

$$VM = coin \rightarrow choc \rightarrow VM$$

This allows *VM* to accept coins, sell chocolate and return to the original state to accept more coins in future transactions.

An external choice operator $\Box$ provides the option of running either of the two processes P or Q when put together as $(P \Box Q)$ where the choice between these two processes is determined by the first event that is performed, which can be chosen by the environment. For instance, if the vending machine also allows users to pay chocolate by cards, then it can be represented as

$$VMC = (coin \rightarrow choc \rightarrow VMC) \Box (card \rightarrow choc \rightarrow VMC)$$

The parallel operator $[\![A]\!]$ is used to force P and Q to run in parallel and synchronise on events in the set of events A, whereas any of their events that are not in A are performed independently. This is written as $(P \,[\![A]\!]\, Q)$. In the vending machine example, a *BUYER* can be represented as

$$BUYER = coin \rightarrow STOP \Box card \rightarrow STOP$$

A buyer can pay for the chocolate by coins or card. The *BUYER* process has to be synchronised with *VMC* process on events *coin* and *card*. The *TRANSACTION* between them can be written as

$$TRANSACTION = BUYER[\{coin, card\}]\,VMC$$

5.3 Formalising interactions

Since the purpose of introducing *CSP* in this research is to capture detailed sequences of people's behaviours, the original transfer roles have to be redefined in order to have a consistency with the *CSP* notations. This does not change the captured details in the original model, as it is just represented from a different perspective.

- People's transfer actions are treated as *events*.
- Individual transfer roles in the model are seen as separate *processes*.
- Similar to a *CSP* system formed by processes and events, KT in the model is seen as a *system*.
- Different options that people have at various decision points are distinguished using the *external* choice operator $\Box$.
- Interactions between different transfer processes are represented as *synchronised events* allowing processes to execute *in parallel*.

In the graphical transfer model (described in Chapter 4), transfer roles are defined according to the nature of the actions taken by people but not the sequential logic of people's behaviours. Before starting the development of the formal *CSP* model, we need to justify whether or not it is necessary to convert all transfer roles into *CSP* processes. First, only two roles can activate KT in the graphical model. A *Knowledge Seeker* can start the KT by searching for required knowledge, while a *Knowledge Provider* can also start the transfer by providing his knowledge to others. Therefore, both *Knowledge Seeker* and *Knowledge Provider* should be formalised as *CSP* processes – *SEEKER* and *PROVIDER*. Second, *Recipient* is treated as a single role in the graphical version of the model, because the recipient of knowledge and knowledge needs are not distinguished. However receiving different contents by a recipient determines different interactions with other involving parties. These interactions are key features of a *CSP* system and have to be represented clearly in the formal model. Consequently, we split the original *Recipient* into two separate roles (*Knowledge Recipient* and *Needs Recipient*) and formalise them as two individual *CSP* processes – *K_RECIPIENT* and *N_RECIPIENT*. Third, there is a role of *Knowledge Absorber* in the graphical model. In reality, a person would never become a *Knowledge Absorber* if he did not take the role of *Knowledge Recipient* previously. Therefore, we can describe an *Knowledge Absorber*'s behaviours in process *ABSORB* and treat it as a subprocess within *K_RECIPIENT*. Similarly, the role of *Needs Assessor* would never be taken by someone if he was not a *Needs Recipient* previously. In our model, a *Needs Assessor*'s behaviours are captured in process *ASSESS* and it is also treated as a sub-process within *N_RECIPIENT*. Fourth, a *Needs Transmitter*'s behaviours are described in process *TRANSMITTER*. It could also been seen as a sub-process within *N_RECIPIENT* in principle. However, we treat it as a separate process in our model because of the following reason. When a person who was previously a *Needs Recipient* takes the role of *Transmitter*, he passes on the knowledge needs to another person who will also act as a *Needs Recipient*. This indicates a different interaction between *TRANSMITTER* and *N_RECIPIENT*. Finally, *Knowledge Repository* is an important part of the low-level model and it interacts with other processes, such as *SEEKER* and *PROVIDER*. It has to be treated as a separate process in the *CSP* model – *REPOSITORY* .

The major moments when people make decisions are highlighted in the graphical model, because this model is developed to guide people's future research on key transfer issues, such as trust and culture. However, these decision points become unnecessary in our *CSP* version of this model. As a general principle in *CSP* modelling, it is sensible to avoid events that do not influence any other behaviour. Only people's options at these decision points are the critical events of each process. This allows us to simplify our formal model without losing any key features of the KT system.

In summary, six separate processes are defined in our formal KT model, including *SEEKER*, *K_RECIPIENT*, *N_RECIPIENT*, *TRANSMITTER*, *PROVIDER* and *REPOSITORY*.

5.4 The process of knowledge seeker

Process *SEEKER* corresponds to the *Knowledge Seeker* role (presented in Section 4.4.2) in our low-level KT model. This process is activated when a person starts to search for required knowledge (described as a subprocess *SEEK*). A *Knowledge Seeker* can either use a *Knowledge Repository* or request knowledge from other people. If he succeeds in searching the repository and retrieves new knowledge, he then becomes a *Knowledge Recipient*. But if he does not find anything, he then returns to the starting point of the entire process and be ready for another *Knowledge Seeker*'s attempt. On the other hand, if the *Knowledge Seeker* had decided to send request to others for help, he would interact with a *Needs Recipient* and wait for responses.

SEEKER can also be a consequent process triggered by other processes. For example, a *Needs Recipient* (process *N_RECIPIENT*) becomes a *Knowledge Seeker* when a new knowledge need is raised. This activates the subprocess *SEEK*.

Knowledge Seeker

process

$SEEKER = SEEK \;\Box\; n\_raised \rightarrow SEEK$

process

$$
\begin{aligned}
SEEK = \; & search\_repository \rightarrow \\
& \quad (s\_succeed \rightarrow k\_retrieved \rightarrow SEEKER \\
& \quad \Box\; s\_failed \rightarrow SEEKER) \\
& \Box\; r\_sent \rightarrow (r\_responded \rightarrow k\_provided \rightarrow SEEKER \\
& \quad \Box\; SEEKER)
\end{aligned}
$$

5.5 The process of knowledge recipient

The *Recipient* role (described in Section 4.4.3) in our low-level model is formally defined as two separate processes (as explained in Section 5.3). The actions performed by a *Knowledge Recipient* are described in process *K_RECIPIENT*. This process is activated when someone previously acted as a *Knowledge Seeker* and successfully retrieved knowledge through a *Knowledge Repository*. It can also be activated if a *Knowledge Provider* has provided someone knowledge either upon his earlier request or without him asking for it.

Then in subprocess *ABSORB* he has to assess the received knowledge and decide how to deal with it. If it corresponds to his earlier request, he will assess its quality and choose either to apply or discard it. If he finds it irrelevant to his earlier request, he then has to check if his knowledge needs remain the same or the received knowledge has helped reveal a new need. In either of the these two cases, he then decides if he will apply the received knowledge or discard it.

Knowledge Recipient

process

$$K\_RECIPIENT = k\_retrieved \rightarrow ABSORB$$
$$\Box\ k\_provided \rightarrow ABSORB$$
$$\Box\ k\_offered \rightarrow ABSORB$$

process

$$ABSORB = k\_relevant \rightarrow (k\_applied \rightarrow K\_RECIPIENT$$
$$\Box\ k\_discarded \rightarrow K\_RECIPIENT)$$
$$\Box\ k\_irrelevant \rightarrow (same\_need$$
$$\rightarrow (k\_applied$$
$$\rightarrow K\_RECIPIENT$$
$$\Box\ k\_discarded$$
$$\rightarrow K\_RECIPIENT)$$
$$\Box\ new\_need$$
$$\rightarrow (k\_applied$$
$$\rightarrow K\_RECIPIENT$$
$$\Box\ k\_discarded$$
$$\rightarrow K\_RECIPIENT)$$
$$)$$

5.6 The process of knowledge provider

Process *PROVIDER* corresponds to the role of *Knowledge Provider* (presented in Section 4.4.7) in our low-level KT model. A *Knowledge Provider* may provide knowledge upon others' requests. He may also choose to share his knowledge without people asking for it in two ways, either offering it to them directly or contributing to knowledge repositories where people can access when they need.

Knowledge Provider

process

$$PROVIDER = k\_offered \rightarrow PROVIDER$$
$$\Box\ provide\_upon\_request \rightarrow PROVIDER$$
$$\Box\ add\_to\_repository \rightarrow PROVIDER$$

5.7 The process of needs recipient

The actions performed by a *Needs Recipient* (as a part of the role of *Recipient* presented in Section 4.4.3) in our low-level model are formalised as process *N_RECIPIENT* here. This process is activated when someone received a knowledge request sent by a *Knowledge Seeker* or transferred by a *Needs Transmitter*. It can also be activated if someone was a *Needs Transmitter* previously and decided to retain the request, so that he can reassess the same request and help the original *Knowledge Seeker* again.

Then in the subprocess *ASSESS*, he has to assess the request and decides how to respond to it. On one hand, if he does not have the requested knowledge and receiving this request helps him raise a new knowledge need, he may start to search for knowledge for himself. On the other hand, if he decides to help the *Knowledge Seeker* who sent the request, he can either respond directly and provide knowledge upon request, or pass on the request to others who may be more capable. Otherwise, he may just ignore the request.

Needs Recipient

process

$$
\begin{aligned}
N\_RECIPIENT = \; & r\_sent \rightarrow ASSESS \\
& \Box\; n\_transferred \rightarrow ASSESS \\
& \Box\; r\_retained \rightarrow ASSESS
\end{aligned}
$$

process

$$
\begin{aligned}
ASSESS = \; & n\_raised \rightarrow N\_RECIPIENT \\
& \Box \\
& r\_noted \rightarrow (r\_responded \rightarrow provide\_upon\_request \\
& \qquad \rightarrow N\_RECIPIENT \\
& \qquad \Box\; passon\_request \rightarrow n\_transferred \\
& \qquad \rightarrow N\_RECIPIENT) \\
& \Box\; r\_ignored \rightarrow N\_RECIPIENT
\end{aligned}
$$

5.8 The process of needs transmitter

Process *TRANSMITTER* formally defines the role of *Needs Transmitter* (presented in Section 4.4.6) in the low-level model. This process is activated when a *Needs Recipient* decides to pass on the knowledge request (that was received earlier) to another person. Then the *Needs Transmitter* can either discard the request or retain it. If he retains the request, he may prefer to reassess it later and decide whether or not to provide further help to the requester. Otherwise if he chooses to discard the request, he then returns to the starting point of the process.

Needs Transmitter

process

$$\begin{aligned} TRANSMITTER = {} & passon\_request \rightarrow n\_transferred \\ & \rightarrow (r\_discarded \rightarrow TRANSMITTER \\ & \quad \Box\ r\_retained \rightarrow TRANSMITTER) \end{aligned}$$

5.9 The process of knowledge repository

The component *Knowledge Repository* (presented in Section 4.3.2) of the low-level KT model is formally represented as process *REPOSITORY*. Without exploring too many details of how a *Knowledge Repository* operates, we only define the events representing its interactions with other transfer roles including *Knowledge Seeker* and *Knowledge Provider*. We treat all operations within a *Knowledge Repository* as internal events that are not visible to external parts of the system.

Knowledge Repository

process

$$\begin{aligned} REPOSITORY = {} & s\_succeed \rightarrow REPOSITORY \\ & \Box\ add\_to\_repository \rightarrow REPOSITORY \end{aligned}$$

5.10 The formal KT model

The above six processes execute in parallel. Several events allow these processes to relate to others and also to form a system representing the formal KT model. This KT system consists of several sub-systems – *NESYS*, *KSSYS*, *KPSYS*, and *KRSYS*.

5.10.1 Exchange of knowledge needs

NESYS represents a subsystem for knowledge needs exchange emphasising the synchronised events (*passon_request*, *n_transferred* and *n_retained*) between *TRANSMITTER* and *N_RECIPIENT*. This system describes a situation when a *Needs Recipient* becomes a *Needs Transmitter* (showing as process *N_RECIPIENT* evolves to process *TRANSMITTER* with event *passon_request*), and then the same *Transmitter* interacts with another *Needs Recipient* on event *n_transferred*. Event *n_retained* takes place when a *Transmitter* retains a knowledge request after passing it on to others. This allows him to become a *Needs Recipient* again. In other words, event *n_retained* allows process *TRANSMITTER* to evolves to process *N_RECIPIENT*.

Knowledge needs exchange

process
$$NESYS = N\_RECIPIENT \; [\![\{passon\_request, n\_transferred, n\_retained\}]\!] \; TRANSMITTER$$

5.10.2 Knowledge supply

KSSYS is a subsystem for knowledge supply where process *PROVIDER* synchronises with process *REPOSITORY* on the event *add_to_repository*. This system captures behaviours of a *Knowledge Provider* adding knowledge into a *Knowledge Repository*.

Knowledge supply

process
$$KSSYS = REPOSITORY \; [\![\{add\_to\_repository\}]\!] \; PROVIDER$$

5.10.3 Synchronising *NESYS* and *KSSYS*

KPSYS allows sub-systems *NESYS* and *KSSYS* to be synchronised. Within this subsystem, process *PROVIDER* and process *N_RECIPIENT* have to be synchronised on event *provide_upon_request*. This event takes place when a *Needs Recipient* chooses to help others with their knowledge requests by acting as a *Knowledge Provider*.

Synchronising NESYS *and* KSSYS

process
$$KPSYS = KSSYS \; [\![\{provided\_upon\_request\}]\!] \; NESYS$$

5.10.4 Knowledge retrieval

KRSYS represents a subsystem for knowledge retrieval capturing a *Knowledge Seeker*'s behaviours in retrieving knowledge from others. Within this subsystem, process *SEEKER* synchronises with process *K_RECIPIENT* on events *k_retrieved* and *k_provided*. The former event takes place when a *Knowledge*

Seeker becomes a *Knowledge Recipient* after he has successfully retrieved knowledge from a *Knowledge Repository*; while the latter event allows process *SEEKER* evolves to process *K_RECIPIENT* once a *Knowledge Seeker* has received knowledge from a *Knowledge Provider*.

Knowledge retrieval

process
$$KRSYS = K\_RECIPIENT \mathbin{[\![} \{k\_retrieved, k\_provided\} \mathbin{]\!]} SEEKER$$

5.10.5 The KT system

KTMODEL is formed when *KPSYS* is in parallel with *KRSYS*. They need to synchronise on events *r_sent*, *n_raised* and *r_responded*, *s_succeed*, and *s_failed*. Event *r_sent* allows a *Knowledge Seeker* to interact with a *Needs Recipient*, so that process *N_RECIPIENT* can be activated. Event *n_raised* takes place when a *Needs Recipient* decides to become a *Knowledge Seeker* to search for new knowledge before helping others with their knowledge requests. This event allows process *N_RECIPIENT* to evolve to process *SEEKER*. Event *r_responded* allows a *Needs Recipient* to respond a *Knowledge Seeker* upon his earlier request for knowledge. Both events *s_succeed* and *s_failed* are the synchronised events between process *SEEKER* and process *REPOSITORY* showing the two possible interactions between a *Knowledge Seeker* and a *Knowledge Repository*. *KTMODEL* can be viewed as the overall KT system.

Knowledge Transfer Model

process
$$KTMODEL = KPSYS \mathbin{[\![} \{r\_responded, s\_succeed, k\_offered, r\_sent, n\_raised\} \mathbin{]\!]} KRSYS$$

5.11 Summary

This chapter describes a formalisation of the low-level KT model and presents a formal version. This formal KT model is presented as one of the main contributions of this research. Process algebra *CSP* was applied for such a formalisation. The dynamics and complexities in KT are well represented using the concurrency theory in *CSP*. This formal model presents KT as a *CSP* system which is formed by six individual *CSP* processes. These processes execute in parallel and synchronise on several events that represent interactions between different parties involved in KT. The choice operator is particularly useful as it allows people's choices at various decision points to be represented correctly in the KT system. The *CSP* notations used in our formal KT model are simple but sufficient to present the transfer details required in this study (more details seen Section 2.5). The formality of this KT model because of the application of *CSP* allows us to study KT in a precise and systematic approach. This also contributes in overcoming the two weaknesses of the graphical low-level KT model (explained in Section 4.7).

Each *CSP* process within this formal KT model can be checked individually to analyse deadlocks, livelocks and refinements, which help provide a certain level of grantee of correctness of our formal KT model. In order to verify our formal KT model, a detailed formal model analysis will be described in the next chapter.

Chapter 6

Verification the Formal KT Model

6.1 Introduction

This chapter is mainly concerned with verifying our formal KT model and reveal any flaws it may have with respect to the purpose of this study. It is organised as following. An overview of model verification and relevant concepts for verifying our formal KT model are introduced in the next section. Then a detailed verification of the model is described in the third section. Not only the desired properties of our model are specified, but also the deadlock and livelock freedoms are checked in this section.

6.2 Model verification

A *CSP* system is designed to satisfy particular requirements, and one of the main benefits of using *CSP* semantics is that it allows this system to be judged against given specifications. Schneider [47] explains that "specification on behaviours describe those executions that are acceptable, an a verification of a system or process P requires an argument to establish that no behaviours of P violates such a specification" [page 195]. Since any *CSP* process is associated with both traces and stable failures, a specification can consist of both traces and failures models. Safety specifications are requirements on traces, which expect no event will occur at an inappropriate point. The failures model supports the expression of liveness specifications. Liveness is expressed in terms of a process's willingness to participate in events. It means that the process should be guaranteed to offer certain events at particular points, where any stable state reached by the process should not refuse those events.

6.2.1 Failures model

Traces model is less capable in identifying the guaranteed responses of a process. The following example shows two processes $P1$ and $P2$ have the same traces but different guaranteed behaviours. Process

$$P1 = a \to STOP \; \Box \; b \to STOP$$

is with an *external* choice and it has the following three possible traces – an empty trace, a trace with a single event a or a trace with a single event b. Process

$$P2 = a \to STOP \sqcap b \to STOP$$

has an *internal* choice and provides the same possibilities as process $P1$. $P1$ will be guaranteed to perform a if this is offered by the environment, and similarly for b. On the other hand, $P2$ may respond differently because of the *internal* choice. For instance, $P2$ may expect to interact on a but only b is offered by the environment. In other words, the actual response from $P2$ is unpredictable, and some of its possibilities may not be guaranteed. In general, processes with the *internal* and *external* choices have the same trace semantics and provide the same possibilities. However they exhibit different behaviours in some contexts. In such a situation, failures model allows a finer form of process observation which can help "make the necessary distinctions and provide the desired information about guaranteed process behaviour" ([47], page 174).

We briefly introduce the *CSP* failures model, which is used for the purpose of analysis in this research. The failures model allows us to reason about events that a process is ready to perform. It is not possible to judge whether a certain event will always be performed by a process as its environment may not allow it to do so. The approach taken in this model is to reason about processes in terms of events that they are not able to (or fail to) perform. A failure (tr,X) of a process P is the set of all events X which P would refuse after performing the events in the sequence tr. The set of all possible failures of P is written as $\mathit{failures}\,[\![P]\!]$. For example, for $a \rightarrow P$ there are two possibilities. First, if a has not occurred then it has performed an empty trace $\langle\rangle$ and is able to refuse any event other than a. Second, event a has occurred in which case the rest of the failures are those of P. More formally,

$$\begin{array}{rcl} \mathit{failures}\,[\![a \rightarrow P]\!] & = & \{(\langle\rangle, X) \mid a \notin X\} \\ & \cup & \{(\langle a \rangle \frown tr, X) \mid (tr, X) \in \mathit{failures}\,[\![P]\!]\} \end{array}$$

where $x \frown y$ denotes appending x with y.

6.2.2 Failures refinement

Process A is said to be failure refined by process B which is written as $A \sqsubseteq_F B$ if all failures of A are also all the failures of B

$$\mathit{failures}\,[\![B]\!] \subseteq \mathit{failures}\,[\![A]\!]$$

For example, we have two processes P and Q. Process P performs as

$$P = a \rightarrow b \rightarrow STOP$$

while

$$Q = a \rightarrow STOP$$

All possible failures of P can be represented as

$$\mathit{failures}\,[\![P]\!] = (\langle\rangle, \{a, b\}) \cup (\langle a\rangle, \{b\}) \cup (\langle a, b\rangle, \{\})$$

whereas all failures of Q are

$$\mathit{failures}\,[\![Q]\!] = (\langle\rangle, \{a\}) \cup (\langle a\rangle, \{\})$$

If we treat process Q as a specification of process P and want to prove P meets such a specification, then we need to either extend process Q so that it allows the extra events of P, or to abstract these extra events from P. The latter solution is preferred normally because of the efficiency when running model-checking tools. Therefore we hide the extra event b from P and its possible failures become

$$\mathit{failures}\,[\![P \setminus \{b\}]\!] = (\langle\rangle, \{a\}) \cup (\langle a\rangle, \{\})$$

Now all failures of Q are also all failures of P after b is hidden which can be written as

$$Q \sqsubseteq_F P \setminus \{b\}$$

In other words, we proved P is failure refined by Q. It can be written as

$$\mathit{failures}\,[\![P]\!] \subseteq \mathit{failures}\,[\![Q]\!]$$

6.2.3 Model-checking using *FDR*

We apply the model-checking technique in our model verification. Model checking is an automatic technique for verifying finite state concurrent systems. Compared with other verification techniques based on automated theorem proving, the most important advantage of model-checking is that this procedure is highly automatic [14].

Within this procedure, a model is represented at an abstract level and the specification is provided to be checked. If the model checker terminates with the answer *true*, it indicates that the model satisfies the specification. However if the model checker give a failed execution, then it shows why the formula is not satisfied. The failed executions during the model-checking procedure are particularly important as it helps finding modelling errors in a system [14].

In this research, the used model checker is a reliable *CSP* model-checking tool named *FDR*. *FDR* stands for Failures Divergences Refinement checker. It was the first tool to utilize the machine-readable dialect of *CSP* [45], and it is designed for automatic analysis of (untimed) *CSP* processes. Its main operation is checking whether or not one *CSP* process refines another. This provides a powerful analysis mechanism, since many important questions about processes can be expressed in terms of refinements. It also permits analysis for particular common properties, such as *deadlock*, *divergence* (livelock), and *determinism*.

We also used another model analysis tool *ProBe* occasionally in this research. *ProBe* is a companion tool for *FDR* [45]. It stands for Process Behaviour Explorer. This tool interprets and animates *CSP* process descriptions, allowing us to interact with a process and thus explore its behaviour patterns. It allows us to synchronize on events, to observe the available options at each stage, to backtrack, and to watch the trace being constructed as the process is executed. However, this tool is not suitable for formal analysis. It is mainly used for us to get a better understanding of *CSP* process descriptions.

6.3 Verifying the formal KT model

Within our model verification process, we first use *FDR* to analyse our KT model and specify the desired properties of the system. Then we check if our model is free from deadlocks and livelocks.

The KT model developed in this study is to present KT in detail so that people could use it to understand and further analyse particular KT problems in different environments. The verification of our model relies on *CSP* failures model. By checking when our model fails to perform KT, we could verify what our KT model is guaranteed to do rather than what it may do. A refinement check on the respective transfer system can be performed using *FDR* with respect to individual specifications. The specifications we chosen to verify our model are mainly concerned with people's changing roles and their options at several decision points during transfer. As explained in Section 2.5, the main observable aspect of KT is reflected as a series of actions performed by various transfer roles. In an idea KT system no role change would be disrupted so that any KT could proceed smoothly. To check if this is the case we perform model-checking on every role change. We identify the clear path of evolution of one role to another (e.g. a *Needs Recipient* becomes a *Knowledge Provider*) to define the related specifications. If a check fails, we would have a counter case showing how this role change has been disrupted so that a flaw of our KT model could be identified. In a similar way we also check if our model is guaranteed to provide people options at various decision points (e.g. a *Seeker*'s alternative choices when his first seeking attempt has failed).

We consider the following ten specifications that our KT system should have. Events in a system's process that do not appear in a specification are explicitly hidden allowing the model checker to observe only events common to both processes. Results of running these model checks using *FDR* are shown in Figure 6.1.

6.3.1 Activating KT system

First we are concerned with people's options when they decide to participate in KT and allow the KT system to be activated. We specify a process to model a person's available choices in this situation. He can choose to perform one of the four following events, *r_sent*, *search_repository*, *k_offered* and *add_to_repository*. This specification can be written as *SPEC*1 below.

Activating KT System

process

$$SPEC1 = r\_sent \rightarrow SPEC1 \;\Box\; search\_repository \rightarrow SPEC1 \;\Box\; k\_offered \rightarrow SPEC1 \;\Box\; add\_to\_repository \rightarrow SPEC1$$

The *FDR* tool allows us to check whether a system always provides a choice of the four initial events.

assert

$$SPEC1 \sqsubseteq_F KTMODEL \setminus \{s\_succeed, s\_failed, provide\_upon\_request, k\_retrieved, k\_provided, k\_relevant, k\_irrelevant, same\_need, new\_need, k\_applied, k\_discarded, passon\_request, n\_raised, r\_noted, r\_ignored, r\_responded, n\_transferred, r\_discarded, r\_retained\}$$

The system *KTMODEL* satisfies the refinement check. This means that every time a person is given a choice of event among *r_sent*, *search_reposiotory*, *k_offered*, and *add_to_repository*. Their action here activates the whole KT system. Formally, the *KTMODEL* never refuses any of the above four events. Thus *SPEC*1 is failure refined by *KTMODEL*

$$failures\,[\![KTMODEL]\!] \subseteq failures\,[\![SPEC1]\!]$$

The analysis result of this property is shown as *SPEC*1 in Figure 6.1.

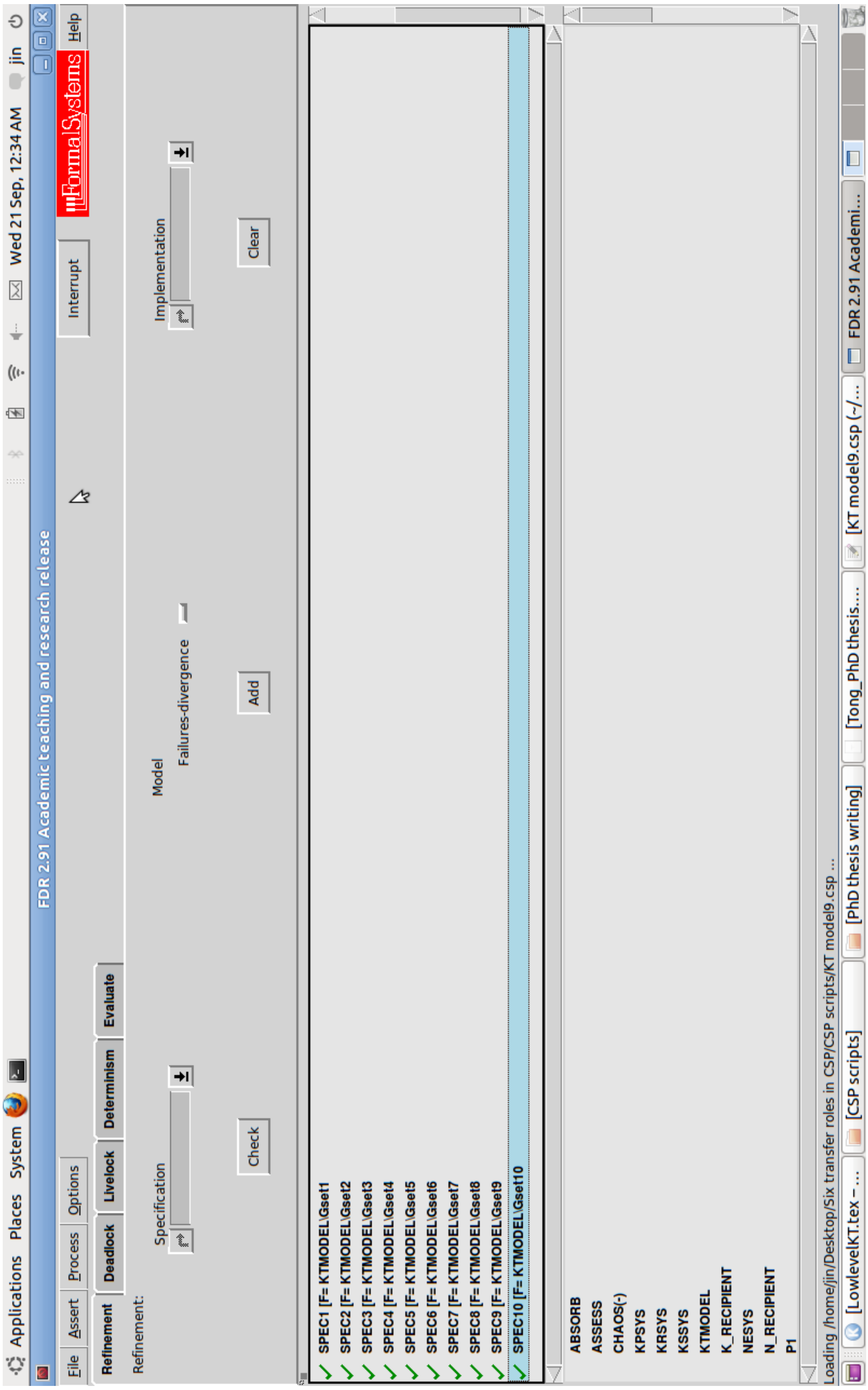


Figure 6.1: A screenshot of ten model refinement checks using *FDR*

6.3.2 A seeker becoming a knowledge recipient

This property is to check whether a *Knowledge Seeker* can become a *Knowledge Recipient* when necessary. Once a *Seeker* successfully found knowledge from a *Knowledge Repository* or received a response from others, he then becomes a *Knowledge Recipient*. Meanwhile, evaluating if the received knowledge is relevant or irrelevant to his knowledge needs should be a part of a *Knowledge Recipient*' actions. Therefore, a choice between events *k_relevant* and *k_irrelevant* should always be given after a *Seeker*'s successful attempt (either event *s_succeed* or event *r_responded*). Since a person can be offered knowledge without any seeking attempt, both events *k_relevant* and *k_irrelevant* can also be performed without the other two events happening. This specification can be written as $SPEC2$ below.

Seeker Becoming Knowledge Recipient

process

$$SPEC2 = s\_succeed \rightarrow (SPEC2 \,\square\, P1) \,\square\, r\_responded \rightarrow (SPEC2 \,\square\, P1) \,\square\, P1$$

process

$$P1 = k\_relevant \rightarrow SPEC2 \,\square\, k\_irrelevant \rightarrow SPEC2$$

assert

$$\begin{aligned} SPEC2 \sqsubseteq_F KTMODEL \setminus \{ & r\_sent, search\_repository, add\_to\_repository, \\ & s\_failed, k\_retrieved, k\_provided, k\_offered, \\ & same\_need, new\_need, n\_raised, r\_noted, \\ & r\_ignored, provide\_upon\_request, passon\_request, \\ & n\_transferred, r\_discarded, r\_retained, \\ & k\_applied, k\_discarded, no\_response\} \end{aligned}$$

The system *KTMODEL* satisfies the refinement check. This means that a person is always given a choice between events *k_relevant* and *k_irrelevant* after a successful seeking attempt. Formally, the *KTMODEL* never refuses any of the above two events after performing either event *s_succeed* or event *r_responded*. This implies that a *Seeker* can become a *Knowledge Recipient* when necessary. Thus $SPEC2$ is failure refined by *KTMODEL*

$$failures\,[\![KTMODEL]\!] \subseteq failures\,[\![SPEC2]\!]$$

This property is checked as $SPEC2$ using *FDR* (shown in Figure 6.1).

6.3.3 A needs recipient becoming a seeker

This property checks if a *Needs Recipient* can become a *Knowledge Seeker* when necessary. It is concerned with two events – *n_raised* from *N_RECIPIENT* and *search_repository* from *SEEKER*. When a *Needs Recipient* becomes aware of a new knowledge need (*n_raised*), he then becomes a *Seeker* and should be given the event *search_repository* as a seeking option. Meanwhile, both of the above two events can be performed recursively because both *SEEKER* and *N_RECIPIENT* are recursive processes. In order to avoid confusion, we hide the other seeking event *r_sent* in this property as it is also a synchronised event showing when process *N_RECIPIENT* is activated by *SEEKER* process. This specification can be written as $SPEC3$ below.

Needs Recipient Becoming Seeker

process

$$\begin{aligned} SPEC3 = {} & n\_raised \rightarrow search\_repository \rightarrow SPEC3 \\ & \Box\; n\_raised \rightarrow SPEC3 \\ & \Box\; r\_sent \rightarrow SPEC3 \end{aligned}$$

assert

$$\begin{aligned} SPEC3 \sqsubseteq_F KTMODEL \setminus \{ & r\_sent, r\_responded, add\_to\_repository, \\ & s\_succeed, s\_failed, k\_offered, k\_retrieved, k\_provided, \\ & k\_relevant, k\_irrelevant, same\_need, new\_need, \\ & k\_applied, k\_discarded, r\_noted, r\_ignored, \\ & provide\_upon\_request, passon\_request, n\_transferred, \\ & r\_discarded, r\_retained, no\_response\} \end{aligned}$$

The system *KTMODEL* satisfies the refinement check. The system *KTMODEL* never refuses event *search_repository* after performing event *n_raised*. This implies that a *Needs Recipient* can become a *Seeker* when necessary. Thus $SPEC3$ is failure refined by *KTMODEL*

$$failures\,[\![KTMODEL]\!] \subseteq failures\,[\![SPEC3]\!]$$

This property is checked as $SPEC3$ using *FDR* (shown in Figure 6.1).

6.3.4 A needs recipient becoming a provider

This property checks if a *Needs Recipient* can become a *Knowledge Provider* when necessary. When a *Needs Recipient* acknowledges one request (r_noted) and decides to help by acting as a *Provider*, then he should be able to perform event $provide\_upon\_request$. Both the above two events can be recursive on their own. This specification can be written as $SPEC4$ below.

Needs Recipient Becoming Provider

process

$$
\begin{aligned}
SPEC4 = {} & r\_noted \rightarrow provide\_upon\_request \rightarrow SPEC4 \\
& \Box\ provide\_upon\_request \rightarrow SPEC4 \\
& \Box\ r\_noted \rightarrow SPEC4
\end{aligned}
$$

assert

$$
\begin{aligned}
SPEC4 \sqsubseteq_F KTMODEL \setminus \{ & search\_repository, r\_responded, add\_to\_repository, \\
& s\_succeed, s\_failed, k\_offered, k\_retrieved, \\
& k\_provided, k\_relevant, k\_irrelevant, same\_need, \\
& new\_need, k\_applied, k\_discarded, n\_raised, \\
& r\_sent, r\_ignored, passon\_request, n\_transferred, \\
& r\_discarded, r\_retained, no\_response\}
\end{aligned}
$$

The system *KTMODEL* satisfies the refinement check. The system *KTMODEL* never refuses event $provide\_upon\_request$ after performing event r_noted. This implies that a *Needs Recipient* can become a *Provider* when necessary. Thus $SPEC4$ is failure refined by *KTMODEL*

$$failures\ [\![KTMODEL]\!] \subseteq failures\ [\![SPEC4]\!]$$

This property is checked as $SPEC4$ using *FDR* (shown in Figure 6.1).

6.3.5 A needs recipient becoming a transmitter

This property is concerned if a *Needs Recipient* can become a *Needs Transmitter* when necessary. When a *Needs Recipient* notes one request (r_noted) and decides to help by acting as a *Transmitter*, then he should be able to perform the event $r\_retained$ as an optional action. This specification can be written as $SPEC5$ below.

Needs Recipient Becoming Transmitter

process

$$SPEC5 = r\_noted \rightarrow r\_retained \rightarrow SPEC5$$
$$\Box\ r\_noted \rightarrow SPEC5$$
$$\Box\ r\_retained \rightarrow SPEC5$$

assert

$$SPEC5 \sqsubseteq_F KTMODEL \setminus \{r\_responded, add\_to\_repository, s\_succeed, s\_failed, k\_offered, search\_repository, k\_retrieved, k\_provided, k\_relevant, k\_irrelevant, same\_need, new\_need, k\_applied, k\_discarded, n\_raised, r\_sent, r\_ignored, provide\_upon\_request, no\_response, passon\_request, n\_transferred, r\_discarded\}$$

The system *KTMODEL* satisfies the refinement check. A person is always given the option to retain the request after transferring it to other. Formally, the system *KTMODEL* never refuses event *r_retained* after performing event *r_noted*. This implies that a *Needs Recipient* can become a *Transmitter* when necessary. Thus *SPEC*5 is failure refined by *KTMODEL*

$$failures\ [\![KTMODEL]\!] \subseteq failures\ [\![SPEC5]\!]$$

This property is checked as *SPEC*5 using *FDR* (shown in Figure 6.1).

6.3.6 Unanswered knowledge request

In reality, a person not always receives responses from others after sending request for knowledge. We need to check if the system allows this case to happen. Event *no_response* should be given as an option once event *r_sent* is performed. This property can be written as *SPEC*6 below.

Unanswered Knowledge Request

process

$$SPEC6 = r\_sent \rightarrow no\_response \rightarrow SPEC6$$
$$\Box\ r\_sent \rightarrow SPEC6$$

assert

$$SPEC6 \sqsubseteq_F KTMODEL \setminus \{add\_to\_repository, s\_succeed, s\_failed, k\_offered, search\_repository, k\_retrieved, k\_relevant, k\_irrelevant, same\_need, new\_need, k\_applied, k\_discarded, n\_raised, r\_noted, r\_ignored, provide\_upon\_request, passon\_request, n\_transferred, r\_discarded, r\_retained, r\_responded, k\_provided\}$$

The system *KTMODEL* satisfies the refinement check. The system *KTMODEL* never refuses event *no_response* after performing event *r_sent*. This means the system allows a *Seeker* to get no response regarding to his knowledge requests. Thus *SPEC*6 is failure refined by *KTMODEL*

$$failures\,[\![KTMODEL]\!] \subseteq failures\,[\![SPEC6]\!]$$

This property is checked as *SPEC*6 using *FDR* (shown in Figure 6.1).

6.3.7 Follow-up knowledge seeking

A *Knowledge Seeker* should be allowed to start another seeking attempt while still waiting for a response after sending a knowledge request. His earlier knowledge request may or may not be responded. This property is designed to check if the system allows a second seeking attempt in this situation. For example, a *Seeker* can choose to search a *Knowledge Repository* (*search_repository*) if he does not hear anything back (*no_response*) regarding to his earlier request. This means a *Seeker* should always be given event *search_repository* as an option after performing event *no_response*. This specification can be written as *SPEC*7 below.

Follow Up Knowledge Seeking

process

$$SPEC7 = no\_response \rightarrow SPEC7 \;\Box\; search\_repository \rightarrow SPEC7$$

assert

$$SPEC7 \sqsubseteq_F KTMODEL \setminus \{r\_sent, r\_responded, add\_to\_repository, s\_succeed, s\_failed, k\_offered, k\_retrieved,$$

$k\_provided, k\_relevant, k\_irrelevant, same\_need,$
$new\_need, k\_applied, k\_discarded, n\_raised,$
$r\_noted, r\_ignored, provide\_upon\_request,$
$passon\_request, n\_transferred, r\_discarded, r\_retained\}$

The system *KTMODEL* satisfies the refinement check. The system *KTMODEL* never refuses event *search_repository* after performing event *no_response*. This means the system allows a *Seeker* to get no response regarding to his knowledge requests and to start a second seeking attempt by using repository. Thus *SPEC*7 is failure refined by *KTMODEL*

$$failures\,[\![KTMODEL]\!] \subseteq failures\,[\![SPEC7]\!]$$

This property is checked as *SPEC*7 using *FDR* (shown in Figure 6.1).

6.3.8 Receiving irrelevant knowledge

People sometimes ask for knowledge from others but then get responses irrelevant to his earlier requests. Although the received knowledge is irrelevant, it may be still valuable. It may be applied for addressing other knowledge needs. This property is designed to check if our KT model allows a situation like this to happen. It is mainly concerned with the sequence of events *k_irrelevant* and *k_applied*. A *Needs Recipient* should always be given the option to perform event *k_applied* after event *k_irrelevant*. Both of the two events can be recursive on their own. This property can be written as *SPEC*8 below.

Receiving Irrelevant Knowledge

process

$$SPEC8 = k\_irrelevant \rightarrow k\_applied \rightarrow SPEC8$$
$$\Box\ k\_irrelevant \rightarrow SPEC8$$
$$\Box\ k\_applied \rightarrow SPEC8$$

assert

$$SPEC8 \sqsubseteq_F KTMODEL \setminus \{r\_sent, r\_responded, add\_to\_repository,$$
$$s\_succeed, s\_failed, k\_offered, search\_repository,$$
$$k\_retrieved, k\_provided, k\_relevant, same\_need,$$
$$k\_discarded, n\_raised, r\_noted, r\_ignored,$$
$$provide\_upon\_request, passon\_request, n\_transferred,$$

$r\_discarded, r\_retained, no\_response, new\_need\}$

The system *KTMODEL* satisfies the refinement check. The system *KTMODEL* never refuses event *k_applied* after performing event *k_irrelevant*. This means the system allows a *Needs Recipient* to apply the received knowledge even when it is not relevant to his earlier knowledge requests. Thus *SPEC*8 is failure refined by *KTMODEL*

$$failures\,[\![KTMODEL]\!] \subseteq failures\,[\![SPEC8]\!]$$

This property is checked as *SPEC*8 using *FDR* (shown in Figure 6.1).

6.3.9 A transmitter to re-assess a knowledge request

This property is concerned with if a *Needs Transmitter* can retain a request to repeat *ASSESS* process even after he has passed it onto another *Needs Recipient*. The system should allow the following sequence of events to happen. A person should always be given the option to perform event *n_transferred* after event *r_noted*. Then he should always be allowed to perform event *r_retained* (after event *n_transferred*). Following event *r_retained*, he should always be given event *r_noted* again as an option to repeat the above sequence of events. Meanwhile, all of the above three events are recursive on their own. This property can be written as *SPEC*9 below.

Transmitter′s ressessment

process

$SPEC9 = r\_noted \rightarrow P4$

process

$P2 = n\_transferred \rightarrow P4$

process

$P3 = r\_retained \rightarrow (SPEC9 \;\Box\; P3)$

process

$P4 = P2 \;\Box\; P3 \;\Box\; SPEC9$

assert

$SPEC9 \sqsubseteq_F KTMODEL \setminus \{r\_sent, r\_responded, add\_to\_repository,$

$s\_succeed, s\_failed, k\_offered, search\_repository,$

$k\_retrieved, k\_provided, k\_relevant, k\_irrelevant,$
$same\_need, new\_need, k\_applied, k\_discarded,$
$n\_raised, r\_ignored, provide\_upon\_request,$
$passon\_request, r\_discarded, no\_response\}$

The system *KTMODEL* satisfies the refinement check. The system *KTMODEL* never refuses event *r_retained* after performing event *n_transferred*, which is always following event *r_noted*. After event *r_retained*, the system never refuses to perform event *r_noted* again. This means the system allows a *Transmitter* to retain a request to repeat *ASSESS* process even after he has passed it onto another *N_RECIPIENT*. Thus *SPEC*9 is failure refined by *KTMODEL*

$$failures\ [\![KTMODEL]\!] \subseteq failures\ [\![SPEC9]\!]$$

This property is checked as *SPEC*9 using *FDR* (shown in Figure 6.1).

6.3.10 Successful KT

The last property is concerned with situations when knowledge is successfully transferred between people. This implies that the following three occasions are allowed in our system for knowledge to be successfully transferred. The first one is when a *Seeker* successfully searched for knowledge from a *Repository* (by taking event *s_succeed*). The second occasion is when a *Seeker* got a response regarding to his earlier request for knowledge (*r_responded*). The final one is when a *Knowledge Recipient* got offered with new knowledge without asking for it (*k_offered*). As clarified in Chapter 4, we recognise a KT is complete and successful only when the transferred knowledge is absorbed and applied by the *Knowledge Recipient*. Therefore a person should always be given the event *k_applied* as an option after performing the above three events for the transfer to be successful. This specification can be written as *SPEC*10 below.

Successful KT

process

$$SPEC10 = s\_succeed \rightarrow P5 \Box r\_responded \rightarrow P5 \Box k\_offered \rightarrow P5$$

process

$$P5 = k\_applied \rightarrow SPEC10 \Box k\_applied \rightarrow P5 \Box SPEC10$$

assert

$$SPEC10 \sqsubseteq_F KTMODEL \setminus \{search\_repository, r\_sent, add\_to\_repository, s\_failed, k\_retrieved, k\_provided, same\_need, new\_need, k\_discarded, n\_raised, r\_noted, r\_ignored, provide\_upon\_request, passon\_request, n\_transferred, r\_discarded, r\_retained, k\_relevant, k\_irrelevant, no\_response\}$$

The system *KTMODEL* satisfies the refinement check. The system *KTMODEL* never refuses event *k_applied* after performing events *s_succeed*, *r_responded* and *k_offered*. Formally, *SPEC*10 is failure refined by *KTMODEL*, where

$$failures\,[\![KTMODEL]\!] \subseteq failures\,[\![SPEC10]\!]$$

This property is checked as *SPEC*10 using *FDR* (shown in Figure 6.1).

6.3.11 Deadlock and livelock freedom

We also use *FDR* to check the system for *deadlocks* and *livelocks*. *Deadlocks* arise when certain processes within a system are awaiting an interaction with other processes before they can continue their own events. This means that some of their events will never take place if related processes are not activated. Such a situation is undesirable as it ultimately halts the execution of a system. *Livelocks* arises when processes descend into an endless sequence of interaction among themselves, excluding any other processes and the external environment. This is particularly undesirable as it means the system gets into an endless cycle of execution with no further progress and possibly an unnecessary consumption of resources. Both problems arise not due to the design of individual processes but due to the way they are combined[47].

Using *FDR*, the KT model has been successfully checked for *deadlock* and *livelock* freedom. This means all processes can be activated in our model and people's KT attempts can always have definite results. This reflects on the feasibility of such a system where all processes will complete no matter what knowledge search results these processes bring (be it successful or failed).

However in reality a *Knowledge Seeker* may send a knowledge request and never get a response. This could be a *deadlock* in the transfer system. On the other hand, if everyone receiving a knowledge request chooses to act as a *Needs Transmitter* by passing it onto another person, the original *Knowledge Seeker* may never get a response. In this situation, the system descends into a *livelock* where the processes *TRANSMITTER* and *N_RECIPIENT* are endlessly repeated. In other words, the KT model presented here reflects an ideal transfer environment. Problematic transfer cases with *deadlock* or *livelock* will be further discussed in future studies.

6.4 Summary

A detailed verification of the formal KT model using *FDR* is described in this chapter. This formal model is verified against ten specifications with the *CSP* failures model. We chose failures model instead of traces model in such a verification simply because it is more appropriate to check what our model is guaranteed to do rather than what it may do. The included specifications represent required properties of the KT model and are selected with respect to the motivation of designing this model in our research.

The formal KT model is also successfully checked for deadlock and livelock freedom using *FDR*. This means that our KT model only presents KT in an ideal environment. However, problematic transfer cases with deadlock or livelock exist in reality. Further investigation into those cases is not included in this research and will be looked at in our future studies.

In reality, KT systems vary and often have deficiencies in many organisations. Their transfer problems can be analysed and demonstrated through model refinement checks against our formal KT model. Such an analysis of KT problems using our formal KT model will be discussed in the next chapter.

Chapter 7

Application to the real-world case study

7.1 Introduction

The formal KT model introduced in the previous chapters captures people's KT behaviours in an ideal scenario. In reality, many organisations experience different transfer problems and often have dysfunctional KT systems. We propose to analyse their transfer problems using a formal approach which analyses a problematic transfer system through *CSP* failures refinement checks against our formal KT model. This chapter aims to demonstrate this formal approach through a case study at a Chinese company named *Lotus*. It is organised as follows. In the next section, we briefly describe the *Lotus* case study. In the third section, we first use a simple *CSP* specification process (that is already failure refined by our formal KT model) to check *Lotus* KT systems in different situations. Then our formal KT model itself is used as a specification process in a failures refinement check to demonstrate another *Lotus* KT dysfunction within the same section. In the fourth section, we explain how to help *Lotus* to overcome its transfer barriers by introducing an agent process to compliment its current KT system. And then finally we reflect the limitations of our application of the formal approach in this research.

7.2 A case study at Lotus

Lotus is a recently founded mobile phone manufacturing company. It is established by a group of 15 active professionals (who worked for different companies previously) from the mobile phone industry in 2005. It designs and manufactures tailor-made mobile phones and other wireless terminal products for markets in China, South America and Europe. Their clients are brand manufactures, mobile phone distributors and small-medium sized wireless product operators.

Lotus is very representative of a typical small organisation in China. KM practices observed in this company reflect on the wider sector in the county. Due to several cultural barriers (such as fear of loosing face, a sense of modesty, hierarchy consciousness, competitiveness and a preference for face-to-face communication) unveiled within *Lotus*, KT was not as effective as expected. Without effective KT, *Lotus* is suffering from several problems which affects its business performance. For example, with an increasing competition in the current mobile phone market, *Lotus* has to shorten its product delivering time in order to win more business contracts. It was identified in the researcher's previous studies that time was often wasted during *Lotus* employees' daily work routines because of their ineffective KT. Improving their KT practice could help improve *Lotus*' business efficiency. More examples can be found in the researchers' previous studies ([60], [59]). *Lotus* managers were very keen to understand what their KT barriers were and how to overcome them. They were fully aware that their KT issue had to be addressed before they can start expanding the company and set a more ambitious business mission.

Lotus is selected to be the case here for applying the proposed formal approach because of the following two reasons. First, *Lotus* is a small but fully functional organisation. Its size determines that its KT problems can be identified easily, and it is still operated normally despite these KT problems. This means that examples taken from *Lotus* will be simple but valid enough to serves the purpose of applying our formal approach here. Second, we have already identified that the cultural issue was the major cause of *Lotus*' KT problems in previous studies [59]. Since many other organisations also experience KT barriers because of the cultural influence (as highlighted in Section 2.3.1), an application to *Lotus* could demonstrate the diagnostic value of our approach and benefit those organisations affected by similar problems.

The examples we took from *Lotus* include shy knowledge seekers, lack of knowledge repositories and unwilling knowledge providers.

7.2.1 Shy knowledge seekers

Some people were too shy to request knowledge from others directly in several occasions. Some senior *Lotus* employees were too embarrassed to ask for help from the junior ones. It was discovered that their knowledge seeking behaviours could be restricted because of their fear of loosing face. A shy *Knowledge Seeker* normally prefers to use external *Knowledge Repositories* to search for answers, such as using Google search engine. In some extreme cases, people use these external databases as their only resources to seek knowledge. A shy *Knowledge Seeker* often fails to access the required knowledge because of their limited resources. Their work efficiency could also be seriously affected because the repository search is too time consuming sometimes. In many occasions, the same piece of knowledge could be accessed much more easily if they chose to request it from others directly.

7.2.2 Lack of knowledge repositories

It was discovered that people prefer to keep their knowledge implicit and share it informally. Many *Lotus* employees believed that through face-to-face communications you are showing more respect to people who actually shared knowledge with you, so that you are building a trustworthy relationship with them and your future requests for help are more likely to be responded. Because of people's preference for face-to-face communications, the effort of establishing an organisational *Knowledge Repository* within *Lotus* has failed. Without a *Knowledge Repository* available in this KT system, both *Knowledge Seekers* and *Providers*' transfer options are restricted. Such a KT system is certainly less effective.

7.2.3 Unwilling knowledge providers

It was also reported that some *Lotus* employees never offer their knowledge to others spontaneously. They only share what they know when such knowledge is requested by others or as part of their job responsibilities (i.e. mentoring a new employee). A strong sense of competitiveness was not the only cause of this kind of behaviours. A sense of modesty and the hierarchy consciousness were also the contributors. An unwilling knowledge provider can critically slow down the KT processes within an organisation, as knowledge is only flowed after work problems have already arisen (normally when people are already in needs of certain knowledge).

7.3 Demonstrating *Lotus* KT dysfunctions

The above three transfer problems observed in *Lotus* can be analysed formally through *CSP* failures refinement checks. We can either use the desired properties of our formal KT model (*CSP* specifications that are already failure refined by *KTMODEL*) or use *KTMODEL* itself as a specification process to check *Lotus* KT systems in different situations. The main purpose is to demonstrate how transfer dysfunctions are caused by specific KT problems using this formal analysis approach. As an example, we first use $SPEC1$ (one of the key properties of our formal KT model presented in Section 6.3) to analyse problematic KT systems in two conditions – A KT system with a shy *Seeker* and a KT system without a *Repository*. Then we explain how to use our formal KT model as a specification process to check a KT system with an unwilling *Provider*.

7.3.1 A KT system with a shy seeker

Shy *Knowledge Seekers* only use *Knowledge Repositories* to search for knowledge. This means that requesting knowledge from others directly is not a shy *Seeker*'s activity. A shy *Knowledge Seeker* within *Lotus* can be defined as:

A Shy Knowledge Seeker

process

$$S\_SEEKER = search\_repository \rightarrow (s\_succeed \rightarrow k\_retrieved \rightarrow S\_SEEKER \;\square\; s\_failed \rightarrow S\_SEEKER)$$

The KT system with such a shy *Knowledge Seeker* can be defined as *FKTMODEL*1.

KT with a Shy Knowledge Seeker

process

$NESYS = N\_RECIPIENT \; [\![\{ passon\_request, n\_transferred, n\_retained \}]\!] \; TRANSMITTEER$

$KSSYS = REPOSITORY \; [\![\{ add\_to\_repository \}]\!] \; PROVIDER$

$KPSYS = KSSYS \; [\![\{ provided\_upon\_request \}]\!] \; NESYS$

$FKRSYS = K\_RECIPIENT \; [\![\{ k\_retrieved, k\_provided \}]\!] \; S\_SEEKER$

$FKTMODEL1 = KPSYS \; [\![\{ r\_responded, s\_succeed, k\_offered, r\_sent, n\_raised \}]\!] \; FKRSYS$

Now we check *FKTMODEL*1 with *SPEC*1.

Activating KT with a Shy Seeker

assert

$SPEC1 \sqsubseteq_F FKTMODEL1 \setminus \{ s\_succeed, s\_failed, r\_responded, k\_retrieved, k\_provided, k\_relevant, k\_irrelevant, same\_need, new\_need, k\_applied, k\_discarded, n\_raised, r\_noted, r\_ignored, n\_transferred, provide\_upon\_request, passon\_request, r\_discarded, r\_retained \}$

The system *FKTMODEL*1 does not satisfy the refinement check. The *FDR* tool demonstrates that the process *FKTMODEL*1 allows the possibility of the events *k_offered*, *add_to_repository* and *search_reposiotory*, but refuses the event *r_sent*. More formally,

$$(\langle\rangle, \{r\_sent\}) \in failures\,[\![FKYMODEL1]\!]$$

whereas

$$(\langle\rangle, \{r\_sent\}) \notin failures\,[\![SPEC1]\!]$$

hence *SPEC*1 is not failure refined by *FKTMODEL*1

$$failures\,[\![FKTMODEL1]\!] \nsubseteq failures\,[\![SPEC1]\!]$$

7.3.2 A KT system without a repository

Without a *Knowledge Repository*, both *Seekers* and *Poviders* behave differently from those in a normal KT system. *Seekers* do not search *Repositories* in this case, whereas *Providers* do not contribute knowledge into *Repositories*.

A Seeker and a Provider without a Repository

process

$$SEEKER\_R = r\_sent \rightarrow (r\_responded \rightarrow k\_provided \rightarrow SEEKER\_R \;\Box\; SEEKER\_R)$$

$$PROVIDER\_R = provide\_upon\_request \rightarrow PROVIDER\_R \;\Box\; k\_offered \rightarrow PROVIDER\_R$$

The KT system without a *Knowledge Repository* can be defined as $FKTMODEL2$.

KT without a Knowledge Repository

process

$$NESYS = N\_RECIPIENT \;[\![\{passon\_request, n\_transferred, n\_retained\}]\!]\; TRANSMITTEER$$

$$FKPSYS = PROVIDER\_R \;[\![\{provided\_upon\_request\}]\!]\; NESYS$$

$$FKRSYS = K\_RECIPIENT \;[\![\{k\_provided\}]\!]\; SEEKER\_R$$

$$FKTMODEL2 = FKPSYS \;[\![\{r\_responded, k\_offered, r\_sent, n\_raised\}]\!]\; FKRSYS$$

Now we check $FKTMODEL2$ with the process $SPEC1$.

Activating KT without a Repository

assert

$$SPEC1 \sqsubseteq_F \text{FKTMODEL2} \setminus \{s\_succeed, s\_failed, r\_responded, k\_retrieved, k\_provided, k\_relevant, k\_irrelevant, same\_need, new\_need, k\_applied, k\_discarded, n\_raised, r\_noted, r\_ignored, n\_transferred, provide\_upon\_request, passon\_request, r\_discarded, r\_retained\}$$

The system $FKTMODEL2$ does not satisfy the refinement check. The *FDR* tool demonstrates that the process $FKTMODEL2$ allows the possibility of the events *k_offered* and *r_sent*, but refuses to perform the events *add_to_repository* and

$search\_reposiotory$. More formally,

$$(\langle\rangle, \{add\_to\_repository, search\_reposiotory\}) \in failures\,[\![FKYMODEL2]\!]$$

whereas

$$(\langle\rangle, \{add\_to\_repository, search\_reposiotory\}) \notin failures\,[\![SPEC1]\!]$$

hence $SPEC1$ is not failure refined by $FKTMODEL2$

$$failures\,[\![FKTMODEL2]\!] \nsubseteq failures\,[\![SPEC1]\!]$$

7.3.3 A KT system with an unwilling provider

Since an unwilling *Provider* does not offer knowledge to others directly or contribute to the *Knowledge Repository* spontaneously, it can be defined as the following:

An Unwilling Knowledge Provider

process
$U\_PROVIDER = provide\_upon\_request \rightarrow U\_PROVIDER$

Because of limited actions of the unwilling *Provider*, a *Knowledge Repository* in this KT system becomes different from the general one. It can be written as:

Knowledge Repository Used by Unwilling Providers

process
$U\_REPOSITORY = s\_succeed \rightarrow REPOSITORY$

The KT system with an unwilling *Provider* can be defined as $FKTMODEL3$.

KT with an Unwilling Knowledge Provider

process

$$\begin{aligned}
NESYS &= N_RECIPIENT \;[\![\{passon_request, n_transferred, \\
&\qquad n_retained\}]\!]\; TRANSMITTEER \\
FKSSYS &= U_REPOSITORY \;[\![\,]\!] \\
&\qquad U_PROVIDER \\
KPSYS &= FKSSYS \;[\![\{provided_upon_request\}]\!]\; NESYS \\
KRSYS &= K_RECIPIENT \;[\![\{k_retrieved, k_provided\}]\!]\; SEEKER \\
FKTMODEL3 &= KPSYS \;[\![\{r_responded, s_succeed, k_offered, \\
&\qquad r_sent, n_raised\}]\!]\; KRSYS
\end{aligned}$$

Now we check *FKTMODEL*3 against our formal transfer model *KTMODEL*.

Analysing KT with an Unwilling Provider

assert
  KTMODEL$\sqsubseteq_F$FKTMODEL3

The system *FKTMODEL*3 does not satisfy the refinement check. The *FDR* tool demonstrates that *FKTMODEL*3 refuses the possibility of all events within *KTMODEL* with exception of *r_sent* and *search_reposiotory*. This is because the system (with an unwilling *Provider*) does not allow events *add_to_repository* and *k_offered* to take place, which results into later refusals of the consequential events. More formally, if the set of refused events is defined as *Rset*.

process

$$Rset = \{add\_to\_repository, s\_succeed, provide\_upon\_request, k\_offered, k\_retrieved, k\_provided, k\_relevant, k\_irrelevant, same\_need, new\_need, k\_applied, k\_discarded, n\_raised, r\_noted, r\_ignored, passon\_request, n\_transferred, r\_discarded, r\_retained, s\_failed, no\_response\}$$

then

$$(\langle\rangle, Rset) \in failures\ [\![FKYMODEL3]\!]$$

whereas

$$(\langle\rangle, Rset) \notin failures\ [\![KTMODEL]\!]$$

hence *KTMODEL* is not failure refined by *FKTMODEL*3

$$failures\ [\![FKTMODEL3]\!] \nsubseteq failures\ [\![KTMODEL]\!]$$

7.4 Improved KT using an agent process

The failures refinement check in Section 7.3.3 formally demonstrated how an unwilling *Knowledge Provider* causes transfer barriers in *Lotus*. The KT system in this case is dysfunctional because of the missing options to the *Knowledge*

Provider. In order to help overcome KT barriers caused by the unwilling *Provider*, we propose to provide the problematic KT system with the missing events (*k_offered* and *add_to_repository*) by introducing an agent *CSP* process (an extra transfer role). Since the current *Provider* is not willing to give away what he knows if it is not requested, then an agent named *Knowledge Extractor* may help. An *Extractor* can identify what kind of knowledge is needed, and then request it from these *Providers* on behalf of the people who need it. He also needs to distribute the requested knowledge throughout the organisation by adding it to the *Repository* and offering it to others directly. Such an agent role can be defined as process *EXTRACTOR*. An *Extractor*' activities of distributing retrieved knowledge could make up the part of functions that an unwilling *Provider* is not able to provide. These distributing actions are described in a sub-process *DISTRIBUTE*.

Knowledge Extractor

process
$$EXTRACTOR = identify\_needs \rightarrow k\_extracted \rightarrow DISTRIBUTE$$
process
$$DISTRIBUTE = k\_offered \rightarrow EXTRACTOR \ \square \ add\_to\_repository \rightarrow EXTRACTOR$$

With the assistance from the *Knowledge Extractor*, the unwilling *Knowledge Provider* is improved.

An improved Knowledge Provider

process
$$I\_PROVIDER = provide\_upon\_request \rightarrow I\_PROVIDER \ \square \ k\_extracted \rightarrow I\_PROVIDER$$

The *Knowledge Seeker* is also enhanced in this case because of the extra seeking option – expressing knowledge needs to a *Knowledge Extractor*.

An enhanced Knowledge Seeker

process
$$E\_SEEKER = E\_SEEK \ \square \ n\_raised \rightarrow E\_SEEK$$

process

$$
\begin{aligned}
E\_SEEK = {} & search\_repository \rightarrow \\
& \quad (s\_succeed \rightarrow k\_retrieved \rightarrow E\_SEEKER \\
& \quad \Box\ s\_failed \rightarrow E\_SEEKER) \\
& \Box\ r\_sent \rightarrow \\
& \quad (r\_responded \rightarrow k\_provided \rightarrow E\_SEEKER \\
& \quad \Box\ E\_SEEKER) \\
& \Box\ identify\_needs \rightarrow E\_SEEKER
\end{aligned}
$$

Process *EXTRACTOR* and process *I_PROVIDER* are synchronised on event *k_extracted*, while event *identify_needs* indicates the synchronisation between process *EXTRACTOR* and process *E_SEEKER*. Including those three processes, an improved KT system in this case can be defined as *IKTMODEL*.

An improved KT system

process

$$
\begin{aligned}
& NESYS = N\_RECIPIENT \ [\![\{passon\_request, n\_transferred, \\
& \qquad n\_retained\}]\!] \ TRANSMITTER \\
& IKSSYS1 = EXTRACTOR \ [\![\{k\_extracted\}]\!] \ I\_PROVIDER \\
& IKSSYS2 = REPOSITORY \ [\![\{add\_to\_repository\}]\!] \ IKSSYS1 \\
& KPSYS = IKSSYS2 \ [\![\{provided\_upon\_request\}]\!] \ NESYS \\
& IKRSYS = K\_RECIPIENT \ [\![\{k\_retrieved, k\_provided\}]\!] \\
& \qquad E\_SEEKER \\
& IKTMODEL = KPSYS \ [\![\{r\_responded, s\_succeed, k\_offered, \\
& \qquad r\_sent, n\_raised, identify\_needs\}]\!] \\
& \qquad IKRSYS
\end{aligned}
$$

Now we check this improved transfer system against our formal KT model (*KTMODEL*) to see if the added agent process does help *Lotus* overcome the transfer barrier identified earlier.

Analysing the improved KT system

assert

$$\text{KTMODEL} \sqsubseteq_F \text{IKTMODEL} \backslash \{identify\_needs, k\_extracted\}$$

The system *IKTMODEL* satisfies the refinement check. The system *IKTMODEL* never refuses all events that are also performed by *KTMODEL*. Formally, *KTMODEL* is failure refined by *IKTMODEL*, where

$$failures\,[\![IKTMODEL]\!] \subseteq failures\,[\![KTMODEL]\!]$$

This means the dysfunctional KT system caused by the unwilling *Provider* within *Lotus* can be improved by introducing a new role named *Knowledge Extractor* in the organisation.

7.5 Limitations of the application

The current analysis of KT problems is mainly carried out informally in the available KM literature. We intend to propose a formal approach for this purpose here. The application to *Lotus* mainly demonstrated how KT dysfunctions were formed due to particular transfer problems in a formal way. These transfer problems had already been identified previously through informal means (e.g. a case study). In our application, the KT analysis using the formal approach is consistent with previous informal case study results. We also suggested relevant improvements to problematic KT systems based on the formal analysis of individual transfer problems (e.g. to include an agent process to make up the functions that its current KT system is missing). However, the examples we took from the *Lotus* case seem too simple to fully demonstrate the formal diagnostic function of the proposed approach. Our application to a real-world problem would be stronger if we could apply the formal approach to a larger company (with more complicated KT problems) or to more than one case studies.

The proposed formal approach can potentially be used in identifying unknown KT problems. Different organisations often practice KT differently. Their KT practices can be reviewed and described as different KT systems using *CSP* notations. Since our formal KT model (*KTMODEL*) represents a KT system in a generic situation, it can be used as a specification process to check these KT systems to identify the problematic parts using a similar approach (*CSP* failures refinement checks). This allows us to identify where KT problems lie within an organisation, so that they could be addressed. However, identifying unknown KT problems using our approach was not explored in our application due to the size of *Lotus* and the simplicity of its KT problems. This can be further investigated when the application to a more complicated real-world case is carried out in our future studies.

7.6 Summary

The main contribution of this chapter is to demonstrate how the formal model developed in previous chapters can be applied to a real-world problem. We used simple examples taken from a case study at a Chinese company (*Lotus*) to describe how this approach can be used for formal analysis of KT problems. This company serves to be a good example of how some cultural issues can become obstacles to effective KT (and in turn affect business efficiency). Our application of the model in this chapter should lay down the foundations of a formal approach that can be transferred to the industry for analysing KT problems. The validity of our model in effectively capturing the problems of KT within real-world organisations is the essential goal here. Admittedly we have only been able to demonstrate for one organisation.

Lotus is a small organisation, and their problems are relatively easy to identify. Our demonstration of the formal model therefore is limited. We aspire to apply our work to more real-world case studies. This should also help to develop and further refine our formal model. Ultimately the diagnostic value of our approach would be realised. This work is ongoing beyond the period of this research study.

Chapter 8

Conclusion and Future Work

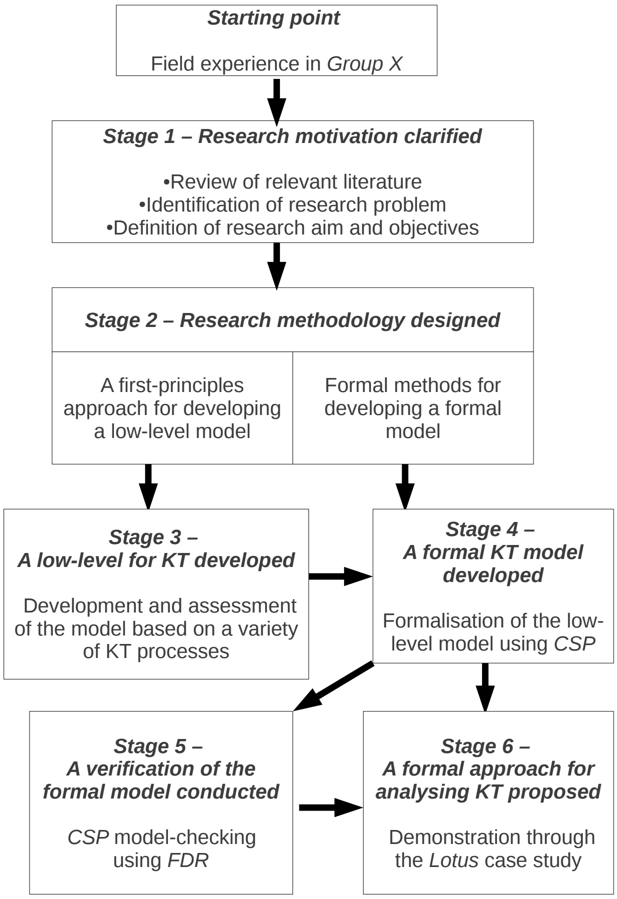


Figure 8.1: A path followed in conducting this research

8.1 Introduction

This chapter summarises the results of this research. First it describes the overview of the path taken in the research. Then it explains how the research questions were answered during the research. The chapter also discusses the contributions and limitations of this research. And finally it highlights areas for further research.

8.2 Research overview

The overall path followed in conducting this research is described as follows (shown as Figure 8.1).

In the first stage of research the review of relevant literature on KT was conducted, the research problem was identified and research purposes were also clarified. Then in the second stage, required research methods and techniques of this research were chosen – a first-principles approach for developing the low-level KT model and formal methods for developing the formal KT model. Following the methodological design, a low-level graphical model for KT was developed in the third stage of the research. This model was developed and assessed based on a collection of various KT cases from both the field and the researcher's own understanding. Next, the formal KT model was developed in stage four. This model was a formalised version of the low-level one using *CSP*. The fifth stage was concerned with verifying the formal KT model. The model verification mainly relied on the *CSP* model-checking technique with the assistance of an analysis tool named *FDR*. Finally in the sixth stage, we proposed a formal approach for analysing KT in different environments. This approach relies on the same model-checking technique (*CSP* failures refinement check) that was used for verifying the formal KT model. It was demonstrated through a case study in an organisation named *Lotus*.

This thesis was also structured to reflected the above research path. From Chapter 2 to 7, each stage with this research path was described in an individual chapter.

8.3 Research questions

All four research questions identified in Chapter 1 were answered during the research.

- *RQ1 – What do we understand by KT?*

  The answers to this question were covered in Chapter 2, 3 and 4. While discussing the concept of KT in Chapter 2, the researcher discovered that the term of KT was often misused, and there was no consistent definition of KT available in the existing literature. It was also highlighted that there was no agreed understanding of when knowledge is considered to be transferred successfully in previous studies. A need for a new definition of KT was clarified. Then the research context was defined, so that the focused aspect of KT was able to be identified in Chapter 3. A definition of KT was then given as part of the low-level graphical model presented in Chapter 4. This model investigated KT at a lower level, and viewed it as the overall process by which knowledge is transferred (and then applied) between people. Such a understanding emphasised the procedural aspect of KT as identified earlier, and also explained when knowledge is considered to be transferred in the context of this research.

- *RQ2 – Why are the current KT models not suitable to be used in analysing KT problems at an individual level?*

  This question was mainly answered in Chapter 2. Current KT understandings from the existing literature were reviewed, and the reasons why they were not suitable for analysing KT problems were explained. It was realized that the current KT models are mainly at a high-level. Related discussions of key KT issues and strategies are often decoupled from the context of KT and only can help us to analyse KT at an organisational level.

- *RQ3 – How does KT take place at an individual level?*

  This question was answered in Chapter 4, Chapter 5 and Chapter 6. Both the low-level graphical KT model and the formal KT model were developed to describe what exactly happened during KT. A detailed description of both models can be found in corresponding chapters. Because of the adoption of *CSP*, the understanding of KT in Chapter 5 is more formal and has a stronger emphasis on the interactions between different parties involved in KT. The formal KT model was verified in Chapter 6. Through *CSP* model-checking (failures refinment check), a set of required properties of a general KT system were identified. These properties helped validate our understanding of how KT takes place.

- *RQ4 – How can we analyse KT problems using a KT model?*

  Both Chapter 6 and Chapter 7 contribute in answering this question. We proposed a formal approach for analysing KT in Chapter 7. This approach largely depends on our formal KT model. In order to analyse specific transfer problems, we check a problematic KT system against our formal KT model (which is used as a specification in the failures refinement checks) to explore how transfer dysfunctions are formed. The set of specifications used in verifying the formal KT model in Chapter 6 can also be used to check a dysfuncational KT system. A cases study at *Lotus* was used as an example to demonstrate this formal approach. In Chapter 7, we also demonstrated how to help an organisation overcome an identified KT barrier by introducing an agent *CSP* process. An improved KT system was then proven to be working well by following the formal approach once again.

  In addition, this question was also partly answered in Chapter 4 where the low-level KT model was presented. It was explained that our low-level KT model can be used to represent various KT situations. This means we can review an organisation's KT practice in an informal way using this model. As part of the model assessment, different KT strategies (such as communities of practice) were demonstrated using the low-level KT model. Although we did not explicitly explain how this could help organisations in choosing appropriate KT strategies, it was highlighted that our low-level model provides a common framework for studying and comparing different KT strategies. This can help people understand their strengths and limitations, so that they could be used more effectively. More discussions on this matter could be found in one of the researcher's papers [58].

8.4 Research contributions

Three primary contributions to knowledge were delivered during this research, including a lower level understanding for KT, a formalised transfer model and a formal approach for analysing KT.

8.4.1 A lower level KT model

A low-level model for KT was developed in this research. What we mean by KT is clearly defined in this model. Unlike other existing understandings, our model avoids defining what kind of knowledge is transferred, but focuses on the procedural aspects of KT. It captures people's roles, decisions, critical actions and interactions with others during KT. It is presented graphically and consists of six modular units representing six key transfer roles. These roles are presented individually rather than in an integrative framework, because the model is designed to represent a wide range of KT situations. These transfer roles can be used as the basic structural elements to form dialogues between different parties involved in KT. An organisation's KT practice can be reviewed by presenting its major KT cases using this low-level model. This model also provides a common framework to study and compare different KT mechanisms. It allows these mechanisms such as knowledge maps and communities of practice to be demonstrated in detail by showing the steps and transactions that take place in the course of transfer. Therefore, suggesting appropriate KT strategies relying on this low-level model becomes easier.

8.4.2 A formalised KT model

A formal KT model was developed using *CSP* in this research. The dynamics and complexities in KT are well represented using the concurrency theory in *CSP*. Different parties involved in KT are treated as *CSP* processes. These processes execute in parallel and form a *CSP* system – the formal KT model.

Communications of events are modelled sequentially and concurrently along with introducing choice, composition and synchronisation. This model is formalised based on the graphical low-level model, and it provides a formal and precise framework in understanding how an organisation practices KT. This formal KT model was verified through *CSP* failures refinement checks with the assistance of a model-checking tool (*FDR*). A list of desired properties (specifications) of our formal KT model were also identified during the verification process. Such a verification provides a certain level of guarantee of the correctness of our formal KT model.

8.4.3 A formal approach for analysing KT

The model analysis of *CSP* failures refinement checking used for verifying our formal KT model was also proposed as a formal approach for analysing KT in different environments. This approach allows us to demonstrate how KT dysfunctions are caused by particular transfer problems (which have been identified previously by informal means) by checking problematic KT systems against our formal KT model or the desired properties (specifications) of the formal model. Relying on this approach, we can demonstrate how to help an organisation overcome its KT barriers by introducing agent *CSP* processes to compliment its current KT system. This approach could also allow us to diagnose unknown KT problems within an organisation. By checking an organisation's KT systems against our formal KT model (which can be seen as an ideal KT system), we can identify where KT problems lie within this organisation, so that they could be addressed.

8.4.4 Significance of the primary contributions

The significance of the above three primary contributions and the relationships between them are illustrated in Figure 8.2.

The above three contributions were delivered in a sequence during this research. The low-level KT model extended current understandings of KT from the literature and was developed first. Then the formal KT model was formalised on the basis of this low-level model. Finally, the formal approach proposed for analysing KT was emerged while developing and verifying the formal KT model.

On one hand, the understanding of KT became more formal as the research progressed. On the other hand, the research outcomes' applicability in analysing KT problems also enhanced significantly during the research. The formality of the understanding allows an organisation's KT practice to be reviewed in a formal and systematic manner, so that analysing specific transfer problems and suggesting relevant KT strategies could be more efficient.

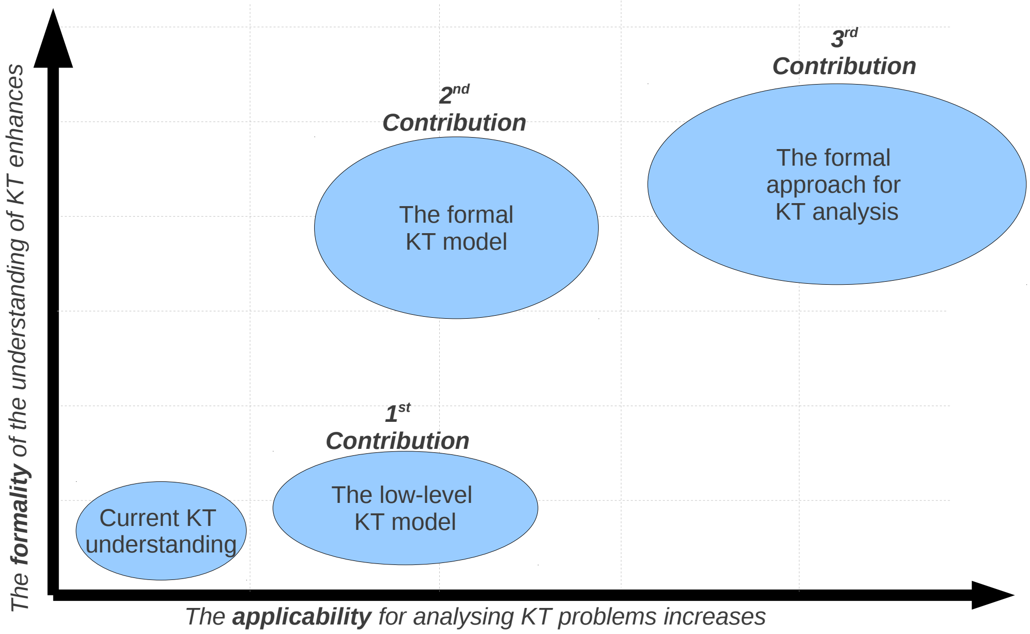


Figure 8.2: The significance of the three primary research contributions

- The formal approach for KT analysis has the highest level of both formality and applicability among the three primary contributions. However, without the successful delivery of the previous two contributions, it was not possible to propose such a formal approach.

- The low-level model broke down KT in detail and presented it at a lower level. This made it much more applicable in studying transfer problems compared to current KT understandings available from the literature, but its formality was not much higher than those from the literature.

- The formal transfer model was much more formal than the low-level graphical KT model because of the adoption of *CSP* notations. Both two versions of the KT model were developed to help people understand an organisation's KT practice, so their roles in analysing transfer problems were similar. However, the formal one's applicability was slightly enhanced because of its potential in helping identify and demonstrate specific transfer problems while using the formal KT analysis approach.

- Compared with the formal transfer model, the proposed formal approach significantly enhanced the applicability in regards to analysing KT problems. Its formality was slightly increased. This is because the refinement checks carried out in the formal approach helped verify the formal model and made it more reliable for use, although they both depended on the application of *CSP*.

8.4.5 Secondary contribution

In addition, the application of *CSP* in this research explored a new direction of studying human knowledge related processes in the KM domain. This research showed that the concurrency theory in *CSP* is suitable to represent the dynamics and complexities within KT and people's KT behaviours can be captured very well using *CSP* notations (e.g. choice operators, synchronisation and communications). Meanwhile, *CSP* model-checking through refinements allow KT to be analysed formally. Similar formal analysis could also be applied in studying other KM processes, such as knowledge innovation and knowledge exploitation. We hope our effort serves to inspire new ideas and approaches to the wider KM community.

8.5 Limitations of the research

We demonstrated how the formal model developed in previous chapters can be applied to a real-world problem in Section 7.5. The *Lotus* case we used in the application may be simple, but it serves to be a valid example for demonstrating how particular KT problems are formed within organisations. Of course, our application would be stronger if we could apply to a larger company with more complicated KT problems or to more than one organisations. In addition, we could not fully demonstrate the diagnostic value of our approach in this research. The examples we had did not allow us to demonstrate how a particular unknown KT problem can be analysed formally. This is due to the simplicity of the *Lotus* case. Since the *Lotus* application showed consistency with previous case study results, we hope that the progress we made in this research serves to be a good foundation for future researchers in the filed of applying formal methods in the KM domain.

The other limitation of this research is that the procedure of using our formal approach for KT analysis seems heavy. Because of the lack of sensibility of our formal KT model to deeper psychological and social factors in KT, the proposed formal approach should be used when the assistance from professional KM practitioners is available. For example, while demonstrating KT dysfunctions in *Lotus*, transfer problems were already identified by the researcher (as a KM expert) before the application of this formal approach. Although the proposed formal approach can be used to identify the problematic parts within an organisation's KT systems and detect where transfer problems lie, analysing why and how these problems exist is still a KM expert's job (e.g. exploring a *Knowledge Provider*'s reluctance in giving away knowledge). In addition, expertise of *CSP* formal modelling is also needed when applying our formal KT analysis approach. Such a heaviness could restrict the applicability of our formal approach for KT analysis. Our research should be seen as an early stage experiment of applying *CSP* formal modelling in studying KT. Although the proposed formal approach still has the space for improvements before it can be used in real business, it does not weaken the contributions of this research.

8.6 Areas of future research

The researcher has identified three areas that are worthy of further research. One is to identify unknown KT problems using the proposed formal approach for KT analysis. Another one is to diagnose KT deadlocks and livelocks within organisations in a similar way. The final one is to further apply *CSP* formal modelling in studying other knowledge processes in the KM domain.

8.6.1 Identification of KT problems

In Section 7.6, the potential of our formal approach in identifying unknown KT problems was highlighted. We could review an organisation's KT practice and describe it as various KT systems. Then we would need to use our formal KT model as a specification to check if it can failure refine these KT systems. If the model-checking failed, then we could identify the problematic parts within these KT systems and find out which transfer roles are dysfunctional. This procedure is slightly heavy, and we did not have enough time to demonstrate it through examples in this research. Therefore, we will collaborate with related KM expertise and experiment this procedure in our future studies.

8.6.2 Diagnosis of KT deadlocks and livelocks

As explained in Section 6.3.11, the formal KT model developed in this research is free from both deadlocks and livelocks according to the *FDR* model-checking results. This model only captures people's behaviours in a general KT situation. It can be seen as an ideal KT system. In reality, KT systems vary and often have deficiencies in many organisations. Some of their KT dysfunctions are normally caused by deadlocks or livelocks in their KT systems. In Chapter 7 where the *Lotus* case study was presented to demonstrate the formal approach for KT analysis, examples on both of the above two kinds of KT problems were not given due to limited research time. Therefore, it is worthy to address this issue in our future research.

8.6.3 Further application of *CSP* in KM

The application of a process algebra such as *CSP* was a novel idea experimented in this research. Results from this study showed that such an application has its potential in studying other human knowledge related processes, such as knowledge innovation process. KM is a multidisciplinary discipline by nature where researchers pick from different fields and areas. Most current KM studies deal with elusive and ambiguous concepts. The lack of consistent views and understandings on these concepts has become a common problem in this domain. A formalisation of definitions, concepts and models could be helpful in addressing this issue, as it can help set a foundational ground of the discipline, and provide a rigorous common language for both the researchers and practitioners. The application of *CSP* was to serve this purpose in this research. Therefore it is worthy to continue exploring it in future KM studies.

8.7 Concluding remarks

This thesis has reported a successful research project. It has made a significant contribution to the body of knowledge, not only in the KM domain but also the formal modelling area. Several research papers were also published to report the findings of the research (see the List of publications).

For individuals who participated in this research, it helped increase their awareness of the importance of KT and provide them with more options with respect to transferring knowledge in their daily work. By getting involved in this research, the organisations (mainly *Lotus* and *Group X*) also benefit from it. Their major transfer barriers were identified during the case studies. Our KT models allowed these barriers to be demonstrated and corresponding suggestions for improvement were made. This project also provided a great opportunity for the researcher to strength her research skills and consolidate her academic background, enabling her to conduct further research in relevant areas.

Finally the author concludes this project with commitment to continue in exploring relevant areas of research.

References

[1] Abrams, L.C., Cross, R., Lesser, E., Levin, D.Z.: Nurturing interpersonal trust in knowledge-sharing networks. Academy of Management Executive 17(4), 64–77 (2003)

[2] Argote, L., Ingram, P.: Knowledge transfer: a basis for competitive advantages in firms. Organizational Behavior and Human Decision Processes 82(1), 150–169 (2000)

[3] Austin, S., Parkin, G.: Formal methods: a survey (1993)

[4] Baeten, J.C.M.: A brief history of process algebra. Report CSR 04-02 (2004)

[5] Bergquist, M., Ljungberg, J.: The power of gifts: organizing social relationships in open source communities. Information Systems 11(4) (2001)

[6] Blackler, F.: Knowledge, knowledge work and organizations: an overview and interpretation. Organization Studies 16(6), 1021–1046 (1995)

[7] Borgatti, S.P., Cross, R.: A relational view of information seeking and learning in social networks. Management Science 49(4), 432–445 (2003)

[8] Boyd, J., Ragsdell, G., Oppenheim, C.: Knowledge transfer mechanisms: a case study from manufacturing. In: Martins, B., Remenyi, D. (eds.) Proceedings of the 8th European Conference on Knowledge Management (ECKM2007). pp. 139–145 (2007)

[9] Brown, J.S., Duguid, P.: Organizational learning and communities of practice. Organization Science 2(1), 40–57 (1991)

[10] Cerone, A., Connelly, S., Lindsay, P.A.: Formal analysis of human operator behavioural patterns in interactive surveillance systems. Software and System Modeling 7, 273–286 (2008)

[11] Cheung, S.L.: Reinventing the bazaar: a natural history of markets. The Economic Journal 113(488), 65–95 (2003)

[12] Chow, C.W., Deng, F.L., Ho, L.L.: The openness of knowledge sharing within organizations: a comparative study of the united states and the people's republic of china. Journal of Management Accounting Research 12, 65–95 (2000)

[13] Christensen, P.H.: Knowledge sharing: moving away from the obsession with best practices. Journal of Knowledge Management 11(1), 36–47 (2007)

[14] Clarke, E.M., Wing, J.M., et al.: Formal methods: state of the art and future directions. ACM Computing Surveys 28(4), 626–643 (1996)

[15] Davenport, T.H., Prusak, L.: Working Knowledge: How Organizations Manage What They Know. Harvard Business School Press, Boston, MA (1998)

[16] Davidson, C., Voss, P.: Knowledge Management: an Introduction to Creating Competitive Advantage from Intellectual Capital. Tandem Press, Auckland (2002)

[17] Disterer, G.: Individual and social barriers to knowledge transfer. In: IEEE 34th Hawaii International Conference on System Science (2001)

[18] Disterer, G.: Fostering knowledge sharing: why and how? In: Proceedings of the IADIS International Conference e-Science 2003 (2003)

[19] Easton, G., Araujo, L.: Market exchange, social structures and time. European Journal of Marketing 28(3), 72–84 (1994)

[20] Foss, N.J., Knudsen, C., Montgomery, C.A.: An exploration of common ground: Integrating evolutionary and strategic theories of the firm. In: Montgomery, C.A. (ed.) Resource-based and Evolutionary Theories of the Firm. pp. 1–17. Kluwer, London (1995)

[21] van Glabbeek, R.J.: Notes on the methodology of ccs and csp. Theoretical Computer Science 177, 329–349 (1997)

[22] Goh, S.C.: Managing effective knowledge transfer: an integrative framework and some practice implications. Journal of Knowledge Management 6(1), 22–30 (2002)

[23] Gold, A., Malhotra, A., Segars, A.: Knowledge management: an organisational capabilities perspective. Journal of Management Information Systems 18(1), 185–214 (2001)

[24] Gooderham, P.N.: Enhancing knowledge transfer in multinational corporations: a dynamic capabilities driven model. Knowledge Management Research and Practice 5, 34–43 (2006)

[25] Granovetter, M.S.: The strength of weak ties. American Journal of Sociology 78(6), 1360–1380 (1973)

[26] Hall, A.: Seven myths of formal methods. IEEE Software 7(5), 11–19 (1990)

[27] Hall, H.: Borrowed theory: applying exchange theories in information science research. Library and Information Science Research 25, 287–306 (2003)

[28] Hoare, T.: The ideal of program correctness: Third computer journal lecture. Computer Journal 50(3), 254–260 (2007)

[29] Holden, N.: Knowledge management: raising the specter of cross-cultural dimension. Knowledge and Process Management 8(3), 155–163 (2001)

[30] King, W.R.: Definition of knowledge transfer. In: G.Schwartz, D. (ed.) Encyclopedia of Knowledge Management. pp. 538–542. Idea Group Inc. (2006)

[31] Lee, J.: The impact of knowledge sharing, organizational capability and partnership quality on is outsourcing success. Information and Management 38(5), 323–335 (2001)

[32] Levin, D., Cross, R.: The strength of weak ties you can trust: the mediating role of trust in effective knowledge transfer. Management Science 50(11), 1477–1490 (2004)

[33] Mauss, M.: The Gift: the Form and Reason for Exchange in Archaic Societies. Routledge, London (1995; 1999)

[34] McGrath, J.E., Argote, L.: Group processes in organizational context. In: Hogg, M.A., Tindale, R.S. (eds.) handbook of social psychology: Group processes. pp. 603–726. Blackwell Publishing (2001)

[35] McMillan, J.: Reinventing the Bazaar: A Natural History of Markets. W. W. Norton, New York and London (2002)

[36] Nelson, R.: On the complexities and limitss of market organization. Review of International Political Economy 10(4), 697–710 (2003)

[37] Newell, S., Robertson, M., Scarbrough, H., Swan, J.: Managing Knowledge Work. PALGRAVE MACMILLAN, New York (2002)

[38] Nonaka, I.: A dynamic theory of organizational knowledge creation. Organization Science 5(1), 14–37 (1994)

[39] Patton, M.Q.: Qualitative Evaluation and Research Methods. Newbury Park, CA: Sage (1990)

[40] Polanyi, M.: Personal Knowledge: Towards a Post-Critical Philosophy. The University of Chicago Press, Chicago (1962)

[41] Porter, M.E.: Toward a dynamic theory of strategy. In: Rumelt, R.P., Schendel, D.E., Teece, D.J. (eds.) Fundamental Issues in Strategy: A Research Agenda. pp. 423–461. Harvard Business School Press, Boston, MA (1994)

[42] Raymond, E.S.: The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. O'Reilly and Associates (1999)

[43] Riege, A.: Actions to overcome knowledge transfer barriers in mncs. Journal of Knowledge Management 11(1), 48–67 (2007)

[44] Robson, C.: Real World Research: A Resource for Social Scientists and Practitioner-Researchers. Blackwell Publishing, 2nd edn. (2002)

[45] Roscoe, A.: The Theory and Practice of Concurrency. Prentice-Hall (Pearson) (2005)

[46] Ruggles, R.: The state of the notion: knowledge management in practice. California Management Review 40(3), 80–89 (1998)

[47] Schneider, S.A.: Concurrent and Real-time Systems: The CSP Approach. John Wiley and Sons, Ltd (2000)

[48] Schulz, M.: The uncertain relevance of newness: organizational learning and knowledge flows. The Academy of Management Journal 44(4), 661–681 (2001)

[49] Shaikh, S.A., Krishnan, P., Cerone, A.: A formal approach to human error recovery. In: The Pre-proceedings of the 2nd International Workshop on Formal Methods for Interactive Systems (FMIS 2007). pp. 101–135 (2007)

[50] Shannon, C.E., Weaver, W.: A Mathematical Model of Communication. University of Illinois Press, Urbana (1949)

[51] Singley, K., Anderson, J.R.: The Transfer of Cognitive Skill. Harvard University Press (1989)

[52] Spender, J.C.: Organisational knowledge, learning and memory: three concepts in search of a theory. Journal of Organisational Change and Management 9(1), 63–78 (1996)

[53] Spender, J.C.: Pluralist epistemology and the knowledge-based theory of the firm. Organization 5(2), 233–256 (1998)

[54] Strauss, A.L., Corbin, J.M.: Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Sage Publications, Inc., 2nd edn. (1998)

[55] Szulanski, G.: Exploring internal stickiness: impediments to the transfer of best practice within the firm. Strategic Management Journal pp. 27–43 (1996)

[56] Szulanski, G.: The process of knowledge transfer: a diachronic analysis of stickiness. Organizational Behavior and Human Decision Processes 82(1), 9–27 (2000)

[57] Thomas, M.: The industrial use of formal methods. Microprocessors and Microsystems 17(1), 31–36 (1992)

[58] Tong, J., Ayres, R.: Knowledge needs: uncharted area in knowledge management. In: Proceeding of the International Conference on Organisational Learning, Knowledge and Capabilities. Amsterdam, Netherlands (2009)

[59] Tong, J., Mitra, A.: Chinese cultural influences on knowledge management practice. Journal of Knowledge Management 13(2), 49–62 (2009)

[60] Tong, J., Shaikh, S.A.: Ict driven knowledge management in developing countries: a case study in a chinese organisation. In: Communications: Wireless in Developing Countries and Networks of the Future. IFIP Advances in Information and Communication Technology, vol. 327, pp. 60–71. Brisbane, Australia (2010)

[61] Watson, S., Hewett, K.: A multi-theoretical model of knowledge transfer in organisations: determinants of knowledge contribution and knowledge reuse. Journal of Management Studies 43(2), 141–173 (2006)

[62] Wenger, E.: Communities of Practice: Learning, Meaning and Identify. Cambridge University Press, Cambridge (1998)

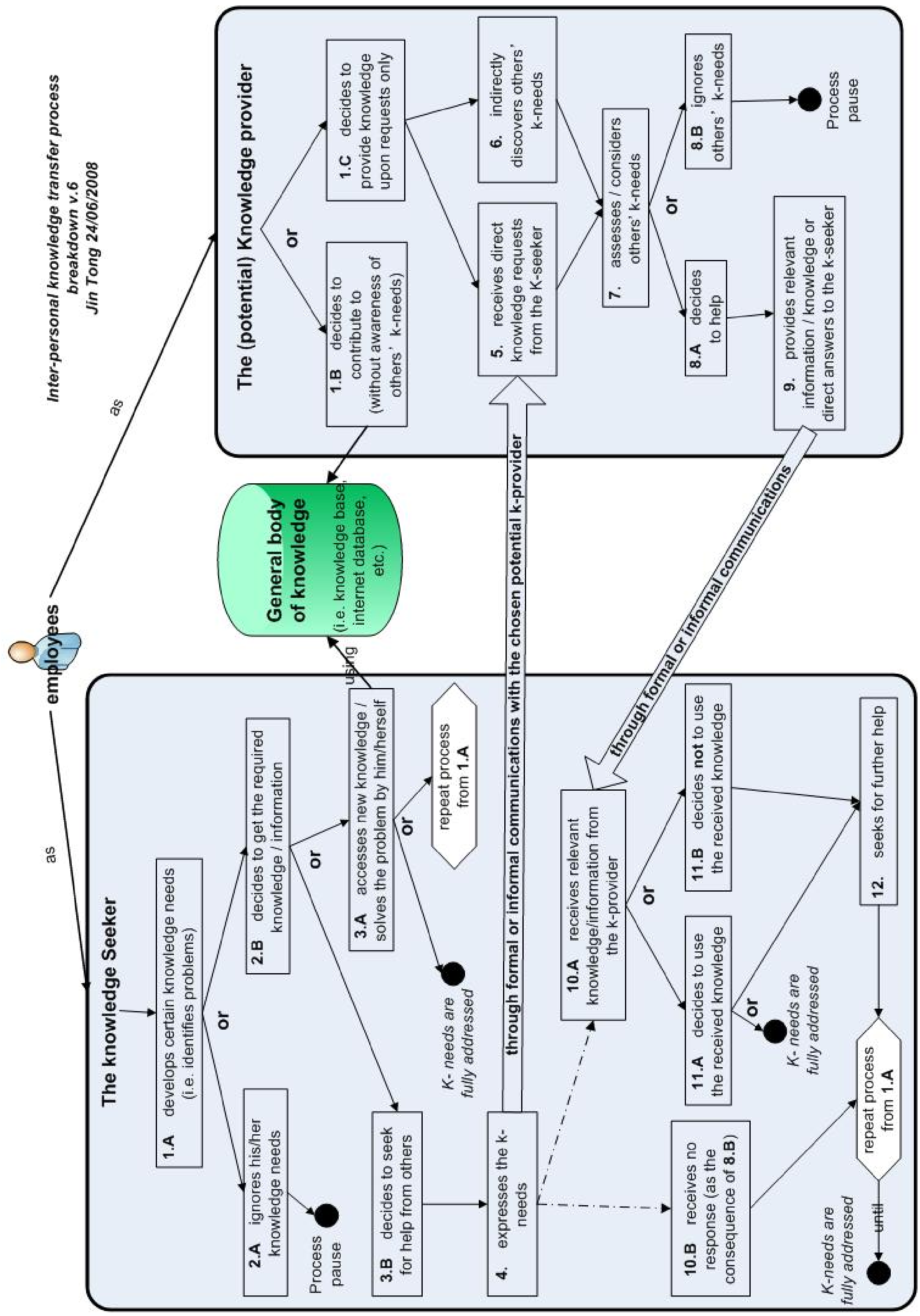
[63] Wenger, E., McDermott, R.A., Snyder, W.: Cultivating Communities of Practice: A guide to Managing Knowledge. Hardvard Business School Press (2002)

[64] Wenger, E., Snyder, W.: Communities of practice: the organizational frontier. Harvard Business Review January - February, 139–145 (2000)

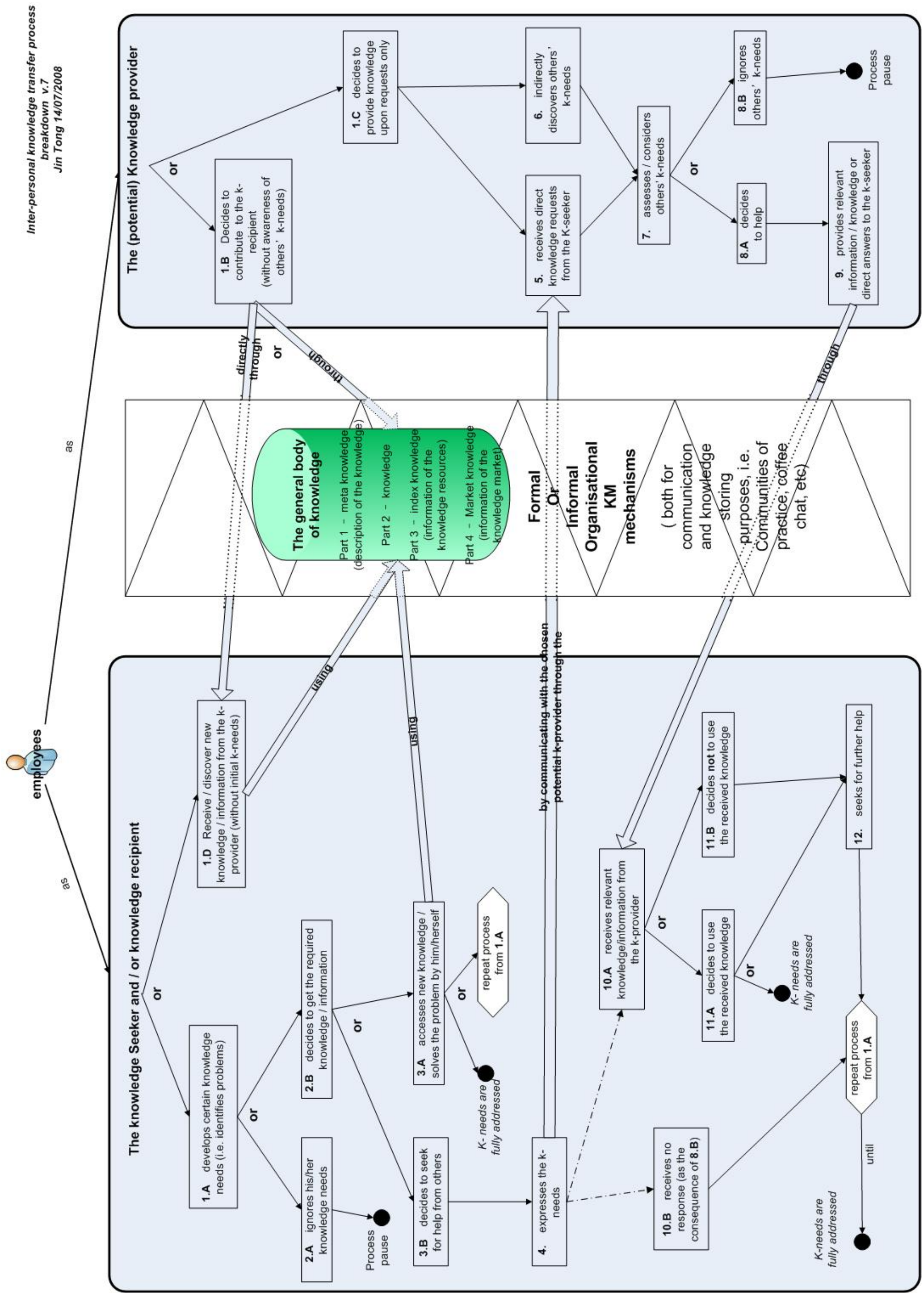
[65] Wing, J.M.: A specifier's introduction to formal methods. IEEE Computer 23(9), 8–24 (1990)

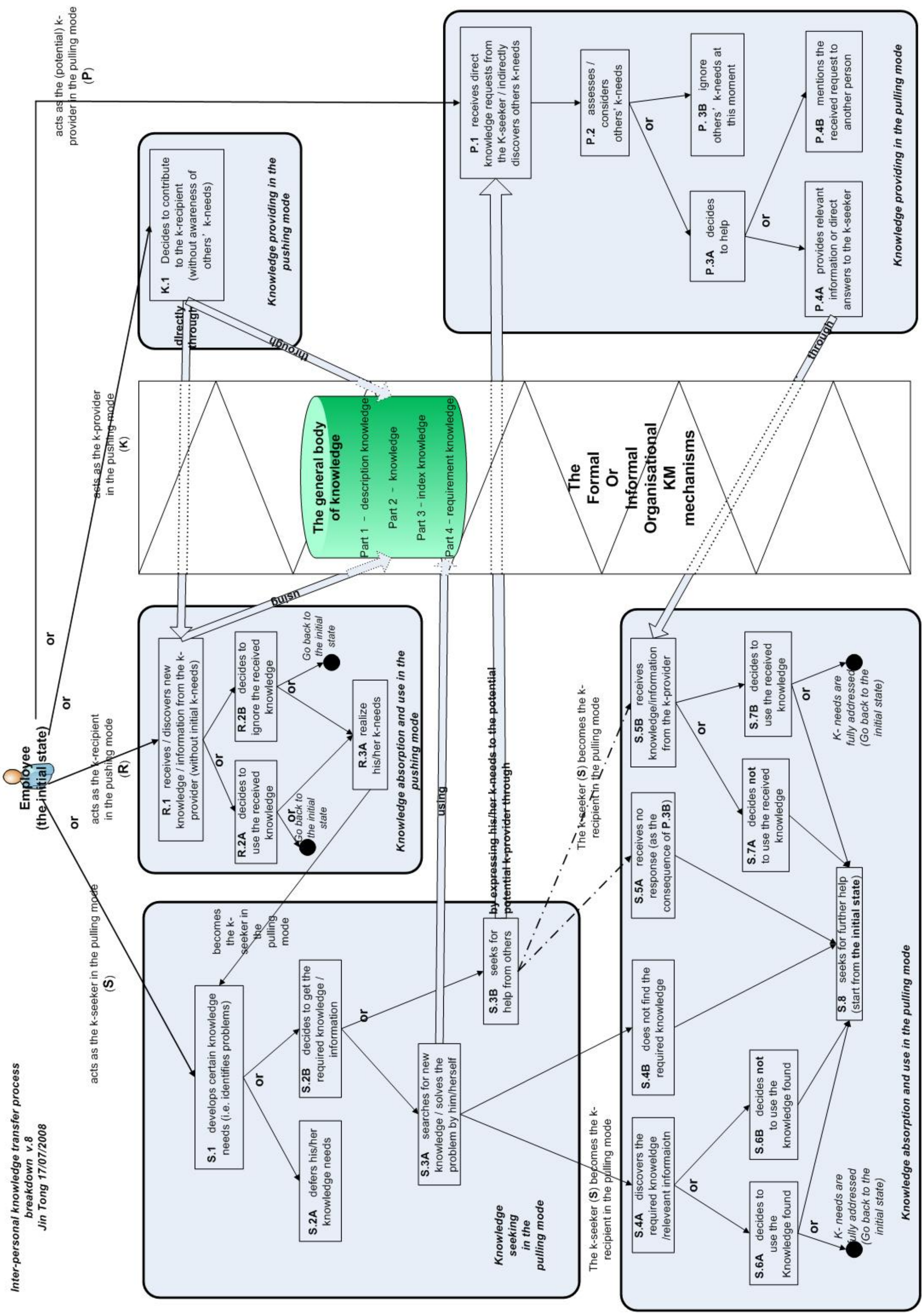
[66] Woodcock, J., Larsen, P.G., Bicarregui, J., Fitzgerald, J.: Formal methods: practice and experience. ACM Computing Surveys V(N), 1–40 (20YY)

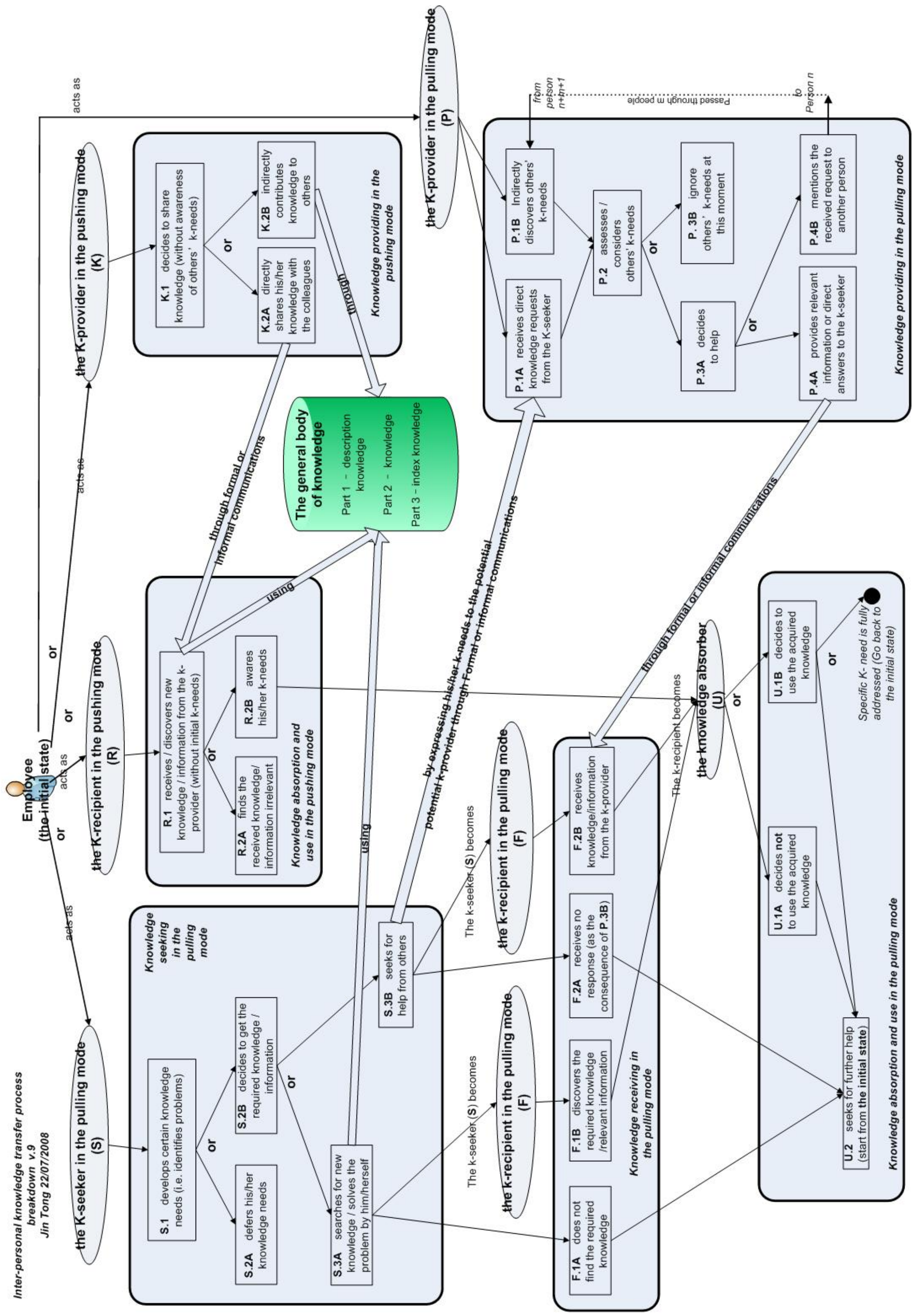
Appendix A – Evolution of the low-level KT model



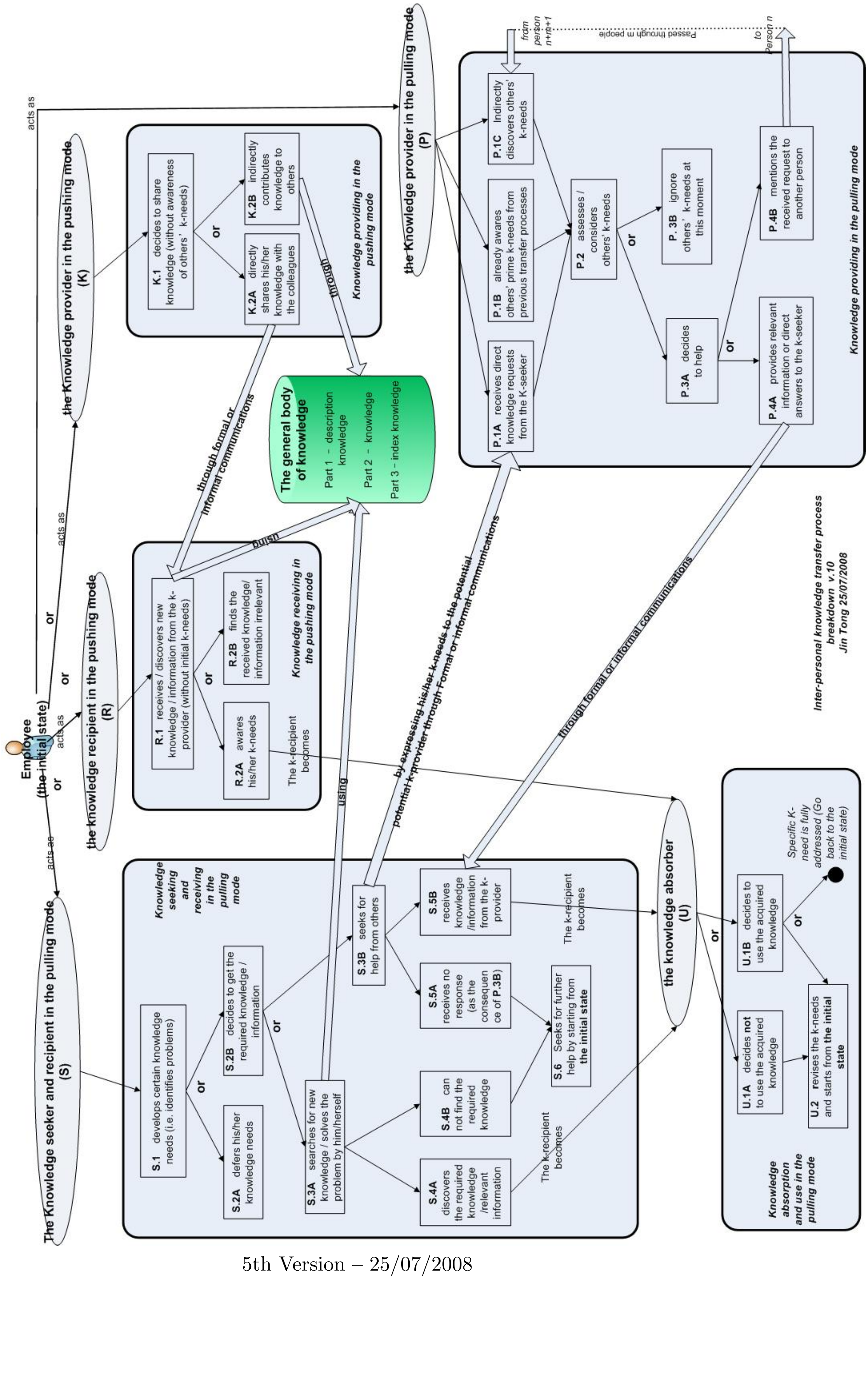
1st Version – 24/06/2008


Inter-personal knowledge transfer process breakdown v.7
Jin Tong 14/07/2008
employees
as
as
The (potential) Knowledge provider
or
1.B Decides to contribute to the k-recipient (without awareness of others' k-needs)
1.C decides to provide knowledge upon requests only
6. indirectly discovers others' k-needs
5. receives direct knowledge requests from the K-seeker
7. assesses / considers others' k-needs
or
8.B ignores others' k-needs
Process pause
8.A decides to help
9. provides relevant information / knowledge or direct answers to the k-seeker
directly through
or
through
through
The general body of knowledge
Part 1 – meta knowledge (description of the knowledge)
Part 2 – knowledge
Part 3 – index knowledge (information of the knowledge resources)
Part 4 – Market knowledge (information of the knowledge market)
Formal Or Informal Organisational KM mechanisms
(both for communication and knowledge storing purposes, i.e. Communities of practice, coffee chat, etc)
The knowledge Seeker and / or knowledge recipient
or
1.A develops certain knowledge needs (i.e. identifies problems)
1.D Receive / discover new knowledge / information from the k-provider (without initial k-needs)
using
or
2.A ignores his/her knowledge needs
Process pause
2.B decides to get the required knowledge / information
or
3.A accesses new knowledge / solves the problem by him/herself
using
or
repeat process from 1.A
K- needs are fully addressed
3.B decides to seek for help from others
4. expresses the k-needs
by communicating with the chosen potential k-provider through the
10.A receives relevant knowledge/information from the k-provider
or
10.B receives no response (as the consequence of 8.B)
11.A decides to use the received knowledge
11.B decides not to use the received knowledge
or
K- needs are fully addressed
12. seeks for further help
repeat process from 1.A
until
K- needs are fully addressed

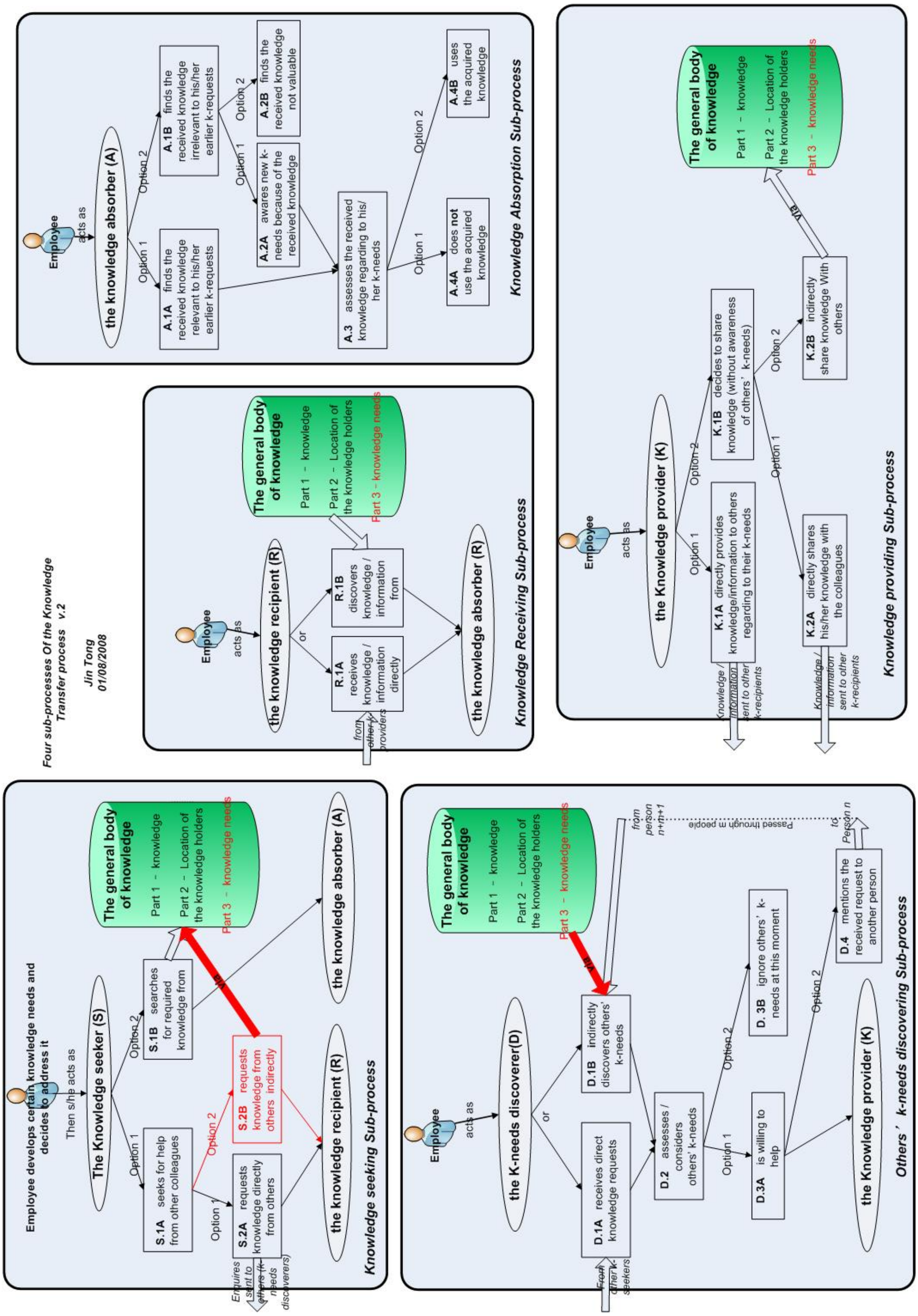

Inter-personal knowledge transfer process breakdown v.8
Jin Tong 17/07/2008
Employee (the initial state)
acts as the (potential) k-provider in the pulling mode (P)
acts as the k-provider in the pushing mode (K)
acts as the k-recipient in the pushing mode (R)
acts as the k-seeker in the pulling mode (S)
or
K.1 Decides to contribute to the k-recipient (without awareness of others' k-needs)
Knowledge providing in the pushing mode
directly
through
push
P.1 receives direct knowledge requests from the K-seeker / indirectly discovers others k-needs
P.2 assesses / considers others' k-needs
P. 3B ignore others' k-needs at this moment
P.3A decides to help
P.4A provides relevant information or direct answers to the k-seeker
P.4B mentions the received request to another person
Knowledge providing in the pulling mode
The general body of knowledge
Part 1 – description knowledge
Part 2 – knowledge
Part 3 - index knowledge
Part 4 – requirement knowledge
The Formal Or Informal Organisational KM mechanisms
R.1 receives / discovers new knowledge / information from the k-provider (without initial k-needs)
R.2A decides to use the received knowledge
R.2B decides to ignore the received knowledge
Go back to the initial state
R.3A realize his/her k-needs
Knowledge absorption and use in the pushing mode
becomes the k-seeker in the pulling mode
S.1 develops certain knowledge needs (i.e. identifies problems)
S.2A defers his/her knowledge needs
S.2B decides to get the required knowledge / information
S.3A searches for new knowledge / solves the problem by him/herself
S.3B seeks for help from others
using
by expressing his/her k-needs to the potential k-provider through
Knowledge seeking in the pulling mode
The k-seeker (S) becomes the k-recipient in the pulling mode
S.4A discovers the required knowledge /relevant information
S.4B does not find the required knowledge
S.5A receives no response (as the consequence of P.3B)
S.5B receives knowledge/information from the k-provider
S.6A decides to use the Knowledge found
S.6B decides not to use the knowledge found
S.7A decides not to use the received knowledge
S.7B decides to use the received knowledge
K- needs are fully addressed (Go back to the initial state)
S.8 seeks for further help (start from the initial state)
Knowledge absorption and use in the pulling mode



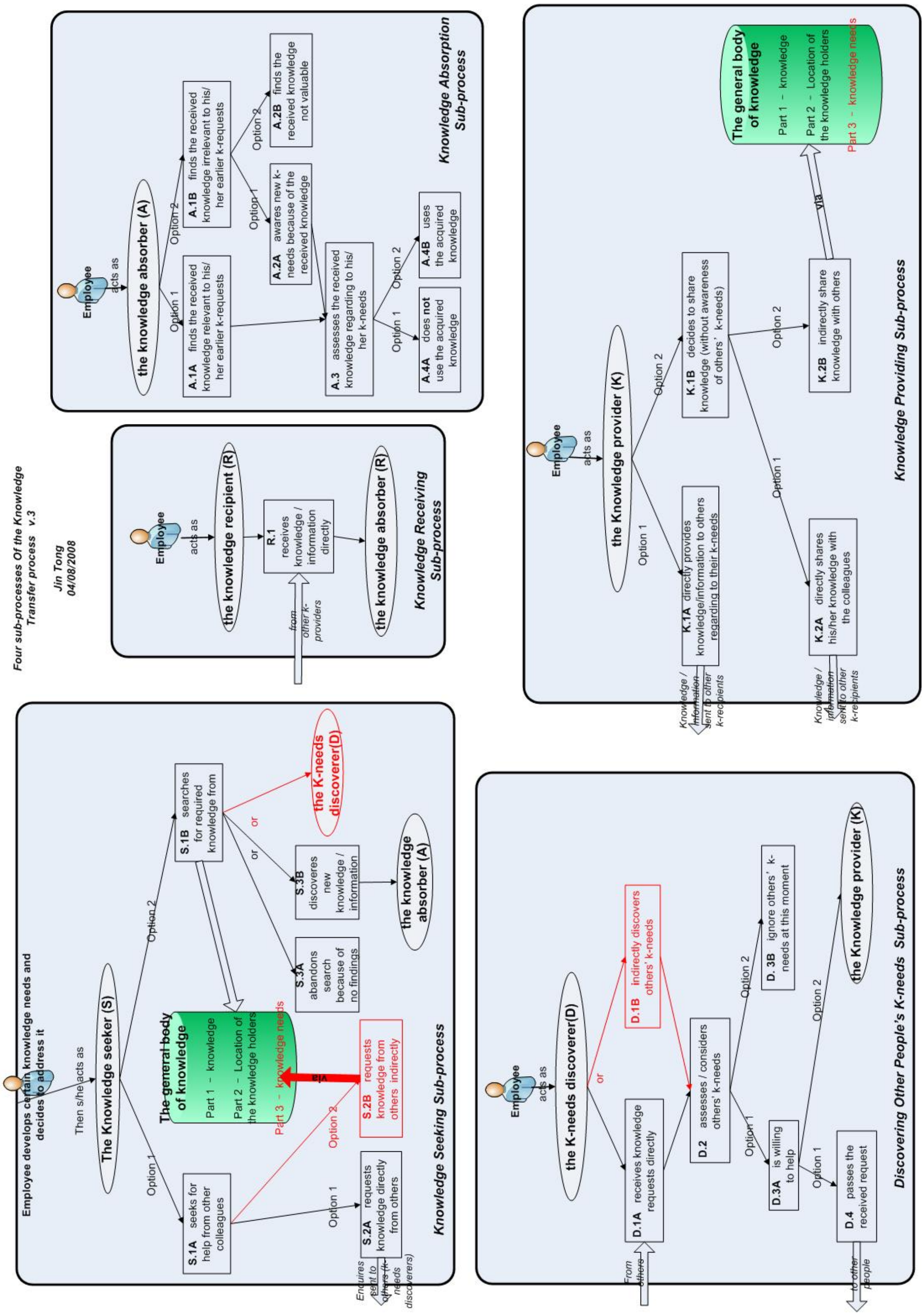
4th Version – 22/07/2008



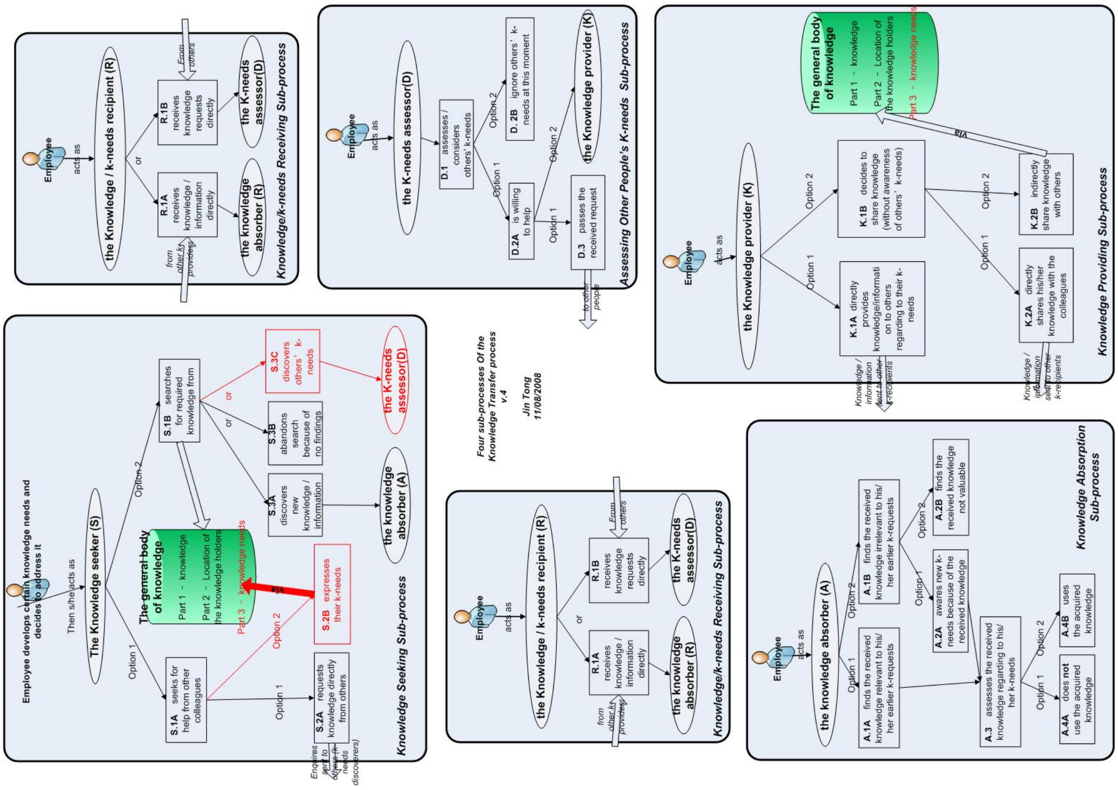
Employee (the initial state)
acts as
or
The Knowledge seeker and recipient in the pulling mode (S)
the knowledge recipient in the pushing mode (R)
the Knowledge provider in the pushing mode (K)
the Knowledge provider in the pulling mode (P)
the knowledge absorber (U)
K.1 decides to share knowledge (without awareness of others' k-needs)
K.2A directly shares his/her knowledge with the colleagues
K.2B indirectly contributes knowledge to others
Knowledge providing in the pushing mode
through
through formal or informal communications
push
The general body of knowledge
Part 1 – description knowledge
Part 2 – knowledge
Part 3 – index knowledge
R.1 receives / discovers new knowledge / information from the k-provider (without initial k-needs)
R.2A awares his/her k-needs
R.2B finds the received knowledge/ information irrelevant
The k-recipient becomes
Knowledge receiving in the pushing mode
P.1A receives direct knowledge requests from the K-seeker
P.1B already awares others' prime k-needs from previous transfer processes
P.1C Indirectly discovers others' k-needs
P.2 assesses / considers others' k-needs
P. 3B ignore others' k-needs at this moment
P.3A decides to help
P.4A provides relevant information or direct answers to the k-seeker
P.4B mentions the received request to another person
from person n+m+1
Passed through m people
to Person n
Knowledge providing in the pulling mode
Inter-personal knowledge transfer process breakdown v.10
Jin Tong 25/07/2008
using
by expressing his/her k-needs to the potential k-provider through Formal or informal communications
through formal or informal communications
Knowledge seeking and receiving in the pulling mode
S.1 develops certain knowledge needs (i.e. identifies problems)
S.2A defers his/her knowledge needs
S.2B decides to get the required knowledge / information
S.3A searches for new knowledge / solves the problem by him/herself
S.3B seeks for help from others
S.4A discovers the required knowledge /relevant information
S.4B can not find the required knowledge
S.5A receives no response (as the consequence of P.3B)
S.5B receives knowledge /information from the k-provider
S.6 Seeks for further help by starting from the initial state
The k-recipient becomes
U.1A decides not to use the acquired knowledge
U.1B decides to use the acquired knowledge
U.2 revises the k-needs and starts from the initial state
Specific K-need is fully addressed (Go back to the initial state)
Knowledge absorption and use in the pulling mode



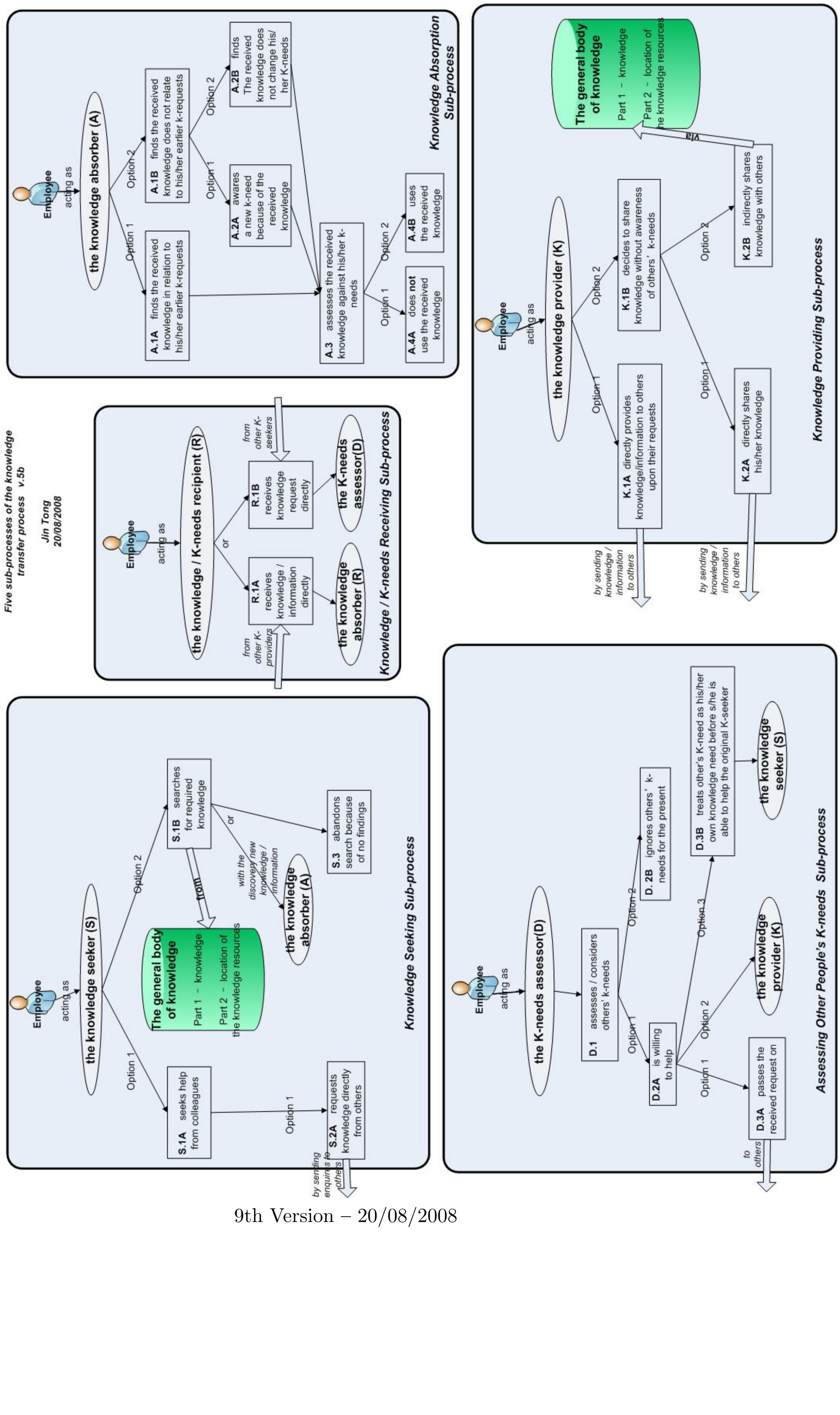
6th Version – 01/08/2008

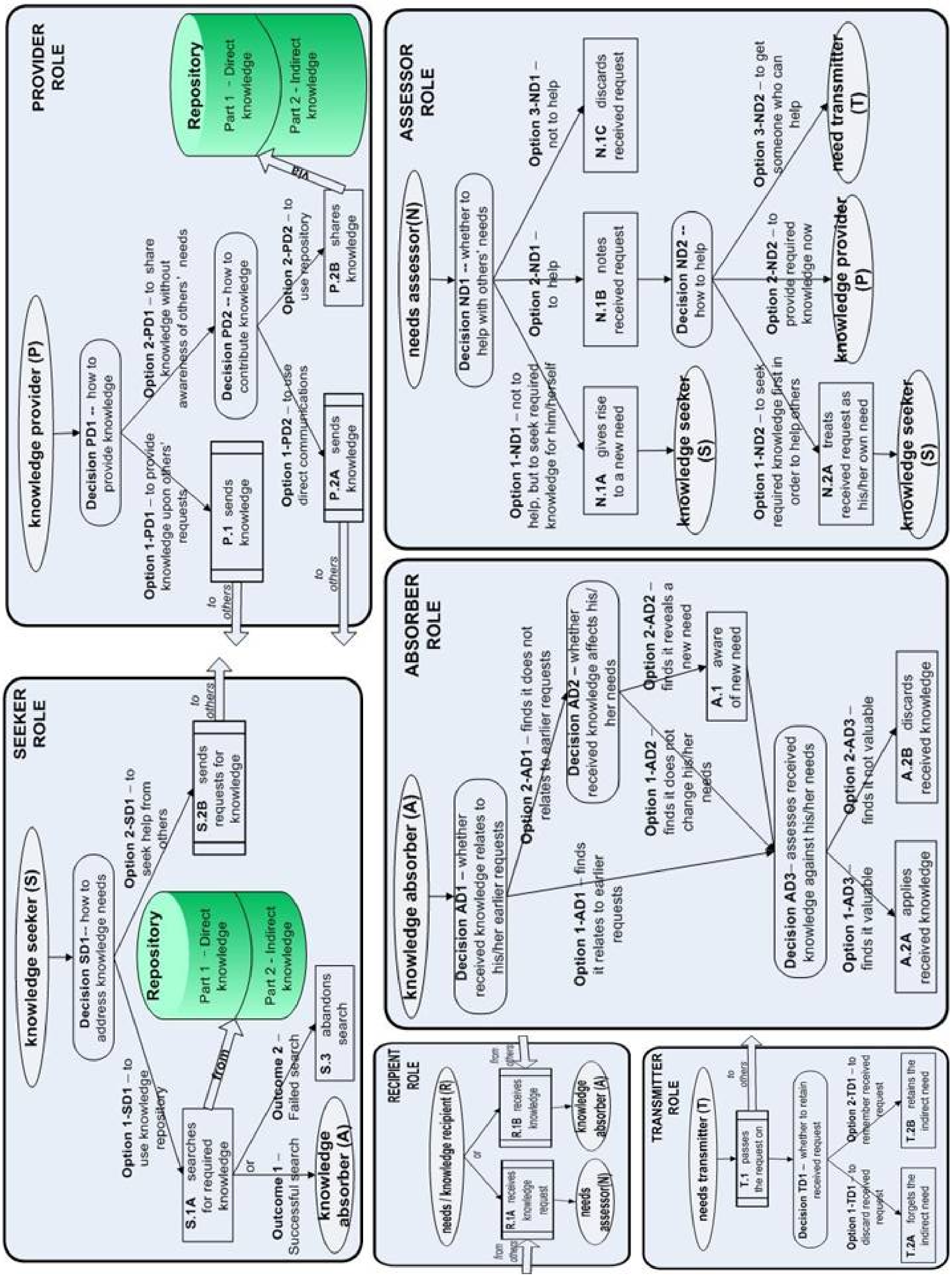


7th Version – 04/08/2008



8th Version – 11/08/2008


Five sub-processes of the knowledge transfer process v.5b
Jin Tong
20/08/2008
Knowledge Absorption Sub-process
Employee
acting as
the knowledge absorber (A)
Option 1
Option 2
A.1A finds the received knowledge in relation to his/her earlier k-requests
A.1B finds the received knowledge does not relate to his/her earlier k-requests
A.2A awares a new k-need because of the received knowledge
A.2B finds The received knowledge does not change his/her K-needs
A.3 assesses the received knowledge against his/her k-needs
A.4A does not use the received knowledge
A.4B uses the received knowledge
Knowledge / K-needs Receiving Sub-process
the knowledge / K-needs recipient (R)
or
R.1A receives knowledge / information directly
R.1B receives knowledge request directly
from other K-providers
from other K-seekers
the knowledge absorber (R)
the K-needs assessor(D)
Knowledge Seeking Sub-process
the knowledge seeker (S)
S.1A seeks help from colleagues
S.1B searches for required knowledge
S.2A requests knowledge directly from others
S.3 abandons search because of no findings
from
with the discovery new knowledge / information
the knowledge absorber (A)
The general body of knowledge
Part 1 - knowledge
Part 2 - location of the knowledge resources
by sending enquires to others
Knowledge Providing Sub-process
the knowledge provider (K)
K.1A directly provides knowledge/information to others upon their requests
K.1B decides to share knowledge without awareness of others' k-needs
K.2A directly shares his/her knowledge
K.2B indirectly shares knowledge with others
via
by sending knowledge / information to others
Assessing Other People's K-needs Sub-process
the K-needs assessor(D)
D.1 assesses / considers others' k-needs
D.2A is willing to help
D. 2B ignores others' k-needs for the present
D.3A passes the received request on
D.3B treats other's K-need as his/her own knowledge need before s/he is able to help the original K-seeker
Option 3
the knowledge provider (K)
the knowledge seeker (S)
to others



PROVIDER ROLE
knowledge provider (P)
Decision PD1 -- how to provide knowledge
Option 1-PD1 – to provide knowledge upon others' requests
Option 2-PD1 – to share knowledge without awareness of others' needs
P.1 sends knowledge
Decision PD2 -- how to contribute knowledge
Option 1-PD2 – to use direct communications
Option 2-PD2 – to use repository
P.2A sends knowledge
P.2B shares knowledge
via
Repository
Part 1 - Direct knowledge
Part 2 - Indirect knowledge
to others
SEEKER ROLE
knowledge seeker (S)
Decision SD1-- how to address knowledge needs
Option 1-SD1 – to use knowledge repository
Option 2-SD1 – to seek help from others
S.1A searches for required knowledge
S.2B sends requests for knowledge
from
or
Outcome 1 – Successful search
Outcome 2 – Failed search
S.3 abandons search
knowledge absorber (A)
ASSESSOR ROLE
needs assessor(N)
Decision ND1 -- whether to help with others' needs
Option 1-ND1 – not to help, but to seek required knowledge for him/herself
Option 2-ND1 – to help
Option 3-ND1 – not to help
N.1A gives rise to a new need
N.1B notes received request
N.1C discards received request
knowledge seeker (S)
Decision ND2 -- how to help
Option 1-ND2 – to seek required knowledge first in order to help others
Option 2-ND2 – to provide required knowledge now
Option 3-ND2 – to get someone who can help
N.2A treats received request as his/her own need
knowledge provider (P)
need transmitter (T)
ABSORBER ROLE
knowledge absorber (A)
Decision AD1 – whether received knowledge relates to his/her earlier requests
Option 1-AD1 – finds it relates to earlier requests
Option 2-AD1 – finds it does not relates to earlier requests
Decision AD2 – whether received knowledge affects his/ her needs
Option 1-AD2 – finds it does not change his/her needs
Option 2-AD2 – finds it reveals a new need
A.1 aware of new need
Decision AD3– assesses received knowledge against his/her needs
Option 1-AD3 – finds it valuable
Option 2-AD3 – finds it not valuable
A.2A applies received knowledge
A.2B discards received knowledge
RECIPIENT ROLE
needs / knowledge recipient (R)
or
R.1A receives knowledge request
R.1B receives knowledge
from others
needs assessor(N)
knowledge absorber (A)
TRANSMITTER ROLE
needs transmitter (T)
T.1 passes the request on
to others
Decision TD1 – whether to retain received request
Option 1-TD1 – to discard received request
Option 2-TD1 – to remember received request
T.2A forgets the indirect need
T.2B retains the indirect need

Appendix B – A Diary Study on KT

Knowledge transfer diary study result analysis

A 5-day diary study was conducted to collect data. Volunteers were invited to record their knowledge transfer experiences in logbooks, so the steps they followed could be analysed. To guide them to complete the logbooks, a covering letter and several open-ended questions (e.g. how did you help him/her with the received request?) were provided. Eight volunteers completed diaries about their knowledge-seeking and knowledge-providing experiences. A total of 74 reported knowledge transfer cases were collected. To process these cases, the model has been used as a coding scheme that contains predetermined categories to encode whether or not a particular behavior (each step in the model) has occurred (Robson, 2002). A typical example of these cases can be paraphrased as the following: "*When having a problem, the participant tried to seek help from a colleague first, but the response received did not help solve the problem. Then he decided to seek help from another colleague, and was still waiting for a response*". Using the model, this example can be encoded as involving the following steps:

S →SD1 → **S.2B** → R → **R.1B** → A → AD1 → AD3 →**A.2B** → S → SD1 → **S.2B** → **still waiting**

The above steps not in bold represent several decision points that have not been explicitly mentioned in the participant's diary, but can be inferred.

Full results of this diary study are listed in the table below.

Case (Participant)	Paraphrase of the case	Total Steps	Steps presented according to the model	Comments
01 (01)	When having a problem, the participant tried to seek help from one colleague first, but the received response did not solve the problem. Then he decided to seek help from another colleague, and was still waiting for	4	S →SD1 → **S.2B** → R → **R.1B** → A → AD1 → AD3 →**A.2B** → S → SD1 → **S.2B** → **still**	The participant's last action – "still waiting" has not been shown in the model. Decision points SD1, AD1, AD3, was not explicitly mentioned in the collected case. *Exploring why he*

	the response.		**waiting**	*did not mention it may be linked to the potential direct use of the model.*
02 (01)	The participant searched the Internet when he has got a problem to solve, then he got the solution.	2	S → SD1→ **S.1A** → A →AD1 → AD3 → **A.2A**	The participant did not explicitly explain how he decided to use the acquired knowledge.
03 (01)	The participant sought help from one colleague to solve his problem	3	S → SD1 → **S.2B** → R → **R.1B** → A → AD1 → AD3 → **A.2A**	
04 (01)	After seeking help from one colleague, the participant received information about whom he should contact to get the solution for his problem. Then he contacted the right knowledge resource and got the response.	5	S → SD1→ **S.2B** → R → **R.1B** → A → AD1 → AD3 → **A.2A**→ S → SD1→ **S.2B** → R → **R.1B** → A → AD1 → AD3 → A.2A	The participant got the solution by seeking help from other people twice. Although the first person did not solve the problem directly, he still provided indirect knowledge (who should be asked) to lead the participant's second KT attempt. He had not explicitly explained if he used the second response or not.
05 (01)	The participant got help from one colleague	3	S →SD1 → **S.2B** → R → **R.1B** → A → AD1 → AD3 → **A.2A**	
06 (01)	The participant received a request, then he helped the requester by suggesting him to contact another colleague who can actually	3	R → **R.1A** → D → ND1 → **N.1B** → ND2 → P →	Although the participant did not provide direct solution for the requester, he gave indirect

	help		**P.1**	knowledge about who can actually help. He did not explicitly explain how he assessed the request, and decided to help.
07 (01)	The participant received a knowledge request. He solved the problem for the requester, instead of providing necessary knowledge.	1	R→**R.1A**	
08 (01)	The participant described the technical actions he followed to solve a problem, not the steps of how he got the knowledge / instruction	0	***Empty case***	He did not explain how he is aware of this problem.
09 (01)	The participant received a request, then he helped by providing knowledge	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → P → **P.1**	
10 (01)	While working on his current task, the participant recalled an earlier request that he received a while ago. He decided to help the requester now by providing a response	2	R → R.1A → N → ND1 → **N.1B** →ND2 → P → **P.1**	The participant did not explicitly explain: 1) when he received the request; 2) what his response at that time; The waiting time in this case can not be shown in the model
11 (02)	The participant described the technical actions he followed to solve a problem, not	0	***Empty case***	

	the steps of how he got the knowledge / instruction			
12 (02)	The participant tried to seek help from a colleague, and then got what he needed	3	S →SD1 → **S.2B** → R → **R.1B** → A → AD1 → AD3 → **A.2A**	
13 (02)	The participant found a solution from the Internet	2	S → SD1 → **S.1A** → A → AD1 → AD3 → **A.2A**	
14 (02)	The participant described the technical actions he followed to solve a problem, not the steps of how he got the knowledge / instruction	0	***Empty case***	
15 (02)	The participant found a solution from the Internet	2	S → SD1 → **S.1A** → A → AD1 → AD3 → **A.2A**	
16 (02)	The participant received a request, then he helped by providing a response	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → P → **P.1**	
17 (02)	The participant received a request. But he tried to clarify the request with the requester before providing direct knowledge	3	R → **R. 1A** → N → ND1 → **clarifying the request** → ND1 → **N.1B** → ND2 → P → **P.1**	The process of clarifying the received request can be presented as seeking knowledge from the requester: S → SD1 → S.2B → R → R.1B → A → AD1 → AD3 → A.2A
18 (02)	The participant received a request, then he helped by providing a response	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → P →	

			P.1	
19 (02)	The participant received a request. He helped the requester by passing on the request to another colleague	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → T →**T.1** → TD1 →T.2B	The participant explained that he believed the other colleague can provider better solution than him. He has not explained how he dealt with the received request after passing it to the third person.
20 (02)	The participant received a request, then he helped by providing a response	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → P → **P.1**	
21 (03)	After having a problem, the participant tried a possible solution found from the Internet. The problem then is solved	3	S → SD1 → **S.1A** → A → AD1 → **AD3** → **A.2A**	The participant assessed the acquired knowledge against his need. So Decision point AD3 has been explicitly mentioned
22 (03)	After an unsuccessful searching on the Internet, the participant sought help from one colleague and got the solution	4	S → SD1 → **S.1A** → S.3 → S →SD1 → **S.2B** → R → **R. 1B** → A → AD1 → AD3 → **A. 2A**	The participant abandoned the current search from the Internet. But this has not been explicitly mentioned
23 (03)	The participant described the technical actions he followed to solve a problem, not the steps of how he got the knowledge / instruction	0	***Empty cases***	
24 (03)	After getting help from one colleague, the participant decided not to use the received	3	S →SD1→ **S.2B** → R → **R.1B** → A → AD1 →	The participant did not explicit explain why she decided not to use

	knowledge, but figuring out her own solution based on her previous knowledge		AD3 → **A.2B** → **try her own solution**	the received knowledge
25 (03)	***Blank***			The participant did not provide the example case.
26 (03)	The participant received a request, then she helped by providing a response	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → P → **P.1**	
27 (03)	**Blank**			The participant did not provide the example case.
28 (03)	The participant received an indirect request (passed by another colleague), then she helped	4	**T. 1 (by another person)** →R → **R.1A** → N → ND1 → **N.1B** → ND2 → P → **P.1**	Another colleague passed on the received request to the participant.
29 (03)	The participant received a request, then she helped by providing a response	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → P → **P.1**	
30 (03)	**Blank**			The participant did not provide the example case.
31 (04)	The participant has a problem, and she decided to search for solution from the internet (No finding yet)	1	S → SD1 → **S.1A** → S.3	Although no solution has been found, the participant has not abandoned the search yet
32 (04)	Following the case in 31/04, the participant continued to search for solution from the	3	S → SD1 → **S.1A** → A	The participant assessed the possible solution, and found out that

	Internet. She tried one possible solution, but decided not to use it		→ AD1 **→ AD3 → A.2B**	it was not the right one.
33 (04)	Following the case in 31/04 and 32/04, the participant revised her k-need. Then a solution has been found from the internet using the revised search terms.	2	**Revise the need** → S → SD1 **→ S.1A** → A → AD1 → AD3 **→ A.2A**	The participant has revised her k-needs, so the search terms are different.
34 (04)	After getting only part of the answer from the Internet, the participant decided to seek help from one colleague. She is still waiting for the response	3	S → SD1 **→ S.1A** → A → AD1 → AD3 **→ A.2A → Revise the need →** S → SD1 **→ S.2B → waiting for the response**	The participant revised her k-needs after acquiring part of the solutions from the Internet. Revising the need is not seen as part of the transfer process in this research. The transfer starts after a need is defined.
35 (04)	The participant decided to get a solution from the Internet. But she tried to get some help from a colleague about the right search terms before starting searching online	3	S → SD1 **→ S.2B** → R → R.1B → A → AD1 → AD3 → A.2A → S → SD1 **→ S.1A** → A → AD1 → AD3 **→ A.2A**	He has not explained if he had received any response when trying to seek help from other, and if he had used the given response.
36 (04)	The participant received a request, then she helped by providing a response	3	R **→ R.1A** → N → ND1 **→ N.1B** → ND2 → P **→ P.1**	
37 (04)	After providing help against a request, the participant received another request from the same person, because the previous	6	R **→ R.1A** → N → ND1 **→ N.1B** → ND2 → P **→ P.1**	The participant (provider) cannot guarantee that her response will be helpful for the requester.

	response did not help the requester. Then she helped again with another answer		**Then** R → **R.1A** → N → ND1 → **N.1B** → ND2 → P → **P.1**	
38 (04)	The participant cannot help with one request received earlier. But she decided to find the solution because this is part of her job responsibility	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → **N.2A** → S (actions not explained)	The participant treated the received request (indirect knowledge need) as her own knowledge need. So she became the seeker after being requested.
39 (04)	The participant received a request, then she helped by providing a response	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → P → **P.1**	
40 (04)	***Blank***			The participant did not provide the example case.
41 (05)	The participant found a solution online	2	S → SD1 → **S.1A** → A → AD1 → AD3 → **A.2A**	
42 (05)	The participant found a solution online	2	S → SD1 → **S.1A** → A → AD1 → AD3 → **A.2A**	
43 (05)	The participant decided to search for answer online for an request which he received earlier, but could not help at that time	5	R → **R. 1A** → N → ND1 → **N.1B** → ND2 → **N.2A** → S → SD1 → **S.1A** → A → AD1 → AD3 → A.2A	The participant did not explicitly explain: 1) when he received the request; 2) what his response at that time; The waiting time in this case can not be shown in the model

			Then N → ND1 → N.1B → ND2 → P → **P.1**	
44 (05)	The participant got help from one colleague	3	S →SD1 → **S.2B** → R → **R.1B** → A → AD1 → AD3 → **A.2A**	
45 (05)	The participant decided to search for answer for an request which he received earlier, but could not help at that time	5	R → **R. 1A** → N → ND1 → **N.1B** → ND2 → **N.2A** → S → SD1 → **S.1A** → A → AD1 → AD3 → A.2A **Then** N → ND1 → N.1B → ND2 → P → **P.1**	
46 (05)	The participant received a request, then he helped by providing a response	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → P → **P.1**	
47 (05)	The participant received a request, but he did not help	2	R → **R.1A** → N → ND1 → **N.1C**	The participant explained that he did not feel confident about his knowledge to help
48 (05)	The participant received a request, then he helped by providing a response	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → P → **P.1**	
49 (05)	The participant received a request, then he helped by providing a response	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → P →	The participant emphasised that the requester's need has been clarified

			P.1	(requests have been revised several times) during the face-to-face communication.
50 (05)	The participant received a request. He discussed it with another colleagues before providing help	3	R → **R.1A** → N → ND1 → **discussing with others** → **N.1B** → ND2 → P → **P.1**	How the participant discussed with other colleagues and what the outcomes are were not explained. The discussion process maybe represented as the following: ND2 →N.2A → S →SD1 → S.2B → R → R.1B → A → AD1 → AD3 → A.2A → N → ND1 This process can be repeated during the discussion
51 (06)	The participant found a solution online	2	S → SD1 → **S.1A** → A → AD1 → AD3 → **A.2A**	
52 (06)	The participant found a solution online	2	S → SD1 → **S.1A** → A → AD1 → AD3 → **A.2A**	
53 (06)	The participant got help from a colleague	3	S →SD1 → **S.2B** → R → **R.1B** → A → AD1 → AD3 → **A.2A**	
54 (06)	The participant tried to get help from a colleague, but still waiting for response	1	S → SD1 → **S.2B** → **waiting for help**	

55 (06)	**Blank**			The participant did not provide the example case.
56 (06)	The participant received a request, then he helped by providing a response	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → P → **P.1**	
57 (06)	The participant received a request, then he helped	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → P → **P.1**	
58 (06)	The participant received a request, then he helped the requester by suggesting him to contact another colleague who can actually help	3	R → **R.1A** → D → ND1 → **N.1B** → ND2 → P → **P.1**	
59 (06)	The participant received a request, then he helped by providing a response	3	R → **R.1A** → D → ND1 → **N.1B** → ND2 → P → **P.1**	
60 (06)	**Blank**			The participant did not provide the example case.
61 (07)	The participant got help from a colleague	3	S →SD1 → **S.2B** → R → **R.1B** → A → AD1 → AD3 → **A.2A**	
62 (07)	The participant found a solution online	2	S → SD1 → **S.1A** → A → AD1 → AD3 → **A.2A**	

63 (07)	The participant got help from a colleague	3	S → SD1 → **S.2B** → R → **R.1B** → A → AD1 → AD3 → **A.2A**	
64 (07)	The participant found a solution online	2	S → SD1 → **S.1A** → A → AD1 → AD3 → **A.2A**	
65 (07)	The participant found a solution online	2	S → SD1 → **S.1A** → A → AD1 → AD3 → **A.2A**	
66 (07)	The participant received a request. He tested the solution before gave it to the requester.	4	R → **R.1A** → N → **ND1** → **N.1B** → ND2 → P → **P.1**	Testing the solution is actually assessing the request to see if he can actually help or not. So decision ND1 has been explicitly mentioned.
67 (07)	The participant received a request, then he helped by providing a response	3	R → **R.1A** → D → ND1 → **N.1B** → ND2 → P → **P.1**	
68 (07)	The participant received a request, and he knew that he could help. But he decided not to help for the present.	3	R → **R.1A** → N → **ND1** → **N.1C** → **to help when is more convenient**	The participant explained that it was a time consuming problem, so he will help later when he is available. Waiting time is shown in the model.
69 (07)	The participant received a request. He helped the requester by passing on the request to another colleague	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → T → **T.1** → TD1 → T.2B	
70 (07)	The participant received a request, then he helped by providing a response	3	R → **R.1A** → D → ND1	

			→ **N.1B** → ND2 → P → **P.1**	
71 (08)	The participant described the technical actions he followed to solve a problem, not the steps of how he got the knowledge / instruction	0	***Empty case***	
72 (08)	The participant found a solution online	2	S → SD1 → **S.1A** → A → AD1 → AD3 → **A.2A**	
73 (08)	The participant found a solution online	2	S → SD1 → **S.1A** → A → AD1 → AD3 → **A.2A**	
74 (08)	The participant tried to get help from a colleague, but still waiting for response	1	S → SD1 → **S.2B → waiting for help**	
75 (08)	The participant tried to get help from a colleague first. While waiting for response, he found a solution online.	3	S → SD1 → **S.2B → waiting for help (**S → SD1 → **S.1A** → A → AD1 → AD3 → **A.2A)**	The participant did not explicitly explain if he is still waiting for response from that colleague.
76 (08)	The participant received a request, then he helped by providing a response	3	R → **R.1A** → D → ND1 → **N.1B** → ND2 → P → **P.1**	
77 (08)	The participant received a request. He helped the requester by passing on the request to another colleague	3	R → **R.1A** → N → ND1 → **N.1B** → ND2 → T →**T.1** → TD1 →T.2B	

78 (08)	The participant received a request, then he helped	3	R → **R.1A** → D → ND1 → **N.1B** → ND2 → P → **P.1**	
79 (08)	The participant received a request, then he helped	3	R → **R.1A** → D → ND1 → **N.1B** → ND2 → P → **P.1**	
80 (08)	The participant received a request, then he helped	3	R → **R.1A** → D → ND1 → **N.1B** → ND2 → P → **P.1**	

Analysis 1

Total cases collected: 74 (80 – 6 'blank')

Codable steps from all collected cases: 197

Uncodable actions described in reported cases: 1) Some of the uncodable actions involve behaviours that do not relate to knowledge (e.g. a sequence of actions of trying different IT techniques to solve one problem) – showing as 'empty' cases; 2) The rest of uncodable steps are mainly concerned with time that is not considered in the design of the model ('waiting step' in the case 01, 24, 34, 54, 68, 74, 75); 3) No other uncodable actions relating to knowledge transfer were observed.

Analysis 2

Total steps in the model: 29 Steps observed in the collected cases: 18

The following 11 steps in the model were not observed: three steps of providing knowledge without any requests (PD1→PD2;

PD2→P.2A; PD2→ P.2B); four steps of absorbing knowledge without any earlier requests (AD1→AD2; AD2→AD3; AD2→A.1; A.1→AD3); three steps of dealing with received requests after passing them to others (T.1→TD1; TD1→ T.1A; TD1→T.1B); and one step of treating other's needs as his own even after deciding not to help (ND1→N.1A). This may be because the questions in the logbooks only focus on how knowledge is transferred when people have immediate problems to solve. The above unobserved situations were not considered in the design of the logbooks. Further investigation is needed to look into these steps by revising questions in the logbooks.

Some decision points were not explicitly mentioned in the collected cases, but can be inferred according to participants' description. Follow-up interviews are required to further clarify these unreported routes and decision points.

Analysis 3

Only few negative examples in the collected cases: 1) only five examples of 'seeking help from someone, then being refused (or still waiting for response)''; 2) only two examples of "willing to help, but cannot help for the present"; 3) no example of "not willing to help"; 4) Only two examples of "receiving help, but decided not to use it". Such results are expected, as participants may not be willing to report these failed cases. Depth investigation on the negative data collection will be conducted in the follow-up interviews. The participants who provided negative examples also explained why they made such decision (i.e. to refuse to help for the present). Similar social factors affecting their decisions are not shown in the model. In addition, limited negative examples can not reveal a broad list of factors affecting people's decisions in the process. However, this is an aspect that the model does not attempt to represent. Major decision-making moments are highlighted to be the guidance for future research.

Analysis 4

Although 15 cases are collected involving more than two parties (including both people and the knowledge base, cases 01, 04, 06, 19, 22, 28, 34, 35, 43, 45, 50, 58, 69, 75, 77), only details of one person's steps are described in these cases. Transactions between different parties are not shown from the collected data. So follow up interviews are necessary to track down the complete knowledge transfer processes in these cases.

Analysis 5

No example was found about the red routes in the extended model – representing knowledge needs. This finding indicates that the approach of representing knowledge needs has been ignored in the organisational KT processes, and it can be an alternative way to try.

Conclusion

The model is valid according the diary study results, although this assessment reveals three major weaknesses of the model: 1) time effects are not shown in the model; 2) Social and cognitive issues affecting people's decisions in the process are not shown in the model; 3) People's actions (steps) on how people define their needs are not explicitly shown in the model.

Appendix C – Logbooks used in the KT diary study

Self-reporting Log Book 1
Knowledge seeking process

PhD research on interpersonal knowledge transfer model

PhD researcher: Jin Tong

Participant No. __

Covering letter

Hello,

This is to invite you to participate in a 5-day self-observation study on knowledge transfer processes at work.

While considering knowledge as one of the most valuable and strategic assets, increasingly organisations are keen to develop effective strategies to improve their knowledge exchange practice. This study attempts to develop a validated model for a better understanding of how knowledge is being transferred in organisations. Based on the results of a literature survey on knowledge transfer, I have developed a detailed knowledge transfer model formed of three sub-processes – knowledge seeking, knowledge sharing (knowledge providing), and knowledge receiving. In order to validate this theoretical model, I would like to use a self-observation (self-reporting) approach to collect the data about people's daily knowledge transfer experience. Participants are invited to complete two (self-completion) logbooks -- a logbook for their knowledge seeking experience and a logbook for their knowledge sharing experience.

This is logbook 1 for you to record your knowledge seeking processes in the next 5 working days. Please take 10 minutes to fill one page per day in this logbook. Your answers will be treated in confidence, and the completed logbooks will be destroyed as soon as the current research is complete.

Your participation is very important and your time is greatly appreciated. Please feel free to contact me if you have any questions or concerns.

Jin Tong.

PhD research student

Coventry University, UK

Mobile: 0788 1920 341

Email: jintongcn@gmail.com

Thank you very much for your help.

Would you like to have a follow-up interview session (approximately 30 minutes) to provide further details about your answers, after you have completed this logbook.

No. ☐[1]

Yes. ☐[2], and I will be available on ______________________________(Please offer me at least two available time slots during next week, if possible)

☐[1] Please tick this box, if you would like to receive a copy of the report of this research.

Thank you very much for your help.

Date: ____________________________________

1. What problem have you experienced today in your work? (i.e. any knowledge gap you identified, any new information you need, or specific technical problems in your work process, etc.)

2. How did you solve above problem? Or if the problem has not been solved yet, what have you done so far and what are you going to do next? (Please write down all major steps that you followed / are going to follow to get the solution)

   Step 1 ____
   ____
   Step 2 ____
   ____
   Step 3 ____
   ____
   Step 4 ____
   ____
   Step 5 ____
   ____
   Step 6 ____
   ____

The problem has been solved ☐[1]

The problem has not been solved yet, and I am at step __ (please specify your current step) ☐[2]

Participant No. __

Date: ____________________________________

1. What problem have you experienced today in your work? (i.e. any knowledge gap you identified, any new information you need, or specific technical problems in your work process, etc.)

2. How did you solve above problem? Or if the problem has not been solved yet, what have you done so far and what are you going to do next? (Please write down all major steps that you followed / are going to follow to get the solution)

   Step 1 ____
   ____
   Step 2 ____
   ____
   Step 3 ____
   ____
   Step 4 ____
   ____
   Step 5 ____
   ____
   Step 6 ____
   ____

The problem has been solved ☐[1]

The problem has not been solved yet, and I am at step __ (please specify your current step) ☐[2]

Participant No. __

Date: ____________________________________

1. What problem have you experienced today in your work? (i.e. any knowledge gap you identified, any new information you need, or specific technical problems in your work process, etc.)

2. How did you solve above problem? Or if the problem has not been solved yet, what have you done so far and what are you going to do next? (Please write down all major steps that you followed / are going to follow to get the solution)

Step 1 ____
____
Step 2 ____
____
Step 3 ____
____
Step 4 ____
____
Step 5 ____
____
Step 6 ____
____

The problem has been solved ☐[1]

The problem has not been solved yet, and I am at step __ (please specify your current step) ☐[2]

Participant No. __

Date: ____________________________________

1. What problem have you experienced today in your work? (i.e. any knowledge gap you identified, any new information you need, or specific technical problems in your work process, etc.)

2. How did you solve above problem? Or if the problem has not been solved yet, what have you done so far and what are you going to do next? (Please write down all major steps that you followed / are going to follow to get the solution)

   Step 1 ____
   ____

   Step 2 ____
   ____

   Step 3 ____
   ____

   Step 4 ____
   ____

   Step 5 ____
   ____

   Step 6 ____
   ____

The problem has been solved ☐[1]

The problem has not been solved yet, and I am at step __ (please specify your current step) ☐[2]

Participant No. __

Date: ______________________________________

1. What problem have you experienced today in your work? (i.e. any knowledge gap you identified, any new information you need, or specific technical problems in your work process, etc.)

2. How did you solve above problem? Or if the problem has not been solved yet, what have you done so far and what are you going to do next? (Please write down all major steps that you followed / are going to follow to get the solution)

   Step 1 ____
   ____

   Step 2 ____
   ____

   Step 3 ____
   ____

   Step 4 ____
   ____

   Step 5 ____
   ____

   Step 6 ____
   ____

The problem has been solved ☐[1]

The problem has not been solved yet, and I am at step __ (please specify your current step) ☐[2]

Participant No. __

Self-reporting Log Book 2
Knowledge sharing process

PhD research on interpersonal knowledge transfer model

PhD researcher: Jin Tong

Participant No. __

Covering letter

Hello,

This is to invite you to participate in a 5-day self-observation study on knowledge transfer processes at work.

While considering knowledge as one of the most valuable and strategic assets, increasingly organisations are keen to develop effective strategies to improve their knowledge exchange practice. This study attempts to develop a validated model for a better understanding of how knowledge is being transferred in organisations. Based on the results of a literature survey on knowledge transfer, I have developed a detailed knowledge transfer model formed of three sub-processes – knowledge seeking, knowledge sharing (knowledge providing), and knowledge receiving. In order to validate this theoretical model, I would like to use a self-observation (self-reporting) approach to collect the data about people's daily knowledge transfer experience. Participants are invited to complete two (self-completion) logbooks -- a logbook for their knowledge seeking experience and a logbook for their knowledge sharing experience.

This is logbook 2 for you to record your knowledge sharing processes in the next 5 working days. Please take 10 minutes to fill one page per day in this logbook. Your answers will be treated in confidence, and the completed logbooks will be destroyed as soon as the current research is complete.

Your participation is very important and your time is greatly appreciated. Please feel free to contact me if you have any questions or concerns.

Jin Tong.

PhD research student

Coventry University, UK

Mobile: 0788 1920 341

Email: jintongcn@gmail.com

Thank you very much for your help.

Participant No. __

Would you like to have a follow-up interview session (approximately 30 minutes) to provide further details about your answers, after you have completed this logbook.

No. ☐[1]

Yes. ☐[2], and I will be available on ______________________________(Please offer me at least two available time slots during next week, if possible)

☐[1] Please tick this box, if you would like to receive a copy of the report of this research.

Thank you very much for your help.

Date: ____________________________________

1. Have you received any knowledge requests from your colleagues or friends today / recently? (e.g. requests to help them to solve any problems, or to provide any information, etc.)

2. How did you receive the above request? (i.e. telephone request, email request, face-to-face conversation, etc. Please specify below)
   ____

3. Is this request related to the job you are doing?
   ☐[1] No ☐[2] Yes

4. Did you provide him/her any information/knowledge upon the above request?

   ☐[1] No, I could not help him/her, because ______________________
   ____
   ____

   ☐[2] Yes, I did help him/her (Please write down all major steps that you followed to provide the help)

Step 1 ____
____
Step 2 ____
____
Step 3 ____
____
Step 4 ____
____
Step 5 ____
____

Date: ____________________________________

1. Have you received any knowledge requests from your colleagues or friends today / recently? (e.g. requests to help them to solve any problems, or to provide any information, etc.)

2. How did you receive the above request? (i.e. telephone request, email request, face-to-face conversation, etc. Please specify below)
   ____

3. Is this request related to the job you are doing?
   ☐[1] No ☐[2] Yes

4. Did you provide him/her any information/knowledge upon the above request?

   ☐[1] No, I could not help him/her, because ______________________
   ____
   ____

   ☐[2] Yes, I did help him/her (Please write down all major steps that you followed to provide the help)

Step 1 ____
____

Step 2 ____
____

Step 3 ____
____

Step 4 ____
____

Step 5 ____
____

Participant No. __

Date: ____________________________________

1. Have you received any knowledge requests from your colleagues or friends today / recently? (e.g. requests to help them to solve any problems, or to provide any information, etc.)

2. How did you receive the above request? (i.e. telephone request, email request, face-to-face conversation, etc. Please specify below)
   ____

3. Is this request related to the job you are doing?
   ☐[1] No ☐[2] Yes

4. Did you provide him/her any information/knowledge upon the above request?

   ☐[1] No, I could not help him/her, because ______________________
   ____
   ____

   ☐[2] Yes, I did help him/her (Please write down all major steps that you followed to provide the help)

Step 1 ____
____
Step 2 ____
____
Step 3 ____
____
Step 4 ____
____
Step 5 ____
____

Participant No. __

Date: ____________________________________

1. Have you received any knowledge requests from your colleagues or friends today / recently? (e.g. requests to help them to solve any problems, or to provide any information, etc.)

2. How did you receive the above request? (i.e. telephone request, email request, face-to-face conversation, etc. Please specify below)

   ____

3. Is this request related to the job you are doing?

   ☐[1] No ☐[2] Yes

4. Did you provide him/her any information/knowledge upon the above request?

   ☐[1] No, I could not help him/her, because ______________________

   ____

   ____

   ☐[2] Yes, I did help him/her (Please write down all major steps that you followed to provide the help)

   Step 1 ____

   ____

   Step 2 ____

   ____

   Step 3 ____

   ____

   Step 4 ____

   ____

   Step 5 ____

   ____

Participant No. __

Date: ___________________________________

1. Have you received any knowledge requests from your colleagues or friends today / recently? (e.g. requests to help them to solve any problems, or to provide any information, etc.)

2. How did you receive the above request? (i.e. telephone request, email request, face-to-face conversation, etc. Please specify below)

____

3. Is this request related to the job you are doing?

☐[1] No ☐[2] Yes

4. Did you provide him/her any information/knowledge upon the above request?

☐[1] No, I could not help him/her, because _____________________

____

____

☐[2] Yes, I did help him/her (Please write down all major steps that you followed to provide the help)

Step 1 ____

____

Step 2 ____

____

Step 3 ____

____

Step 4 ____

____

Step 5 ____

____

Participant No. __

Appendix D – Follow-up interviews for the model development

Follow-up interview plan

21 November 2008

Purposes of the follow-up interviews

Because of the design of previous diary study, only one party's actions in the knowledge transfer processes are recorded in the logbooks. The follow-up interviews firstly aim to clarify the complete process involving all parties in these reported cases. Secondly, examples of some paths in the model are not collected in the diary study, so the follow-up interviews also attempt to collect these examples and explore why the participants make their decisions.

Questions for all participants –

1. Is knowledge sharing activity (eg. Seeking help from others, share what you know with others) often happening in your work environment?
2. Do you think knowledge sharing is important to you and the organisation?
3. Have you ever not helped with other people's requests for knowledge? Why? – Any examples for this situation? (Cases 47(05), 68(07) showed examples)
4. Have you ever decided not to help with others' requests, but still to look for required knowledge for yourself? Why do you do that?
5. What do you normally do with received requests after you passed them on to other people? Still bearing it in mind just in case you can help directly in the future, or just forgetting it?
6. Have your knowledge requests ever been left un-responded? Do you know the reasons for that? – Any examples for this situation? (In cases 01(01), 34(04), 54(06), 74(08), participants were still waiting for responses to their earlier requests, so have they received responses at the end? Have they been ignored?)
7. Have you ever provided knowledge to someone or shared knowledge with others without them asking you for that? Why do you do that? Any examples?
8. Have you ever learned new knowledge while browsing online or discussing/chatting with someone, without an intention to search for such knowledge (or without realizing that you actually need such knowledge)? Any examples?
9. Have you ever decided not to use others' responses to your earlier requests for knowledge? Why? Any examples? (Cases 24(03), 32(04) showed examples)
10. When you receive / discover new knowledge, do you normally check if it is relating to your earlier requests or it is something that you did not realise that you needed?

Further clarification with the following collected cases from the diary study: 04(01), 06(01), 19(02), 28(03), 32(04), 35(04), 38(04), 43(05), 45(05),

Follow-up interview plan

21 November 2008

50(05), 58(06), 69(07), 75(08), 77(08).

1. Who else has involved in these cases?
2. Do you know any details of how they participated in your reported cases?
3. Can I contact them for further details of your reported cases? Or you prefer to investigate the complete process by yourself and let me know later.